

Social Media Integration in Video Games. GameNshare: A Social Overlay for Desktop Games.

JOANA SOFIA SAMPAIO OSÓRIO

Outubro de 2015

Social Media Integration in Video Games

GameNshare: A Social Overlay for Desktop Games

Joana Sofia Sampaio Osório

**Dissertation for the
Master degree in
Computer Engineering, Specialization
in Graphics Systems and Multimedia**

Supervisor: Filipe de Faria Pacheco, PhD

Jury:

President:

Isabel de Fátima Silva Azevedo, PhD

Vowels:

António Abel Vieira de Castro, PhD

Filipe de Faria Pacheco, PhD

Porto, October 2015

Dedictory

This thesis is dedicated to my parents, brothers, nephew, nanny "Quê Quê" and to my great family who have supported me since the beginning of my master's studies.

This thesis is also dedicated to all those who believe in me and in the richness of learning.

Finally, but not the least, this thesis is dedicated to my beloved and geeky boyfriend António who has been an amazing source of motivation and inspiration.

Abstract

The ever increasing popularity of social media makes it a promising source for the personalization of gameplay experiences. Furthermore, involving social network friends in a game can greatly enrich the satisfaction of the player and also attract potential novel players to a game.

This master thesis describes a social overlay designed for desktop games, called GameNshare. It allows players to easily capture and share with multiple social networks game-related screenshots, videos and stories. Additionally, it also provides asynchronous multiplayer game mechanics to directly integrate social network friends in the game. GameNshare was designed to interact with the users in a non-intrusive way allowing them to be in complete control of what is shared. It prevents unsolicited sharing of messages, a key problem in social media integration tools, by the use of built-in message monitoring and anti-spam measures.

GameNshare was specially designed for players aged from 18 to 25 years that are regular users of Twitter and Facebook. It was tested by a group of 10 individuals from the target age range that were surveyed to capture their insights on the use of the social overlay. The implemented GameNshare features were well accepted by the testers that were also useful in highlighting features for future development. GameNshare ultimate goal is to make players look and ask for social integration and allow them to take full advantage of their social communities to improve gaming experiences.

Keywords: Social media integration, Desktop games, Social overlay, Social networks, Gameplay experiences

Resumo

A crescente popularidade dos media sociais torna-os um recurso promissor de personalização das experiências de um jogo. Envolver os amigos das redes sociais num jogo pode não só, enriquecer notoriamente a satisfação do jogador, mas também, atrair potenciais novos jogadores para o jogo.

Esta dissertação descreve um *software* de integração dos media sociais chamado GameNshare, que é destinado a jogos de computador em ambientes de trabalho Windows, Mac OS X ou Linux. Este *software* permite aos jogadores facilmente capturar e partilhar imagens de ecrã, vídeos e histórias relacionadas com o jogo em diversas redes sociais. Para além disso, o GameNshare também fornece mecanismos multijogador assíncronos que integram diretamente os amigos das redes sociais no jogo. Contrariamente à maioria dos sistemas de integração dos media sociais, o GameNshare foi projetado para interagir com os utilizadores de forma não-intrusiva, permitindo-lhes ter controlo absoluto do que é partilhado.

Um dos problemas chave na área de integração dos media sociais é a partilha de mensagens não solicitadas (SPAM). Numa tentativa de ultrapassar este problema o GameNshare oferece mecanismos de monitorização das mensagens partilhadas através do uso deste *software* e medidas anti-*spam* para a prevenção de mensagens duplicadas. O GameNshare foi desenhado para jogadores entre os 18 e 25 anos que são utilizadores regulares do Twitter e do Facebook. Como forma de avaliar as necessidades do público-alvo, o GameNshare foi testado por jogadores desta faixa etária. Nestes testes foram avaliadas entre outras questões o funcionamento, confiabilidade, intrusividade e eficiência do protótipo e concluiu-se que, em relação a estes aspetos, a maioria das funcionalidades implementadas pelo GameNshare funcionaram bem. O principal objetivo do GameNshare é fazer com que os jogadores vejam a integração social como forma de melhorar as experiências de jogabilidade. O GameNshare irá permitir aos jogadores tirar o máximo proveito das suas comunidades sociais para melhorar as suas experiências no jogo.

Palavras-chave: Integração dos media sociais, Jogos de computador, *Software* de integração dos media sociais, Redes sociais, Experiências no jogo

Acknowledgments

My special thanks go to Tiago Magalhães, creator of “Adventure! The Paladin Order” game and also for APO’s test team for all the help and dedication in the construction of this prototype.

I would also like to thank my supervisor Filipe Pacheco for always being available to help me develop this project.

Finally, I would like to show my gratitude to all my professors, colleagues, family and friends that somehow supported me during this work and contributed to its success.

Index

1	Introduction	1
1.1	Goals.....	2
1.2	Contributions.....	2
1.3	Overview of Dissertation	2
2	Background on Social Media Integration	5
2.1	Definition and Classification of Social Media	6
2.2	Strategies of Social Media Integration.....	7
2.3	Social Media Integration Challenges	8
2.4	Social networks APIs and SDKs	9
2.5	Similar Systems and GameNshare Progress	10
2.5.1	Xbox 360: Social Network Integration (2009)	10
2.5.2	Overwolf and Evolve: Social Overlays (2011).....	11
2.5.3	Call of Duty: Elite and Facebook Integration (2011)	13
2.5.4	Final Fantasy XIII-2: Outerworld Services (2012)	14
2.5.5	PS4: Share button (2013)	14
3	Requirements Specification.....	16
3.1	Functional requirements	16
3.1.1	Record a Video.....	18
3.1.2	Take a screenshot.....	19
3.1.3	Share Gameplay	20
3.1.4	Craft	26
3.1.5	Passive party system.....	31
3.2	Non-functional requirements.....	35
3.2.1	Hardware and Software requirements.....	35
3.2.2	Performance requirements	35
3.2.3	Security requirements.....	36
4	System Design and Architecture	37
4.1	Desktop game client	38
4.1.1	GameNshare integration with Desktop Game Application (APO)	38
4.1.2	Social overlay tools	40
4.2	External Social Networks.....	42
4.2.1	Facebook’s Graph API	43
4.2.2	Twitter REST API	45

4.2.3	Google APIs.....	46
4.2.4	Twitch API.....	48
4.3	Web Server.....	48
4.3.1	Database.....	49
4.3.2	Web services.....	50
5	Implementation and Evaluation.....	55
5.1	Alpha version.....	56
5.1.1	Share Menu.....	56
5.1.2	Record and Upload Video.....	56
5.1.3	Capture and Upload Screenshots.....	60
5.1.4	Share Game Log.....	62
5.2	Alpha testing.....	64
5.2.1	Results.....	64
5.2.2	Conclusions.....	65
5.3	Beta version.....	66
5.3.1	Anti-spam measures.....	66
5.3.2	Graphic improvements.....	67
5.3.3	Personal game stories.....	68
5.3.4	Passive party system.....	69
5.3.5	Progress of the missing features.....	71
5.4	Beta testing.....	71
5.4.1	Results.....	72
5.4.2	Conclusions.....	77
6	Conclusions and Future work.....	79
	References.....	81
	Annexes.....	87
	Annex A - Testers Invitational Form.....	89
	Annex B - Alpha Testing Survey GameNshare Sections.....	99
	Annex C - Beta Testing Survey GameNshare Sections.....	111
	Annex D - InForum 2015 Paper.....	125

Figure Index

Figure 1 - Contrasting the seven functional blocks of different social network websites	7
Figure 2 - Xbox 360 Facebook Application	11
Figure 3 - Overwolf inside Allods Online	12
Figure 4 - Call of Duty: MW3 with Facebook integration	13
Figure 5 - Final Fantasy XIII-2 1.03 (Boss achievement)	14
Figure 6 - Share Options of PS4	15
Figure 7 - "Share" button PS4 controller	15
Figure 8 - Average of test team opinions (scale of 1 to 5) about broadcast gameplay (orange), craft system (blue) and passive party system (grey)	17
Figure 9 - Use case diagram of social overlay prototype with prioritization scale	18
Figure 10 - Use case diagram of share gameplay	20
Figure 11 - Use case diagram of "Craft"	27
Figure 12 - Use case diagram of "Party"	31
Figure 13 - System architecture of GameNshare	37
Figure 14 - Navigation chart/Game flow of GameNshare integration with APO	38
Figure 15 - Class diagram of Share Menu integration with APO	39
Figure 16 - Class diagram of capture screen integration with APO	39
Figure 17 - Class diagram of craft and party integration with APO	40
Figure 18 - Class diagram of public media services integration - client side	40
Figure 19 - Class diagram of database integration - client side	41
Figure 20 - Class diagram of the FFmpeg integration - client side	41
Figure 21 - Graph API Explorer	43
Figure 22 - Exchanging OAuth 2.0 tokens on Facebook	44
Figure 23 - Twitter pin-based authentication	45
Figure 24 - Sequence Diagram of Twitter authentication flow	46
Figure 25 - Google APIs Explorer	46
Figure 26 - Google OAuth 2.0 flow for installed applications	47
Figure 27 - GameNshare Entity Framework Data Model	50
Figure 28 - Class diagram of Social Networks MVC Controllers	50
Figure 29 - Class diagram of Database MVC Controllers	51
Figure 30 - Share Menu (alpha version)	56
Figure 31 - Select Video to Upload to Facebook, Twitter and YouTube (alpha version)	58
Figure 32 - Share APO' video to Facebook, Twitter and YouTube (alpha version)	59
Figure 33 - Samples of posted videos in the social networks: (a) Facebook video; (b) Twitter video; (c) YouTube video	59
Figure 34 - User prompt for upload video to YouTube and share the link in Twitter	60
Figure 35 - Select Screenshot to Upload (alpha version)	61
Figure 36 - Share a screenshot to Facebook and Twitter (alpha version)	61
Figure 37 - Samples of posted screenshots in the social networks: (a) Facebook screenshot; (b) Twitter screenshot	62

Figure 38 - Sample of “gamelog.json” file	63
Figure 39 - Upload story to Facebook and Twitter (alpha version): (a) Share windows of game log option after select “Started playing” story; (b) “Started playing” story in Twitter and Facebook	63
Figure 40 - Alpha testing: Product awareness (scale of 1 to 5).....	64
Figure 41 - Alpha testing: Product problems and limitations (open answer).....	65
Figure 42 - Duplicate Twitter status error - GameNshare client side.....	66
Figure 43 - Rejected YouTube video (duplicate upload)	66
Figure 44 - Alert message of duplicate twitter story.....	67
Figure 45 - Recording a video in Overworld Scene	67
Figure 46 - Share Menu (beta version).....	68
Figure 47 - Add a story (beta version): Enter a story description	68
Figure 48 - Add a story (beta version): GameLog with a personal story.....	68
Figure 49 - Passive party system (beta version): (a) Synchronize social network friends; (b) Select a friend.....	69
Figure 50 - Passive party system (beta version): Use a friend	69
Figure 51 - Friends permissions of passive party system	70
Figure 52 - Facebook private message requesting a game ore.....	71
Figure 53 - Beta testing: Product awareness (scale of 1 to 5).....	72
Figure 54 - Beta testing: Video Record and Upload evaluation (scale of 1 to 5)	73
Figure 55 - Beta testing: Complaints and Suggestions - Video Record and Upload (open answer)	73
Figure 56 - Beta testing: Screen Capture and Upload evaluation (scale of 1 to 5)	74
Figure 57 - Beta testing: Complaints and Suggestions - Screen Capture and Upload (open answer).....	75
Figure 58 - Beta testing: Share Game Log evaluation (scale of 1 to 5)	75
Figure 59 - Beta testing: Complaints and Suggestions - Share Game Log (open answer).....	76
Figure 60 - Beta testing: Passive Party System evaluation (scale of 1 to 5).....	76
Figure 61 - Beta testing: Friend permissions evaluation (scale of 1 to 5)	77

Table Index

Table 1 - Prioritization scale of functional requirements	16
Table 2 - "Record Video" use case brief description.....	18
Table 3 - "Record Video" use case flow-of-events.....	19
Table 4 - "Take a screenshot" use case brief description	19
Table 5 - "Take a screenshot" use case flow-of-events	20
Table 6 - "Share Gameplay" use case brief description	21
Table 7 - "Share Gameplay" use case flow-of-events.....	21
Table 8 - "Upload Video" use case brief description	22
Table 9 - "Upload Video" use case main flow	22
Table 10 - "Upload Screenshot" use case brief description.....	23
Table 11 - "Upload Screenshot" use case main flow	23
Table 12 - "Share Game Log" use case brief description	24
Table 13 - "Share Game Log" main flow and "add story" alternative flow	24
Table 14 - Alternative flows 1 and 2 of the essential "Share Gameplay" use cases	25
Table 15 - Alternative flows 3, 4 and 5 of the essential "Share Gameplay" use cases	25
Table 16 - "Broadcast Gameplay" use case brief description	26
Table 17 - "Broadcast Gameplay" use case main flow.....	26
Table 18 - "Craft" use case brief description	27
Table 19 - "Craft" use case flow-of-events	28
Table 20 - "Accept requests" use case brief description	28
Table 21 - "Accept requests" use case main flow and alternative flow	29
Table 22 - "Send requests" use case brief description	30
Table 23 - "Send requests" use case flow-of-events	30
Table 24 - "Passive party/team system" use case brief description.....	31
Table 25 - "Passive party/team system" use case flow-of-events.....	32
Table 26 - "Use friends' bonus" use case brief description	33
Table 27 - "Use friends' bonus" use case flow-of-events	33
Table 28 - "Change friends' permissions" use case brief description.....	34
Table 29 - "Change friends' permissions" use case flow-of-events.....	34
Table 30 - GameNshare system requirements	35
Table 31 - Main REST endpoints of users.....	52
Table 32 - Main REST endpoints of social network requests.....	53
Table 33 - Main REST endpoints of party relationships (part 1)	53
Table 34 - Main REST endpoints of party relationships (part 2)	54
Table 35 - Compatibility table of GameNshare features	55

Code Index

Code 1 - Convert JSON string into model objects with Reflection.....	54
Code 2 - Screenshot movie functions: (a) start recording; (b) save video sequence	57
Code 3 - Write a video frame with Ruslan-B FFmpeg.AutoGen.....	58
Code 4 - Take a screenshot function.....	60
Code 5 - Intersecting APO's Twitter users with player's Twitter friends	70

Acronyms and Nomenclature

API	Application Programming Interface
APO	Adventure! The Paladin Order
CAN-SPAM	Controlling the Assault of Non-Solicited Pornography and Marketing
SPAM	Sending and Posting Advertisement in Mass
COPPA	Children’s Online Privacy Protection Act
CRUD	Create, Read, Update and Delete
FFIEC	Federal Financial Institutions Examination Council
FFMPEG	Fast Forward Moving Picture Experts Group
FPS	Frames per second
HTTP	Hypertext Transfer Protocol
HUD	Heads-up display
JSON	JavaScript Object Notation
KB	KiloByte
MD5	Message-Digest algorithm 5
MVC	Model-view-controller
PC	Personal Computer
PS4	PlayStation 4
REST	Representational State Transfer
SDK	Software Development Kit
TCPA	Telephone Consumer Protection Act
UI	User Interface

1 Introduction

Social media content and the large number of users in social networks are one of the most promising sources for the personalization of game play experiences [Kaplan, and Haenlein, 2010]. Furthermore, inviting social network friends to influence the game experience with new contents and participation in gameplays can increase the awareness of players' activities outside the game and increase the appeal of the game to users that are not currently playing. This fact can also promote the creation of collaborative environments to improve the game quality for every participant [Konert, et al., 2014].

To support the creation of new and innovative ways to integrate social media with desktop games, a prototype of a social overlay, called GameNshare was developed and tested. This prototype was integrated in the context of the multiplatform game, the *Adventure! The Paladin Order* (APO)[Clockwork Inc., 2015a]. It was designed for all desktop players, but with a focus in the age group between 18 and 25 that according to Duggan, et al. [Duggan, et al., 2015] is included in the most popular range of ages for the users of Twitter and Facebook.

This manuscript, as well as the project to which it refers was produced for the dissertation of the Master's Degree in Computer Engineering, specialization in Multimedia and Graphic Systems of the Instituto Superior de Engenharia do Porto (ISEP). GameNshare had as main goal the development of a user-friendly, non-intrusive and multiplatform social overlay for desktop games to allow players to enhance their gaming experiences using the social communities. Therefore, this master thesis represents an essential contribution for the development of social media integration in games.

1.1 Goals

Considering that there are several successful stories of the use of social integration tools for game enhancement, the work described in this dissertation intends to implement and evaluate a social overlay for desktop games that combines and integrates the different existing solutions to solve common limitations. Among the different problems that this research intends to bypass, the following ones can be highlighted [Osório, and Pacheco, 2015]: (1) Existence of spam-like social network promotion systems; (2) Lack of asynchronous gameplay and multiplayer mechanics with social networks friends in desktop games; (3) Lack of multiplatform social overlays fully integrated with desktop games.

This dissertation proposes a social overlay for desktop games (GameNshare) which, unlike most of existing systems, is designed to have non-intrusive interactions with the users, allowing them to be in complete control of what is shared and used by the social communities.

It enables not only to easily capture and share game experiences on multiple social networks, but also includes asynchronous multiplayer mechanics that integrate social networks friends with the game. The goal of this master thesis is to allow players to make the best use of their social communities to enhance their desktop gaming experience.

1.2 Contributions

This dissertation aims to contribute to the area of Cultural and Entertainment Computing with the implementation of a social overlay for multiplatform desktop games that is targeted for regular users of Twitter and Facebook. GameNshare was developed in the context of APO [Clockwork Inc., 2015a] and tested by a group of 10 players aged between 19 and 25.

In summary, this dissertation contributes with: (1) an overview of existing social network game overlays and the presentation of an innovative implementation approach that integrates perspectives and views from existing systems and novel features (Chapter 2-4); (2) a detailed description of an implementation of a social overlay for Unity3D [Unity Technologies, 2015f] multiplatform desktop games, named GameNshare (Chapter 5 - Sections 5.1 and 5.3); (3) evaluation of the GameNshare prototype, aiming to capture the target audience experiences a feedback (Chapter 5 - Sections 5.2 and 5.4).

1.3 Overview of Dissertation

This dissertation is structured in 6 chapters.

Following the introduction, chapter 2 reviews the state-of-the-art of the related literature and compares the existing systems with GameNshare.

Chapters 3-5 are devoted to the presentation of GameNshare including its requirement specifications, system design and architecture, implementation and evaluation.

Finally, chapter 6 summarizes the main conclusions and future work perspectives.

2 Background on Social Media Integration

Full social media integration in games is frequently seen in online games that can be played through the social networks communities. Usually, these games contain multiplayer features or asynchronous gameplay mechanics. These games are often implemented as browser games or as mobile game apps [Shin, and Shin, 2011]. Social network games are between the most played games in the world, with titles such as FarmVille [zyngaAdmin, 2014], Mafia Wars [Hof, 2010], The Sims Social [Lynley, 2011] and Candy Crush [Andrew, 2013].

Recently, players have realized that the gaming community can be expanded directly through the use of existing social media sites like YouTube [Google, 2015a], Twitter [Twitter Inc., 2015a] and Facebook [Facebook, 2015a]. Gamers started to record and upload their gameplays to YouTube and to invite friends to comment and share gaming approaches and strategies. These new habits created business niches that are starting to be explored. As an illustration, Rooster Teeth created a spinoff of their company called Achievement Hunter, that is exclusively dedicated to the upload of gameplay videos of their own staff playing games competitively [Fjællingsdal, 2014]. Live-streaming of gameplay, particularly Twitch.Tv [Twitch Interactive, Inc., 2015], has also become popular lately [Pires, and Simon, 2015]. Twitch created the concept of “gamer celebrities” that collect in real time an enormous amount of fans to watch and comment live games. Both game developers and players have profited from the popularity of live-streaming. One of the top leaders on Twitch.tv in 2013, Sacriel, has a regular audience of more than 2,000 simultaneous viewers on his game streams. This has allowed him to make his income solely on revenues from YouTube and Twitch.tv and becoming a professional “social network gamer” [Brunelle, 2013].

To answer to this emerging reality, social media integration in video games has progressively become more available in game consoles such as Microsoft Xbox [Mack, 2009], Sony PlayStation 4 [Kelion, 2013] and also on desktop games [Takahashi, 2011; Echobit LLC, 2015]. Usually, the social features for these types of platforms are media sharing, live-streaming and exchanging playing experiences.

The remainder of this chapter presents a review of the state of the art of the related literature and work with emphasis on three topics: (1) definition and classification of social media; (2)

strategies of social media integration; (3) social networks APIs and SDKs; (4) similar systems and prototype progress.

2.1 Definition and Classification of Social Media

According to Andreas Kaplan and Michael Haenlein [Kaplan, and Haenlein, 2010] social media is defined as *"a group of Internet-based applications that build on the ideological and technological foundations of Web 2.0, and that allow the creation and exchange of user-generated content."* Social media tools and technologies can have different forms and formats, including forums, weblogs, wikis, social networks, podcasts, photos, pictures and videos. Among the different main categories of social media, the following ones can be highlighted: (1) blogs; (2) micro-blogs; (3) collaborative projects; (4) content communities; (5) social networking sites; (6) social news websites; (7) virtual worlds.

Blogs are a particular type of websites that are mainly used as diaries. The users of these platforms can comment, add texts, graphic, videos, links or web pages, e.g. Blogger [Blogger, 2015]. Micro-blogs are a special type of blogs that can be viewed as a transition between blogs and social network sites. This blogs were initially developed to share and read short messages (140 characters) with other users, e.g. Twitter [Twitter Inc., 2015a].

Collaborative projects are those in which users collaborate directly with each other to produce new content in real time, e.g. Wikipedia [Wikipedia, 2015]. Content communities are platforms in which users can share different multimedia contents such as pictures, videos, music. Some examples of these communities are Instagram [Systrom, and Krieger, 2014], YouTube [Google, 2015a], Twitch [Twitch Interactive, Inc., 2015] and Spotify [Spotify AB, 2015].

Social network sites can be represented by the graphs of the users' connections. With this type of applications users can create personal profiles, invite friends, share different type of contents and send messages to each other, e.g. Facebook [Facebook, 2015a] and Google+ [Google, 2015b]. Social news websites, also known as crowdsourcing news, allow the instantaneous sharing of news and contents from anywhere on the web, using submitting, voting and commenting, e.g. Reddit [Reddit, 2015].

Virtual worlds are three-dimensional environments "in which users can appear in the form of personalized avatars and interact with each other as they would in real life" [Kaplan, and Haenlein, 2010]. Virtual worlds can be classified in two types: virtual games worlds, e.g. World of Warcraft [Blizzard Entertainment, Inc., 2015] and virtual social worlds, e.g. Second Life [Linden Research, Inc., 2015].

2.2 Strategies of Social Media Integration

To specify the different types of social media user experience, Kietzmann et al. conceptualized a honeycomb with seven functional build blocks that defines how social media differ according to the focus placed on one or all seven proposed blocks: identity, conversations, sharing, presence, relationships, reputation and groups [Kietzmann, et al., 2011]. This model can help developers and companies understanding on what social platforms they should be more present and how they should use them as strategic tools.

Figure 1 illustrates the honeycomb of Kietzmann *et al.* with four examples: LinkedIn, Foursquare, YouTube and Facebook. The greater is the importance of each social media functionality on the site, the darker is the block colour [Kietzmann, et al., 2011].

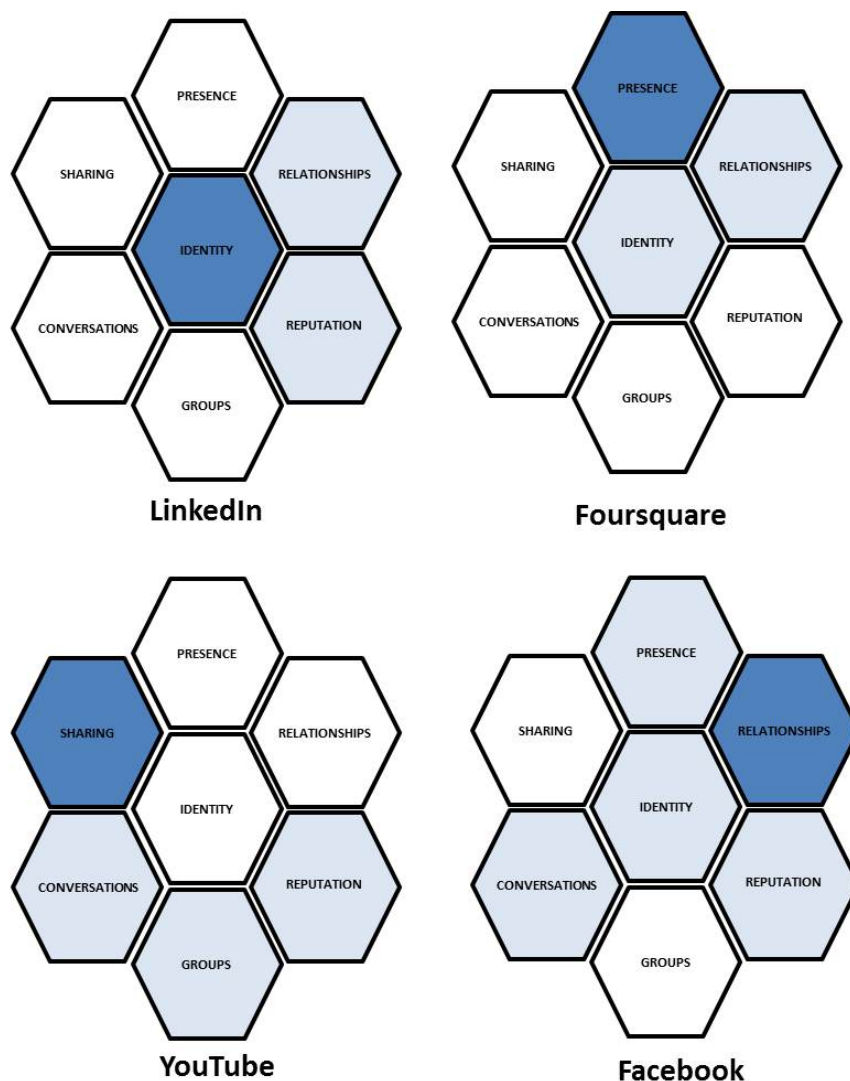


Figure 1 - Contrasting the seven functional blocks of different social network websites¹

¹<http://busandadmin.uwinnipeg.ca/silvestrepdfs/PDF06.pdf>

Based on this model Facebook [Facebook, 2015a] can be incorporated in almost all contexts and is used to promote long-term relationships with target audience. In this platform there is a strong connection between identity and relationships, because the higher the identity is rated by a social media community, the higher the relationships will also be rated. Facebook users have specific profiles for self-promotion (real identity vs. virtual identity) and communicate (share or post) motivations, causes, events and activities with other friends. On the other hand, Twitter [Twitter Inc., 2015a] is more focused in the conversation than in the identity of its users. In this platform, users "tweet" short messages which are often real-time status updates. Most of these messages are short-term, so there is no obligation to answer. Further, Twitter is a good place to share real-time scores and stories of games.

The Kietzmann model can equally be applied to YouTube [Google, 2015a]. This social network is centred in the sharing of videos. Originally, YouTube only allowed users to upload homemade videos, but recently it is also used to improve the marketing of companies and for the sharing of music videos, series and movies. In opposition to Facebook that uses "likes" for reputation, YouTube reputation is based on "view counts" and "ratings". The group relationships have a great importance in this social community, so this platform is indicated for the sharing of user experiences.

2.3 Social Media Integration Challenges

Social network sites are already used by several companies not only to support the creation of brand communities, but also for marketing research in the area of netnography. The study of netnography can permit among other opportunities the creation of good strategies for the social media integration and consequently, improve the product quality for every user [Kaplan, and Haenlein, 2010].

However, in order to successfully interact with their customers, companies should review some supervisory guidance about privacy compliance challenges. The "Social Media: Consumer Compliance Risk Management Guidance" released on 11th December 2013 by Federal Financial Institutions Examination Council (FFIEC) of the U. S. government [Federal Financial Institutions Examination Council (FFIEC), 2013] is an example of a guidance. It includes some privacy rules that have particular significance in the context of social media spaces. These guidelines are only designed for financial institutions, but can be applied to other institutions' and products contexts.

FFIEC guidelines identifies certain crucial issues that financial institutions should include in their social media compliance programs. Among these issues can be highlighted the following ones: (1) Complaint Submission and Processing; (2) GLBA Privacy Rules and Interagency Data Security Guidelines; (3) Controlling the Assault of Non-Solicited Pornography and Marketing (CAN-SPAM) and Telephone Consumer Protection Act (TCPA); (4) Children's Online Privacy Protection Act (COPPA).

“Complaint Submission and Processing” FFIEC guidelines remind institutions that they should implement a process for the monitoring of the information that is posted to the institutions’ social media sites or contracted by a third-party vendor². However, according to these guidelines is not expected that the institutions monitor all communications about them, which include complaints or inquiries.

Social media sites allow institutions to collect and access to information about customers. Therefore, FFIEC recommends institutions to clearly divulge their privacy policies in their sites. The institutions may face reputation risks if they are not transparent about the privacy policies that their social media sites or third-party vendors use.

FFIEC guidance suggests institutions to review the CAN-SPAM Act and the TCPA to control the sending of unsolicited communication to their customers via social media activities. According to the acts cited by FFIEC, organizations should establish certain requirements in order to send unsolicited commercial messages to their consumers using social media activities.

In relation to COPPA, FFIEC remembers that if the social media activities of the institutions collect or use information about children under 13 years of age, may be imposed responsibilities on operators of online services. As such, institutions should clarify this issue in their privacy policies.

In addition to the privacy compliance challenges, the intrusiveness of social media can also be a challenge to their integration in products. Two surveys performed by the Marketing Executives Network Group reveal that 55% of senior marketing executives and 52% of consumers consider that social media is intrusive. Besides this, these surveys also show that 39% of senior marketing executives agree that the information collected by the social media analytics is not useful to their businesses [Marketing Executives Networking Group (MENG), 2015]. Intrusiveness is also a key issue in the integration of social media in games. For example, when Zynga forced its users to spread the games virally on Facebook, the popularity of their games began to decline. In 2012, Zynga shares had huge losses and the company was practically defamed by the gaming press for its intrusive and exploitative practices [MacCallum-Stewart, 2014].

2.4 Social networks APIs and SDKs

"Application Programming interfaces "(APIs) combined with" Software Development kits "(SDKs) are effective ways of direct communication between companies and their customers’ social network platforms [Muller, and Thiesing, 2011]. This type of communication protocols is officially available for several companies like Facebook, Twitter, Google and Amazon.com (actual owner of Twitch).

The **Facebook’s Graph API** [Facebook, 2015f] is the low-level HTTP-based API of Facebook Platform that allows developers to read and write data into Facebook. To interact with Facebook

² This recommendation can also be applied to third-party applications of the companies.

Graph API developers need to use the OAuth 2.0 protocol [Hardt, 2012] to obtain a Facebook access token for each user. This API can be used with HTTP Requests or SDKs. Facebook offers official SDKs for iOS, Android, JavaScript, PHP and for Unity3D Web Player and mobile, however a wide range of unofficial and open-source SDKs can be used for other environments, e.g. Facebook SDK C# [Outercurve Foundation, 2014].

The **Twitter REST API** [Twitter Inc., 2015d] is the main application programming interface that allows developers to read and write Twitter platform data. To connect with Twitter REST API developers need to use OAuth 1.0 protocol [Hammer-Lahav, 2010], this protocol is used to acquire an access token for each user. This API can be used with HTTP Requests or with libraries. Twitter offers official libraries for the integrate Twitter in Java. Besides this, an extensive range of unofficial libraries are available for other languages, e.g. Twitterizer [Twitterizer, 2015].

Google APIs [Google, 2015e] is a group of APIs provided by Google that allow the communication of third-party applications with Google services. Google APIs use the OAuth 2.0 protocol for authentication and authorization. These APIs can be integrated with third-party applications directly with HTTP Requests or with client libraries. Google provides official client libraries for Java, JavaScript, .NET, Objective-C, PHP and Python [Google, 2015f].

Twitch API [Justin.tv Inc, 2013] allow developers to integrate Twitch content and Twitch streams into their applications. This API uses the OAuth 2.0 protocol for authentication and authorization and can be accessed directly by the REST API and by the use of the JavaScript SDK.

2.5 Similar Systems and GameNshare Progress

The concept of social integration in video games is not new and there are several successful stories of solutions that take advantage of social integration for the promotion of video games within their target audience.

This section presents, in a chronological order, examples illustrating the evolution of social integration in video games over the time. These examples vary greatly, not only in their degree of complexity and temporal resolution, but also in the financial investment applied to the development of each platform.

2.5.1 Xbox 360: Social Network Integration (2009)

On November 17th of 2009 [Mack, 2009], Xbox 360 presented an integration with Facebook, Twitter and Last.fm for Xbox Live Gold subscribers. Initially these services were only available to users above the age of 18, however later Microsoft made the services available to all users with parental permissions.

Facebook and Twitter applications allowed the user very limited functionality when compared to their web counterparts, the users could only update their status, comment, like friends' posts and view friends' pictures (Figure 2, 1).

Additionally, users could use “Xbox Live Friend Finder” (Figure 2, 2) to see which Facebook friends used Xbox Live [Loguidice, and Loguidice, 2012].

The passive party system of GameNshare (Chapter 5 - Section 5.3.4) is based on Xbox Live Friend Finder, but goes beyond finding social network friends and allows players to engage friends in the asynchronous multiplayer game mechanics.

Figure 2 presents the main features of Xbox of the Xbox 360 Facebook Application.



Figure 2 - Xbox 360 Facebook Application³

For unclarified reasons, on 16th of October 2012 [Raby, 2012] the Facebook and Twitter applications were removed from Xbox Live and users were limited to the access of Facebook and Twitter through the Xbox 360 web browser.

2.5.2 Overwolf and Evolve: Social Overlays (2011)

In August 2011 [Takahashi, 2011], Overwolf Ltd. announced a social overlay for Windows desktop games, called Overwolf. It is a free software platform for Windows desktop games that includes a variety of social features inside the game environment that allow users to browse contents, share images and statuses on Facebook and Twitter, upload videos to YouTube and watch them and make Skype calls to friends. Players can also record games and stream their gameplay to Twitch.

Figure 3 presents an example of Overwolf integration with Allods Online [My.Com B.V, 2015].

³ <http://www.webpronews.com/facebook-and-twitter-head-to-the-xbox-2009-11/>

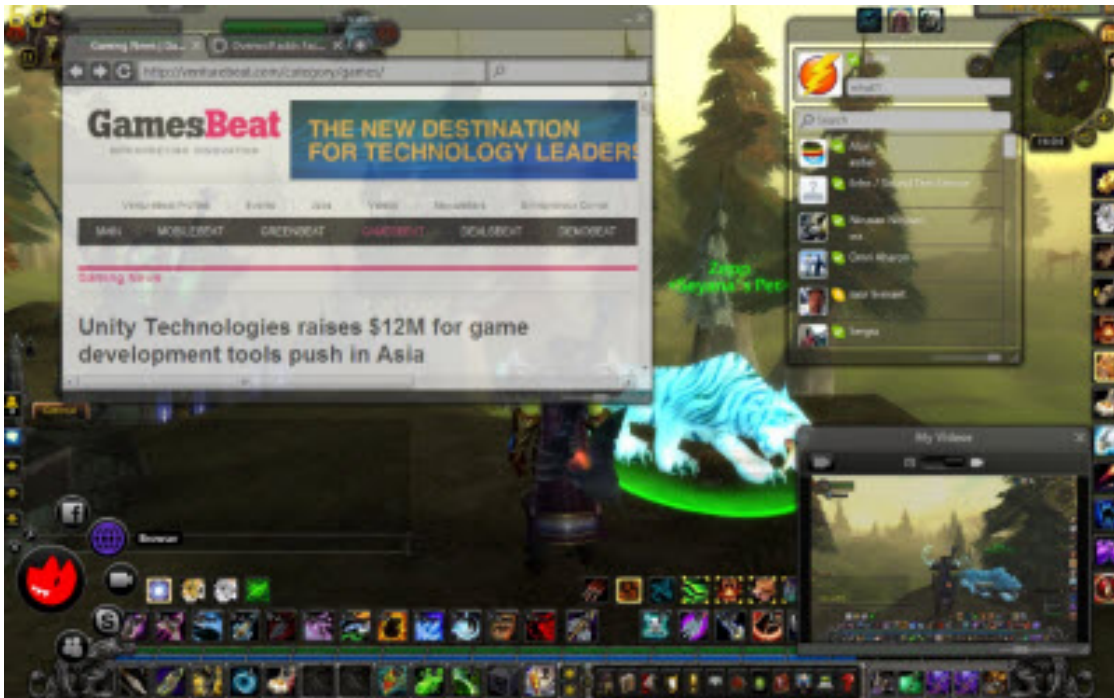


Figure 3 - Overwolf inside Allods Online⁴

Currently, Overwolf provides an AppStore [Overwolf, 2015a] with a wide range of free social network applications, however some applications are very limited comparatively to other social overlays, for example the chat does not allow setting a global status or detecting the games that friends are playing. This social overlay offers official support for more than 1000 different desktop games [Overwolf, 2015b], but its major limitation is only being available for Windows platforms. GameNshare provides support for the common share and screen capture actions, however to bypass the Overwolf compatibility limitation it was developed using Unity3D [Unity Technologies, 2015f], which offers support to Windows, Mac OSX and Linux.

Later in 2011, a company called Echobit released Evolve [K900 (Product Owner of Evolve), 2015]. Evolve is a social gaming platform for Windows desktop games that provides an easy access to a social overlay using a hotkey. Beyond the tools supported by the majority of the social overlays, such as browser in web, share contents, broadcast and chat with friends, this platform includes a new feature called “Party”. With this feature, players can search friends’ game statuses and create parties for multiple games at the same time. Evolve officially supports more than 4300 games, four social networks: Twitter, Facebook, YouTube and Twitch and four platform distribution services: Steam, Battle, Origin and Play Station Network [Echobit LLC, 2015]. The main limitation of Evolve besides the absence of a multiplatform version is the lack of APIs and SDKs to easily customize and integrate this tool in games. Similar to this overlay, GameNshare includes a “share” hotkey and allows integration of Twitter, Facebook and YouTube, which according to Ann Hurk [Hurk, 2013] are the most used social media platforms. Contrary to Evolve, GameNshare tools can be exported as Unity3D assets [Unity Technologies,

⁴ <http://venturebeat.com/2011/08/02/overwolf-launches-social-overlay-for-online-games/>

2015b] and integrated in different Unity3D development environments for desktop games with Windows, MAC OS X and Linux system.

2.5.3 Call of Duty: Elite and Facebook Integration (2011)

In 8th of November 2011 [Pereria, 2011], Activision a subsidiary of Beachhead Studios responsible for the worldwide famous game “Call of Duty” launched the service Call of Duty Elite. Call of Duty Elite was an online service for the multiplayer portion of the game present in titles such as, “Call of Duty: Modern Warfare 3”. This service contained some social networking capabilities like tracking scores, creating custom leader boards and creating groups of player friends. Additionally, Call of Duty Elite subscription also included the integration with Facebook accounts. This integration was a more complex version of the “Xbox Live Friend Finder” (Chapter 2 – Section 2.5.1), that allowed players to be notified when their Facebook friends were playing “Call of Duty” and also to communicate with friends that were on Facebook and not on native communication apps of the gaming devices, e.g. Xbox Live.

Figure 4 presents a screenshot of Facebook integration with the Call of Duty MW3 on multiplayer mode.

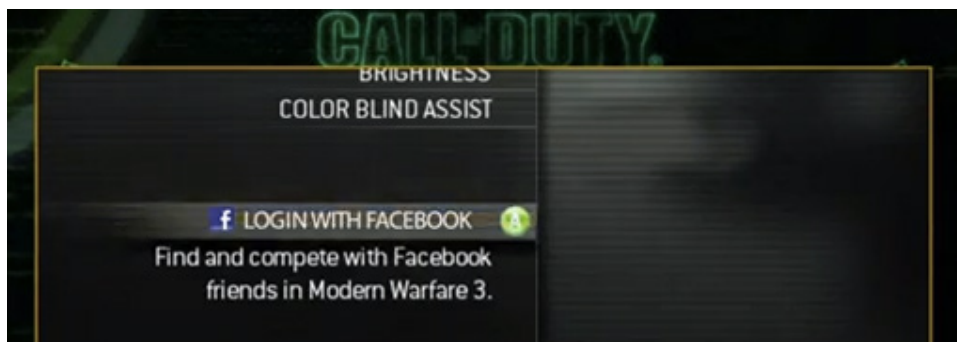


Figure 4 - Call of Duty: MW3 with Facebook integration⁵

Call of Duty Elite was shut down on 28th of February 2014, because according to Activision [Activision 2014]: “*Call of Duty Elite was designed to work with prior Call of Duty games and has begun winding down support for those titles since late last year, which ends on February 28, 2014. We’ve taken all the key leanings and fan feedback to create the Call of Duty app for Call of Duty®: Ghosts, which takes some of the best and most popular features of Call of Duty® Elite and introduces new elements such as Call of Duty Clan Wars*”. The Call of Duty: Ghosts mobile [Activision Publishing, 2014] application is a simplified version of the Call of Duty Elite services. In opposition to Call of Duty Elite, this application does not provide Facebook or other social network integration; instead it communicates directly with the Call of Duty: Ghosts multiplayer mode to allow players to compete and collaborate with rivals and friends.

Similarly to Call of Duty Elite, the passive party feature of GameNshare (Chapter 5, Section 5.3.4), lets players to collaborate with social network friends inside the game, but goes beyond real-

⁵ <http://www.ign.com/videos/2011/11/03/modern-warfare-3-on-facebook-elite-integration>

time play and introduces the concept of asynchronous multiplayer game mechanics. This allows gamers to cooperate with friends even if they are not currently playing the game.

2.5.4 Final Fantasy XIII-2: Outerworld Services (2012)

In 3th of February 2012, Square Enix announced the "Outerworld Services" to post story progress and party monsters (trophies) to Twitter or Facebook on Xbox 360 and PlayStation 3. Figure 5 presents a sample of the sharing of a "boss achievement" in the wall of the Facebook.

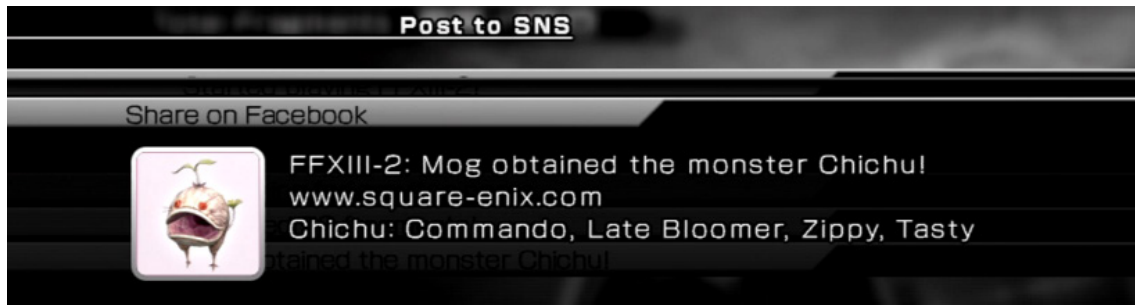


Figure 5 - Final Fantasy XIII-2 1.03 (Boss achievement)⁶

Outerworld Services is one of the ways of communication in "Lightning Returns: Final Fantasy XIII", with these services players can show their customized Lightning (main protagonist), location and character snapshots, bosses' Battle Scores and also send items for sale or buy items from another players using Facebook and Twitter accounts [Final Fantasy Wiki, 2014]. The items that were sold and exchanged with this system were neither removed from inventory nor allowed to receive game money. The major goal of this system is to help friends attaching the items to social network messages.

The concept of the Outerworld Services and in particular the sales system, was adopted in the conceptualization of the GameNshare's craft system (Chapter 3, Section 3.1.4) with a different reward system for players.

2.5.5 PS4: Share button (2013)

Sony took a decisive step forward in social media integration in their games by incorporating a "Share" button in the hardware of the PlayStation 4 (PS4) game controller [Kelion, 2013]. This button located in the front of the controller was revealed during the worldwide presentation of PS4 on 20th of February of 2013 [Kelion, 2013]. It allows players to easily share their gaming experiences directly to the PlayStation network, Facebook, Twitter, Twitch and USTREAM [IGN, 2014]. Besides this, players can also use the "Share play" feature. This feature allows gamers to share the screen and play with other friends in the same room. Figure 6 presents the share options of the PS4 sharing system.

⁶ <http://www.gamepur.com/news/6961-final-fantasy-xiii-2-patch-103-availale-now-includes-facebook-integration.html>

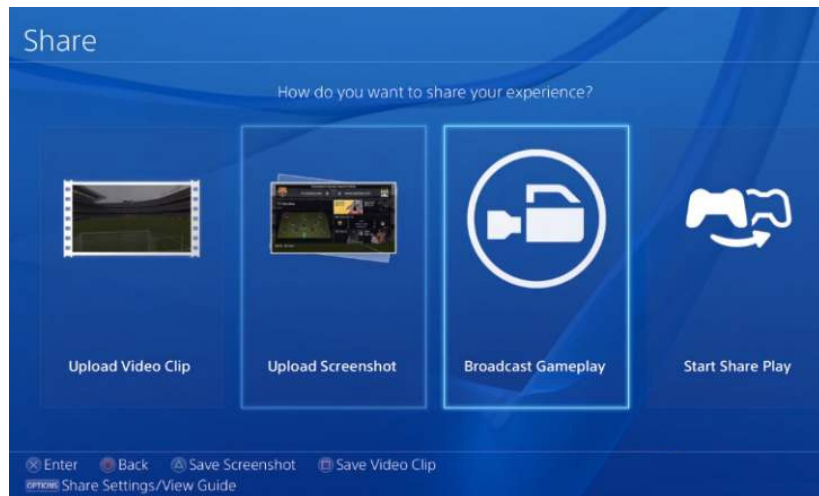


Figure 6 - Share Options of PS4⁷

While games like “Call of Duty” (Chapter 2 - Section 2.5.3) and Final Fantasy XIII-2 (Chapter 2 - Section 2.3.4) use special modes and menus to allow gamers to play, edit and save game experiences and to share them with others, the PS4 sharing function makes it even simpler by having a dedicated button in the controller (Figure 7).



Figure 7 - “Share” button PS4 controller⁸

This "Share" button is the first existing social feature of its kind and is analogous to the “share” hotkey of GameNshare (Chapter 5 – Section 5.3.2, Figure 46), however instead of only sharing gameplay experiences, a party feature integrating asynchronous game mechanics was also developed to include social networks friends in the game environment.

⁷ <http://blog.us.playstation.com/2014/10/24/check-out-share-play-in-action-on-ps4/>

⁸ <https://www.playstation.com/en-us/explore/accessories/dualshock-4-wireless-controller/>

3 Requirements Specification

The key goal of GameNshare is to use the social media communities to enhance the gameplay experience of the players. The involvement of communities and friends can promote sharing of game strategies and achievements among players and avoid eventual states of irritation or boredom in the player that may drive him lose interest in the game [Schell, 2014].

It is relevant to emphasize that the developed GameNshare’s features are entirely optional in the game. Players can use them in accordance with the circumstances to improve gameplay experience of APO [Clockwork Inc., 2015a]. Requirements specification for these features will be presented in two sections: functional requirements and non-functional requirements. The first section presents what the prototype should do and the second section describes how the prototype should work.

3.1 Functional requirements

According to their importance for GameNshare, the functional requirements were prioritized into three categories (essential, conditional and optional). Below is presented the prioritization scale of functional requirements, as well as their descriptions (Table 1).

Table 1 - Prioritization scale of functional requirements

Scale level	Description
Essential	Critical requirements, without them the product is not acceptable.
Conditional	These requirements would improve the product quality.
Optional	The requirements that would be nice to have, but are not essential to the product.

As GameNshare was integrated into the APO for testing proposes, the agreement on essential requirements was done through meetings with the game developing team [Clockwork Inc., 2015a]. The prioritizing of conditional and optional requirements was based on the opinions of a group of GameNshare and APO testers. These opinions were collected using the alpha version survey with a Likert scale (Annex B, section 4, 6 and 7). The group of testers was composed by

10 individuals (8 boys and 2 girls) with ages between 19 and 25 that were not involved in the development of GameNshare or APO and were externally recruited for the preliminary testing phases. The invitation form for testers as well as the alpha version survey can be consulted in Annexes A and B, respectively. This survey followed the Qualtrics recommendations for the use of two concept testing methods: (1) new product concept tests; (2) product modification or upgrade tests [Qualtrics, 2015]. Qualtrics is a private research software company that has one of the world's biggest enterprise survey platform [Bort, 2014].

The requirements with an average score in the survey between 2.5 and 5 (in a scale of 1 to 5) were considered "conditional requirements". Contrarily, all the requirements with average scores between 1 and 2.5 were considered "optional requirements".

The opinions of the group of testers about requirements related with "Broadcast Gameplay" had an average score of 3.5 (Figure 8, orange). The "Craft" requirements, their opinions had an average score of 4.1 (Figure 8, blue). Finally, the opinions about requirements associated with the "Passive party system" had an average result of 2.1 (Figure 8, grey).

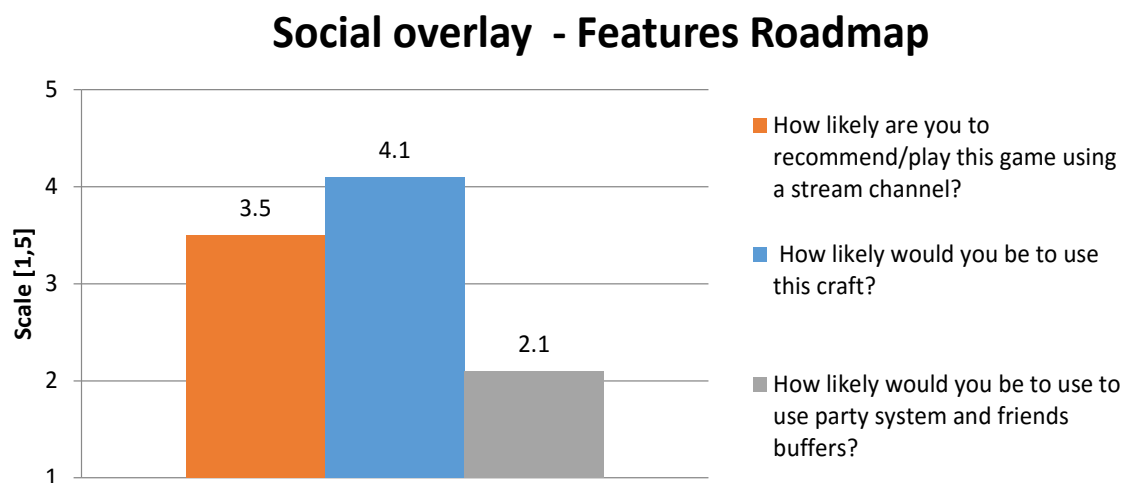


Figure 8 - Average of test team opinions (scale of 1 to 5) about broadcast gameplay (orange), craft system (blue) and passive party system (grey)

The result of the prioritization of use case-based requirements can be seen in diagram below (Figure 9). The next sections (Chapter 3 - Sections 3.1.1 - 3.1.5) are dedicated to the presentation of these requirements.

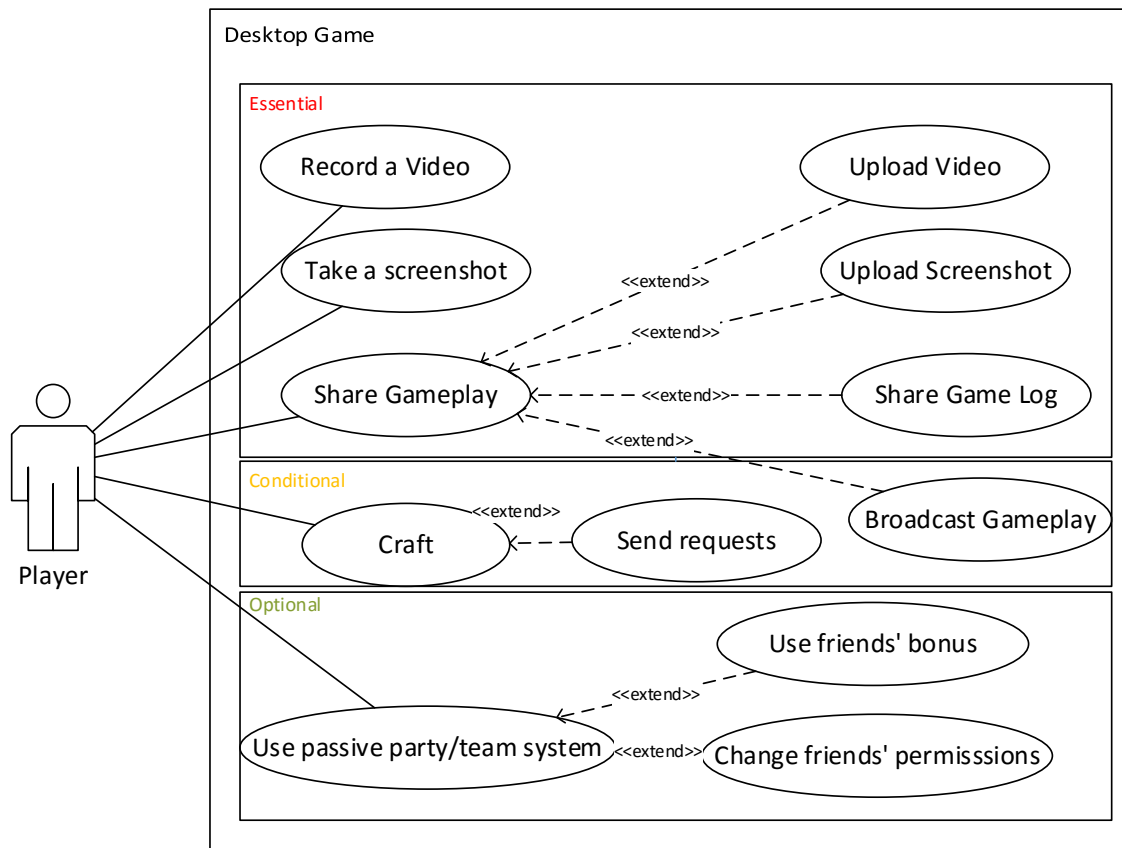


Figure 9 - Use case diagram of social overlay prototype with prioritization scale

3.1.1 Record a Video

Table 2 presents a brief description of the “Record Video” use case.

Table 2 - “Record Video” use case brief description

Title of fields	Description of fields
Name	Record a video.
Summary	The player records a gameplay video.
Actor	Player.
Classification	Essential

The “Record a video” use case classified as essential (Table 2), allows players to record short videos (up to 15 minutes) during the game.

Players can record using the record shortcut or the print screen button of the heads-up display (HUD) menu. Table 3 presents the flow-of-events of the “Record a video” use case.

Table 3 - "Record Video" use case flow-of-events

Flow	Description
Main flow	<ol style="list-style-type: none"> 1. This use case starts when the player presses the record button or shortcut. The system displays a "rec" icon to notify the player that the video recording has started. 2. The player stops recording using the record button or the shortcut. The system hides the "rec" icon to notify the player that the video recording ended. 3. The system displays a success message when the video is rendered. The use case ends.
Alternative flow (starts after the main flow - step 1)	<ol style="list-style-type: none"> 1. The player tries to record a new video when a video is already being recorded or rendered. The system displays an error message. The use case ends.

3.1.2 Take a screenshot

Table 4 provides a brief description of "Take a screenshot" use case.

Table 4 - "Take a screenshot" use case brief description

Title of fields	Description of fields
Name	Take a screenshot.
Summary	The player captures a screenshot.
Actor	Player.
Classification	Essential.

The "Take a screenshot" use case, classified as essential (Table 4), allows players to take screenshots of their game experiences. Players can take screenshots using the print screen shortcut or the print screen button of the HUD menu.

Table 5 presents the flow-of-events of the "Take a screenshot" use case.

Table 5 - "Take a screenshot" use case flow-of-events

Flow	Description
Main flow	<ol style="list-style-type: none"> 1. This use case starts when the player presses the print screen button or shortcut. 2. The system displays a success message when the screenshot is rendered. The use case ends.
Alternative flow (starts after the main flow - step 1)	<ol style="list-style-type: none"> 1. The system failed to render a screenshot for unidentified reasons, for example, for the lack of permissions to access to screenshot folder. The system displays an error message. The use case ends.

3.1.3 Share Gameplay

Figure 10 presents a use case diagram with the interactions between players and social network users when the gameplay is shared.

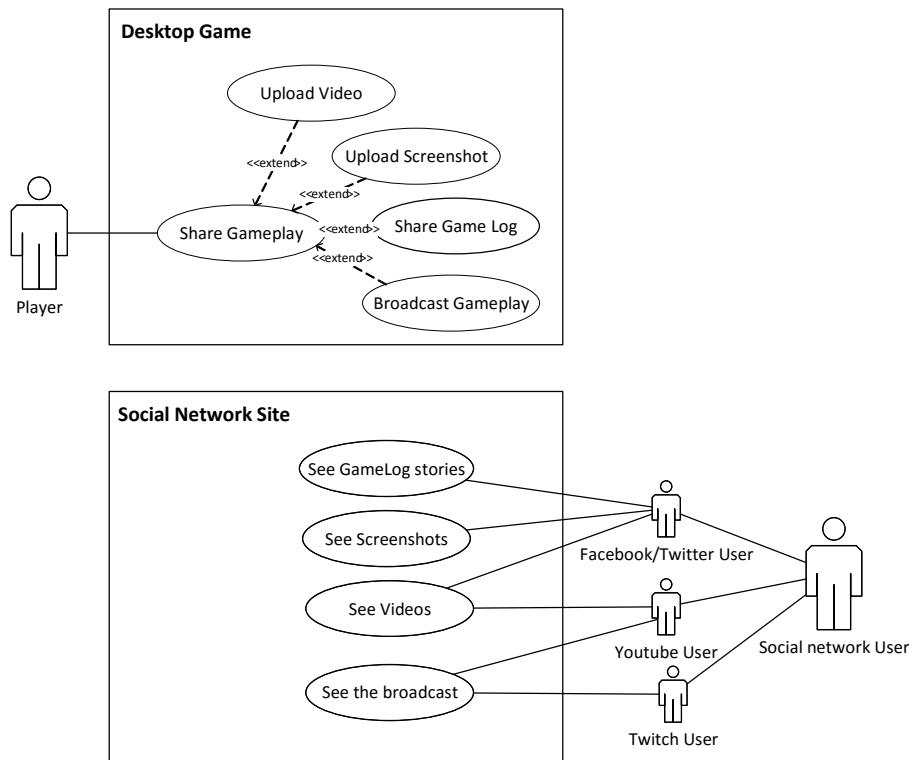


Figure 10 - Use case diagram of share gameplay

Players can share on multiple social networks screenshots, videos and game-related stories and stream their games online. Table 6 presents a summary of "Share Gameplay" use case.

Table 6 - "Share Gameplay" use case brief description

Title of fields	Description of fields
Name	Share Gameplay.
Summary	The player shares a game experience.
Actor	Player.
Classification	Essential.

The use case "Share Gameplay" classified as essential (Table 6), resulted from the "Share Menu" feature.

This menu is displayed by clicking in the social button from the game options menu or from the Share Menu shortcut.

It has four options: (1) "Upload Video"; (2) "Upload Screenshot"; (3) "Share Game Log"; (4) "Broadcast Gameplay". Table 7 below presents the flow-of-events of the "Share Gameplay" use case.

Table 7 - "Share Gameplay" use case flow-of-events

Title of fields	Description of fields
Main flow	<ol style="list-style-type: none"> 1. This use case starts when the player clicks in the "social button" from the game options menu or from the "Share Menu" shortcut. The system displays the Share Menu with four options. 2. The system determines that the internet connection is available and shows a green Wi-Fi icon to notify the player. 3. The system determines that the player has a valid login session in a social network and shows a logout button of this social network. The use case ends.
Alternative flow 1 (starts after the main flow - step 1)	<ol style="list-style-type: none"> 1. The system determines that the internet connection is unavailable and shows a red Wi-Fi icon to notify the player. The use case ends.
Alternative flow 2 (starts after the main flow - step 2)	<ol style="list-style-type: none"> 1. The system determines that user has no valid login session in a social networks and does not shows a logout buttons. The use case ends.
Extension points	<ol style="list-style-type: none"> 1. Upload Video; 2. Upload Screenshot; 3. Share Game Log; 4. Broadcast Gameplay.

3.1.3.1 Upload Video

Table 8 summarizes the “Upload Video” use case.

Table 8 - “Upload Video” use case brief description

Title of fields	Description of fields
Name	Upload Video.
Summary	The player uploads a video to a social network.
Actor	Player.
Classification	Essential.

With “Upload Video”, classified as essential (Table 8), players can select and upload gameplay videos to social networks.

Table 9 describes the main flow of “Upload Video” use case.

Table 9 - “Upload Video” use case main flow

Flow	Description
Main flow	<ol style="list-style-type: none"> 1. This use case starts when the player clicks in the “upload video button” from the “Share Menu”. The system displays a list with the player’s videos. 2. The player selects a video from the list. The system shows a pre-visualization of the upload. 3. The player changes the video title and enters a video description. 4. The player selects one or more social networks. 5. The player clicks in the “share button”. 6. The system determines that internet connection is available. 7. The system determines that the login sessions of the social networks are valid. 8. The system determines that the video was successfully shared and displays a success message. The use case ends.
Alternative flow (starts after the main flow - step 2)	<ol style="list-style-type: none"> 1. The player does not change video title and enters a video description. The use case jumps to step 4.

3.1.3.2 Upload Screenshot

Table 10 presents a brief description of the “Upload Screenshot” use case.

Table 10 - “Upload Screenshot” use case brief description

Title of fields	Description of fields
Name	Upload Screenshot.
Summary	The player uploads a screenshot to a social network.
Actor	Player.
Classification	Essential.

The “Upload Screenshot” use case, classified as essential (Table 10), allows the upload of player-selected screenshots to social networks. Table 11 contains the description of the “Upload Screenshot” use case main flow.

Table 11 - “Upload Screenshot” use case main flow

Flow	Description
Main flow	<ol style="list-style-type: none"> 1. This use case starts when the player clicks in the “upload screenshot” button from the Share Menu. The system displays a list with the player’s screenshots. 2. The player selects a screenshot from the list. The system shows a pre-visualization of the upload. 3. The player enters a screenshot description. 4. The player selects one or more social networks. 5. The player clicks in the “share button”. 6. The system determines that the internet connection is available. 7. The system determines that the login sessions of the social networks are valid. 8. The system determines that the screenshot was successfully shared and displays a success message. The use case ends.
Alternative flow (starts after the main flow - step 2)	The player does not change video title and enters a video description. The use case jumps to step 4.

3.1.3.3 Share Game Log

Table 12 presents a brief description of the “Share Game Log” use case.

Table 12 - “Share Game Log” use case brief description

Title of fields	Description of fields
Name	Share Game Log.
Summary	The player shares a game story to a social network.
Actor	Player.
Classification	Essential.

The “Share Game Log” use case, classified as essential (Table 12), resulted from the “share game log menu” feature and is dedicated to the sharing of achievements, game stories and other information in the social networks. Additionally, inside this menu, players can create their own stories and share them in the social networks. The “add story” alternative flow was added to the requirements in the beta version, it was suggested by 9% of the testers in the alpha testing. The results of alpha testing can be consulted in the section 5.2.1 of this dissertation.

Table 13 describes the main flow and the “add story” alternative flow of the “Share Game Log” use case.

Table 13 - “Share Game Log” main flow and “add story” alternative flow

Flow	Description
Main flow	<ol style="list-style-type: none"> 1. This use case starts when the player clicks in the “share game log button” from the “Share Menu”. The system displays a list with the player’s stories. 2. The player selects a story from the list. The system shows a pre-visualization of the share. 3. The player enters a story description. 4. The player selects one or more social networks. 5. The player clicks in the share button. 6. The system determines that internet connection is available. 7. The system determines that the login sessions of the social networks are valid. 8. The system determines that the story was successfully shared and displays a success message. The use case ends.
Alternative flow 1 (starts after the main flow - step 1)	<ol style="list-style-type: none"> 1. The player clicks in the “add story” button. The system shows a form to add a new story. 2. The player writes a new story. The system adds the story to the game log. The use case jumps to main flow - step 2.

3.1.3.4 Shared Alternative Flows

“Upload Video Clip”, “Upload Screenshot” and “Share Game Log” use cases share the same four alternative flows: (1) no social network selected; (2) internet connection unable; (3) invalid login session; (4) duplicate post error; (5) unsuccessful error.

Table 14 and Table 15 present the description of these alternative flows.

Table 14 - Alternative flows 1 and 2 of the essential “Share Gameplay” use cases

Flow	Description
Alternative flow 1 (starts after the main flow - step 3)	1. The player does not select any social network and tries to share the post. The system displays an error message. The use case jumps to main flow - step 3.
Alternative flow 2 (starts after the main flow - step 6)	1. The system determines that internet connection is unable. The system displays an error message. The use case ends.

Table 15 - Alternative flows 3, 4 and 5 of the essential “Share Gameplay” use cases

Flow	Description
Alternative flow 3 (starts after the main flow - step 7)	<ol style="list-style-type: none"> 1. The system determines that the login sessions of the social networks are invalid. 2. The system shows a browser instance with a login windows. 3. The player enters a correct username and password. 4. The system shows a browser instance asking for the applications permissions. 5. The player accepts the applications permissions. The use case jumps to main flow - step 8.
Alternative flow 4 (starts after the main flow - step 8)	<ol style="list-style-type: none"> 1. The player shares a duplicate post of a previously shared. The system shows an alert message. 2. The player refuses to share again. The use case ends. 3. The player wants to share again. The use case jumps to main flow - step 8.
Alternative flow 5 (starts after the main flow - step 8)	1. The system determines that the post was unsuccessfully shared to the social networks. The system displays an error message. The use case ends.

3.1.3.5 Broadcast Gameplay

Table 16 presents a brief description of the “Broadcast Gameplay” use case.

Table 16 - “Broadcast Gameplay” use case brief description

Title of fields	Description of fields
Name	Broadcast Gameplay
Summary	The player streams gameplays to the social networks.
Actor	Player
Classification	Conditional

The “Broadcast Gameplay” use case allows live social network broadcasting (streaming) of gameplays. This use case was classified as conditional (Table 16) and resulted from the “broadcast gameplay menu” feature.

Table 17 presents the main flow of the “Broadcast gameplay” use case.

Table 17 - “Broadcast Gameplay” use case main flow

Flow	Description
Main flow	<ol style="list-style-type: none">1. This use case starts when the player clicks in the “broadcast gameplay” from the “Share Menu”. The system displays a list with social networks available.2. The player selects a social network. The system displays the configuration menu.3. The player configures the stream settings and clicks in the “start broadcasting” button. The use case ends.

3.1.4 Craft

Figure 11 presents a use case diagram with the interactions between players and social network friends in the craft.

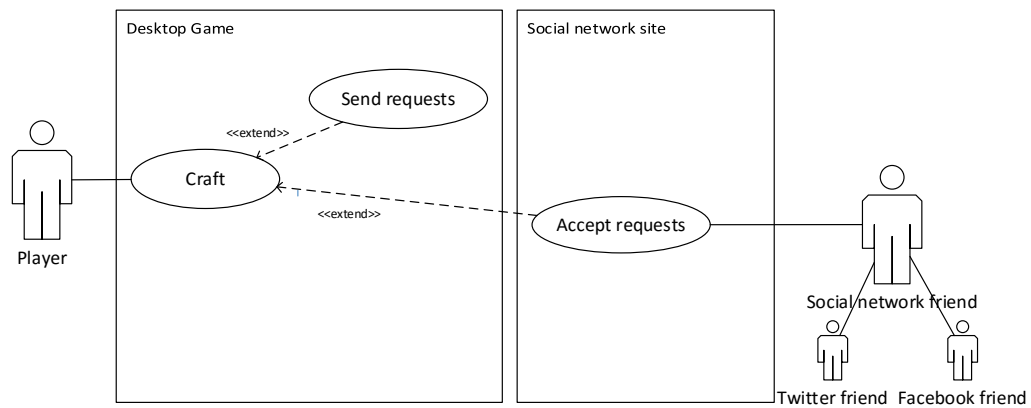


Figure 11 - Use case diagram of "Craft"

Players can ask social networks friends that also play the game for help to complete missions, quests and collections. Table 18 presents a brief description of the "Craft" use case.

Table 18 - "Craft" use case brief description

Title of fields	Description of fields
Name	Craft
Summary	The player send or accept a craft request.
Actor	Player
Classification	Optional

The "Craft" use case was classified as a conditional (Table 18). It enables players to send and accept social network friends' requests, asking for missing game elements to create new game items or to complete collections, quests or missions. "Craft" is only available through the "craft" shortcut or button in the HUD menu. Table 19 describes the "Craft" use case.

Table 19 - "Craft" use case flow-of-events

Flow	Description
Main flow	<ol style="list-style-type: none"> 1. This use case starts when the player clicks in the "craft" button from the HUD menu or in the "craft" shortcut. The system shows a list of social networks that can be synced. 2. The player chooses to synchronize friends of one or more social networks. 3. The system determines that the internet connection is available. 4. The system checks if the user already has a valid login session in that social networks and shows a logout button of these social networks. 5. The system shows a list social networks friends that play with the option: "Send a request".
Alternative flow 1 (starts after the main flow - step 2)	<ol style="list-style-type: none"> 1. The system determines that the internet connection is unable and shows an error message. The use case ends.
Alternative flow 2 (starts after the main flow - step 3)	<ol style="list-style-type: none"> 1. The system determines that the login sessions of the social networks are invalid. 2. The system shows a browser instance with a login windows. 3. The player enters a correct username and password. 4. The system shows a browser instance asking for the applications permissions. 5. The player accepts the applications permissions. The use case jumps to main flow - step 4.
Extension points	<ol style="list-style-type: none"> 1. Accept requests. 2. Send requests.

3.1.4.1 Accept requests

Table 20 presents a brief description of the "Accept requests" use case.

Table 20 - "Accept requests" use case brief description

Title of fields	Description of fields
Name	Accept requests
Summary	The player accept a craft request of a friend.
Actor	Player
Classification	Conditional

The “Accept requests” use case was classified as conditional (Table 20). In this use case players can accept friends’ craft requests through social network private messages. Table 21 presents a description of the flow-of-events of the “Accept requests” use case.

Table 21 - “Accept requests” use case main flow and alternative flow

Flow	Description
Main flow	<ol style="list-style-type: none"> 1. This use case starts when the player sees his private messages and opens the craft request link. The system shows a previsualization of the craft request. 2. The player chooses to accept the craft request. 3. The system determines that the player has a valid social network session. 4. The system determines that the player can accept that craft request. 5. The system exchanges the item of the player with friend. The system displays a success message. The use case ends.
Alternative flow 1 (starts after the main flow - step 2)	<ol style="list-style-type: none"> 1. Player does not have the request item. The system displays an error message. The use case ends.
Alternative flow 2 (starts after the main flow - step 2)	<ol style="list-style-type: none"> 1. The system determines that the player has an invalid social network session. 2. The system shows a browser instance with login form. 3. The player enters a correct username and password. 4. The system shows a browser instance asking for the applications permissions. 5. The player accepts the applications permissions. The use case jumps to main flow - step 4.
Alternative flow 3 (starts after the main flow - step 3)	<ol style="list-style-type: none"> 1. The system failed to exchange the item with the friend. The system displays an error message. The use case ends.

3.1.4.2 Sends requests

Table 22 presents a brief description of the “Send requests” use case.

Table 22 - “Send requests” use case brief description

Title of fields	Description of fields
Name	Send requests
Summary	The player send craft request to a friend.
Actor	Player
Classification	Conditional

The “Send requests” use case, classified as conditional (Table 22), allows players to send a private messages to social network friends asking for specific game elements. Table 23 presents the flow-of-events of the “Send requests” use case.

Table 23 - “Send requests” use case flow-of-events

Flow	Description
Main flow	<ol style="list-style-type: none">1. This use case starts when the player clicks in the “Send a request” button of a social network friend. The system shows a previsualization of the craft request.2. The player clicks in the “send” button. The system sends a private message to the social network friend with the craft request.3. The system determines that the private message was effectively sent and shows success message. The use case ends.
Alternative flow (starts after the main flow - step 2)	<ol style="list-style-type: none">4. The system determines that the private message was unsuccessfully sent to the social networks. The system displays an error message. The use case ends.

3.1.5 Passive party system

Figure 12 presents a use case diagram with the interactions between players and social network friends in the passive party system.

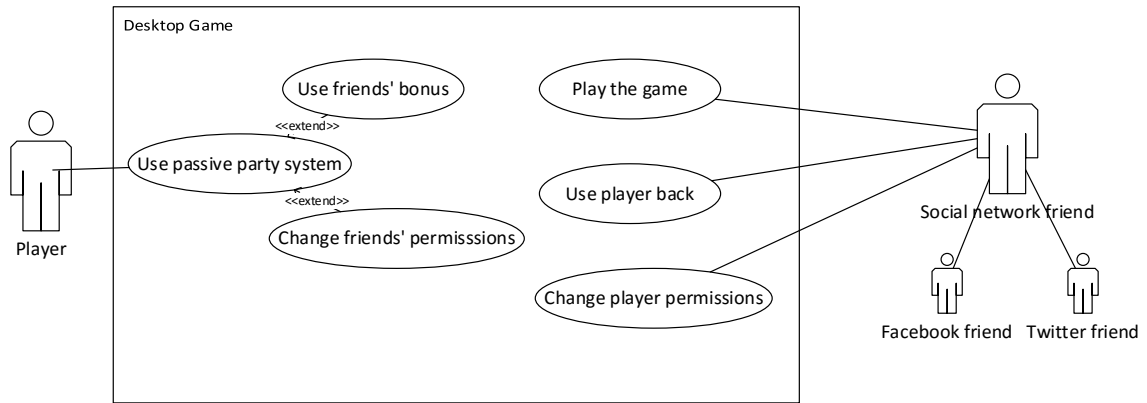


Figure 12 - Use case diagram of "Party"

Players can make teams with their social networks friends that play the game in game-related experiences, e.g. combat.

Table 24 - "Passive party/team system" use case brief description

Title of fields	Description of fields
Name	Passive party/team system
Summary	The player use friends' bonus or change friends' permississions.
Actor	Player
Classification	Optional

The passive party/team system classified as an optional requirement (Table 24), allows social network friends to act as passive members by performing actions even when not playing the game. These actions result in different game bonus, for example exclusive attacks, that can be used to improve game-related experiences, e.g. combats. Similar to craft, this system is only available in the HUD menu or through the "party" shortcut.

Table 25 presents the flow-of-events of the "Passive party/team system" use case.

Table 25 - "Passive party/team system" use case flow-of-events

Flow	Description
Main flow	<ol style="list-style-type: none"> 1. This use case starts when the player clicks in the "party" button from the HUD menu or in "party" shortcut. The system shows a list of social networks that can be synced. 2. The player chooses to synchronize friends of one or more social networks. 3. The system determines that the internet connection is available. 4. The system checks if the user already has a valid login session in that social networks and shows a logout button of these social networks to notify user about that. 5. The system shows a list social networks friends that play with two options in each friend: "Select" and "Friend permissions". The use case ends.
Alternative flow 1 (starts after the main flow - step 2)	<ol style="list-style-type: none"> 1. The system determines that the internet connection is unable and shows an error message. The use case ends.
Alternative flow 2 (starts after the main flow - step 3)	<ol style="list-style-type: none"> 1. The system determines that the login sessions of the social networks are invalid. 2. The system shows a browser instance with a login windows. 3. The player enters a correct username and password. 4. The system shows a browser instance asking for the applications permissions. The player accepts the applications permissions. The use case jumps to main flow - step 4.
Alternative flow 3 (starts after the main flow - step 5)	<ol style="list-style-type: none"> 1. The player clicks in the refresh button. The use case jumps to main flow - step 3.
Alternative flow 3 (starts after the main flow - step 6)	<ol style="list-style-type: none"> 1. The player clicks in the configuration button. The use case jumps to main flow - step 1.
Extension points	<ol style="list-style-type: none"> 1. Use friends' bonus. 2. Change friends' permissions.

3.1.5.1 Use friends' bonus

Table 26 summarize the "Use friends' bonus" use case.

Table 26 - "Use friends' bonus" use case brief description

Title of fields	Description of fields
Name	Use friends' bonus
Summary	The player use the friend's bonus.
Actor	Player
Classification	Optional

The "Use friends' bonus" use case classified as optional (Table 26), let players to use the social network friends to improve the game experiences, e.g. combat. Players can only use a friend bonus each time in game experiences that allow the use of passive party system. Table 26 summarize the flow-of-events of the "Use friends' bonus" use case.

Table 27 - "Use friends' bonus" use case flow-of-events

Flow	Description
Main flow	<ol style="list-style-type: none"> 1. This use case starts when the player clicks in the "select" button of a social network friend. 2. The system determines that the friend can be successful used and save it in the party list. 3. The system shows a success message to notify player that the friend can be used and changes the status of the relationship player-friend to enable. 4. The player enters in a party system's game experience and chooses to use the friend bonus. 5. The system deletes the friend from the party list. 6. The system changes the status of the relationship friend-player to disable. The use case ends.
Alternative flow 1 (starts after the main flow - step 1)	<ol style="list-style-type: none"> 1. The player tries to use two friends at the same time. The system shows an error message. The use case ends.
Alternative flow 2 (starts after the main flow - step 1)	<ol style="list-style-type: none"> 1. The system determines that the friend cannot be used and shows an error message. The use case ends.
Alternative flow 3 (starts after the main flow - step 5)	<ol style="list-style-type: none"> 1. The system determines that the player can use the friend for an unlimited number of times. 2. The system changes the status of relationship friend-player to enable. The use case ends.

3.1.5.2 Change friends' permissions

Table 28 presents a brief description of "Change friends' permissions" use case.

Table 28 - "Change friends' permissions" use case brief description

Title of fields	Description of fields
Name	Change friends' permissions
Summary	The player changes the friend's permissions.
Actor	Player
Classification	Optional

The "Change friends' permissions" use case, classified as optional (Table 28), allows the player to change the permissions of their friends. By default, players' friends can only be used one time and then they have to use the friend back. In addition, players can give "free passes" to let friends use them for an unlimited number of times. Table 29 presents the description of flow-of-events of the "Change friends' permissions" use case.

Table 29 - "Change friends' permissions" use case flow-of-events

Flow	Description
Main flow	<ol style="list-style-type: none"> 1. This use case starts when the player clicks in the "friend permissions" button of a social network friend. The system shows the list of friend's permissions. 2. The player changes the friend's permissions. 3. The system determines that the friend's permissions were successful changed and shows a success message. The use case ends.
Alternative flow 1 (starts after the main flow - step 1)	<ol style="list-style-type: none"> 1. The player clicks in the exit hotkey (ESQ) or clicks in friend permission button. The system closes the list of friend's permissions. The use case ends.
Alternative flow 1 (starts after the main flow - step 2)	<ol style="list-style-type: none"> 1. The system determines that the friend's permissions were unsuccessful changed and shows an error message. The use case ends.

3.2 Non-functional requirements

Among the different non-functional requirements of GameNshare the following can be highlighted: (1) support APO integration and full compatibility with Unity3D [21] for desktop games; (2) full Windows compatibility and when possible multiplatform support (Windows, Mac OS X and Linux).

The remaining part of this section presents a detailed description of the non-functional requirements needed to achieve the project's goals. The requirements were divided into four categories: (1) hardware and software requirements; (2) performance requirements; (3) security requirements.

3.2.1 Hardware and Software requirements

The GameNshare game client side must be compatible with APO Game and with almost all desktop environments [Unity Technologies, 2015e]. Regarding to the web instances of GameNshare they must be compatible with IE, Google Chrome, Firefox and Safari. Table 30 presents the minimum system requirements of GameNshare.

Table 30 - GameNshare system requirements

Attribute	Specifications
Operating system	Windows XP+ (mandatory), Mac OS X 10.7+ (optional), Ubuntu 12.04+ (optional), SteamOS Beta+ (optional).
Graphics card	Graphics cards with DX9 (shader model 2.0) capabilities.
Central Processing Unit (CPU)	Processors with support for SSE2 instruction set.
Communication	Internet connection (Ethernet or Wi-Fi connection).
Browser Support	IE, Google Chrome, Firefox and Safari.
Social networks Support	Facebook, Twitter, Google (YouTube) and Twitter ⁹ .

In what regards to programming languages, GameNshare client side must be written in a coding language compatible with Unity modified version of Mono 2.6 [Mono, 2009].

3.2.2 Performance requirements

The speed of communication between the game, web services and the social networks must be adequate. Players should expect a reasonable response time after a social network login, typically not exceeding than 15 seconds. The response time of social networks screenshots, videos and story must not be greater than 3 minutes. Depending on the size of the files and stories, greater response times are allowed.

⁹ Although Twitch is not currently being supported by GameNshare, it was conceptualized as a software requirement.

Depending on the number of friends, the response time to synchronize social networks friends must be reasonable, usually not greater than 30 seconds.

3.2.3 Security requirements

The server on which the web services and database reside must have their own security measures to prevent unauthorized read and write accesses. The players' PCs on which game client application resides does not require any special built-in protection.

GameNshare needs to provide a privacy policy page to its players. The use of social network usernames and passwords by the player will be done on the social network systems and thus is external to GameNshare. However, to ensure a greater security and confidentiality of the user public information in the database, a secret key of the game client application must be created. This key must be encrypted and sent in all the web services calls that request or send database information.

4 System Design and Architecture

GameNshare system architecture can be divided into three components. The first one is the desktop game client which includes the desktop game application and the social overlay tools. The second component is the web server. The web server includes the web services that receive and send information to the social overlay and the database that is used to store the information about the players' requests and social network friends' relationships. Lastly, the third component are the external social networks which include all the public media services.

Figure 13 presents the system architecture of GameNshare.

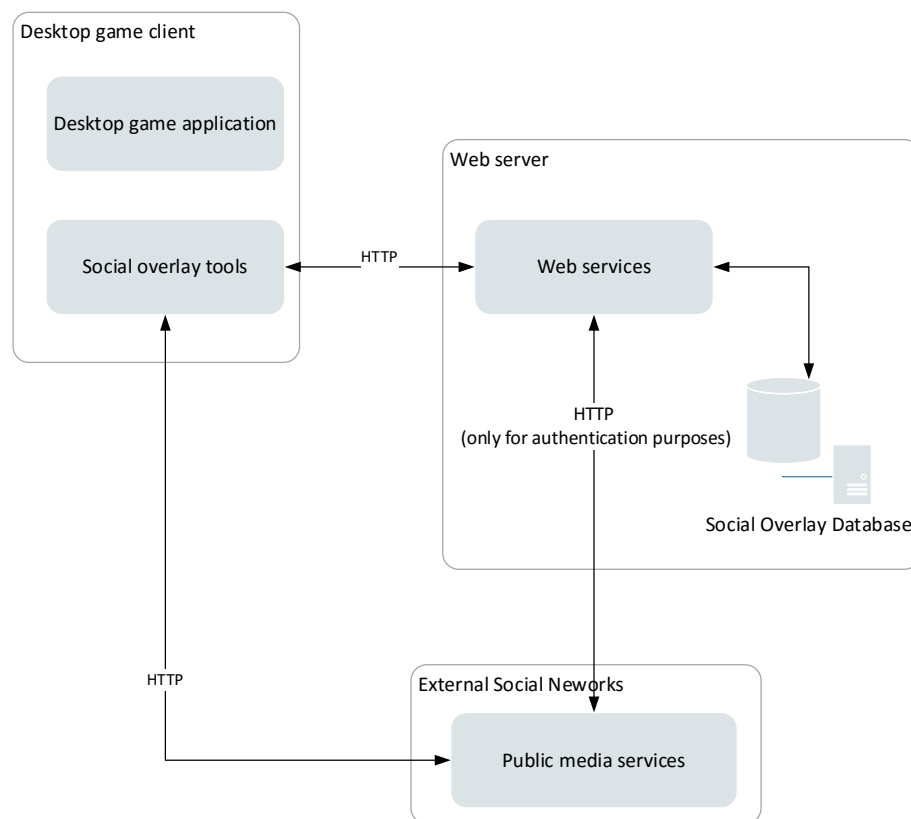


Figure 13 - System architecture of GameNshare

This chapter describes the design and architecture of GameNshare based on the requirements that were proposed. The next sections provide a detailed description of each of the components.

4.1 Desktop game client

Desktop game client is composed by two components. First one is the desktop game application, which is the part of desktop game client that contains the game's core logic and mechanics. The second one are the social overlay tools, which include all GameNshare features. This section presents a description of the design and architecture of desktop game client with emphasis in two topics: (1) GameNshare integration with Desktop Game Application (APO); (2) Social overlay tools.

4.1.1 GameNshare integration with Desktop Game Application (APO)

Figure 14 presents a simplified navigation chart that includes integration of share, record, screenshot capture, craft, party system tools with the desktop game application APO. This game flow can be adapted in accordance to different game environments to create specific game tools or to customize the existing ones.

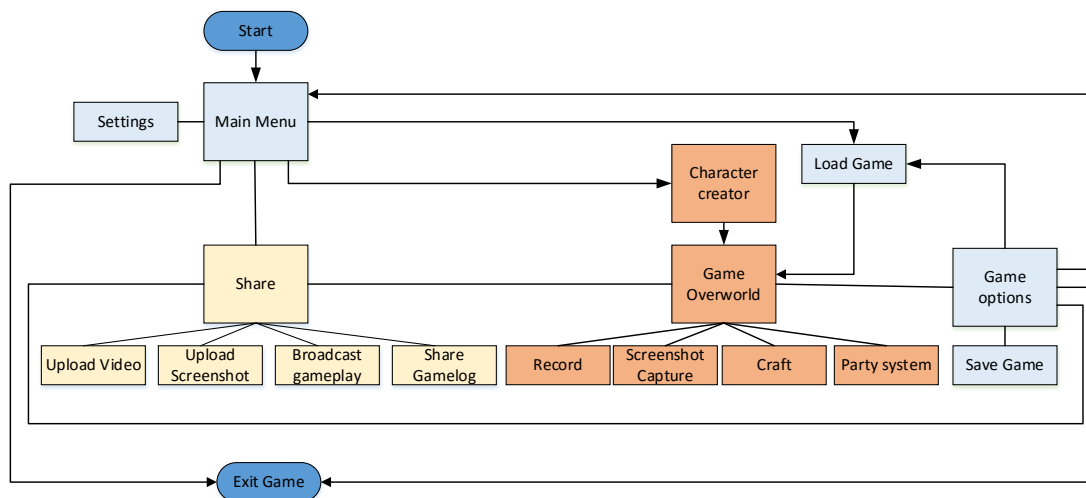


Figure 14 - Navigation chart/Game flow of GameNshare integration with APO

Figure 15 presents a simplified class diagram of the integration of “Share Menu” with APO.

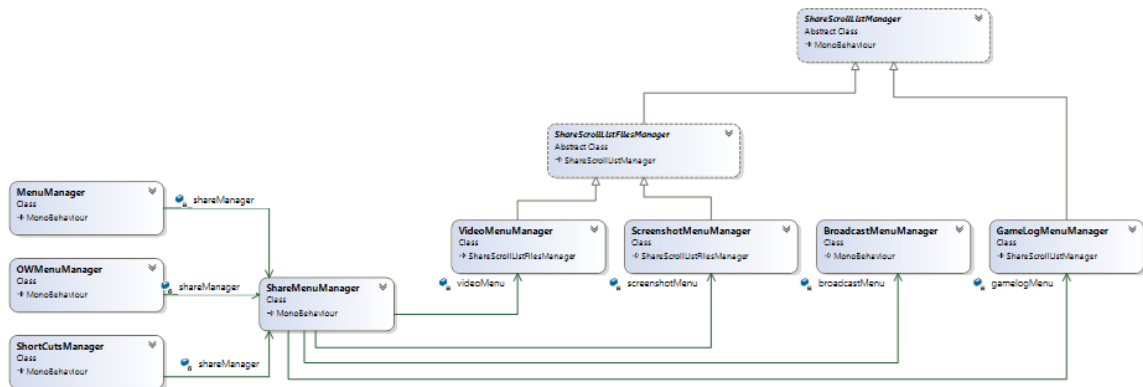


Figure 15 - Class diagram of Share Menu integration with APO

“Share Menu” (Figure 15, ShareMenuManager) includes three scroll list menus (Figure 15, ShareScrollListManager) for the sharing of videos, screenshots and game-related stories and a menu for the broadcasting of the gameplay (Figure 15, BroadcastMenuManager).

In addition, videos and screenshot scroll list menu support the previewing of the files with thumbnails (Figure 15, ShareFileScrollListManager).

Players can access to the “Share Menu” by clicking in the share option of the main menu (Figure 15, MenuManager) or game option menus (Figure 15, OWMenuManager) or directly using the “Share Menu” hotkey (Figure 15, ShortCutsManager).

Figure 16 presents a simplified class diagram of the integration of capture screen tools with APO.

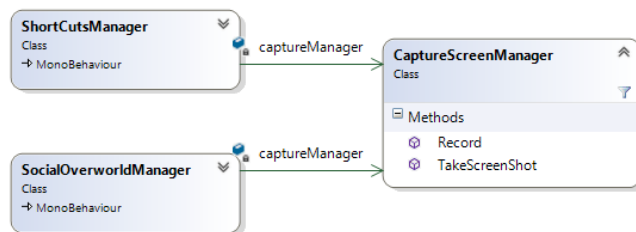


Figure 16 - Class diagram of capture screen integration with APO

Record and screenshot capture tools (Figure 16, CaptureScreenManager) allow players to do video record and screenshot capture at any moment of the game using hotkeys (Figure 16, ShortCutsManager) or through the record and capture buttons in the overworld scene (Figure 16, SocialOverworldManager), which is the area that interconnects all the game locations, e.g. combat areas.

Figure 17 presents a simplified class diagram of the integration of capture screen tools with APO.

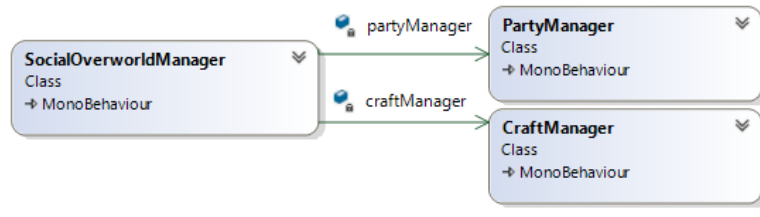


Figure 17 - Class diagram of craft and party integration with APO

Finally, the craft (Figure 17, CraftManager) and the party (Figure 17, PartyManager) tools are only accessible inside the game overworld (Figure 13, SocialOverworldManager). With the craft, players can exchange items with social network friends' in order to complete missions, quests or collections. With passive system, players can use their social network friends' as bonus in different game-related experiences, e.g. combats.

All the scripts that need to be attached to Game Objects [Unity Technologies, 2015c] such as GameNshare's managers derive from the base class MonoBehaviour [Unity Technologies, 2015d]. MonoBehaviour contains all the common actions of the Game Objects in Unity Environments. To prevent the creation of more than one manager in the game scenes, all the managers implement the Singleton pattern [Skeet, 2013].

4.1.2 Social overlay tools

The social overlay tools are all the features of GameNshare's client side, which include share gameplay, record, screenshot capture, craft and party. All these tools are Unity3D Assets [Unity Technologies, 2015b] for desktop games and are written in C#.

Share gameplay, craft and party tools include support for read and write data in the public media services, e.g. Facebook Graph API (Figure 18, FacebookManager), Twitter API (Figure 18, TwitterManager) and Google APIs (Figure 18, GoogleManager). Figure 18 presents the simplified class diagram of the integration of public media services in the GameNshare's client side.

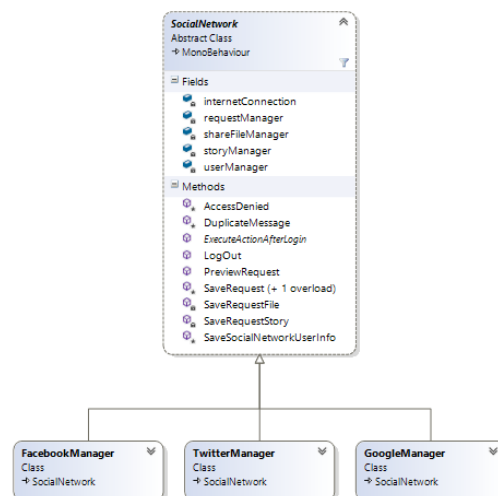


Figure 18 - Class diagram of public media services integration - client side

In addition, share gameplay, craft and party communicate with web services for grant permissions to players' social networks accounts and also to store and receive information about their social network requests (Figure 19, RequestManager) and social network friends (Figure 19, PartyRelationshipManager) in the database. Figure 19 presents a simplified class diagram of the database integration in GameNshare's client side.

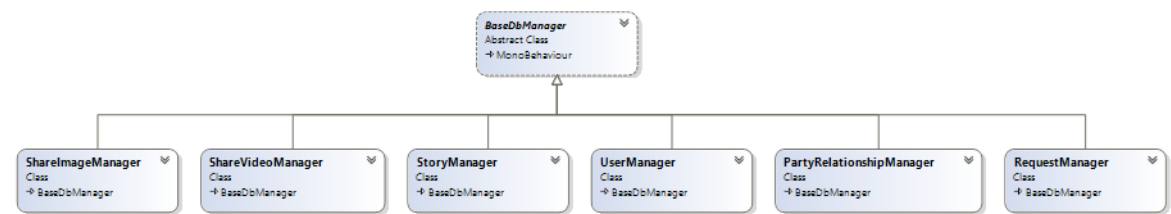


Figure 19 - Class diagram of database integration - client side

The choice of additional development technologies for the implementation of social overlay tools was based only in the evaluation of the requirements specification and project needs.

Screenshot Capture was implemented using the native capture function [Unity Technologies, 2015a] of Unity3D. This function is compatible with Windows, Mac OSX and Linux. Regarding to video recording it was determined that the two best development options were: (1) the use of DirectShowNET [DirectShowNet, 2010]; and (2) the use of FFMPEG [Tomar, 2006]. As DirectShowNET is only available for Windows applications, so it was decided to use only the FFMPEG library, as it supports Windows, MAC OS X and Linux. Figure 20 presents a simplified class diagram of the FFMPEG integration in GameNshare client side.

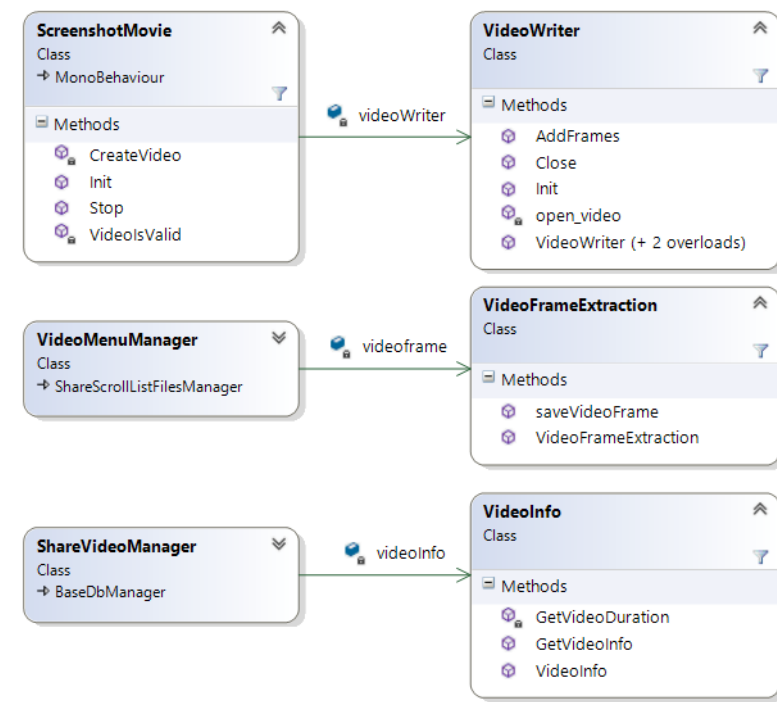


Figure 20 - Class diagram of the FFMPEG integration - client side

FFMPEG library for video recording (Figure 20, VideoWriter) was implemented with the help of Ruslan-B FFmpeg.AutoGen [Balanukhin, 2015], which is an open-source wrapper for FFMPEG 2.5.2, compatible with Mono. This wrapper is also used for the previewing of video thumbnails (Figure 20, VideoFrameExtraction) and for the saving of video info in the database (Figure 20, VideoInfo). The classes “VideoFrameExtraction”, “VideoWriter” and “VideoInfo” include support for direct communication with the Ruslan-B FFmpeg.AutoGen wrapper.

4.2 External Social Networks

The social overlay tools, as well as the web server are capable of interacting with the existing social networks. The credentials of the users are only used inside of social networks sites to acquire access tokens and are not stored in the database.

GameNshare includes support for the integration of Facebook’s Graph API v2.2 [Facebook, 2015h] and Twitter REST API v1.1 [Twitter Inc., 2015d] for the sharing of videos, images and stories and also for obtaining the social network friends list. In addition to these three features the integration of Google+ using Google+ Domains API [Google, 2015d] and Google+ API [Google, 2015c] was also approached, but proven unsuccessful, because neither of these APIs provide full write access to the Google+ profile of a player. Additional details concerning Google+ integration can be found in section 4.2.3.

GameNshare also offers support for Google APIs v3 (YouTube API v3 with Google API v3 OAuth2 Service [Google, 2015i]) for the sharing of videos. YouTube support was also used for the sharing of the videos in Twitter, because until May 21, 2015 [Bulava, 2015] the Twitter REST API did not support the direct upload of videos¹⁰.

Social network APIs and policies are constantly being updated and changing. Thus, frequent code maintenance of the integration of these APIs within GameNShare is inevitable. There is no information about the deprecation date of Google V3 and Twitter 1.1 APIs, but the integration of GameNshare with Facebook should work without any critical code changes until 25th March of 2017 [Facebook, 2015c].

Broadcast integration was also studied making use of Twitch API [Justin.tv Inc, 2013] and YouTube Live Streaming API [Google, 2015j]. These APIs were not successfully incorporated, mainly because of their requirement for real-time video encoding. The sections 4.2.3 and 4.2.4 present the results and conclusions of the studies conducted towards the integration of live broadcasting in GameNshare.

¹⁰ This API feature was released after the date of the integration of the upload of videos to Twitter. Other features of beta version with higher priority were being implemented at this date not permitting the update of the integration code.

The next sections are devoted to the presentation of the GameNshare integration with each of these public media services, i.e., Facebook's Graph API, Twitter REST API, Google APIs and Twitch API.

4.2.1 Facebook's Graph API

Graph API presents a simplification of Facebook social graph and provides a large number Facebook actions for external applications. These actions include for example, query data, post new messages, upload photos and videos and get user public information [Facebook, 2015f].

Figure 21 shows an example of a query to user data using the Graph API Explorer tool [Facebook, 2015e].



Figure 21 - Graph API Explorer¹¹

To interact with Graph API, developers' applications need to use the Facebook authentication and authorization flow to obtain an access token for each user. This flow uses OAuth 2.0 protocol [Hardt, 2012] and provides the same login mechanism for web, mobile and desktop applications.

The integration of Facebook in GameNshare follows a solution proposed by Paul Price [Price, 2013]. This approach uses a web server to handle the OAuth 2.0 tokens exchange with HTTP Requests and a compatible version of Facebook SDK C# [Outercurve Foundation, 2014] for Mono 2.6 [Mono, 2009] to query Graph API. Figure 22 presents the exchanging of short-term token for a long-term token purposed by Paul Price [Facebook, 2015b]:

¹¹ <https://developers.facebook.com/tools/explorer/145634995501895/>

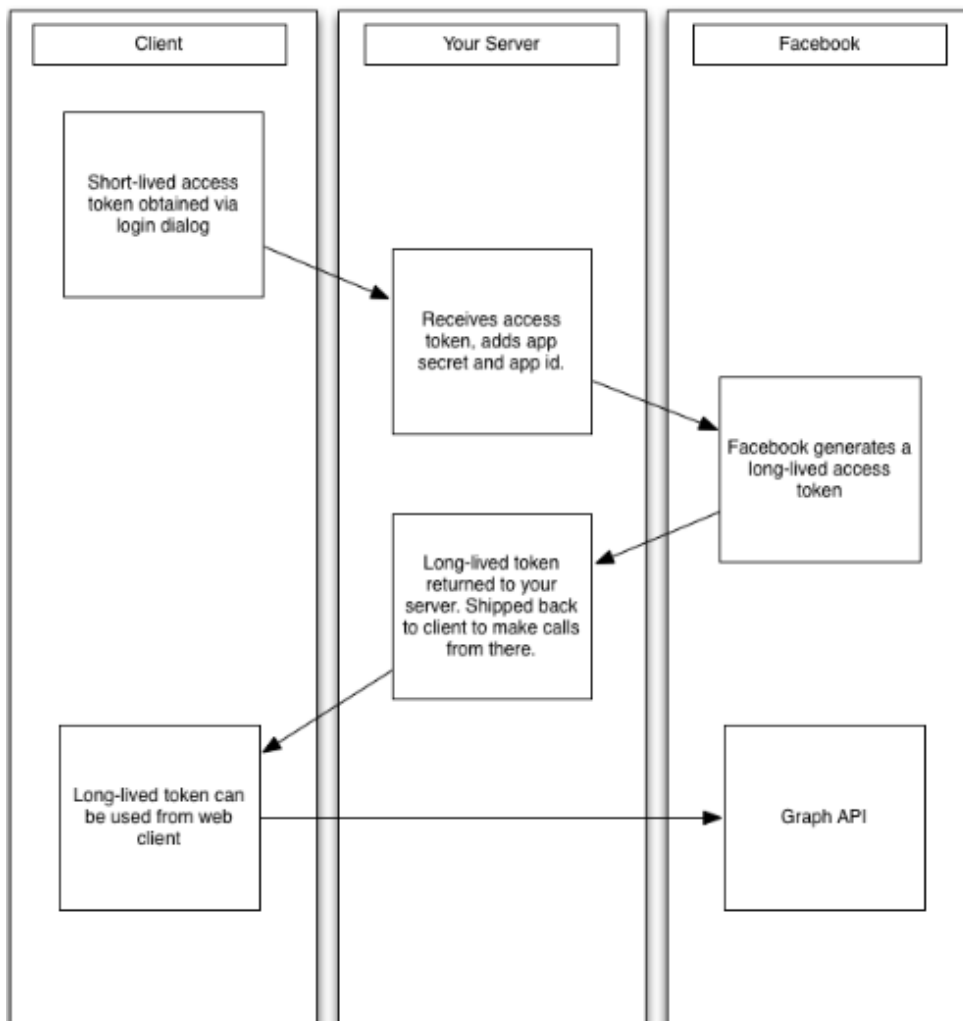


Figure 22 - Exchanging OAuth 2.0 tokens on Facebook¹²

The main disadvantage of the solution developed by Paul Price is that the flow of authentication and authorization is restarted every time Graph API methods are used. A new access token (long-lived access token) is requested in every Graph API calls, even if it is not expired. To overcome this limitation GameNshare implements a reformulation of Paul Price solution with the following modifications: (1) the internet connection is validated every time that the player does Facebook requests; (2) the login prompt and the generation of a access token are requested in case of being the first time that the players use Facebook in the game or if they do not have a valid Facebook session; (3) the last Facebook session is stored between game sessions until the user decides to logout or revokes application access; (4) Facebook session is validated every time that the players make Facebook requests to check if a new long-lived access token is needed; (5) the implementation of the Facebook SDK C# wrapper and the web server implementations were reformulated according to GameNshare's requirements specification

¹² <https://developers.facebook.com/docs/facebook-login/access-tokens>

4.2.2 Twitter REST API

The Twitter REST API includes among others actions the following: create a new Tweet, upload a new photo and read public user information and data [Twitter Inc., 2015d]. To interact with REST API, developers' applications need to use the one Twitter authentication flows to obtain an access token for each user. This flow uses OAuth 1.0 protocol [Hammer-Lahav, 2010] and provides the different authentication mechanisms for web, mobile and desktop applications.

To build a flexible integration of Twitter with GameNshare it was decided to do an implementation that combines both the "Let's Twitter in Unity" asset [Yang, 2014] and Paul Price Facebook solution [Price, 2013]. "Let's Twitter in Unity" is a free and open-source Unity asset based on Twitterizer library [Twitterizer, 2015], that helps developers to deal with the OAuth 1.0 protocol authentication of Twitter API 1.1. This asset only provides a pin-based authentication flow (Figure 23), so users have to manually insert a pin code in the game [Twitter Inc., 2015c].

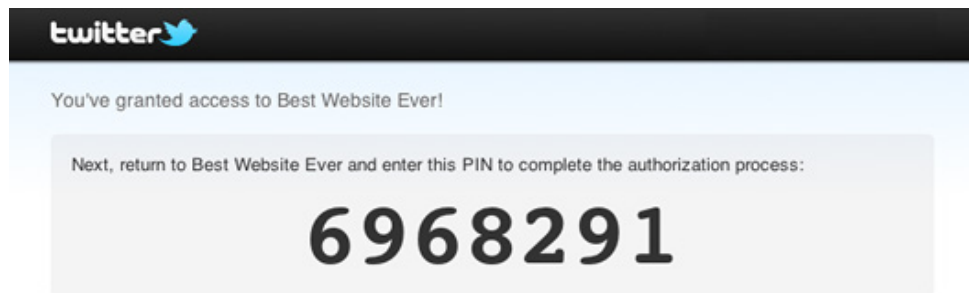


Figure 23 - Twitter pin-based authentication¹³

To overcome the pin-based limitation the GameNshare integration with Twitter uses a web server and includes the following steps: (1) Step 1 - player makes the first twitter call (Figure 24, step 1); (2) Step 2 - the internet connection is validated (Figure 24, step 2); (3) Step 3 - a request-token is solicited to the API using a HTTP POST call (Figure 24, step 3); (4) Step 4 - API provides the request-token (Figure 24 step 4); (5) Step 5 - API redirects the player to the login prompt¹⁴ (Figure 24, step 5); (6) Step 6 - the default browser of the player opens and he/she logs into Twitter account and accepts application permissions (Figure 24, step 6); (7) Step 7 - the player is redirected to a GameNshare's web server page (Figure 24, step 7 and 7.1); (8) Step 8 - the OAuth-verifier is passed to the game and the web server shows a success message to the player (Figure 24, step 8); (9) Step 9 - the request token and OAuth-verifier obtained in steps 4 and 8, respectively, are converted to a access token (Figure 24, step 9); (10) Step 10 - the access token is used for GameNshare to do HTTP requests to Twitter API (Figure 24, step 10). Figure 24 presents the sequence diagram of the Twitter authentication flow.

¹³ <https://dev.twitter.com/oauth/pin-based>

¹⁴ OAuth tokens are stored until they turn invalid or the user decides to logout. The login prompt and the request-token (steps 1-9) are only requested in the first time that players use Twitter in the game or if they do not have a valid Twitter session.

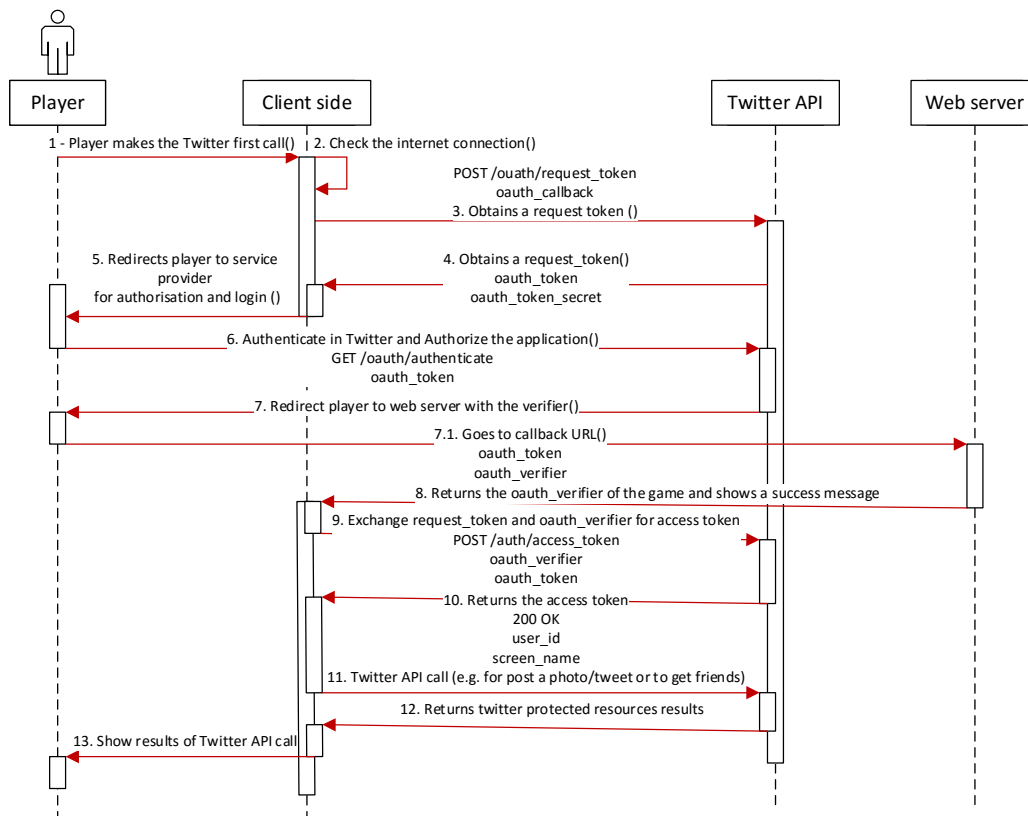


Figure 24 - Sequence Diagram of Twitter authentication flow

4.2.3 Google APIs

Google APIs enable the direct communication between third-party applications and Google services. Among these services the following can be highlighted: YouTube, Google+, Google Search, Gmail and Google Maps. These APIs offer a wide range of functionalities which include for example, analytics and the access to user data [Google, 2015e].

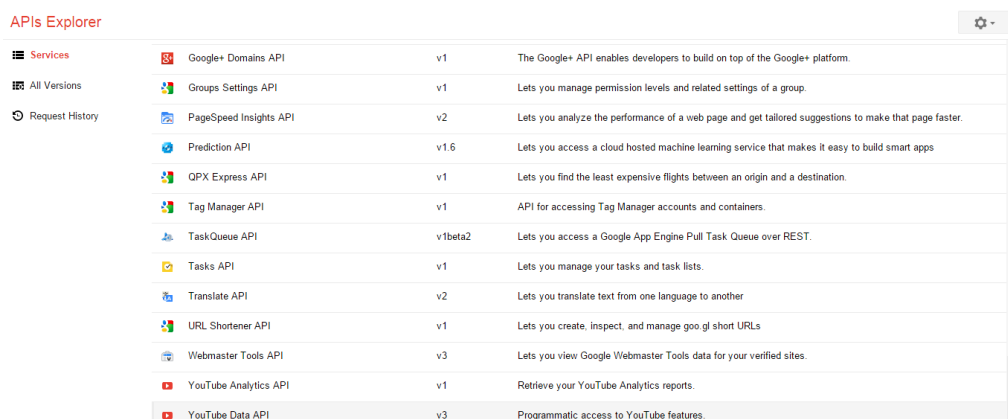


Figure 25 - Google APIs Explorer¹⁵

¹⁵ <https://developers.google.com/apis-explorer/#p>

Similarly to Facebook (Chapter 4, section 4.2.1), Google APIs use the OAuth 2.0 protocol [Hardt, 2012] to provide an access token for each user. Figure 26 presents the Google OAuth 2.0 flow for installed applications [Google, 2015h].

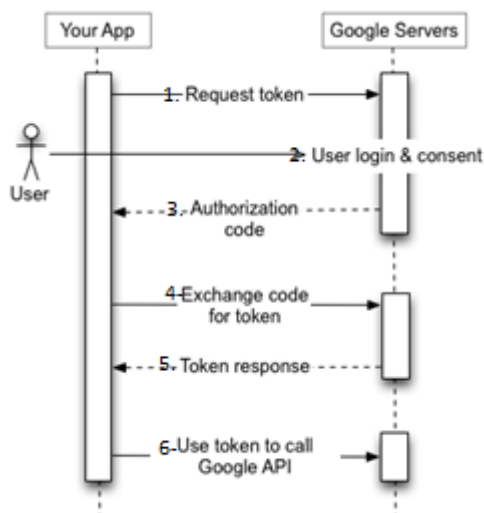


Figure 26 - Google OAuth 2.0 flow for installed applications¹⁶

For the integration of Google APIs v3 authentication flow in GameNshare a solution based on the Twitter REST API approach was developed (Chapter 4, section 4.2.2). This solution combines the use of Google APIs with HTTP Requests and Paul Price Facebook solution [Price, 2013] and includes the following steps: (1) the internet connection is validated every time that players do Google requests; (2) a HTTP GET request to Google server is done to obtain an authorization token (Figure 26, step 1); (3) the default browser of the player opens and he/she logs in with Google account and accepts application permissions¹⁷ (Figure 26, step 2); (4) player is redirected to a web server page that shows him a success message, meanwhile the authorization token obtained via login dialog is exchanged for a access token and a refresh token (Figure 26, step 3 and 4); (4) Google server returns the access token (Figure 26, step 5); (5) the client-side polls the GameNshare web server to check if the access token for that player is already available; (6) the access token is used to do HTTP requests to Google APIs (Figure 26, step 6); (7) access and refresh tokens are stored until they are valid or until the user decides to logout.

Comparing to Facebook and Twitter APIs, an additional step needs to be done to verify login sessions of the players in Google API, because its access tokens have a shorter living time and need to be updated more often using a specific refresh token. This refresh token is only provided by Google server the first time that the user logs in with APO's Google application.

Google APIs v3 authentication flow was only successful integrated with YouTube API v3. Concerning the Google+ integration with GameNshare for the upload of videos, screenshots and stories, Google only allows full-write access to social media management companies[Google,

¹⁶ <https://developers.google.com/identity/protocols/OAuth2InstalledApp>

¹⁷ Similar to Twitter and Facebook, the login prompt and request of authorization token are only asked the first time that players use their Google account in the game or if they have an invalid Google session.

2015g], registered independent software vendors or official Google Apps [Google, 2015d], so integration was not possible at this moment.

Concerning the broadcast support Google provides an API, called YouTube Live Streaming API [Google, 2015j]. This API allows developers to manage live events on YouTube and also uses the OAuth 2.0 protocol for authorization and authentication. To integrate YouTube Live Streaming API with third-party applications, players need to have live stream enabled in their YouTube accounts, besides this videos need to be encoded in real time, a resource intensive process, and sent to YouTube Live Stream. The FFmpeg library used in GameNshare for video recording supports the encoding of videos in real time; however the complex and specific programming necessary for this integration was not feasible to do within the timeframe of this thesis.

4.2.4 Twitch API

Twitch API enables developers to do single sign-in integration with Twitch and to retrieve data about live streams. This API uses OAuth 2.0 protocol for the authentication and authorization.

Amazon.com offers official integration with the Twitter REST API and also with JavaScript SDK [Justin.tv Inc, 2013] for web solutions. To support direct integration of Twitch REST API with desktop game applications, the videos need to be encoded in real time and sent to the API. The same reasons presented in section 4.2.3 for the lack of integration of YouTube Live Streaming API also apply here. The complexity and lack of experience with FFmpeg programming for real time encoding did not allow the integration of Twitch in GameNshare within the timeframe of this thesis.

Besides the mentioned API, Amazon.com also provided support for a Twitch PC SDK that was compatible with APO, however this SDK was discontinued. According to Twitch staff [Fugiman, 2014] *“Since PC based broadcasting software has improved so much over the past year, we no longer recommend studios spend the time integrating game streaming directly into their PC game clients. Instead, we recommend studios to encourage their userbases to download third party broadcast solutions such as OBS, and promote their streaming community within their game with in-game viewing integrations and through all other communication channels such as social media and email”*. Taking this statement in account the study of this SDK and other possible integrations with Twitch for stream broadcasting were suspended.

4.3 Web Server

The web server uses a basic ASP.NET MVC website and a MySQL database for the storing of players' public information in the social networks and also for the monitoring of their requests. In addition to the support of different HTTP request methods (e.g. GET, POST, PUT), the ASP.NET MVC application also includes a JSON library to send and receive data information in this format.

The web server component is composed by two different tiers: the user interface component (UI component), which is the part that includes the views for the presentation of user interface layer to the players and the backend component, which is the part of the web server that provides the business and logic layer. The UI component provides the interface of the web browser instances in the GameNshare client side and also presents the basic information about the game and privacy policy [Clockwork Inc., 2015b]. The backend component is responsible for ensuring the communication between the desktop game client and the other two components of the web server: the Database and the UI Component. Besides this, the backend component also includes support for authentication and authorization of GameNshare client side in the social networks.

The database is responsible for the persistence and monitoring of players' activity in social networks and in the game, for example, for obtain the statuses of social networks requests and for prevent duplicate social networks requests. This database is also used for the storing of social networks' public information of the players (e.g. name, gender, username and user ID). Among other situations the storing of player information can be particular helpful to query specific social network users that play APO, for example, for obtain all the Twitter users that play APO. In the next two sections will be detailed presented the two main components of the Web Server: (1) "Database"; (2) "Web services".

4.3.1 Database

GameNshare Database was stored in ClearDB service [SuccessBricks, Inc., 2015] with a free plan, limited to 4 simultaneous connections at same time. ClearDB is a MySQL database service that is designed to reduce the necessity of the management of MySQL databases. Among other features, ClearDB ensure that MySQL is always available and online and provide a layered software stack to prevent database conflicts, such as circular replication [Jeremy, 2013]. To allow database modifications with a reduce number of code changes. Database integration is implemented with the Entity Framework (Repository pattern) [Castro, et al., 2007] and uses the Reflection pattern [Buschmann, et al., 2013] to iterate and save the database layer entities.

Figure 27 presents GameNshare Entity Framework Data Model. GameNshare database is a relational database that stores information about user, party/team relationships with social network friends, social network requests and all other related data. This database currently contains a total of 8 tables. Among these tables the following one can be highlighted: (1) "user"; (2) "request"; (3) "party_relationship". The first table, defines the player, it contains all the general information of GameNshare users such as Name, Gender, Age range, social network IDs, usernames and profile pictures. The second table defines the group of social network requests, it stores the requests for uploading files and story and in the future it will also store users craft requests. The third table define all the party relationships between the players and the social networks friends.

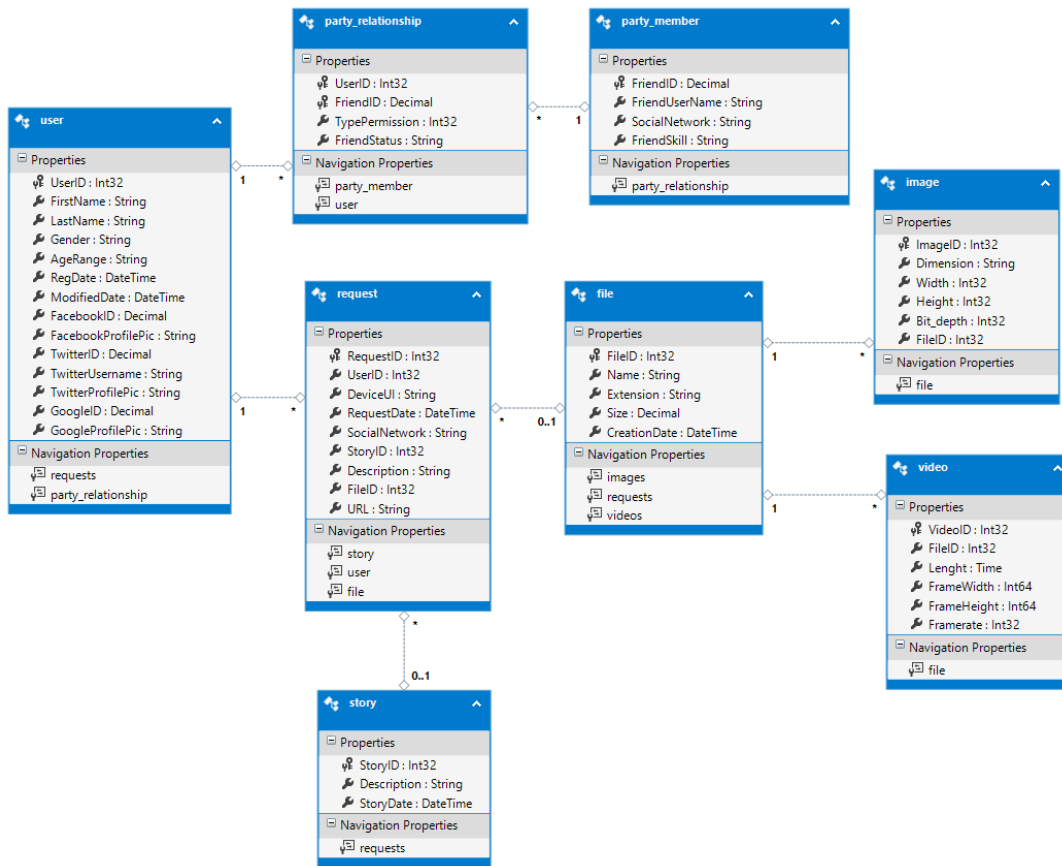


Figure 27 - GameNshare Entity Framework Data Model

As the focus of this master thesis was the integration of the social media with desktop games, the GameNshare database was implemented as simple as possible. This database does not contain stored procedures, triggers, functions and only supports database programming features of ClearDB service.

4.3.2 Web services

Web services provide support for Facebook, Twitter and Google authentication flows. These flows are described in section 4.2. Figure 28 presents a simplified class diagram with the MVC Controllers involved in this process.

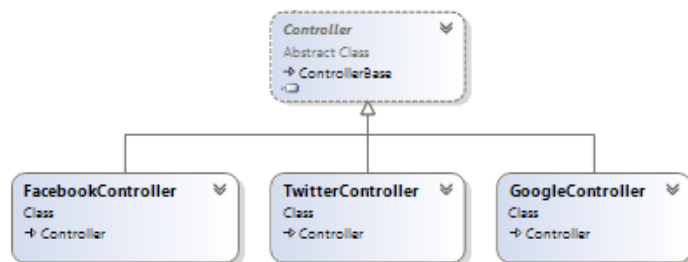


Figure 28 - Class diagram of Social Networks MVC Controllers

Besides this web services ensure the communication between GameNshare client side and the database. Similarly, to social networks APIs, to ensure greater security and confidentiality of the information in the database, was created a “secretKey” of GameNshare game client application. This key is encrypted with MD5 hashing and need to be passed in all the web services calls that request or send database information. Among the MVC Controllers integrated in the database integration the following ones can be underlined: (1) usersController; (2) requestsController; (3) partyRelationshipsController. Figure 29 presents a class diagram with the main database MVC Controllers.

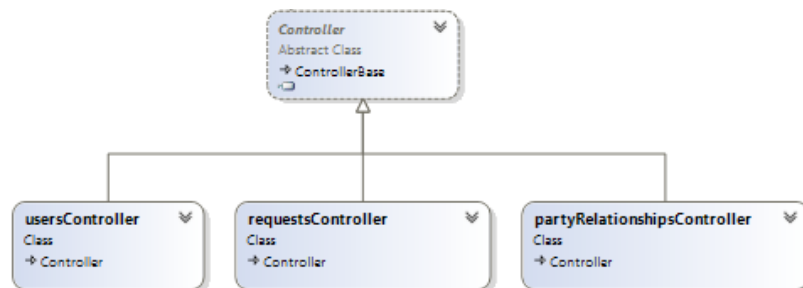


Figure 29 - Class diagram of Database MVC Controllers

The “usersController” is responsible for all the CRUD (Create, Read, Update and Delete) actions of “user” table. Among other actions this Controller is responsible for create a user, update a user, check if a user exists, check if a social network user is a player and for retrieve user’s public information in the social networks. Additionally, it also allows to list all the Facebook, Twitter and Google users that play the game. Table 31 presents the main REST endpoints of “usersController”.

Table 31 - Main REST endpoints of users

URL	HTTP	POST Body	Result
http://tdmei.azurewebsites.net/users/create	POST	JSON string (user and secretKey).	Create a new user entry.
http://tdmei.azurewebsites.net/users/update	POST	JSON string (user and secretKey).	Update an existing user entry.
http://tdmei.azurewebsites.net/users/userExists	POST	JSON string (UserID and secretKey).	Check if a user entry already exists.
http://tdmei.azurewebsites.net/users/fbUserExists	POST	JSON string (UserID and secretKey).	Check if a Facebook user entry already exists ¹⁸ .
http://tdmei.azurewebsites.net/users/getFbProfilePic	POST	JSON string (FacebookID, secretKey).	Get the Facebook profile picture of a user ¹ .
http://tdmei.azurewebsites.net/users/getUsersFb	POST	JSON string (secretKey).	Get all Facebook users ¹ .

The “requestsController” is responsible for all the CRUD actions of “request” table. Among other actions this Controller is responsible for create a request entry and for check if a request is duplicate. In addition to this is also possible to create stories with “storiesController” and to create files with “filesController”. As a future development it will also provide support of craft requests of the users.

Table 31 presents the main REST endpoints of “requestsController”, “storiesController” and “filesController”.

¹⁸ This method also has a version for Twitter and Google.

Table 32 - Main REST endpoints of social network requests

URL	HTTP Verb	POST Body	Result
http://tdmei.azurewebsites.net/requests/create	POST	JSON string (request and secretKey)	Create a new request entry.
http://tdmei.azurewebsites.net/requests/duplicateRequest	POST	JSON string (request and secretKey)	Check if a request is duplicate.
http://tdmei.azurewebsites.net/stories/create	POST	JSON string (story and secretKey)	Create a new story entry.
http://tdmei.azurewebsites.net/files/create	POST	JSON string (file and secretKey)	Create a new file entry ¹⁹ .

The “partyRelationshipsController” is responsible for all the CRUD actions of “partyRelationship” table.

Among other actions this Controller is responsible for create new party relationship between players and social networks friend and for the change and retrieving of the status and permissions of party members. In addition, with “partyMemberController” is possible to create new party members.

Table 33 and Table 34 present the main REST endpoint of “partyRelationshipsController”.

Table 33 - Main REST endpoints of party relationships (part 1)

URL	HTTP	POST Body	Result
http://tdmei.azurewebsites.net/partyRelationships/create	POST	JSON string (relationship and secretKey)	Create a new party relationship entry. ²⁰
http://tdmei.azurewebsites.net/partyRelationships/changeFriendStatus	POST	JSON string (status and secretKey)	Change a friend status.

¹⁹ “ImagesController” and “VideoController” have similar methods to create videos and images.

²⁰ “PartyMemberController” also has a version of this method for create new party members.

Table 34 - Main REST endpoints of party relationships (part 2)

URL	HTTP Verb	POST Body	Result
http://tdmei.azurewebsites.net/partyRelationships/changeFriendPermission	POST	JSON string (permission and secretKey)	Change a friend permission.
http://tdmei.azurewebsites.net/partyRelationships/getFriendStatus	POST	JSON string (secretKey)	Get a friend status.
http://tdmei.azurewebsites.net/partyRelationships/getFriendPermission	POST	JSON string (secretKey)	Get a friend permission.

To reduce the code changes of MVC Database Controllers, the data access layer was implemented with Reflection pattern [Buschmann, et al., 2013]. This pattern was used to iterate entities and to convert JSON string to data model objects and vice-versa. Code 1 presents an extract of ASP.NET code used in GameNshare that shows the use of Reflection to convert JSON strings into model objects.

```

1.     public static void ConvertJSONToObject(Object o, JsonObject jsonAttributes)
2.     {
3.         //Get all attributes of o
4.         PropertyInfo[] propertiesUsers = o.GetType().GetProperties();
5.         //Map jsonAttributes to o attributes
6.         foreach (string key in jsonAttributes.Keys)
7.         {
8.             foreach (PropertyInfo attribute_obj in propertiesUsers)
9.             {
10.                //Check if the json key is equal to the Object attribute name (e.g currentUser.AgeRange = userJson["AgeRange"])
11.                if (key.Equals(attribute_obj.Name))
12.                {
13.                    string valueKey = jsonAttributes[attribute_obj.Name].ToString();
14.                    //Cast to json string to a valid type
15.                    if (!String.IsNullOrEmpty(valueKey))
16.                    {
17.
18.                        Type t = Nullable.GetUnderlyingType(attribute_obj.PropertyType) ?? attribute_obj.PropertyType;
19.                        var converter = TypeDescriptor.GetConverter(t);
20.                        if (converter.CanConvertFrom(valueKey.GetType()))
21.                        {
22.
23.                            object safeValue = (valueKey == null) ? null : converter.ConvertFrom(valueKey);
24.                            //Set the object attribute value
25.                            attribute_obj.SetValue(o, safeValue, null);
26.                        }
27.                    }
28.                }
29.            }
30.        }
31.    }

```

Code 1 - Convert JSON string into model objects with Reflection

5 Implementation and Evaluation

GameNshare was developed in C# with Unity3D. In addition, was used FFMPEG library for the video recording and a modified version of Facebook SDK C# for the integration with Facebook. The GameNshare web server was developed in ASP.NET and use a MVC web application to ensure the communication between the client side and the database and external social network components.

GameNshare include support for Facebook, Twitter and YouTube. This social overlay can be executed on Windows, Mac OSX and Linux systems. Record feature is not currently compatible with Mac OSX and Linux system, because it requires different installations of FFMPEG libraries for each system. Table 35 presents a compatibility table for each of GameNshare feature.

Table 35 - Compatibility table of GameNshare features

Feature	Windows XP+ x86/x64	Mac OS X 10.7+ x86/x64	Ubuntu 12.04+ x86/x64
UI components	✓	✓	✓
Capture Screenshots	✓	✓	✓
Record Videos (without sound)	✓	✗	✗
Facebook Share (upload videos, screenshots and stories)	✓	✓	✓
Twitter Share (upload videos, screenshots and stories)	✓	✓	✓
YouTube Share (upload videos)	✓	✓	✓
Party System (Facebook and Twitter friends)	✓	✓	✓

5.1 Alpha version

The key goal of alpha version was to provide a preview version of GameNShare in order to fix critical limitations and to capture roadmap features and target audience needs.

In this first stage of GameNShare only the essential requirements were implemented: (1) Share Menu (without broadcast support); (2) Record and Upload Videos; (3) Capture and Upload screenshots; (4) Share Game Log.

The next sections of this chapter present GameNShare features of alpha version.

5.1.1 Share Menu

In the alpha version to share videos, screenshots and game-related stories in Facebook, Twitter and Google accounts, the players had access to the following menu:

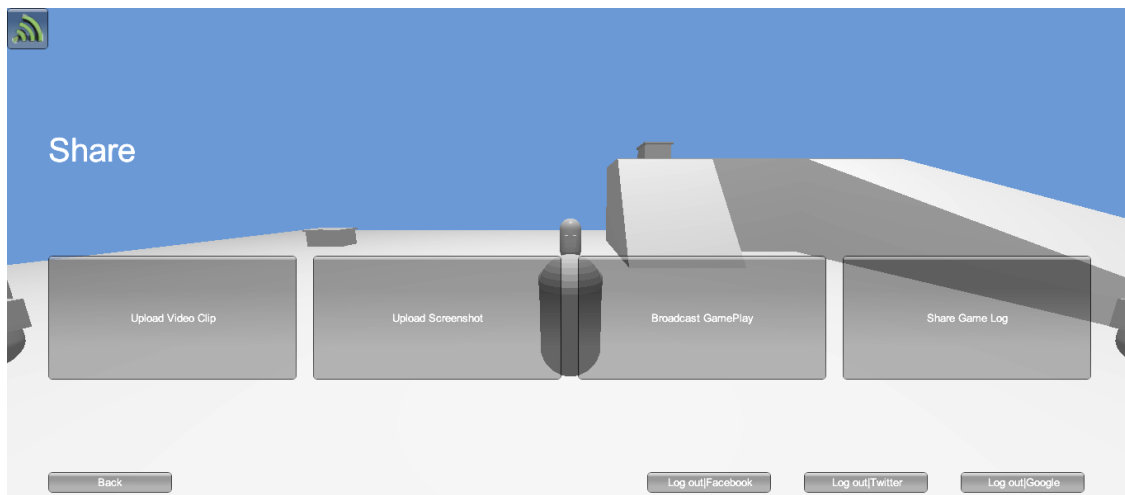


Figure 30 - Share Menu (alpha version)

The “Share” menu was in a very early stage, but already included “Upload Video Clip”, “Upload Screenshot”, “Broadcast Gameplay” and “Share Game Log” options with Broadcast Gameplay option disabled. To open this menu, players can click in “share button” from the menus or use the “Share Menu” shortcut. All the videos, screenshots and stories captured with these features are presented to the user in a reverse chronological order (from newest to oldest).

5.1.2 Record and Upload Video

This version already supports the record of short video clips (up to 15 minutes). GameNShare videos can be recorded in any part of game and uploaded to Facebook (Chapter 4, Section 4.2.1), Twitter (Chapter 4, Section 4.2.2) and YouTube (Chapter 4, Section 4.2.3).

To record videos players have to press the record button or record shortcut. To capture a video was adapted a Unify Community script called "ScreenshotMovie" [Unify Community, 2014]. This script allows to capture a sequence of images at a constant frame rate. To provide a fast upload,

videos were only rendered with a rate of 10 frames per second, but this value can be set to bigger values anytime. Code 2 presents an extract of code “ScreenshotMovie” script used in GameNshare project.

```

1. public void Init()
2. {
3.     if (isRecord)
4.     {
5.         // Set the playback framerate
6.         Time.captureFramerate = ResourceManager.VIDEO_FRAME_RATE;
7.         //Image Increment of screenshots sequence
8.         imageIncrement = 0;
9.         // The folder we place all screenshots inside.
10.        realTempFolder = Application.dataPath + "/" + tempFolder;
11.        //A video is already rendering
12.        if (Directory.Exists(realTempFolder) && Directory.GetCreationTime(realTempFolder).Date == DateTime.Today.Date)
13.        {
14.            render = true;
15.            isRecord = false;
16.            Stop();
17.        }
18.        else
19.        {
20.            //Folder increment
21.            int folderIncrement = 1;
22.            while (Directory.Exists(realTempFolder))
23.            {
24.                // If the folder exists we will append numbers to create an empty folder.
25.                realTempFolder = realTempFolder + folderIncrement;
26.                folderIncrement++;
27.            }
28.            // Create the temp folder
29.            Directory.CreateDirectory(realTempFolder);
30.            // Create APO Video folder
31.            if (!Directory.Exists(Application.dataPath + ResourceManager.VIDEO_FOLDER))
32.            {
33.                Directory.CreateDirectory(Application.dataPath + ResourceManager.VIDEO_FOLDER);
34.            }
35.        }
36.    }
37. }

```

(a)

```

1. //Update run in all the frame
2. public void Update()
3. {
4.     //Take a print in all frames
5.     if (isRecord)
6.     {
7.         //Name of screenshot
8.         var name = String.Format("{0}/shot{1}.png", realTempFolder, imageIncrement);
9.         //Increment the image index of the sequence
10.        imageIncrement++;
11.        //Capture the screenshot
12.        Application.CaptureScreenshot(name, sizeMultiplier);
13.    }
14. }

```

(b)

Code 2 - Screenshot movie functions: (a) start recording; (b) save video sequence

Ruslan-B FFmpeg.AutoGen wrapper was used to render the sequence of images to a video. FFMPEG library provides some tutorial lessons to output a video from an image sequence using FFMPEG [Bellard, 2001; Bellard, 2003], that were the base of all “VideoWriter” class methods. Among other steps, in order to write a video frame from an image using FFMPEG the following steps are needed: (1) convert the pixel format of the images to a supported pixel format of FFMPEG videos (YUV420P); (2) encode the image to a compressed frame using “avcodec_encode_video2” method; (3) check if image was successful buffered in the video; (4) write the compressed frame to the video. Code 3 presents an extract of code of the video encoding using Ruslan-B FFmpeg.AutoGen wrapper.

```

1. private void write_video_frame(string imageName)
2. {
3.     int ret;
4.     SwsContext* sws_ctx = null;
5.     AVCodecContext* c = video_st->codec;
6.
7.     if (sws_ctx == null)
8.     {
9.         //Convert from BGR24 to YUV420P
10.        sws_ctx = FFmpegInvoke.sws_getContext(c->width, c->height, AVPixelFormat.AV_PIX_FMT_BGR24, c->width, c->height, AVPixelFormat.AV_PIX_FMT_YUV420P, 0, null, null, null);
11.        if (sws_ctx == null)
12.        {
13.            throw new Exception("Could not initialize the conversion context");
14.        }
15.    }
16.    /* as we only generate a YUV420P picture, we must convert it to the codec pixel format if needed */
17.    fixed (AVPicture* src_picture_p = &src_picture)
18.    {
19.        fill_image(src_picture_p, imageName);
20.        fixed (AVPicture* dst_picture_p = &dst_picture)
21.        {
22.            FFmpegInvoke.sws_scale(sws_ctx, &src_picture_p->data_0, src_picture_p->linesize, 0, c->height, &dst_picture_p->data_0, dst_picture_p->linesize);
23.            src_picture = *(src_picture_p);
24.            dst_picture = *(dst_picture_p);
25.        }
26.    }
27.
28.    if ((oc->format->flags & FFmpegInvoke.AVFMT_RAWPICTURE) > 0)
29.    {
30.        /* Raw video case - directly store the picture in the packet - FFmpeg will change in the future for fix this */
31.        AVPacket pkt;
32.        FFmpegInvoke.av_init_packet(&pkt);
33.        pkt.flags |= FFmpegInvoke.AV_PKT_FLAG_KEY;
34.        pkt.stream_index = video_st->index;
35.        pkt.data = dst_picture.data_0;
36.        pkt.size = sizeof(AVPicture);
37.        /*write a video frame with that picture
38.        ret = FFmpegInvoke.av_interleaved_write_frame(oc, &pkt);
39.    }
40.    else
41.    {
42.        AVPacket pkt;
43.        pkt.data = (byte*)0;
44.        pkt.size = 0;
45.        int got_packet;
46.        FFmpegInvoke.av_init_packet(&pkt);
47.
48.        /* encode the image */
49.        ret = FFmpegInvoke.avcodec_encode_video2(c, &pkt, frame, &got_packet);
50.        if (ret < 0)
51.        {
52.            throw new Exception("Error encoding video frame");
53.        }
54.
55.        /* If we got the packet (got_packet>0) and its size is bigger than 0, it means the image was buffered. */
56.        if (got_packet > 0 && pkt.size > 0)
57.        {
58.            pkt.stream_index = video_st->index;
59.            /* Write the compressed frame to the media file. */
60.            ret = FFmpegInvoke.av_interleaved_write_frame(oc, &pkt);
61.        }
62.        else
63.        {
64.            throw new Exception("Error while writing video frame");
65.        }
66.    }
67. }

```

Code 3 - Write a video frame with Ruslan-B FFmpeg.AutoGen

To upload a video clip the player needs to (Figure 31 to Figure 33): (1) Select a video from the list; (2) Choose a social network (Facebook, YouTube or Twitter or all the three); (3) (Optional step) Enter a comment; (4) (Optional step) Change video title; (5) Click in the share button.

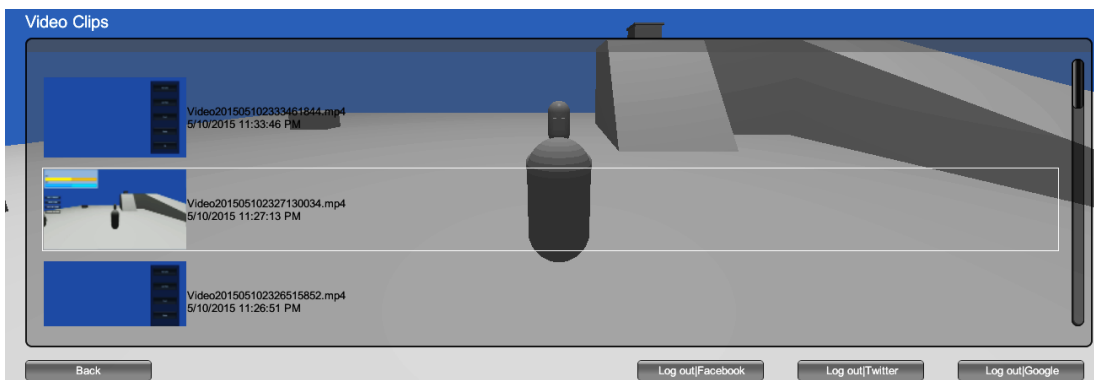


Figure 31 - Select Video to Upload to Facebook, Twitter and YouTube (alpha version)



Figure 32 - Share APO' video to Facebook, Twitter and YouTube (alpha version)

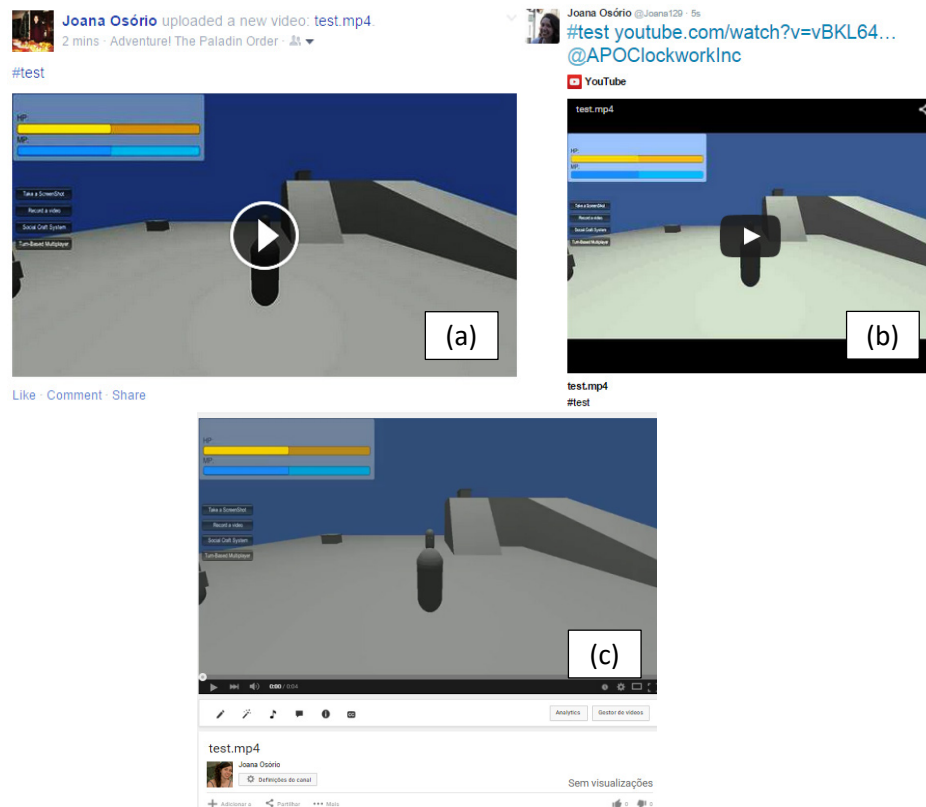


Figure 33 - Samples of posted videos in the social networks: (a) Facebook video; (b) Twitter video; (c) YouTube video

In the case of Twitter, players are asked to upload their videos to their youtube account and share the link to Twitter (Figure 34), because until 21th of May 2015 [Bulava, 2015] Twitter API did not support the direct upload of videos. This option can be saved for next requests.

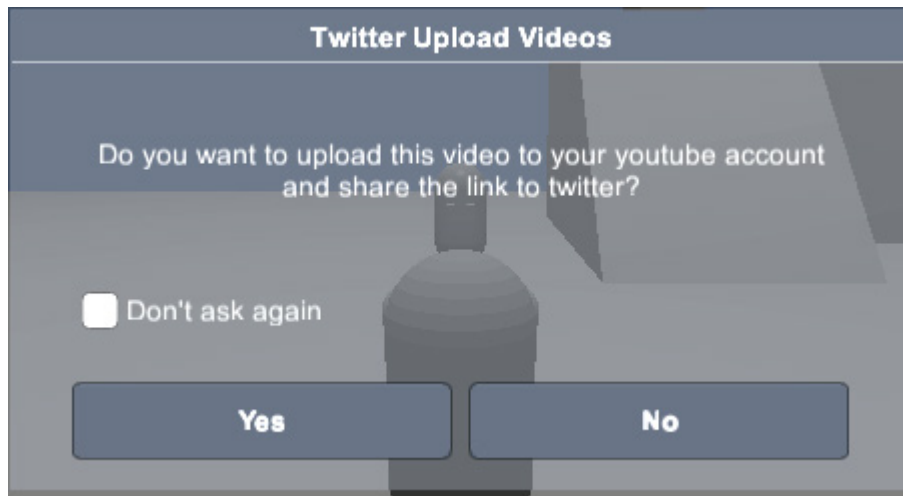


Figure 34 - User prompt for upload video to YouTube and share the link in Twitter

5.1.3 Capture and Upload Screenshots

The Alpha version already integrated the capture and uploads of screenshots to Facebook (Chapter 4, Section 4.2.1) and Twitter (Chapter 4, Section 4.2.2). With this feature, players can take screenshots while playing the game and upload them to social networks.

To take a screenshot players have to press screenshot button (or screenshot shortcut). Screenshot Capture uses the native capture function [Unity Technologies, 2015a] of Unity3D. Code 4 presents an extract of code of the “TakeScreenShot” function used.

```

1. //Take screenshot is used with screenshot bt and printscreen shortcut
2. public void TakeScreenShot()
3. {
4.     //Check if APO screen folder already exists
5.     string filePath = Application.dataPath + ResourceManager.SCREEN_FOLDER;
6.     string name = ResourceManager.SCREEN_FILE_NAME;
7.     string extension = ResourceManager.SCREEN_EXTENSION;
8.     if (!Directory.Exists(filePath))
9.     {
10.        Directory.CreateDirectory(filePath);
11.    }
12.
13.    //Save screenshot with timestamp
14.    var filePathScreenShot = filePath + name + System.DateTime.Now.ToString("yyyyMMddHHmmssffff") + extension;
15.    Application.CaptureScreenshot(filePathScreenShot);
16.
17.    //Wait for printscreen file and show a success message when it is rendered
18.    NotificationManager notification = NotificationManager.Instance();
19.    try
20.    {
21.        notification.InitializeFileWatcherPrintScreen(filePathScreenShot);
22.    }
23.    catch (IOException)
24.    {
25.        notification.ShowGenericErrorNotification();
26.    }
27. }

```

Code 4 - Take a screenshot function

To upload a screenshot the player needs to (Figure 35 to Figure 37): (1) Select a screenshot; (2) Select a social network (Facebook, Twitter or both); (3) (Optional step) Enter a comment; (4) Click in the share button.

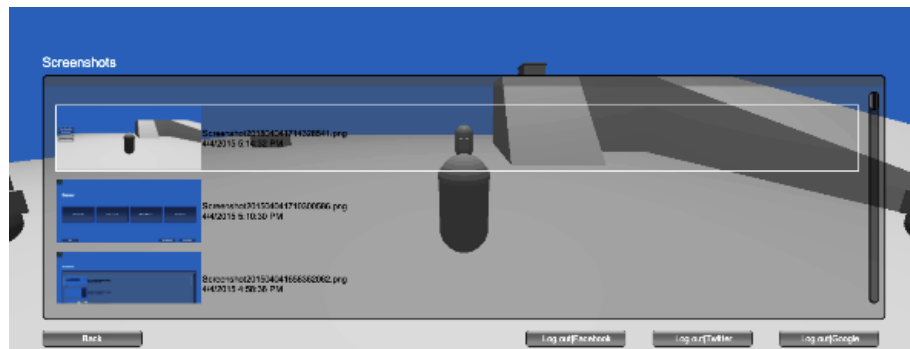


Figure 35 - Select Screenshot to Upload (alpha version)



Figure 36 - Share a screenshot to Facebook and Twitter (alpha version)

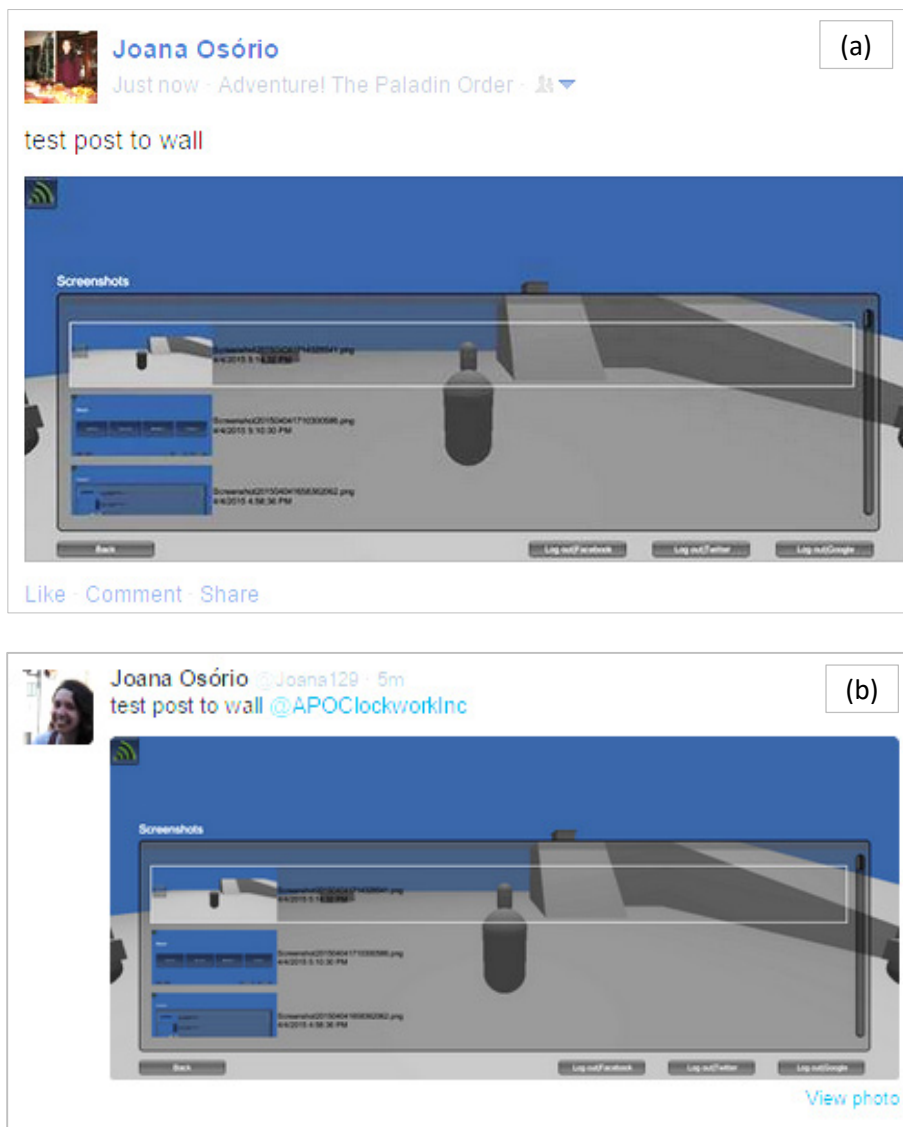


Figure 37 - Samples of posted screenshots in the social networks: (a) Facebook screenshot; (b) Twitter screenshot

5.1.4 Share Game Log

GameNshare alpha version included the share of game-related stories to Facebook (Chapter 4, Section 4.2.1) and Twitter (Chapter 4, Section 4.2.2). With this menu, players can share information about their trophies, quests, missions and other game events on social networks.

Game-related stories are automatically captured and stored by the GameNshare client side in a “gamelog.json” file. These stories are later read in the “Share GameLog” menu. This file contains the description of the stories and the date on which they were captured. Figure 38 presents a sample of the “gamelog.json” file.

```

1  {
2    "Stories": [
3      {
4        "description": "Started playing Adventure! The Paladin Order",
5        "date": "18/10/2015 21:13:03"
6      },
7      {
8        "description": "Started quest \"It's all Ogre now\"",
9        "date": "18/10/2015 21:13:12"
10     },
11     {
12       "description": "I love this game!",
13       "date": "18/10/2015 21:13:34"
14     }
15   ]
16 }
17

```

Figure 38 - Sample of “gamelog.json” file

The Game Log tool captures by default, some common stories of the games, such as “Started playing a game”. To share a story the player needs to (Figure 39): (1) Select a story; (2) Select a social network (e.g. Facebook, Twitter or both); (3) (Optional step) Enter a comment; (4) Click in the share button.



Figure 39 - Upload story to Facebook and Twitter (alpha version): (a) Share windows of game log option after select “Started playing” story; (b) “Started playing” story in Twitter and Facebook

5.2 Alpha testing

GameNshare has undergone a preliminary testing phase called alpha testing. During this phase, GameNshare and APO game were evaluated with survey (Annex B), using Google Forms. To build this survey were followed Qualtrics recommendations [Qualtrics, 2015].

The alpha tests were performed by a group of 10 players (8 boys and 2 girls) aged between 19 and 25 that are members of APO’s test team. To evaluate GameNshare, the testers received a link with game documentation and instructions (Annex A). The players executed the tests by themselves without direct or indirect supervision and after testing they filled up the Google Form.

Alpha testing was focused on the workflow desires and user needs for this kind of system more than on usability and graphics. Although the main purpose of the alpha testing was to identify the main limitations of GameNshare, this testing also gave some insights to product awareness and future work. The next two sections present the results and the main conclusions of the alpha testing.

5.2.1 Results

The survey had three parts that evaluated the GameNshare: (A) Product Awareness; (D) Social overlay; (E) Suggestions and Feedback Ideas (optional section). Only the relevant results of each part will be presented.

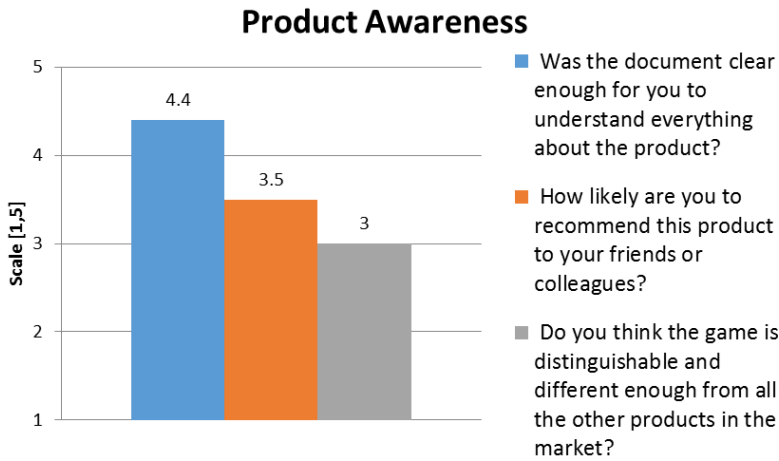


Figure 40 - Alpha testing: Product awareness (scale of 1 to 5)

Regarding the product awareness (Figure 40) testers found that the documentation and overall of the GameNshare tools work well and they would recommend the game to their friends and colleagues (Figure 40, average score of 3.5). However they did not consider the game distinguishable enough (Figure 40, average score of 3), mainly because only the common actions of social overlays were implemented and also because of the poor graphics and usability.

What would you change?

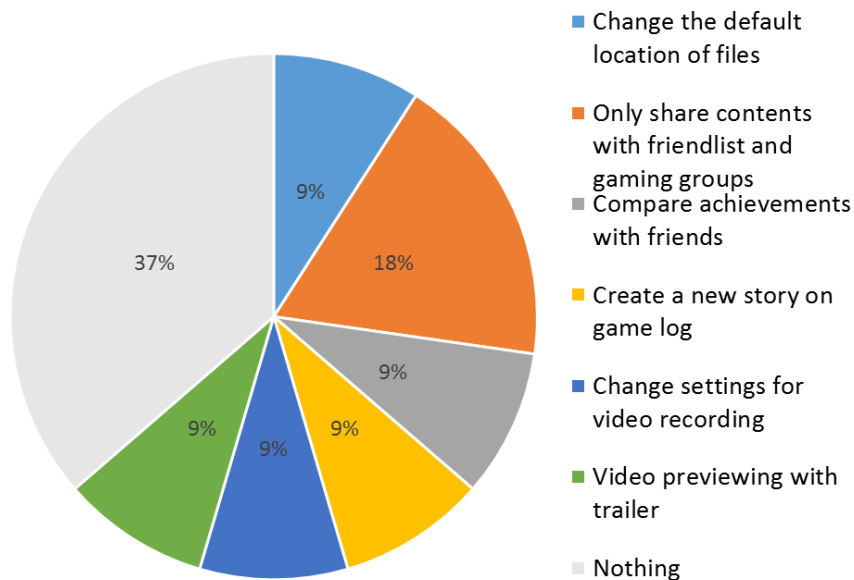


Figure 41 - Alpha testing: Product problems and limitations (open answer).

The major part of testers (37%, Figure 41) did not report any limitation or problems with the features implemented in the alpha version. Some testers (18%, Figure 41) did not like to post directly to their social network “wall” and would prefer to post only to their friend list or gaming groups to prevent the creation of spam. A small part of testers (9%, Figure 41) wanted to change the default path where videos and screenshots are stored, compare achievements with friends and create personalized stories in game log. Lastly, they also requested (9%, Figure 41) to change the settings of video recording, e.g. change the video frame rate and to view a video trailer instead of showing only a thumbnail.

Clockwork Inc. team suggested roadmap features for the options of Share Menu, from which can be underlined the following: (1) record the recent minutes of gameplay continuously and automatically recorded; (2) record the best moments of the game automatically; (3) capture screenshots of the best moments of the game automatically; (4) post to specific group/communities; (5) use private message to share screenshot; (6) use private message to share game log message. These suggestions were classified by the testers with an average score of 3.7 or above in a scale of 1 to 5. Besides the roadmap features suggested test team recommended GameNshare to provide: (1) support for Stream; (2) support for Skype; (3) support for specific game community such as Achievement hunter and Speedrunning community; (4) support for Reddit; (5) support for Vine.

5.2.2 Conclusions

In this first stage only the essential requirements were implemented and significant development was needed in beta version to improve these requirements, particularly to what regards to anti-spam measures, graphics and usability of GameNshare.

All the suggestions, problems and bugs reported were evaluated and prioritized taking into account not only the target audience desires and needs, but also the resources available and degree of development complexity. From this analysis was decided that the beta version would only implement as new features: (1) “Broadcast gameplay”; (2) “Craft an item”; (3) “Use passive party system” and as improvements: (1) Anti-spam measures; (2) New graphics and menus; (3) creation of personalized stories in game log.

5.3 Beta version

In major goal of beta version was to repair the critical limitations of alpha version and also to provide support for broadcast, craft and passive party system.

However, due to time constraints and lack of human resources only the “Passive party system” requirement was fully implemented. The progress of the other two remaining features is presented in the section 5.3.5.

In this section will be presented the alpha version improvements, the passive party system and finally, the progress of broadcast and craft system.

5.3.1 Anti-spam measures

Anti-spam measures were implemented, because 18% testers (Figure 41) did not like to post directly to their “walls” to prevent their friends of receiving spam posts of their game experience.

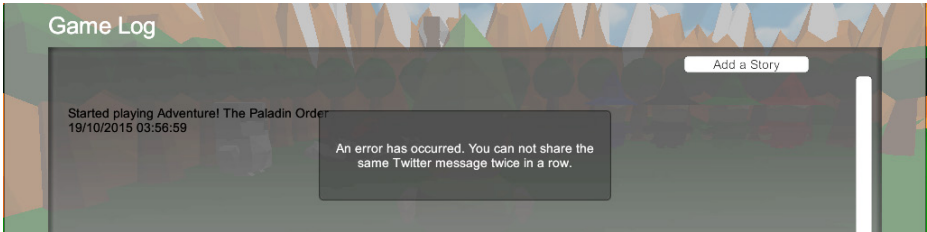


Figure 42 - Duplicate Twitter status error - GameNshare client side

Facebook and Twitter APIs do not allow sharing two consecutive equal statuses and this was already approached in the alpha version (Figure 42).

However, if the statuses were not uploaded immediately after each other, users could upload two or more duplicate statuses.

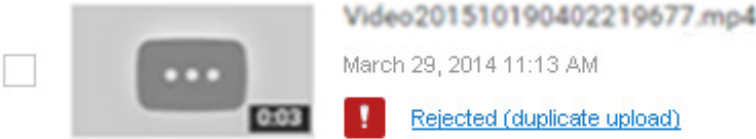


Figure 43 - Rejected YouTube video (duplicate upload)

YouTube API does not prevent the upload of duplicate video, because it only detects it after the upload is done, however it displays a “duplicate error” message (Figure 43).



Figure 44 - Alert message of duplicate twitter story

To prevent the two situations described above, social network requests are stored in the database and the player receives an alert message (Figure 44) to prevent the upload of duplicate images, video and stories. Players can share duplicate posts if they want to or chose to permanently ignore all the next alert messages about duplicate requests with “Don’t show again” checkbox.

5.3.2 Graphic improvements

Regarding graphic improvements of GameNshare in the overworld scene, the “record” and “take a screenshot” buttons were added to HUD, instead of only appearing in the top corner of this scene. The record notification also appears in this menu in order to reduce player distraction. Figure 45 presents the HUD menu while a video is recording.



Figure 45 - Recording a video in Overworld Scene

Concerning to “Share Menu” more information about the overlay were added, e.g. shortcut access and an opaque background to provide to the player the impression of a “Layer”. The buttons were also changed to a modern cubic style.



Figure 46 - Share Menu (beta version)

5.3.3 Personal game stories

This feature was suggested in alpha testing by 9% of the testers and is an upgrading of “Share Game Log” menu.



Figure 47 - Add a story (beta version): Enter a story description

With this features players can create and add their own stories to game log. In order to create personal game stories, players need to (Figure 47): (1) Open the “Share Game Log” menu; (2) Click in “Add new story” button; (3) Enter a story description; (4) Click in “Add” button. These stories are automatically add to “gamelog.json” file with the date of the capturing.

Figure 48 presents the Game Log menu with “I love this game” personal story.



Figure 48 - Add a story (beta version): GameLog with a personal story

5.3.4 Passive party system

Players can use their Facebook and Twitter friends that play the game as attack skills in combats²¹. To accomplish this, players need to (Figure 49 and Figure 50): (1) Click in party button of HUD menu or use party shortcut; (2) Synchronize social network friends (Facebook friends, Twitter friends or both); (3) Select a friend with skill associated to it; (4) Click in the friend photo to use it combat.



Figure 49 - Passive party system (beta version): (a) Synchronize social network friends; (b) Select a friend



Figure 50 - Passive party system (beta version): Use a friend

²¹ In order to test this system was added a dummy strong attack skill that is same to all the friends.

One of reasons given by alpha testers for not wanting the party system in the beta version was the overpowering of their characters in game experiences, i.e. in combats. This fact would make their combats too easy and reduce their combat experience. To prevent this effect, players can only use a friend skill at a time.

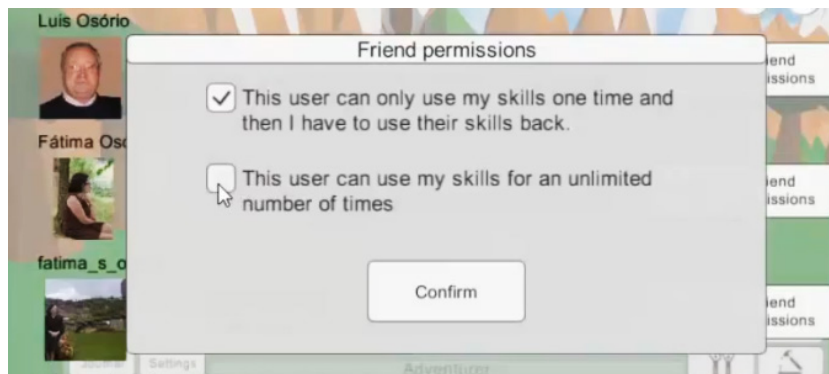


Figure 51 - Friends permissions of passive party system

Each friend can associate a permission status to the player (Figure 51). By default, players can only use their friends one time and then they have to use them back. In addition, players can give “free passes” to friends. With “free passes” players can be used by friends for an unlimited number of times.

To integrate Facebook friends in the passive party system, Facebook API already provide “/me/friends” method [Facebook, 2015g] that only returns the friends of the user that play the game.

Regarding to Twitter API it only provides “friends/ids” method [Twitter Inc., 2015b] that returns all the friends of the player. To overcome this limitation all Twitter users of APO are obtained from the GameNshare database and then they are intersected with all the Twitter friends to the player. To increase system performance this operation was implemented using parallel computing features of C#. Code 5 presents an extract of code used to interest Twitter users of APO with player’s Twitter friends.

```

1. public List<string> IntersectAllTwFriendsWithDbTwUsers(List<string> allTwFriendsList, List<string> databaseTwList)
2. {
3.     List<string> twUserIntersect = new List<string>();
4.     Parallel.ForEach(allTwFriendsList, friendID1 =>
5.     {
6.         Parallel.ForEach(databaseTwList, friendID2 =>
7.         {
8.             if (friendID1 == friendID2)
9.             {
10.                lock (twUserIntersect)
11.                {
12.                    twUserIntersect.Add(friendID1);
13.                }
14.            }
15.        });
16.    });
17.    return twUserIntersect;
18. }

```

Code 5 - Intersecting APO’s Twitter users with player’s Twitter friends

The friends’ pictures are obtained from the GameNshare database and they are saved the first time that players use Facebook or Twitter. These pictures are only requested to Facebook or

Twitter API and updated in database, in case of they have more than one day or do not exist in the user table or if database exceed the limited of connections.

5.3.5 Progress of the missing features

In what regards to “Broadcast Gameplay” requirement were study YouTube Live Streaming API and Twitch API, the detailed results of these studies can be consulted in section 4.2.3 and 4.2.4, respectively. Concerning to YouTube Live Streaming API integration, as it shares the same authentication flow as Youtube API v3, the only missing thing is the encoding of videos in real time. The encoding in real time can be done in the future using Ruslan-B FFmpeg.AutoGen wrapper. This encoding can also be used to integrate Twitch API in GameNshare.

Regarding to “Craft”, the Facebook and Twitter friend finders were developed and then used in “Passive Party System” and also to build a draft of craft mechanism that allow players to send Facebook private message to Facebook friends that play the game. However, substantial development is required to integrate the “Craft” with GameNshare database and APO. Figure 52 presents a draft of the craft system.

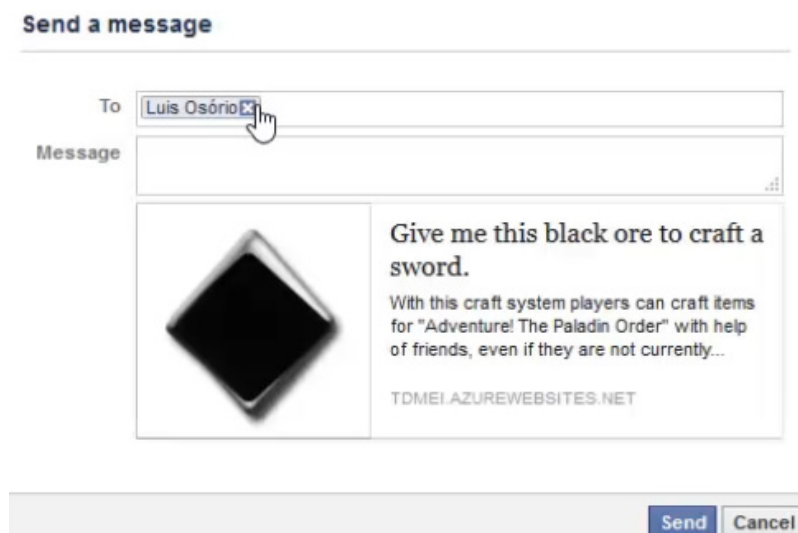


Figure 52 - Facebook private message requesting a game ore

5.4 Beta testing

Beta testing was the second testing phase of GameNshare. Similarly, to the alpha testing the updated version of the game, which included APO and GameNshare, was evaluated by a group of 10 players (8 boys and 2 girls) aged between 19 and 25. These players were the same test team of the alpha testing, because GameNshare needed to be tested with a specific target audience.

In this phase the testers received a link with the game, instructions and documentation [Clockwork Inc., 2015c] about the new features and know bugs and limitations. In the beta

testing, the players also executed the tests by themselves without any supervision and after this they filled up a survey using Google Forms (Annex C).

The focus of this testing was to provide specify feedback about functionality (e.g. ease of use, adaptability, intelligibility), efficiency (e.g. integrity, precision, performance) and reliability (e.g. content quality) of each of the features developed. However, particularly in questions of open answer, the testers also provided some opinions related to graphics, usability, product awareness and future work. Sections 5.4.1 and 5.4.2 present the main results and conclusion of this phase.

5.4.1 Results

This survey contained four parts that evaluated GameNshare: (A) Product Awareness; (E) Share Menu; (F) Party system and friend skills; (G) Suggestions and Feedback Ideas (optional section). Each of these parts contained a specific number of tests. The survey involved a visualization of two 3-min trailers of GameNshare and total interaction of 15 minutes with the game.

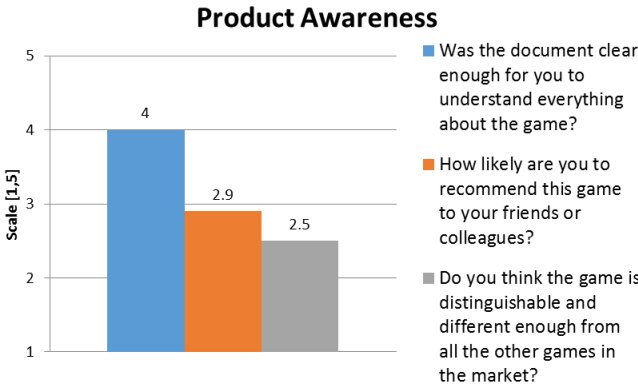


Figure 53 - Beta testing: Product awareness (scale of 1 to 5)

Relatively to product awareness (Figure 53) and comparatively to alpha testing, the testers complained about missing detailed documentation of the GameNshare features (average score of 4, Figure 53). The players considered that the overall of the GameNshare tools work well. However, they would not recommend this game very likely (average score of 2.9, Figure 53), neither have they considered that is distinguishable enough (average score of 2.5, Figure 53), mainly because of graphics aspects and also because one of the most desired features (average score of 3.5, Figure 8), broadcast, is missing in this version. They found out that the graphics of menus and buttons of GameNshare were inadequate for a fantasy game and they would like to broadcast the game to Twitch.

In the “Share Menu” part of the survey, the testers had to watch a 3-min video trailer of “Share Menu” and then they had to perform 3 testes: (1) record a video and share in social networks; (2) upload a screenshot and share in social networks; (3) share a game log story. All the tests included a group of instructions and recommendations. Figure 54 presents a summary of the “Video Record and Upload” beta testing.

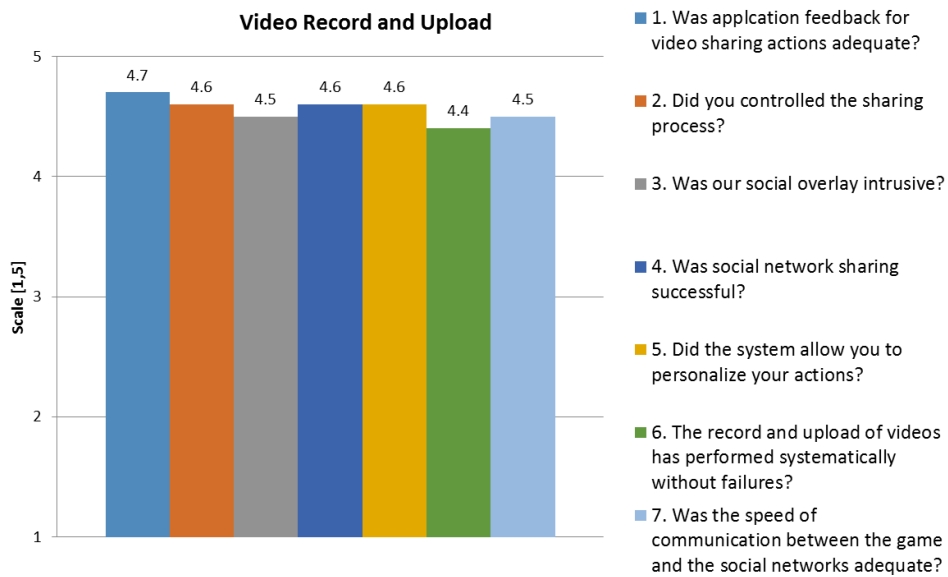


Figure 54 - Beta testing: Video Record and Upload evaluation (scale of 1 to 5)

Regarding to functionality and reliability of “Video Record and Upload”, the major part of testers considered that the upload and video record worked well (Figure 54), but were reported a fewer situations (18%, Figure 55) of interface freezes on the upload of the video.

Beside this, the majority of testers reported some FPS drops (37%, Figure 55) in the game, while recording that did not affect the recording and uploading of the video.

Testers also suggested that videos should have a higher quality (36%, Figure 55), i.e. increasing the frame rate to at least 25 FPS. In what regards to intrusiveness, a minor part of the testers prefer to accept the permissions and enter credentials on social networks every time a video is shared (9%, Figure 55). Figure 55 presents the limitations and problems reported by the testers in this test.

Video Record and Upload - Complaints and Suggestions

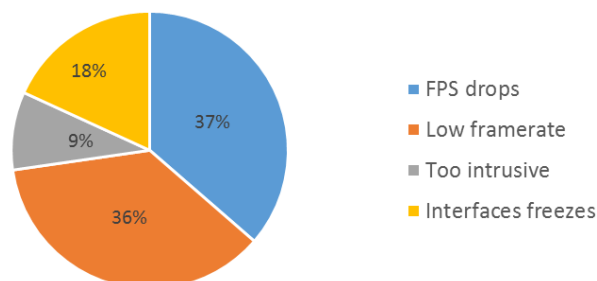


Figure 55 - Beta testing: Complaints and Suggestions - Video Record and Upload (open answer)

In what regards to efficiency of “Video Record and Upload”, the testers found that the speed of communication between the game and social network was adequate (average score of 4.5, Figure 54). The average time for the login process in social networks was 18 seconds (17.85).

For the upload of videos, the average time was 3 seconds (2.5) to videos with a less than one minute (maximum of 40 seconds) and 6 seconds for the videos with more than two minutes (maximum of 00:2:15).

The testers suggested that upload of video should have an average time of 1 second (1.25) to videos with less than one minute (maximum of 40 seconds) and 4 seconds (3.67) for the videos with more than two minutes (maximum of 00:2:15).

Figure 54 presents a summary of the “Screenshot capture and upload” beta testing.

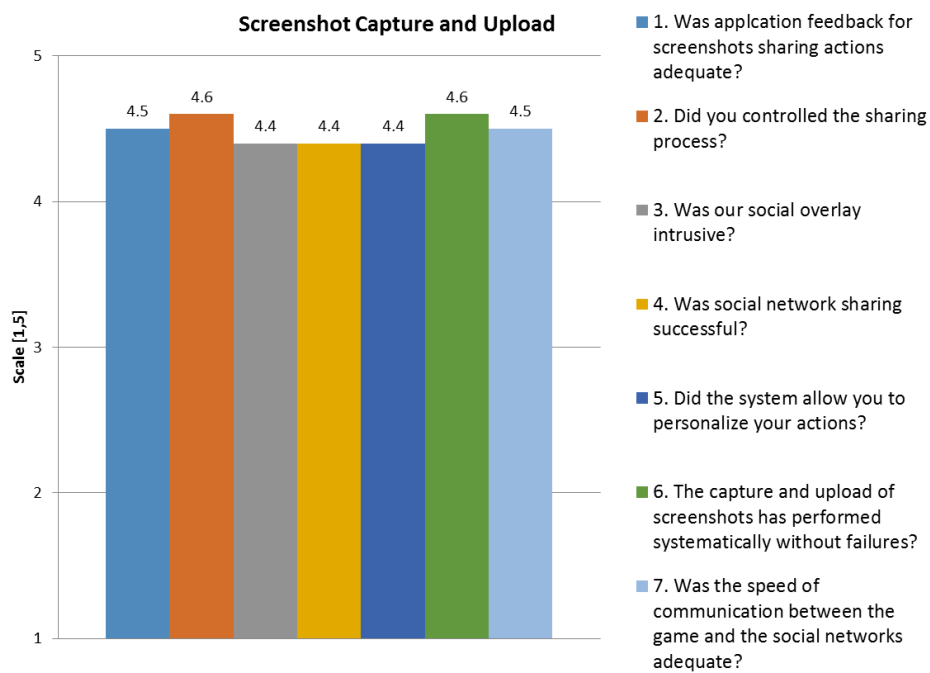


Figure 56 - Beta testing: Screen Capture and Upload evaluation (scale of 1 to 5)

The testers considered that concerning to functionality and reliability the “Screenshot Capture and Upload” tool worked well (Figure 56). A significant share of testers did not report any suggestion or complains (Figure 57, 30%), but the majority would like to have more personalization of their actions, i.e. change the location where the screenshots are saved (Figure 57, 40%) and add feelings status (Figure 57, 30%).

The testers also reported some interfaces freezes in the upload of the screenshots (Figure 57, 10%). Figure 57 presents the complaints and suggestions reported by the testers in this test.

Screen Capture and Upload - Complaints and Suggestions

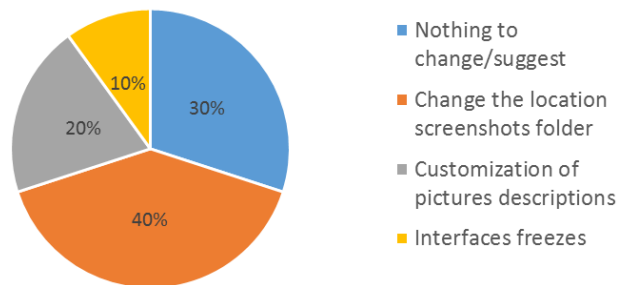


Figure 57 - Beta testing: Complaints and Suggestions - Screen Capture and Upload (open answer)

The testers concluded that the speed of communication between the game and social network was adequate (average score of 4.5, Figure 57). For screenshots with less than 100 KB (minimum 66.7 KB) the average time of upload was 3 seconds (2.5) and the testers suggested that the upload for these type of files should have an average time of 2 seconds (1.5).

Regarding to screenshots with more than 100 KB (maximum of 277KB) the average time of the upload was 4 seconds (3.8) and the testers think that the upload for this type should have an average time of 2 seconds (1.8).

Figure 58 presents a summary of the “Share Game Log” beta testing.

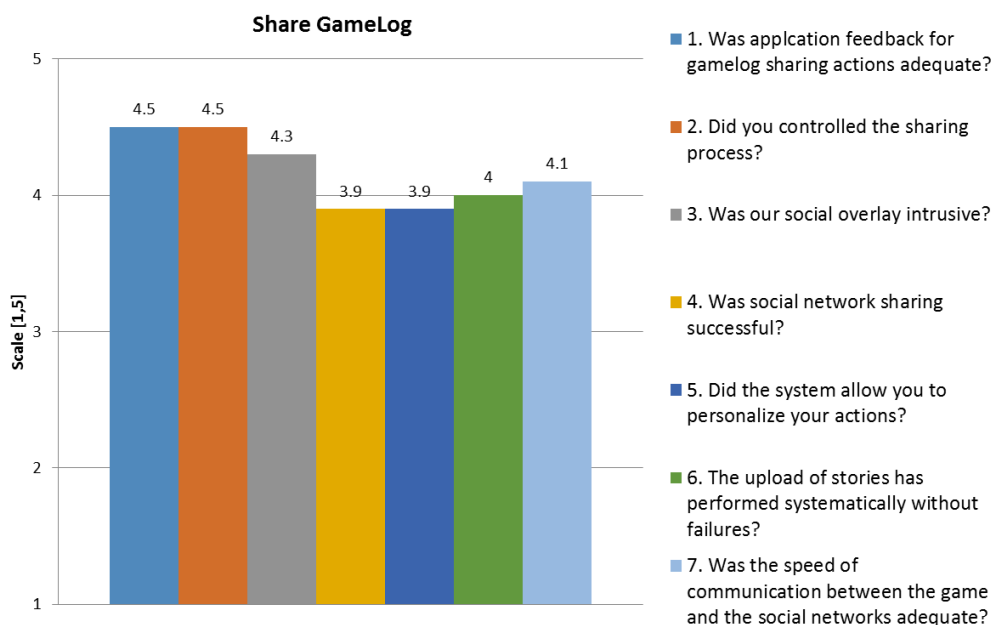


Figure 58 - Beta testing: Share Game Log evaluation (scale of 1 to 5)

In final part of the “Share Menu” tests some users reached the requests limits for share with Facebook API (Figure 59, 20%) and their calls for Facebook with GameNshare were blocked for 30 minutes with the message “User request limit reached on Facebook” [Facebook, 2015d].

This API limitation was already documented in list of “know bugs and limitations” provided for beta testing [Clockwork Inc., 2015c]. They also reported some interface freezes (Figure 59, 10%) in the upload of stories and they suggested that the creation of personal stories (Figure 59, 30%) should be more customized.

Figure 59 presents the complaints and suggestions reported in the “Share Game Log”.

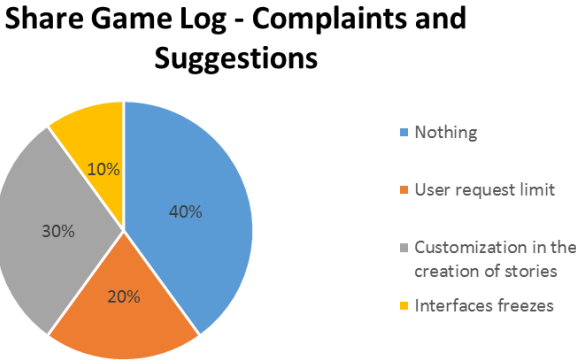


Figure 59 - Beta testing: Complaints and Suggestions - Share Game Log (open answer)

The testers considered that the speed of communication in “Share Game Log” menu between the game and social network was adequate (average score of 4.5, Figure 54). The average time for sharing for stories with less than 15 characters (minimum of 4 characters) was 1 second (1.3) and the testers suggested that the sharing should have an average of 2 seconds (1.83). For stories with more than 15 characters (maximum of 52 characters) was 2 seconds (1.87) and the testers suggested that the average should be 2 seconds (1.62).

In section of “Party system and friend skills” the testers performed two tests: (1) Use friends’ skills in combat; (2) Change friends’ permissions of the friends. In order to execute these tests, the players watched a 3-min video trailer of “Party system and friend skill” and follow some instructions and recommendations. Figure 60 presents a summary of the “Party system and friends skills” beta testing.

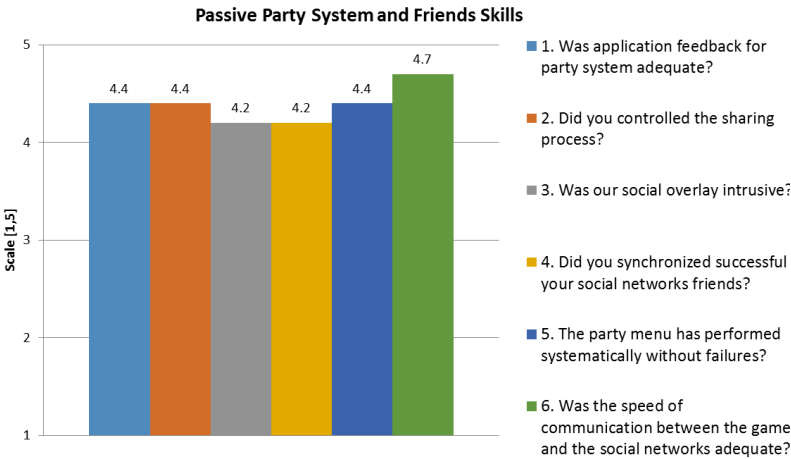


Figure 60 - Beta testing: Passive Party System evaluation (scale of 1 to 5)

The overall of passive party system tests worked well (average scores of 4.2 or above, Figure 60). As the exclusive dummy skill added to party system for the tests purposes, was too strong the only limitation reported by testers was that it was overpowering of their characters. In order to solve that, testers suggested that the friends' skills should affect the stats of the enemy turning it stronger.

The testers considered that the speed of communication between the game and social networks was adequate (average score of 4.7, Figure 60). The social network friends that play the game (minimum of 1 friend and maximum 5 friends) were synchronized with an average time of 40 seconds (37.7) and testers considered that for a maximum of 5 friends the average time of time should be 31 seconds (30.61).

Figure 61 presents the summary of the friends' permissions tests.

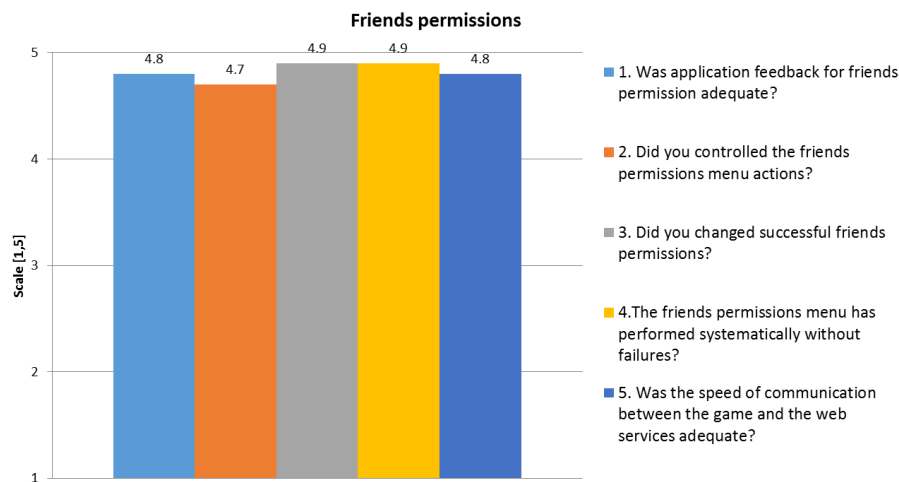


Figure 61 - Beta testing: Friend permissions evaluation (scale of 1 to 5)

Regarding to friends' permissions the testers considered that the menus and buttons of used in friends permissions were inadequate for a fantasy game. In these tests other limitations, suggestions or bugs were not reported.

For the testers, the speed of communication between the game and web services was adequate (average score of 4.8, Figure 61). The average time to change the permission of a friend to "free pass" was 2 seconds (2.3) and the testers considered the average should be 2 seconds (1.83).

5.4.2 Conclusions

The beta testing phase conducted to a bigger engagement of the test team with the game and the social overlay, which resulted in specific feedback to each of the features implemented. The next step of this work is the evaluation and prioritization of the suggestions and complaints reported by the testers.

According to the testers to increase the awareness and attractiveness of GameNshare, substantial improvements needed to be done in the graphics and art of the "Share Menu" and

“Party system and friends’ skills” features. The visual aspect of these tools needed to more closely resemble the graphical style of the game. This was expected since in the time-frame of this thesis it was not possible to create graphical themes to increase the visual attractiveness of GameNshare. Another feature that could not be concluded was the integration of Twitch for live broadcasting of gameplays and this was also pointed out by the testers as a disadvantage.

The “Share Menu”, as well as the “Party system and friends’ skills” were considered for the majority of the testers as non-intrusive and with satisfactory speed of communication between the game, the web services and social networks. It is interesting to notice that the testers consider that these features need an increasing level of personalization in the actions provided to the players. Specifically, 30% of the testers mentioned (Figure 57 and Figure 59) that the “Share Menu” should give to players more options for the customization of their stories, videos and screenshots for example, with the integration of Facebook feeling statuses (moods). In the “Party system and friends’ skills” some game strategies should be designed to prevent the lack of game balance (overpowering), because this complaint was pointed by all the members of test team.

There were also reports of particular issues regarding to interface freezes or FPS drops that should be tested in specific devices and fixed in the next versions. In addition it was also mentioned that the recording of the videos and capturing of screenshots should allow choosing different quality and resolution (36%, Figure 55) and more configurations options, e.g. change the location of the screenshots/videos (Figure 57, 40%).

Overall, with this beta testing analysis it was concluded that the implemented GameNshare tools worked well, however significant work is needed in order to enrich the existing features and also to develop specific missing features, such as broadcast.

6 Conclusions and Future work

The focus of this work was to support the integration of social media in desktop games. In order to do this a prototype, named GameNshare, was developed and tested. It supports not only the common features of the existing social overlays for desktop games such as the sharing of screenshots, videos and stories, but also the integration of social network friends in the mechanics of desktop games.

Social media integration is already used by several organizations and companies for the promotion of their brands, for the strengthening of user communities and improvement of their experiences. However, social media integration is commonly associated with issues (Chapter 2, section 2.2) such as: (1) lack of privacy compliance or (2) social media intrusiveness. GameNshare was designed to enhance desktop gaming by social network integration in a way that avoids these common problems of social media tools. This was accomplished through the delivery of good privacy policies and the implementation of non-intrusive and anti-spam mechanics that allow the players to be in full control of what is shared with social media communities.

GameNshare was specifically targeted for people aged from 18 to 25 years that are regular users of social networks like Facebook and Twitter. It was evaluated in the context of a desktop game (APO) by a group of 10 players within the target age range.

Although the size of the test group did not allow a representative population sample and a statistical analysis of the results from the surveys, it was concluded that in general the features implemented in GameNshare were useful. Additionally was also concluded that, at least in that experiments, GameNshare was found to positively respond to several needs of the testers.

GameNshare is in its second stage (beta version), therefore significant development is still required to improve aspects such as the graphic style of the "Share Menu" and the "Passive party system" and also to implement specific additional features. Furthermore, the recording of videos needs to improve some aspects related to the compatibility and usability that currently impair its ability to capture sound and to be executed in Mac OS X and Linux systems.

Even though GameNshare was not conceived for people under the age of 13 ("Children"), the target audience includes players of different age groups enhancing the care that is needed in

the development of the user interface. In future versions the user interface needs to be more player-friendly, customizable and also more graphically integrated with the style and environments of the game in which it is being played.

As future work, it is purposed to perform a study for the implementation and integration of live broadcast, craft and also additional approaches of asynchronous multiplayer mechanics with social network friends.

The GameNshare prototype was presented as an oral communication in the session of Cultural Contents and Computing (CCC) of the 7th Portuguese Symposium on Computer Sciences (INForum'15) [Inforum'15, 2015b] and selected for publication in the conference proceedings. It was also awarded the prize for best paper of its session and nominated for the best paper of the congress award [Inforum'15, 2015a]. The published paper can be consulted in the Annex D of this dissertation.

In all, the work performed in this thesis allowed concluding that GameNshare has the potential to play an essential role in the integration of a social networks in Windows, Linux and Mac OS X desktop games. In the future developmental stages of GameNshare, a tool that will allow an innate interaction of the player with desktop games and social networks is envisioned also contributing for the development of Cultural Computing, an exciting and emerging field of human computer interaction.

References

- [Activision2014] Activision, 2014. Call of Duty Elite Shutdown FAQ - Activision Support. <http://is.gd/yrAzEG> [accessed: Feb. 2015].
- [Activision Publishing,2014] Activision Publishing, 2014. Call of Duty®: Ghosts || Call of Duty Anywhere. <https://www.callofduty.com/ghosts/features/mp/app> [accessed: Mar. 2015].
- [Andrew,2013] Andrew, W., 2013. Half a billion people have installed “Candy Crush Saga.” *Vox Media*. <http://is.gd/H7qWtG> [accessed: May 2015].
- [Balanukhin,2015] Balanukhin, R., 2015. Ruslan-B/FFmpeg.AutoGen. <http://is.gd/yR1ZFW> [accessed: Jun. 2015].
- [Bellard,2001] Bellard, F., 2001. FFMpeg: doc/examples/decoding_encoding.c Source File. https://ffmpeg.org/doxygen/trunk/decoding__encoding_8c-source.html [accessed: Apr. 2015].
- [Bellard,2003] Bellard, F., 2003. FFMpeg: output_example.c Source File. https://www.ffmpeg.org/doxygen/0.5/output__example_8c-source.html [accessed: May 2015].
- [Blizzard Entertainment, Inc.,2015] Blizzard Entertainment, Inc., 2015. World of Warcraft. <http://eu.battle.net/wow/en/> [accessed: Oct. 2015].
- [Blogger,2015] Blogger, 2015. Blogger: Blogger Panel. <https://www.blogger.com/home> [accessed: Oct. 2015].
- [Bort,2014] Bort, J., 2014. The Guy Who Turned Down \$500 Million For His Startup Just Landed A \$1 Billion Valuation. *Business Insider*. <http://www.businessinsider.com/qualtrics-becomes-1-billion-business-2014-9>.
- [Brunelle,2013] Brunelle, W., 2013. Social Media and Video Games. <http://is.gd/iNRL2K> [accessed: Feb. 2015].
- [Bulava,2015] Bulava, J., 2015. REST API now supports native video upload | Twitter Blogs. *Twitter Blogs*. <https://blog.twitter.com/2015/rest-api-now-supports-native-video-upload> [accessed: Oct. 2015].
- [Buschmann, et al.,2013] Buschmann, F. et al., 2013. *Pattern-Oriented Software Architecture, A System of Patterns*, John Wiley & Sons.

- [Castro, et al.,2007] Castro, P., Melnik, S. and Adya, A., 2007. ADO.NET Entity Framework: Raising the Level of Abstraction in Data Programming. In *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*. SIGMOD '07. New York, NY, USA: ACM, pp. 1070–1072. <http://doi.acm.org/10.1145/1247480.1247609> [accessed: Oct. 2015].
- [Clockwork Inc.,2015a] Clockwork Inc., 2015. Adventure! The Paladin Order. <http://tdmei.azurewebsites.net/> [accessed: Jun. 2015].
- [Clockwork Inc.,2015b] Clockwork Inc., 2015. Privacy Policy. <http://tdmei.azurewebsites.net/Privacy> [accessed: Oct. 2015].
- [Clockwork Inc.,2015c] Clockwork Inc., 2015. S.O.S_SocialOverlay. http://tdmei.azurewebsites.net/Content/Docs/S.O.S_SocialOverlay.txt [accessed: Oct. 2015].
- [DirectShowNet,2010] DirectShowNet, 2010. DirectShowNet library. <http://directshownet.sourceforge.net/about.html> [accessed: Apr. 2015].
- [Duggan, et al.,2015] Duggan, M. et al., 2015. Demographics of Key Social Networking Platforms. *Pew Research Center: Internet, Science & Tech*. <http://www.pewinternet.org/2015/01/09/demographics-of-key-social-networking-platforms-2/> [accessed: Jul. 2015].
- [Echobit LLC,2015] Echobit LLC, 2015. Evolve. <https://www.evolvehq.com/welcome> [accessed: Jun. 2015].
- [Facebook,2015a] Facebook, 2015. About Facebook. <https://www.facebook.com/facebook/info> [accessed: May 2015].
- [Facebook,2015b] Facebook, 2015. Access Tokens. <https://developers.facebook.com/docs/facebook-login/access-tokens> [accessed: Apr. 2015].
- [Facebook,2015c] Facebook, 2015. Facebook Platform Changelog. <https://developers.facebook.com/docs/apps/changelog> [accessed: Oct. 2015].
- [Facebook,2015d] Facebook, 2015. Graph and Ads API Rate Limiting. <https://developers.facebook.com/docs/marketing-api/api-rate-limiting> [accessed: Oct. 2015].
- [Facebook,2015e] Facebook, 2015. Graph API Explorer - Facebook for Developers. <https://developers.facebook.com/tools/explorer/145634995501895/> [accessed: Oct. 2015].
- [Facebook,2015f] Facebook, 2015. Graph API Overview. <http://is.gd/RmaFCy> [accessed: Apr. 2015].
- [Facebook,2015g] Facebook, 2015. Graph API /user/friends. <https://developers.facebook.com/docs/graph-api/reference/v2.2/user/friends> [accessed: Oct. 2015].
- [Facebook,2015h] Facebook, 2015. Using the Graph API v2.2. <https://developers.facebook.com/docs/graph-api/using-graph-api/v2.2> [accessed: Oct. 2015].
- [Federal Financial Institutions Examination Council (FFIEC),2013] Federal Financial Institutions Examination Council (FFIEC), 2013. FDIC: FIL-56-2013: Social Media: Consumer Compliance Risk Management Guidance. <https://www.fdic.gov/news/news/financial/2013/fil13056.html> [accessed: Oct. 2015].

- [Final Fantasy Wiki,2014] Final Fantasy Wiki, 2014. Outerworld Services. *Final Fantasy Wiki*. http://finalfantasy.wikia.com/wiki/Outerworld_Services [accessed: Oct. 2015].
- [Fjællingsdal,2014] Fjællingsdal, K., 2014. *Let's Graduate - A thematic analysis of the Let's Play phenomenon*, <http://www.diva-portal.org/smash/record.jsf?pid=diva2%3A752412&dswid=7814> [accessed: Feb. 2015].
- [Fugiman,2014] Fugiman, 2014. Twitch PC SDK no longer supported - SDK - Twitch Developer Forums. <https://discuss.dev.twitch.tv/t/twitch-pc-sdk-no-longer-supported/1267> [accessed: Oct. 2015].
- [Google,2015a] Google, 2015. About YouTube. *About Youtube*. <https://www.youtube.com/yt/about/> [accessed: May 2015].
- [Google,2015b] Google, 2015. Google+ - About. <https://plus.google.com/+googleplus/about> [accessed: Oct. 2015].
- [Google,2015c] Google, 2015. Google+ API | Google+ Platform for Web | Google Developers. <https://developers.google.com/+web/api/rest/> [accessed: Oct. 2015].
- [Google,2015d] Google, 2015. Google+ Domains API | Google Developers. <https://developers.google.com/+domains/> [accessed: Oct. 2015].
- [Google,2015e] Google, 2015. Google APIs Explorer. <https://developers.google.com/apis-explorer/#p/> [accessed: May 2015].
- [Google,2015f] Google, 2015. Google APIs Libraries - Google APIs Discovery Service — Google Developers. <https://developers.google.com/discovery/libraries> [accessed: May 2015].
- [Google,2015g] Google, 2015. Pages API - Partner Application Form. <https://developers.google.com/+web/api/rest/pages-signup> [accessed: Oct. 2015].
- [Google,2015h] Google, 2015. Using OAuth 2.0 for Installed Applications - Google Identity Platform — Google Developers. <https://developers.google.com/identity/protocols/OAuth2InstalledApp> [accessed: May 2015].
- [Google,2015i] Google, 2015. YouTube Data API | Google Developers. <https://developers.google.com/youtube/v3/> [accessed: Oct. 2015].
- [Google,2015j] Google, 2015. YouTube Live Streaming API Overview | YouTube Live Streaming API | Google Developers. <https://developers.google.com/youtube/v3/live/getting-started> [accessed: Oct. 2015].
- [Hammer-Lahav,2010] Hammer-Lahav, E., 2010. The oauth 1.0 protocol. <http://tools.ietf.org/html/rfc5849>, [accessed: Jun. 2015].
- [Hardt,2012] Hardt, D., 2012. The OAuth 2.0 authorization framework. <http://tools.ietf.org/html/rfc6749.html> [accessed: Jun. 2015].
- [Hof,2010] Hof, R., 2010. Social Games: From the Farm to the Mafia. <http://is.gd/PkAlZK> [accessed: Jan. 2015].
- [Hurk,2013] Hurk, A.M. van den, 2013. *Social Media Crisis Communications: Preparing for, Preventing, and Surviving a Public Relations #FAIL*, Que Publishing.

- [IGN,2014] IGN, 2014. Social Sharing and Connectivity - PlayStation 4 Wiki Guide - IGN. <http://is.gd/Apo11u> [accessed: May 2015].
- [Inforum'15,2015a] Inforum'15, 2015. Prémio INForum 2015 — INForum2015. <http://inforum.org.pt/INForum2015/premios-inforum> [accessed: Sep. 2015].
- [Inforum'15,2015b] Inforum'15, 2015. Programa — INForum2015. <http://inforum.org.pt/INForum2015/programa> [accessed: Sep. 2015].
- [Jeremy,2013] Jeremy, S., 2013. Circular Replication in MySQL (Learning MySQL). https://blogs.oracle.com/jsmyth/entry/circular_replication_in_mysql [accessed: Oct. 2015].
- [Justin.tv Inc,2013] Justin.tv Inc, 2013. Twitch Developers Program. <http://dev.twitch.tv/> [accessed: Oct. 2015].
- [K900 (Product Owner of Evolve),2015] K900 (Product Owner of Evolve), 2015. Looking for Evolve, the game from Turtle Rock? Sorry, but you're in the wrong place. <http://is.gd/samZVx> [accessed: Jun. 2015].
- [Kaplan, and Haenlein,2010] Kaplan, A.M. and Haenlein, M., 2010. Users of the world, unite! The challenges and opportunities of Social Media. *Business Horizons*, 53(1). <http://www.sciencedirect.com/science/article/pii/S0007681309001232> [accessed: Feb. 2015].
- [Kelion,2013] Kelion, L., 2013. PlayStation 4 console announced by Sony in New York. *BBC News*. <http://www.bbc.com/news/technology-21526450> [accessed: May 2015].
- [Kietzmann, et al.,2011] Kietzmann, J.H. et al., 2011. Social media? Get serious! Understanding the functional building blocks of social media. *Business horizons*, 54(3), pp.241–251.
- [Konert, et al.,2014] Konert, J., Göbel, S. and Steinmetz, R., 2014. Video Game Personalization via Social Media Participation. In S. Göbel & J. Wiemeyer, eds. *Games for Training, Education, Health and Sports*. Springer International Publishing, pp. 35–46. http://link.springer.com/chapter/10.1007/978-3-319-05972-3_5 [accessed: Oct. 2014].
- [Linden Research, Inc.,2015] Linden Research, Inc., 2015. About | Second Life. <http://secondlife.com/whatis/> [accessed: Oct. 2015].
- [Loguidice, and Loguidice,2012] Loguidice, B. and Loguidice, C., 2012. *My Xbox: Xbox 360, Kinect, and Xbox LIVE*, Que Publishing.
- [Lynley,2011] Lynley, M., 2011. Electronic Arts' The Sims Social hits 4.6 million daily players a week after launch. <http://is.gd/U8K8UZ> [accessed: May 2015].
- [MacCallum-Stewart,2014] MacCallum-Stewart, E., 2014. *Online Games, Social Narratives*, Routledge.
- [Mack,2009] Mack, C., 2009. Facebook and Twitter Go Live on Xbox 360. *SocialTimes*. <http://is.gd/Px19hr> [accessed: May 2015].
- [Marketing Executives Networking Group (MENG),2015] Marketing Executives Networking Group (MENG), 2015. Digital Marketing Survey: What do leading brand executives think. *MENG*. <http://mengonline.com/digital-marketing-survey-what-do-leading-brand-executives-think/> [accessed: Feb. 2015].
- [Mono,2009] Mono, 2009. Release Notes Mono 2.6. <http://is.gd/X3nm76> [accessed: Jun. 2015].

- [Muller, and Thiesing,2011] Muller, F. and Thiesing, F., 2011. Social networking APIs for companies #x2014; An example of using the Facebook API for companies. In *2011 International Conference on Computational Aspects of Social Networks (CASoN)*. 2011 International Conference on Computational Aspects of Social Networks (CASoN). pp. 120–123.
- [My.Com B.V,2015] My.Com B.V, 2015. Game | Allods Online. <http://allods.my.com/en/game> [accessed: Oct. 2015].
- [Osório, and Pacheco,2015] Osório, J. and Pacheco, F., 2015. Social Media Integration in Video Games: A Social Overlay for Desktop Games. http://www.cister.isep.ipp.pt/docs/social_media_integration_in_video_games__a_social_overlay_for_desktop_games/1146/view.pdf [accessed: Oct. 2015].
- [Outercurve Foundation,2014] Outercurve Foundation, 2014. About the Facebook SDK for .NET. <http://facebooksdk.net/docs/about/> [accessed: Apr. 2015].
- [Overwolf,2015a] Overwolf, 2015. Overwolf Apps. <http://store.overwolf.com/apps/> [accessed: Oct. 2015].
- [Overwolf,2015b] Overwolf, 2015. Overwolf supported games. <http://www.overwolf.com/supported-games/> [accessed: Apr. 2015].
- [Pereria,2011] Pereria, C., 2011. Call of Duty’s Long-Rumored Subscription Service is Here. *1UP*. <http://is.gd/s0gekk>.
- [Pires, and Simon,2015] Pires, K. and Simon, G., 2015. YouTube live and Twitch: a tour of user-generated live streaming systems. In *Proceedings of the 6th ACM Multimedia Systems Conference*. ACM, pp. 225–230. <http://perso.telecom-bretagne.eu/gwendalsimon/data/MMSysDataSets-final10.pdf> [accessed: May 2015].
- [Price,2013] Price, P., 2013. Connecting to Facebook from your Unity desktop game. <http://www.ifc0nfig.com/accessing-facebook-unity-game-2/> [accessed: Apr. 2015].
- [Qualtrics,2015] Qualtrics, 2015. Concept Testing Surveys - How to Conduct a Concept Test | Qualtrics. <http://www.qualtrics.com/research-suite/concept-testing/> [accessed: Oct. 2015].
- [Raby,2012] Raby, M., 2012. Microsoft retires Facebook, Twitter Xbox apps. <http://is.gd/Fu4Vil>.
- [Reddit,2015] Reddit, 2015. reddit.com: about reddit. <https://www.reddit.com/about/> [accessed: Oct. 2015].
- [Schell,2014] Schell, J., 2014. *The Art of Game Design: A Book of Lenses, Second Edition* 2 edition., Boca Raton: A K Peters/CRC Press.
- [Shin, and Shin,2011] Shin, D.-H. and Shin, Y.-J., 2011. Why do people play social network games? *Computers in Human Behavior*, 27(2), pp.852–861.
- [Skeet,2013] Skeet, J., 2013. Implementing the Singleton Pattern in C#. In *C# in Depth, 3rd Edition*. Shelter Island, NY: Manning Publications.
- [Spotify AB,2015] Spotify AB, 2015. Contact - Spotify. <https://www.spotify.com/pt/about-us/contact/> [accessed: Oct. 2015].
- [SuccessBricks, Inc.,2015] SuccessBricks, Inc., 2015. ClearDB Developer Center - Frequently Asked Questions. https://www.cleardb.com/developers/help/faq#general_0 [accessed: Oct. 2015].

- [System, and Krieger,2014] System, K. and Krieger, M., 2014. About Us • Instagram. *Instagram*. <http://instagram.com/about/us/> [accessed: Nov. 2014].
- [Takahashi,2011] Takahashi, D., 2011. Overwolf launches social overlay for online games. *VentureBeat*. <http://is.gd/7Txi33> [accessed: Apr. 2015].
- [Tomar,2006] Tomar, S., 2006. Converting Video Formats with FFmpeg. *Linux J.*, 2006(146), p.10–.
- [Twitch Interactive, Inc.,2015] Twitch Interactive, Inc., 2015. About Twitch. <http://www.twitch.tv/p/about> [accessed: May 2015].
- [Twitter Inc.,2015a] Twitter Inc., 2015. About Twitter. <https://about.twitter.com/> [accessed: May 2015].
- [Twitter Inc.,2015b] Twitter Inc., 2015. GET friends/ids | Twitter Developers. <https://dev.twitter.com/rest/reference/get/friends/ids> [accessed: Oct. 2015].
- [Twitter Inc.,2015c] Twitter Inc., 2015. PIN-based authorization | Twitter Developers. <https://dev.twitter.com/oauth/pin-based> [accessed: Oct. 2015].
- [Twitter Inc.,2015d] Twitter Inc., 2015. REST APIs. <https://dev.twitter.com/rest/public> [accessed: Jun. 2015].
- [Twitterizer,2015] Twitterizer, 2015. Twitterizer. <http://www.twitterizer.net> [accessed: Jun. 2015].
- [Unify Community,2014] Unify Community, 2014. ScreenshotMovie - Unify Community Wiki. <http://wiki.unity3d.com/index.php/ScreenshotMovie> [accessed: Oct. 2015].
- [Unity Technologies,2015a] Unity Technologies, 2015. Application.CaptureScreenshot. <http://is.gd/Nx3NYM> [accessed: Jun. 2015].
- [Unity Technologies,2015b] Unity Technologies, 2015. Submitting your content to the Asset Store. <http://unity3d.com/asset-store/sell-assets> [accessed: Jun. 2015].
- [Unity Technologies,2015c] Unity Technologies, 2015. Unity - Manual: GameObject. <http://docs.unity3d.com/Manual/class-GameObject.html> [accessed: Oct. 2015].
- [Unity Technologies,2015d] Unity Technologies, 2015. Unity - Scripting API: MonoBehaviour. <http://docs.unity3d.com/ScriptReference/MonoBehaviour.html> [accessed: Oct. 2015].
- [Unity Technologies,2015e] Unity Technologies, 2015. Unity - System Requirements. <https://unity3d.com/unity/system-requirements> [accessed: Oct. 2015].
- [Unity Technologies,2015f] Unity Technologies, 2015. What is Unity? <https://unity3d.com/unity> [accessed: May 2015].
- [Wikipedia,2015] Wikipedia, 2015. Main Page. *Wikipedia, the free encyclopedia*. https://en.wikipedia.org/w/index.php?title=Main_Page&oldid=664887982 [accessed: Oct. 2015].
- [Yang,2014] Yang, Y., 2014. Let's Tweet In Unity. <https://www.assetstore.unity3d.com/en/#!/content/536> [accessed: Jun. 2015].
- [zyngaAdmin,2014] zyngaAdmin, 2014. FarmVille. <https://company.zynga.com/games/farmville> [accessed: May 2015].

Annexes

Annex A - Testers Invitational Form

Adventure! The Paladin Order

Alpha Phase: Conceptual Tests

Game Summary

You are a villager, a nobody stowed away by society. Without a family or anyone to support you, you then decided to throw away your life in to the Paladin Order of the Sun an order dedicated to cleanse the world of evil forces. With this video game we wish to provide the players community with a geeky, funny and visually cute experience, satisfying the more casual players, but also reaching out for the more experienced fans who miss the days of Final Fantasy 7 and all of the other turn based games.

About Us

We are a pair of students from Portugal who are currently creating a video game for their master's thesis on Computer Sciences in the specialization of Multimedia and Graphical Systems at the Instituto Superior de Engenharia do Porto.

Game Design

With every passing day there is an increasing number of people playing video games, either due to the evolution of technology towards mobile systems or thanks to the fact that technology is becoming gradually more accessible to everyone. With an ever increasing and differentiated group of players, it becomes more and more important to develop video games that are accessible, challenging, discernible and modular independently of the group of players and the environment in question. Therefore, with this project, we wish to develop a video game whose game design is focused on the decisions of the player. By play testing the game thoroughly throughout its different phases and by presenting different scenarios to the testers we can gauge whether the situation satisfies the conditions we established before or if there is something that needs to be changed. It is vital for the success of this thesis that the test group contains players from different social backgrounds with different expectations, as it is not possible to have a game who is accessible and challenging if we just test a group of highly experienced gamers, disregarding the other social groups. Game design wise, the game is an old school turn-based game with elements of exploration in between.

Social Features

Nowadays, social media contents and the large number of users in social networks are one of the most promising sources for the personalization of gameplay experiences. Furthermore, the invitation of social network friends to influence the game experience with new contents and participation in gameplays can increase the awareness of players' activities outside the game and consequently raise the curiosity of the game to users that are not currently playing. This fact can also contribute to the creation of collaborative environments to improve the game quality of every participant. However,

social media integration can quickly evolve into a simple promotion system for the players who connect their games to social networks accounts, so developers need to carefully consider the benefits and disadvantages of social media as an opt-in experience.

Our players can choose not to use the social media plugin, but if they use, they may benefit with that. We do not force the users to connect to any social network to play and we also do not push users to share information in order to progress in the game. Our goal is to make players look and ask for social integration. Unlike most social sharing systems designed for games our Share Menu interacts with user in non-intrusive way allowing them to choose and share only the best game moments. With this menu, players can share screenshots, videos and even game stories with friends using social platforms. Players can also take advantage of their social communities to improve their game experiences. They can get better game equipment using social craft system or create small parties to help them fighting in the combats with social party system.

Final product

At the end of the development stage we expect to have a playable demo for the game. This demo will contain the very first main quest with a dungeon to explore, and several side quests that heavily impact the world in which the players are in.

Game so far

In the following chapter we will talk a little bit about the state of the project so far. It is worth noting that the project is in a very structural phase, all you see here will not reflect the game in its final stage.

Overworld

The game will have a third-person perspective that the players can use to investigate the world around them. Mechanics wise, there is a smart camera in place, gravity and slide implemented as well as NPC interaction and treasure finding so far.

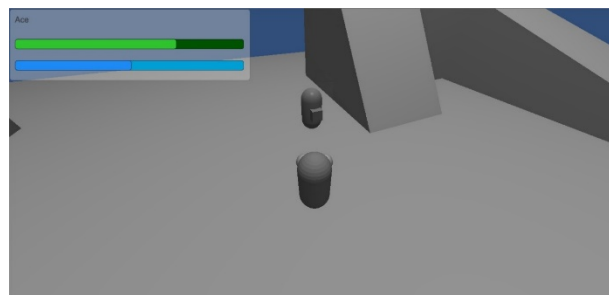


Figure 1 Character in Overworld space

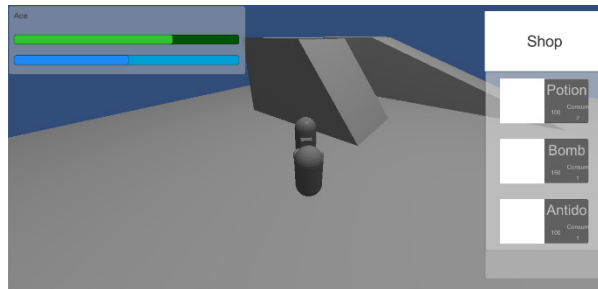


Figure 2 Character interacting with a Shop NPC

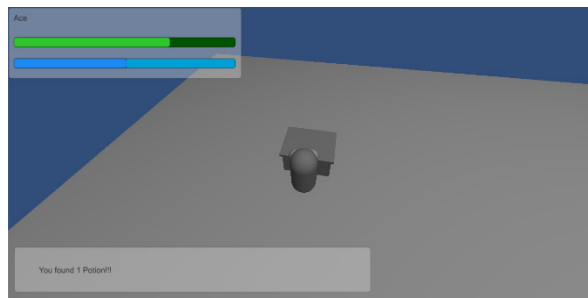


Figure 3 Character found an item!

Combat

Combat is made in a turn-base fashion with a highly intelligent AI. The enemies learn with the players' actions that are made each turn and adapt to defeat the player as quickly as possible.



Figure 4 Early representation of combat

Share Menu

To share videos, screenshots and game history information, all the player needs is a social network account. The players will have access to a menu (by clicking in a share button or share shortcut) similar to the following:

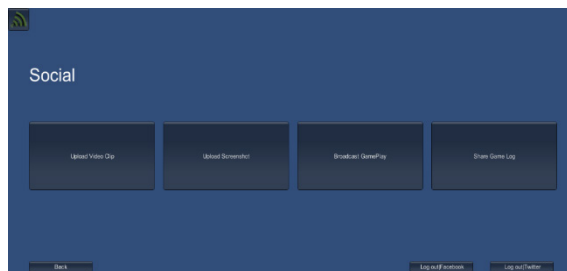


Figure 5 Draft of Share Menu

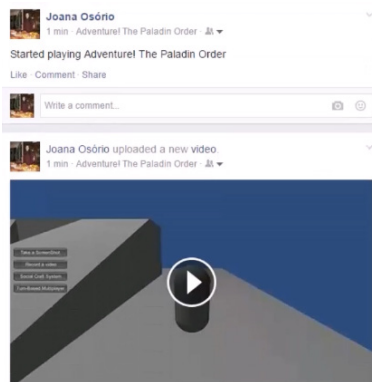


Figure 6 Share experiences on Facebook

Screenshots

The players can take screenshots while playing a game and upload them to social networks. To take a screenshot players have to press and hold Screenshot button (or screenshot shortcut) for at least 1 second.

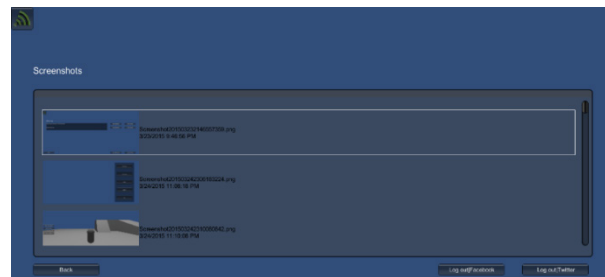


Figure 7 Draft of Screenshot Menu

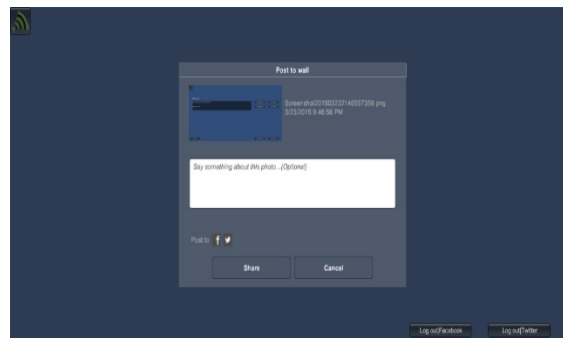


Figure 8 Draft of Share Screenshot Dialog

Videos

The players can record short video clips (1~15 minutes) while playing a game and upload them to social networks. To record players have to press and hold the record button (or record shortcut) for at least 1 second.

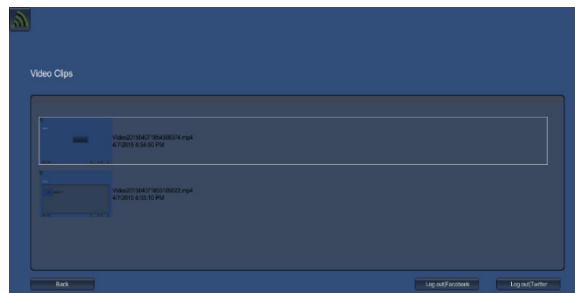


Figure 9 Draft of Video Menu

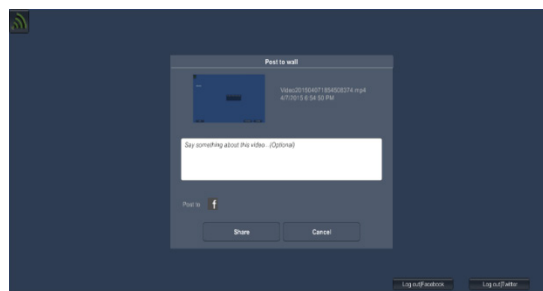


Figure 10 Draft of Share Video Dialog

Game Log

Players can share information about trophies and game events on social networks. This information is presented in a history board similar to the following:

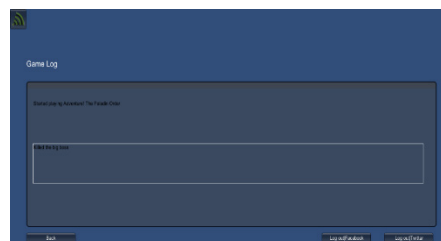


Figure 11 Draft of Game Log

What is expected from you

When performing tests, these are divided in to two separate groups:

- Extensive tests;
- Short tests.

Extensive tests consist of testing big chunks of the game that normally require more time and a more in-depth feedback. Here are some examples:

- Newly implemented combat mechanics;
- Complex social features;
- Different engine quirks;
- Story and quest related adventures.

Short tests consist of testing short features of the game that are spontaneously added as a reflect to something that has been forgotten or fixed, or in the case of something that was suggested by one of the testers

and thus was modified or added after it was already implemented. Here are some examples:

- Bug fixes;
- Functionality corrections;
- Small improvements made based off testing feedback.

Furthermore, we expect to receive as much feedback as possible via honest constructive criticism as well as reporting any bugs, errors or quirks that should be attended.

Functions to perform

As a tester, you are expected to participate in all of the different testing phases through which the game is going to pass through in its development. However, it is fine if you find yourself in a position where you cannot perform a certain test and thus have to skip it.

This next paragraph is very important. Each testing phase will work differently from one another, some will only consist of filling questions from a form to help plan future features, while others we will ask you to perform a skype call with you (screen sharing is all we need) so we can assist, see and analyze your gameplay, your reactions and probably complains. This is what it's called a testing session, where you play and test a specific part of the game and we analyze and decide how to fix or improve it based on your experience.

In the case of which you don't feel comfortable in performing a testing session with us, we ask you to record a playing session of yourself testing and report (along with the video) where you had the most difficulty, what are yours complains, as well as other aspects.

Form

If you have reached this chapter and you are still interested in becoming a tester for our game, please fill the form bellow with the information we ask you so you can be added to our testers list and we can contact you in the future.

Leave this field in blank if you want to keep as anonymous, this field is only required for university formal purposes.

Personal Name: _____

This next field is obligatory as we require some form of public identity to identify you.

Nickname*: _____

Provide us with some form of contact, the most accessible method we ask for is your email.

Your email*:

In this field please describe what your availability to perform these tests is, not only in a time perspective but also in any foreseen problems. We expect the development of the game to finish sometime around October so if there will be any instances in which you are lacking time or aren't available for some reason let us know so we can plan ahead.

Availability*:

If you have any additional comments you would like to add, please write them in this section.

Additional comments:

Contacts

After filling the form in the previous chapter, please email this document back to us so we can receive your information and add you to our list of testers. If you have any questions you'd like to ask before sending us this document, don't hesitate in emailing us.

Contact email: contactclockworkinc@gmail.com

Personal website: <http://tdmei.azurewebsites.net/>

Annex B - Alpha Testing Survey GameNshare Sections

Alpha Test V0.01 - Conceptual Analysis

This is the first test phase for the game Adventure! The Paladin Order. In the form we will ask you a very diversified group of questions that will help us determine what is the best approach towards building our game. The questions are all conceptual and are ideas that the game developers wish to implement. Through this form we can determine what ideas to develop (and which to leave behind), as well as what is the best way to develop them in the most interesting way in your perspective as a player.

*** Required**

A. Product Awareness

This section focus on how interesting and well built the ideas and fundamentals of this project were made.

1. Was the document clear enough for you to understand everything about the game? *

Rate how easy it was to understand what the game is about and what it will be (mechanics wise) based off of the document given.

1 2 3 4 5

Didn't understand Understood everything

2. How likely are you to recommend this game to your friends or colleagues? *

If the game was on a retail stage and free to play, how likely would you recommend a friend to purchase it.

1 2 3 4 5

Wouldn't recommend Highly recommend

3. Do you think the game is distinguishable and different enough from all the other games in the market? *

Given the current market of video games would you consider this game to be unique enough to be different, or too similar to all other games.

1 2 3 4 5

Unoriginal, not unique Very unique

Alpha Test V0.01 - Conceptual Analysis

* Required

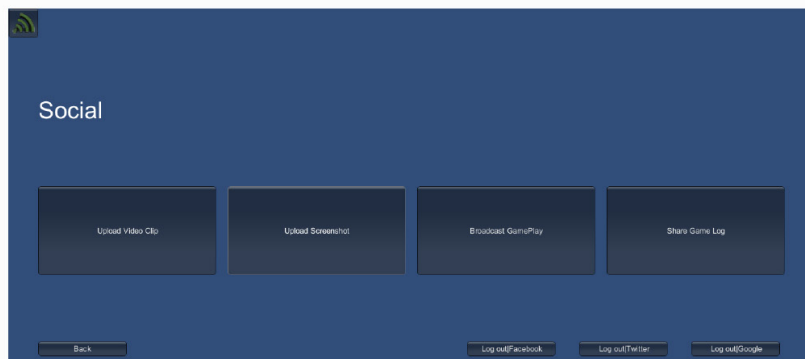
D. Social Overlay

Our social overlay allows players to easily capture and share on multiple social networks screenshots, videos and even game-related stories. In the future, they may also get better game equipment using craft system and use passive party system/friends buffers to help them fighting in the combats or with game quests.

For this section we ask that you see each of the questions presented as a social feature for the game, and then select what you think would you like to have as USER (and not as tester) in that situation/feature.

1. Share Menu

To share videos, screenshots, gameplays and game log information, all the player needs is a social network account. The players will have access to a menu similar to the following:



1.1. What would you change in this menu?

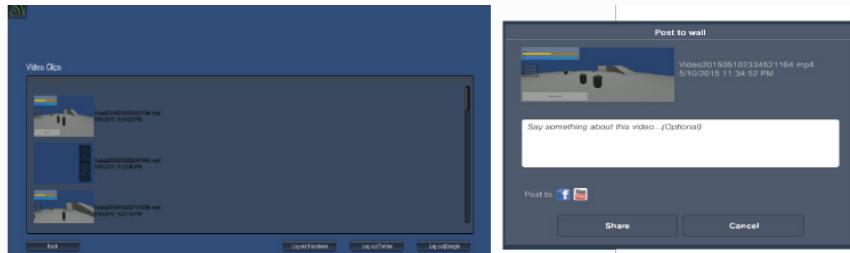
In this initial phase, please try to ignore aspects related to usability and visual appeal and focus only in the functionality dimension. Is there any feature that you are looking and you can't find? What do you complain about that doesn't exist yet?

2. Upload Video Clips

The players can record short video clips (3s~15 minutes) while playing a game and upload them to social networks. To record players have to press and hold the record button (or record shortcut) for at least 1 second.

To upload a video clip the user will need only to:

1. Click the Upload Video Clip menu;
2. Select a video;
3. Select a social network (e.g. Facebook, Twitter, Youtube);
4. (Optional step) Enter a comment;
5. Click in the share button.



2.1. Would you like to have the recent minutes of gameplay continuously and automatically recorded? *

Would you like to have temporary videos of the recent 5 minutes and then choose if you want to save and upload them to online services (e.g. facebook and youtube)?

1 2 3 4 5

Very Probably Not Definitely

2.2. Would you like to have the best moments of your game automatically recorded? *

Suppose that we can collect information of your best moments and automatically recorded them. Would you like us to do that?

1 2 3 4 5

Very Probably Not Definitely

2.3. Would you like to post to specific groups/communities? *

Would you like to post your videos, for example, in specific gaming groups or communities?

1 2 3 4 5

Very Probably Not Definitely

2.3.1. Could you provide examples?

If you would like to post videos to specific groups/communities, could you provide examples of these groups/communities?

2.4. In what other social networks/online services would you like to share your gameplay videos?

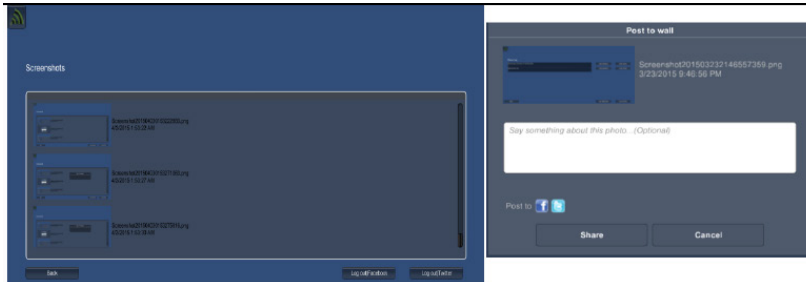
In addition to Facebook, Youtube (and eventually Twitter), in which other social networks/online services would you like to share your game videos?

3. Upload Screenshots

The players can take screenshots while playing a game and upload them to social networks. To take a screenshot players have to press and hold Screenshot button (or screenshot shortcut) for at least 1 second.

To upload a screenshot the user needs to:

1. Click the Upload Screenshot menu;
2. Select a screenshot;
3. Select a social network (e.g. Facebook, Twitter);
4. (Optional step) Enter a comment;
5. Click in the share button.



3.1. Would you like to have the best moments of your game automatically captured? *

Suppose that we can collect information of your best moments and automatically take a print screen of them. Would you like us to do that?

1 2 3 4 5

Very Probably Not Definitely

3.2. Would you like to post to specific groups/communities? *

Would you like to post your screenshots, for example, in specific gaming groups or communities?

1 2 3 4 5

Very Probably Not Definitely

3.2.1. Could you provide examples?

If you would like to post screenshots to specific groups/communities, could you provide examples of these groups/communities?

3.3. Would you like to use private messages to share your print screens? *

Would you like to use, for example, to use facebook chat/twitter direct message to share your print screens with your friends?

1 2 3 4 5

Very Probably Not Definitely

3.4. In what other social networks/online services would you like to share your screenshots?

In addition to Facebook, Twitter (and eventually Google+), in which other social networks/online services would you like to share your screenshots?

Alpha Test V0.01 - Conceptual Analysis

* Required

4. Broadcast Gameplay

In the future, players will be able to broadcast (stream) gameplay online on Youtube (and probably in Twitch and Ustream).

To broadcast a gameplay the user needs to:

1. Select Broadcast gameplay menu;
2. Select a social network (e.g. Youtube, Twitch and Ustream);
3. Prepare Broadcast configurations;
4. Select "Start Broadcasting".

4.1. How likely are you to recommend this game using a stream channel? *

If the game was on a retail stage how likely would you recommend a friend to purchase it using a stream channel?

1 2 3 4 5

Wouldn't recommend Highly recommended

4.2. Would you like to have a "facecam" feature? *

Would you like us to use your webcam to record yourself playing the game during the live stream?

1 2 3 4 5

Very Probably Not Definitely

4.3. Would you like to use microphone?

Would you like to use microphone while playing to record commentaries?

1 2 3 4 5

Very Probaly Not Definitely

4.4. In what other social networks/online services would you like to stream?

In addition to Youtube, Twitch and Ustream, in which other social networks/online services would you like to stream?

5. Share Game Log

Players can share information about trophies and game events on social networks. This information is presented in a game log board similar to the presented below.

To share a story the user needs to:

1. Click the Game log menu;
2. Select a story;
3. Select a social network (e.g. Facebook, Twitter);
4. (Optional step) Enter a comment;
5. Click in the share button.



5.1. Would you like to have a "share in real-time" button to share all game stories automatically? *

Suppose that you could activate a "share in real-time" button to share all game stories automatically, would you like to have that? Remember that this button would be disabled by default and could be turned on/off anytime.

1 2 3 4 5

Very Probably Not Definitely

5.2. Would you like to post to specific groups/communities? *

Would you like to post your game log stories, for example, in specific gaming groups or communities?

1 2 3 4 5

Very Probably Not Definitely

5.2.1. Could you provide examples?

If you would like to post to specific groups/communities, could you provide examples of these groups/communities?

5.3. Would you like to use private messages to share your game log stories? *

Would you like to use, for example, to use facebook chat/twitter direct message to share your game log stories with your friends?

5.4. In what other social networks/online services would you like to share your stories?

In addition to Facebook, Twitter (and eventually Google+), in which other social networks/online services would you like to share your game log stories?

6. Crafting

Crafting is a method to do new inventory items using different materials that can be found in the game. Suppose that you can also craft items faster with the help of social networks.

6.1. Craft System with social interactions

Think that your character can craft new inventory items in two ways: looking for different materials in the game or using social networks as a resource. Please remember the second way is optional!

6.1.1. How likely would you be to use this craft? *

Would you like to use the craft described above?

1 2 3 4 5

Not at all likely Very likely

6.1.2. Would you like to use Facebook in craft? *

Think that your character is crafting an item and you really want to craft an item faster, would you like to somehow use your Facebook account to do that?

- Yes.
- No.

6.1.3. Besides Facebook, which social networks would you like to use? *

Think that your character is crafting an item, would you like to somehow use others social networks as resource to help you out?

- Twitter.
- Google+/Youtube.
- Twitch.
- None.
- Other:

Alpha Test V0.01 - Conceptual Analysis

* Required

7. Passive Party System and Friends Buffers

The players can do special combat combos and get special friends buffers. This combos and friends buffers are optional and can be used in fights or game quests.

7.1. Passive Party System and Friends Buffers with social interactions

The players can do some special combat combos with the help of social network friends. These friends act as passive party members and must perform certain actions (e.g. play the game or accept a request) to achieve a specific combo/buffer. This combos can be used in boss/mobs fights or game quests.

7.1.1. How likely would you be to use to use party system and friends buffers? *

Would you like use social networks as resource to help you in game quests or boss/mobs fights?

1 2 3 4 5

Not at all likely Very likely

7.1.2. Would you like to use Facebook in party system? *

Would you like to use Facebook as resource to help you in games quests or fights?

- Yes.
- No.

7.1.3. Besides Facebook, which social networks would you like to use? *

Would you like to somehow use others social networks as resource in game party system?

- Twitter.
- Google+/Youtube.
- Twitch.
- None.
- Other:

Alpha Test V0.01 - Conceptual Analysis

E. Suggestions and Feedback Ideas

We are always open to all kinds of feedback, in fact we encourage you to give us all of your ideas so we can further improve our project! Use this section to provide us with any new ideas we could implement or if you think we ought to do something in a different way.

1. What would you change?

From what you know of the game, what features would you change or do differently.

2. Are our future ideas appealing?

Given what was presented in this form that is not yet implemented in the game, would you suggest any changes or if you find these ideas unappealing how could we make them more interesting.

3. What would you like to see added to the game in the future?

If you have any ideas of your own that you would like to see added to the game, write to us so we can consider adding them in a future version of the game.

Annex C - Beta Testing Survey GameNshare Sections

Beta Test

This questionnaire takes approximately 20 minutes (with the visualization of videos included) and involves a 15 minutes' interaction with the game. It consists on 7 parts: A- Product Awareness, B- Character Creator, C- Overworld scenario, D- Combat system, E- Social Overlay - share menu, F- Social overlay - Party system and friends skills and G- Suggestions and Feedback Ideas.

To carry out this survey you need to:

- 1- Download our game: <http://tdmei.azurewebsites.net/Content/Game/APO.zip> (This zip contains a .exe with the game and a folder with game data. We recommend you to run game with resolution 1024x600.)
- 2- Have an account on Twitter or Facebook and optionally on Youtube (Google).
- 3- When requested in the tests (inside the game), temporarily accept our applications permissions in social networks.
- 4- When requested in the tests (inside the game), temporarily share some game contents in social networks.
- 5- Follow <https://twitter.com/Joana129> or <https://twitter.com/APOClockworkInc> on twitter OR add <https://www.facebook.com/joana.s.osorio> as facebook friend.
- 6- (Optional) If you have doubts about new features, bugs or limitations of social overlay, please download the following notes: http://tdmei.azurewebsites.net/Content/Docs/S.O.S_SocialOverlay.txt

It is worth noting that the focus of the game design aspect of the project (chapters, B,C and D) has shifted from a real game oriented experience to more of a study of an adaptive game design. With this, keep in mind that some aspects (such as the user interface) are less emphasized.

Thank you for your interest in being tester of our game and for filling out the questionnaire. Your answers will be confidential and very important for an appropriate framework of our master thesis.

* Required

A. Product Awareness

This section focus on how interesting and well built the ideas and fundamentals of this project were made.

1. Was the document clear enough for you to understand everything about the game? *

Rate how easy it was to understand what the game is about based on the document given.

1 2 3 4 5

Didn't understand Understood everything

2. How likely are you to recommend this game to your friends or colleagues? *

Consider that the game was on a retail stage and free to play.

1 2 3 4 5

Wouldn't recommend Highly recommend

3. Do you think the game is distinguishable and different enough from all the other games in the market? *

Given the current market of video games would you consider this game to be unique enough to be different, or too similar to all other games.

1 2 3 4 5

Unoriginal, not unique Very unique

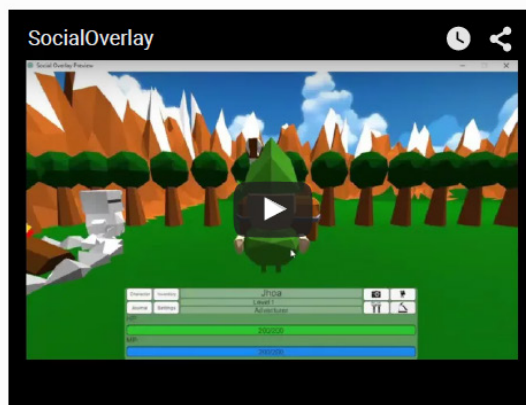
Beta Test

* Required

E. Social Overlay - Share menu

In this section we will ask you to perform three tasks: 1 - upload a video and share in social networks; 2- upload a screenshot and share in social networks; 3- share a gamelog story. In these three tests use the same character of the sections B-C. Take your time in the tests, but beware that we may request you some time estimates about the communication with game and social networks.

Please watch this 3-min trailer (without sound) about share menu features before testing the application.



1. Upload Video Clips

In this task you will record a video up to 15 minutes and share it on twitter, facebook and/or youtube.

1. Click F2 to start recording a video.
2. Click F2 to stop video recording.
3. Wait for message "A new video was added to your APO_Videos folder."
4. Press "TAB" to open share menu.
5. Click on "Upload Video Clip" option.
6. Select your video from video list.
7. Optional - Write a post description and change video title.
8. Share video in twitter or/and facebook. Please note that to share a video to twitter you need to upload it to your youtube account.
9. Accept applications permissions.
10. Optional - Share video to youtube.
11. Wait for success sharing messages.
12. Check the posts in twitter, facebook and/or youtube.

1.1. Was application feedback for video sharing actions adequate? *

Did you receive the success sharing messages and all the feedback that you need?

1 2 3 4 5

Strongly disagree Strongly agree

1.2. Did you controlled the sharing process? *

Everything went as expected? The posts were adequate?

1 2 3 4 5

Strongly disagree Strongly agree

1.3. Was our social overlay intrusive? *

Do you think that our social overlay was too intrusive in the upload of video clips?

1 2 3 4 5

Yes, it was No, not at all

1.4. Was social network sharing successful? *

Could you share videos on Twitter and/or Facebook and/or Youtube?

1 2 3 4 5

Strongly disagree Strongly agree

1.5. Did the system allow you to personalize your actions? *

Could you change video title and insert a post description?

1 2 3 4 5

Strongly disagree Strongly agree

1.6. The record and upload of videos has performed systematically without failures? *

Did you successful record and upload a video without failures or game crashes?

1 2 3 4 5

Strongly disagree Strongly agree

1.7. Was the speed of communication between the game and the social networks adequate? *

The speed of communication between the game and facebook, twitter or youtube was adequate or too slow?

1 2 3 4 5

Strongly disagree Strongly agree

1.7.1. How long was your video? *

What was your video length (HH:MM:SS) ?

1.7.2. How long did it take to login and upload to social networks? *

Approximately, how long it took to login in the social networks? And to upload the video?

1.7.3. In a perfect world how long should it take? *

In a perfect world how long you think should take the login? And the upload?

1.8. Please give us your suggestions or complaints about this feature.

Beta Test

* Required

2. Upload Screenshots

Instructions/Recommendations:

Take a screenshot and share it on Twitter and/or Facebook.

1. Use print screen button or PrtSc shortcut to take a printscreen
2. Wait for message "Screenshot captured"
3. Press "TAB" to open share menu.
5. Click on "Upload Screenshot" option.
6. Select your screenshot from screenshots list.
7. Write a post description (optional step).
8. Share screenshot in twitter or/and facebook.
9. Wait for success sharing messages.
10. Check the posts in twitter and/or facebook.

2.1. Was application feedback for screenshots sharing actions adequate? *

Did you receive the success sharing messages and all the feedback that you need?

1 2 3 4 5

Strongly disagree Strongly agree

2.2. Did you controlled the sharing process? *

Everything went as expected? The posts were adequate?

1 2 3 4 5

Strongly disagree Strongly agree

2.3. Was our social overlay intrusive? *

Do you think that our social overlay was too intrusive in the upload of screenshots?

1 2 3 4 5

Yes, it was No, not at all

2.4. Was social network sharing successful? *

Could you share screenshots on Twitter and/or Facebook?

1 2 3 4 5

Strongly disagree Strongly agree

2.5. Did the system allowed you to personalize your actions? *

Could you insert a post description?

1 2 3 4 5

Strongly disagree Strongly agree

2.6. The capture and upload of screenshots has performed systematically without failures? *

Did you successful capture and upload a screenshot without failures or game crashes?

1 2 3 4 5

Strongly disagree Strongly agree

2.7. Was the speed of communication between the game and the social networks adequate? *

The speed of communication between the game and facebook and/or twitter was adequate or too slow?

1 2 3 4 5

Strongly disagree Strongly agree

2.7.1. What was the size of your screenshot? *

What was the size in KB of your screenshot?

2.7.2. How long did it take to upload? *

Approximately how long it took to upload the screenshot?

2.7.3. In a perfect world how long should it take? *

In a perfect world how long you think should take the upload?

2.8. Please give us your suggestions or complaints about this feature.

Beta Test

* Required

3. Share Game Log

Instructions/Recommendations:

Share a story to twitter or facebook.

1. Press "TAB" to open share menu.
5. Click on "Share game log" option.
6. Select a story from gamelog.
7. Write a post description (optional step).
8. Share screenshot in twitter or/and facebook.
9. Wait for success sharing messages.
10. Create new story using add new story button.
11. Share screenshot in twitter or/and facebook.
12. Wait for success sharing messages.
13. Check the posts in twitter and/or facebook.

3.1. Was application feedback for gamelog sharing actions adequate? *

Did you receive the success sharing messages and all the feedback that you need?

1 2 3 4 5

Strongly disagree Strongly agree

3.2. Did you controlled the sharing process? *

Everything went as expected? The posts were adequate?

1 2 3 4 5

Strongly disagree Strongly agree

3.3. Was our social overlay intrusive? *

Do you think that our social overlay was too intrusive in the sharing of stories?

1 2 3 4 5

Yes, it was No, not at all

3.4. Was social network sharing successful? *

Could you share stories on Twitter and/or Facebook?

1 2 3 4 5

Strongly disagree Strongly agree

3.5. Did the system allowed you to personalize your actions? *

Could you create your own stories and insert post descriptions?

1 2 3 4 5

Strongly disagree Strongly agree

3.6. The upload of stories has performed systematically without failures? *

Did you successful capture, create and share a story without failures or game crashes?

1 2 3 4 5

Strongly disagree Strongly agree

3.7. Was the speed of communication between the game and the social networks adequate? *

The speed of communication between the game and facebook and/or twitter was adequate or too slow?

1 2 3 4 5

Strongly disagree Strongly agree

3.7.1. What was the length of your story? *

How many characters of text (letters) have your story? You can also paste your story, if do not want to count.

3.7.2. How long did it take to upload? *

Approximately how long it took to upload the story?

3.7.3. In a perfect world how long should it take? *

In a perfect world how long you think should take the upload?

3.8. Please give us your suggestions or complaints about this feature.

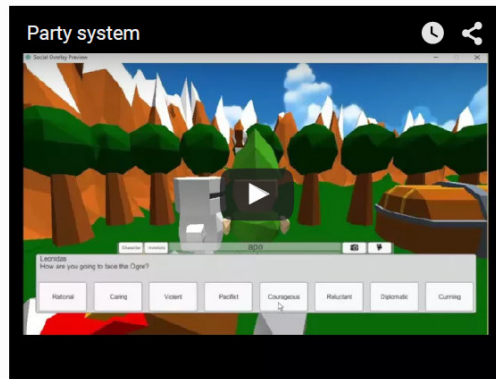
Beta Test

* Required

F. Social overlay - Party system and friend skills

In this section we will ask you to perform two tasks: 1- use friend skills in combat; 2- give a free pass to a friend. In these two tests use the same character of the sections B-C. Take your time in the tests, but beware that we may request you some time estimates about the communication with game and social networks/web servers.

Please watch this 3-min trailer (without sound) about friends skills features before testing the application.



1. Use Friends skills

Instructions/Recommendations:

Use facebook and/or twitter friends.

- 1.1. To synchronize twitter friends, follow @Joana129 and/or @APOClockworkinc on twitter. You can only synchronize with twitter friends that play the game. If you failed to synchronize twitter friends say it in your answers and try to refresh or sync them again.
- 1.2. To synchronize facebook friends add <https://www.facebook.com/joana.s.osorio> on facebook. You can only synchronize with facebook friends that play the game. If you failed to synchronize twitter friends say it in your answers and try to refresh or sync them again.
2. Use "P" shortcut or click in party button (button with two person). You may open and close party menu several times, to test the speed communication with web servers and social networks.
3. Select a friend.
4. Talk with guard NPC and choose, for example, a violent behaviour.
5. Talk with one of mages NPCs (only one will talk with you).
6. Equip items, press E and then click in the equipments.
7. Talk with ogre.
8. Click in the friend photo to use it combat.

1.1. Was application feedback for party system adequate?

Did you receive all the feedback that you need?

1 2 3 4 5

Strongly disagree Strongly agree

1.2. Did you controlled the party menu actions?

Everything went as expected?

1 2 3 4 5

Strongly disagree Strongly agree

1.3. Was our social overlay intrusive? *

Do you think that our social overlay was too intrusive in the party menu?

1 2 3 4 5

Yes, it was No, not at all

1.4. Did you synchronized successful your social networks friends?

Did you synchronized successful your social network friends (Twitter and/or Facebook) that play the game?

1 2 3 4 5

Strongly disagree Strongly agree

1.5. The party menu has performed systematically without failures?

Did you synchronized your social network friends that play the game, without failures or game crashes?

1 2 3 4 5

Strongly disagree Strongly agree

1.6. Was the speed of communication between the game and the social networks adequate? *

The speed of communication between the game and facebook and/or twitter was adequate or too slow?

1 2 3 4 5

Strongly disagree Strongly agree

1.6.1. How many friends that play the game do you have? *

How many facebook and/or twitter friends that play the game do you have?

1.6.2. How long did it take to synchronize your friends? *

Approximately how long it took to synchronize your friends?

1.6.3. In a perfect world how long should it take? *

In a perfect world how long you think should take the synchronize?

1.7. Please give us your suggestions or complaints about this feature.

Beta Test

* Required

2. Friends permissions

Instructions/Recommendations:

Give a free pass to a friend

- 1.1. To synchronize twitter friends, follow @Joana129 and/or @APOClockworkInc on twitter.
- 1.2. To synchronize facebook friends add <https://www.facebook.com/joana.s.osorio> on facebook.
2. User "P" shortcut or click in party button (button with two person). You can open and close party menu several times, to test the speed communication with web servers and social networks.
3. Click in a "friend permissions" button.
4. Set permissions of a friend to "This user can use my skills for an unlimited number of times"
5. Wait for success message.
6. Close party menu (press P or Esq).
7. Reopen party menu (click in party button or P).
8. Click in step 3 "friend permissions" button.
9. Check if friend have its permissions set to "This user can use my skills for an unlimited number of times". If this doesn't happen, please say it in your answers and try changing your friend's permissions a few more times.

2.1. Was application feedback for friends permission adequate?

Did you receive all the feedback that you need?

1 2 3 4 5

Strongly disagree Strongly agree

2.2. Did you controlled the friends permissions menu actions?

Everything went as expected?

1 2 3 4 5

Strongly disagree Strongly agree

2.3. Did you changed successful friends permissions?

Did you changed successful your friend permissions to free pass?

1 2 3 4 5

Strongly disagree Strongly agree

2.4. The friends permissions menu has performed systematically without failures?

Could you see and change your permissions friends without failures or game crashes?

1 2 3 4 5

Strongly disagree Strongly agree

2.5. Was the speed of communication between the game and the web services adequate? *

When you reopen the menu to see if friend permissions changed after you give him a free pass, they changed instantly or it was too slow?

1 2 3 4 5

Strongly disagree Strongly agree

2.5.1. How long did it take to change your friends permissions? *

Approximately how long it took to change your friends permissions?

2.5.2. In a perfect world how long should it take? *

In a perfect world how long should it take to change your friends permissions?

2.6. Please give us your suggestions or complaints about this feature.

Beta Test

G. Suggestions and Feedback Ideas

We are always open to all kinds of feedback, in fact we encourage you to give us all of your ideas so we can further improve our project! Use this section to provide us with any new ideas we could implement or if you think we ought to do something in a different way.

1. What would you change?

From what you know of the game, what features would you change or do differently.

2. What would you like to see added to the game in the future?

If you have any ideas of your own that you would like to see added to the game, write to us so we can consider adding them in a future version of the game.

Annex D - InForum 2015 Paper

Social Media Integration in Video Games: A Social Overlay for Desktop Games

Joana Osório and Filipe Pacheco,

Polytechnic Institute of Porto (ISEP-IPP), Computer Engineering Department, Porto, Portugal
{1100594, ffp}@isep.ipp.pt

Abstract. The ever increasing popularity of social media makes them a promising source for the personalization of gameplay experiences. Furthermore, involving social network friends in a game can greatly enrich the satisfaction of the player and also attract potential novel players to a game. This paper describes a social overlay designed for desktop games. It allows players to easily capture and share on multiple social networks screenshots, videos and even game-related stories. Unlike most social sharing systems our social overlay is designed to interact with the user in a non-intrusive way allowing him/her to be in complete control of what is shared. Our goal is to make players look and ask for social integration. The development of this social overlay will allow players to take full advantage of their social communities to improve their gaming experience.

Keywords: Social media integration, Desktop games, Social overlay, Social networks, Gameplay experiences

1 Introduction

Gamers often compete, collaborate and share stories of gaming accomplishments. Over the years this leads to game developers realizing that they have to support the growth of gaming as a social hobby. This fact can also contribute to the creation of collaborative environments to improve the game quality of every participant [1]. However, social media integration can quickly evolve into a spam-like promotion system [2] for the players who connect their games to social networks accounts, so developers need to carefully consider the benefits and disadvantages of social media as an opt-in experience.

To promote social media integration in desktop games we have developed and tested a social overlay in the context of a small multiplatform game project, called Adventure! The Paladin Order (APO) [3], however it intends to be applied to different desktop game environments. Our social overlay is designed for desktop players of all age groups, but with a focus in young people (typically aged from 18 to 25) that are regular users of social networks like Facebook and Twitter [4]. With our overlay, players will always be able to choose not to use the social features, but if they opt to use them, they will have benefits such as the capacity to capture and share their game experiences in different social networks and in the future, the creation of collaborative environments with social network friends for the personalization of their game experiences.

This paper describes a user-friendly, non-intrusive and multiplatform social overlay for desktop games to allow players to enhance their gaming experiences using the social communities. It is structured in 7 sections. Following this general introduction section 2 reviews the state-of-the-art of the related literature and compares the existing systems with our social overlay prototype. Sections 3-6 are devoted to present our social overlay including its requirement specifications, system design, prototype implementation and prototype alpha evaluation. Finally, in section 7 we summarize the conclusions and main future work.

2 Background on Social Media Integration

Full social media integration in games is often seen in online games that can be played through social networks. Usually these games include multiplayer features or asynchronous gameplay mechanics. These games are frequently implemented in the web browser or in mobile device apps [5]. Social network games are among the most popular games in the world, with games such as FarmVille [6], Mafia Wars [7], The Sims Social [8] and Candy Crush [9].

Recently, the gamers have realized that they can expand the gaming community directly through YouTube, Twitter, Facebook and other popular social media sites. Gamers now often record and upload their gameplays to YouTube to receive comments and share gaming strategies and principles. These new habits created business opportunities that are starting to be explored. For example, Achievement Hunter, a Rooster Teeth spinoff, is exclusively dedicated to the upload gameplay videos to the Let's Play YouTube Channel [10]. Live-streaming, particularly Twitch.Tv, has also become popular lately [11], it fostered the appearance of “gamer celebrities” that congregate in real time a large numbers of fans to watch and comment the game live. Gamer Sacriel was one of the top leaders on Twitch.tv in 2013 and he has a regular audience of more than 2,000 simultaneous viewers on his game streams. This has allowed him to become a professional gamer making his income solely on revenues from YouTube and Twitch.tv [12].

To address this new reality, social media integration in video games evolved to a new dimension and it is increasingly available on game consoles, such as Microsoft Xbox [13] or Sony's PlayStation 4 [14] and also on desktop games [15, 16]. Typically, the social features for these types of platforms are media sharing, live broadcasting and exchanging playing experiences.

2.1 Strategies of social media integration

Kietzmann et al. [17] presented a model that stratifies different types of social media accordingly to the focus placed on seven functional blocks: identity, conversations, sharing, presence, relationships, reputation and groups. Exploring this model can help developers and companies to better design their social network targeting strategies. Based on this model Facebook is used to promote long-term relationships with the target audience and can be integrated in almost all contexts. In this platform there is a strong connection between identity and relationships. In the gaming context Facebook-users have specific profiles for self-promotion (real identity vs. virtual identity) and share motivations, causes, events and activities with friends. On other hand, Twitter is more focused in the conversation than in the identity of users. In this platform, users "tweet" short and real-time messages which are often scores, real-time status updates or game stories with no obligation to answer. Kietzmann model can equally be applied to YouTube, which is centred in the sharing of previously recorded videos. Initially, YouTube only allowed users to upload homemade videos, but now is also used to improve the marketing of companies and for sharing game videos. Unlike Facebook that uses "likes" for reputation, YouTube reputation is based on "view counts" and "ratings". In this platform the group relationships really matter, so this type of platform is also indicated for user experience sharing.

2.2 Similar Systems and Prototype Progress

Presently, there are several successful stories of the use of social integration tools for game promotion and enhancement. In the last decade these tools vary greatly on its characteristics, complexity and dedicated financial investment.

One of the first relevant examples of game-related social media integration was revealed on November 17th of 2009 upon the presentation of Xbox 360. It had native applications for Facebook, Twitter and Last.fm [13]. The Facebook application allowed gamers to update their status, comment, like and view friends' pictures. Additionally, “Xbox Live Friend Finder” allowed gamers to identify Facebook friends that used Xbox Live [18]. For unclarified reasons, on October 16th 2012 Facebook and Twitter applications were removed from Xbox Live limiting users to access Facebook and Twitter through the Xbox 360 web browser [19].

In August 2011, Overwolf Ltd. announced a free social overlay for Windows desktop games [15] that included a variety of social features inside the game environment such as the ability to browse contents, share pictures and statuses on Facebook and Twitter (including game-related stories and pictures), upload and watch YouTube videos and make Skype calls. It also allowed game recording and live gameplay streaming to Twitch. Currently, Overwolf provides official support for more than 1000 different Windows desktop games [20] and a store with a wide range of free social network applications. However, some of these applications for example the ones for conversation or ongoing game detection are still limited in functionality comparatively to other social overlays and there is a complete lack of versions for other platforms other than Windows. In our social overlay we also provide support for the common share and screen capture actions, however to bypass the Overwolf compatibility limitation, our overlay was developed using Unity3D [21] and offers support to Windows, Mac OSX and Linux. Later in this year, Echobit released Evolve [22] a gaming platform for Windows desktop games with a hotkey accessible social

overlay. Beyond the basic social overlay functions (e.g. gameplay sharing and chatting) this platform developed a new feature called “Party” that can be initiated for multiple games at the same time. It allows players to search the game statuses of their friends and create parties (teams) in the games that they are playing. Presently, Evolve officially supports more than 4300 games, four social networks: Twitter, Facebook, YouTube and Twitch and for four platform distribution services: Steam, Battle, Origin and PSN [16]. The major limitation of Evolve beside the absence of a multiplatform version, is the lack of APIs and SDKs to easily customize and integrate this tool in games. Similar to this overlay, our prototype includes a “share” hotkey and allows integration of Twitter, Facebook and YouTube, which according to Ann Hurk [23] are the most used social media platforms. Contrary to Evolve, we can export our social overlay tools as Unity3D assets [24] and integrate them in different Unity3D development environments for desktop games.

More recently, Sony took a decisive step forward in social media integration in their games by incorporating a "Share button" in the back of the PlayStation 4 (PS4) game controller. This button was revealed during the worldwide presentation of PS4 on February 20th of 2013 [14] and allows players to easily share their gaming experiences directly to the PlayStation network, Facebook, Twitter, Twitch and USTREAM. This "Share" button is the first existing social feature of its kind and is analogous to our “share” hotkey, however instead of only share gameplay experiences, we will also provide asynchronous game mechanics to integrate social networks friends in the creation of new game elements and in the obtainment of different game bonus.

3 Requirements Specifications

The main goal of our social overlay is to enhance the gameplay experience of the players using social media communities as resource. The involvement of friends and communities can promote sharing of game strategies and achievements among players and avoid eventual irritation or boredom states in the player that may drive him to stop playing [25].

Requirements were divided into two types: functional requirements and non-functional requirements. The first type of requirements presents what the prototype should do and the second type describes how the prototype should work.

The functional requirements were prioritized into three categories (essential, conditional and optional), according to their importance for the prototype. Table 1 presents the prioritization scale of functional requirements, as well their descriptions.

Table 1. Prioritization scale of functional requirements.

Scale level	Description
Essential	Critical requirements, without them the product is not acceptable.
Conditional	These requirements would improve the product quality.
Optional	The requirements that would be nice to have, but are not indispensable to the product.

As the social overlay is being integrated in Adventure! The Paladin Order (APO) for tests purposes, the agreement on essential requirements was made informally. In what regards to conditional and optional requirements the prioritizing was based on the opinions of APO’s test team that were collected using a survey. APO’s test team are an external group of 10 young players with ages between 19 and 25 that were recruited by our Clockwork Inc. team [3] for the preliminary testing phases. We considered “optional requirements” all the requirements with average between 1 and 2.5. The requirements with average between 2.5 and 5 were considered “conditional requirements”. The opinions of APO’s test team about use case-based requirements related with “Broadcast Gameplay” have an average result of 3.5 (Fig.1) in a scale of 1 to 5. Regarding to the “Craft” use case-based requirements, their opinions have an average result of 4.1 (Fig.1) in a scale of 1 to 5. Finally, the opinions about use case-based requirements associated with “Use passive party system” have an average result of 2.1 (Fig.1) in a scale of 1 to 5.

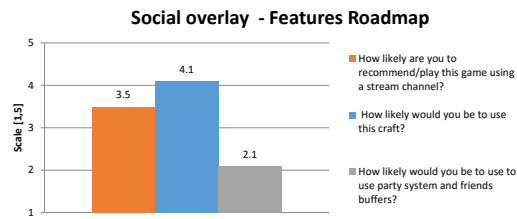


Fig. 1. Average of APO’s test team opinions (scale of 1 to 5) about broadcast gameplay (orange), craft system (blue) and passive party system (gray).

Below we present our use case diagram, as well the prioritization scale of use case-based requirements.

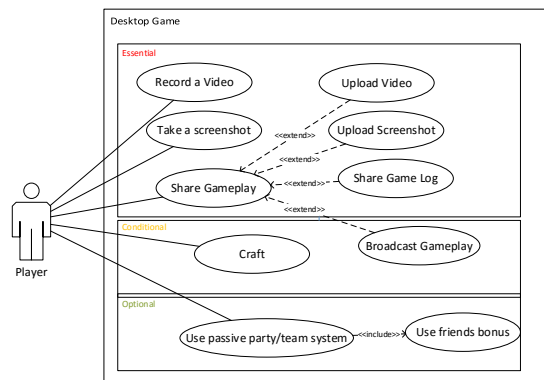


Fig. 2. Use case diagram of social overlay prototype with prioritization scale.

“Record video” and “Take a screenshots” use cases, classified as essential, allow players to record short videos (up to 15 minutes) and take screenshots during the game. The resulting videos and screenshots can be uploaded to social networks using the “upload video menu” and “upload screenshot menu”.

The use case “Share Gameplay”, classified as essential, resulted from the “Share Menu” feature. This menu has four options: (1) “Upload Video”; (2) “Upload Screenshot”; (3) “Broadcast Gameplay”; (4) “Share Game Log”. It is presented by clicking the social button from the game options menu or the Share Menu shortcut. “Upload Video” is classified as essential and resulted from the “video clips menu” feature. It allows players to select and upload gameplay videos to social networks. Similarly, “Upload Screenshot” results from an “upload screenshot menu” feature. It allows the social network upload of player-selected screenshots. The “Share Game Log” use case resulted from the “share game log menu” feature and is dedicated to the social networks sharing of achievements, game stories and other information. The “Broadcast gameplay” use case, classified as conditional, resulted from the “broadcast gameplay menu” feature. It allows live social network broadcasting (streaming) of gameplays.

Our social overlay will also allow player to craft items or collections faster using social networks. Crafting is a method to create new game items or to complete collections, using different game elements that can be found in the game such as materials, pieces and cards. This feature resulted in “Craft” use case that was classified as a conditional. It enables players to send private messages asking missing game elements from their social network friends.

Finally, the passive party/team system, classified as an optional requirement, allows social network friends to act as passive members by performing actions even when not playing the game. These actions result in different game bonus that can be used to improve game-related experiences, e.g. get more attack for each friend that have the game installed. These bonuses are optional and can be used to improve game-related experiences.

Among the different non-functional requirements of the social overlay we highlight the following: (1) support APO integration and full compatibility with Unity3D [21] for desktop games; (2) full Windows compatibility and when possible multiplatform support (Windows, Mac OS X and Linux).

4 System Design and Architecture

This section describes the design and architecture of the social overlay based on the requirements that were proposed.

Our system can be divided into three components. The first one is the desktop game client which includes the desktop game application and the social overlay tools. The second component is the web server. The web server includes the web services that receive and send information to the social overlay and the database that is used to store the information about the players' requests. Lastly, the fourth component are the external social networks which include all the public media services. Fig. 3. shows the architecture of our system.

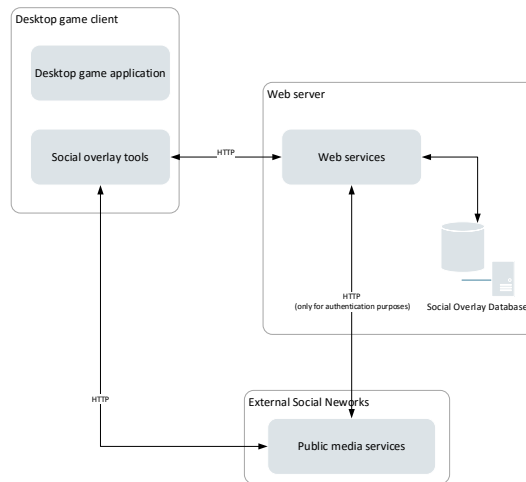


Fig. 3. System architecture for the social overlay integration in a desktop game.

4.1 Desktop game client

Desktop game client consists in two components: the desktop game application component, which is the part of desktop game client that contains the game's core logic and mechanics and the social overlay tools, which include all the social overlay features. To maintain some game styles attributes, the UI components or styles of each of these two components can be shared.

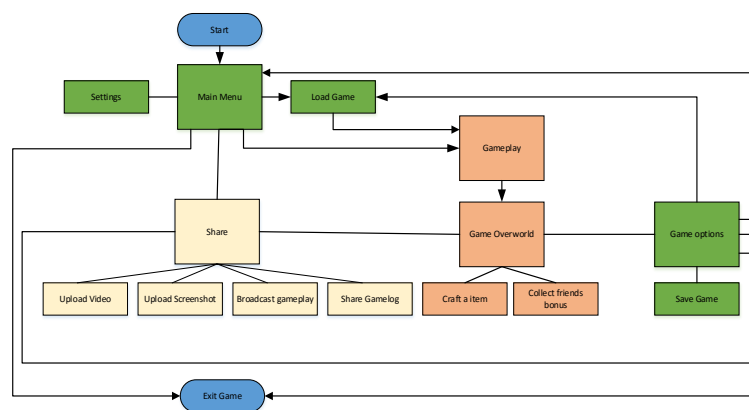


Fig. 4. Navigation chart/Game flow of social overlay integration with APO.

Fig. 4. presents a simplified navigation chart that includes integration of share, craft and passive party system tools with desktop game application. The first one includes a menu for the sharing of videos, screenshots and game-related stories and for the broadcasting of the gameplay. Players can access to the Share Menu by clicking in the share option of the game menus or directly using the Share Menu hotkey. The second and third tools are only accessible inside the game overworld, which is the area that

interconnects all the game locations, e.g. combat areas. With the craft, players can request help of social network friends' to create new inventory items and finally with passive party, players can collect social network friends' bonus to use game quests or fights. Additionally, social overlay allows the video record and screenshot capture in any part of the game using hotkeys or through the record and capture buttons. This game flow can be adapted in accordance to different game environments to create specific game tools or to customize the existing ones.

4.2 Social overlay tools

The social overlay tools include support for read and write data in the public media services (e.g. Graph API, Twitter API and Google API). In addition, the social overlay tools communicate with web services for grant permissions to players' social networks accounts and also to store and receive information about their social network requests. These tools are Unity3D Assets [24] for desktop games and are written in C#.

The choice of additional development technologies for the implementation of social overlay tools was made only based in the evaluation of the technical implementation specifications and in the verification of which tools responded better to the project needs. Screenshot Capture was implemented using the native function [26] of Unity3D for this effect. This function is compatible with Windows, Mac OSX and Linux. Regarding to video recording was concluded that the two best development options were: (1) the use of DirectShowNET [27]; (2) the use of FFmpeg [Tomar, 2006]. As DirectShowNET is only available for Windows applications was decided to use the FFmpeg library, because it supports Windows, MAC OSX and Linux. This library was implemented with the help of Ruslan-B FFmpeg.AutoGen [29], which is an open-source wrapper for FFmpeg 2.5.2, compatible with Mono. In what regards to share, craft and passive party tools were analysed technologies for three types of social network integrations: Facebook integration, Twitter integration and Google integration. As these social overlay features are intimately related to the "external social networks" component, they will be presented in the next section.

4.3 External Social Networks

The social overlay tools, as well the web server are capable to interact with existing social networks. The credentials of the users are only used inside of social networks sites to acquire access tokens and are not stored. Our system includes support for the integration of Facebook's Graph API, Twitter REST API and Google APIs.

The **Facebook's Graph API** [30] is the low-level HTTP-based Application Programming Interface (API) of Facebook Platform that allow developers to read and write data into Facebook, to interact with it developers' applications need to use the OAuth 2.0 protocol [31] to obtain a Facebook access token for each user. Regarding to Facebook integration was analysed a solution proposed by Paul Price [32]. This solution uses a web server to handle the tokens exchange with HTTP Requests and a Mono 2.6 [33] compatibility version of Facebook SDK C# [34] to access to Graph API. The main disadvantage of the solution provided by Paul Price is that every time Graph API methods are used a new access token is requested, even if it is not expired. To overcome this disadvantage was decided to implement a reformulation of Paul Price solution with the following changes: (1) The internet Connection is validate every time that the players do Facebook requests; (2) The login prompt and generation of access token are only requested the first time that players use Facebook in game; (3) The last Facebook session is stored between game sessions until user decides to logout or revokes application access; (4) Facebook session is validate before the user makes Facebook requests to check if a new access token need to be requested; (5) the wrapper of Facebook SDK C#, as well the web server implementation were reformulated according to technical implementation specifications.

The **Twitter REST API** [35] is the main application programming interface that allows developers to read and write Twitter platform data, to interact with it developers' applications need to use OAuth 1.0 protocol [36], similar to Facebook this protocol is used to acquire an access token for each user. As none of libraries analysed were compatible with mono 2.6, to build a more flexible integration of Twitter was decided to do a new implementation that combines both "Let's Twitter in Unity" asset [37] and Paul Price Facebook solution [32]. "Let's Twitter in Unity" is a free and open-source Unity asset that helps developers to deal with OAuth 1.0 protocol authentication of Twitter API 1.1. This asset only provides a pin-based authentication flow, so users have to manually insert a pin code in the game. To overcome this limitation our Twitter integration uses a web server and includes the following steps: (1) The internet Connection is

validated every time that gamers do Twitter requests; (2) A HTTP POST calling a request token to the API is done. The login prompt and generation of request token are only requested the first time that players use Twitter in game; (3) default user browser is opened and he logins with Twitter account and accepts application permissions; (4) user is redirected to a web server page and success message is shown, meanwhile the OAuth verifier is passed back to the game; (5) the request token and OAuth verifier are converted to a access token; (6) The access token is used to do HTTP requests to the Twitter API; (7) OAuth tokens are stored until they are valid or until the user decides to do logout.

Google APIs [38] is a group of Application Programming Interfaces (APIs) provided by Google that allow the communication of third-party applications with Google services. Similar to Facebook, Google APIs use the OAuth 2.0 protocol for authentication and authorization. For the integration of Google services was decided to develop a new implementation that combine the use of Google APIs with HTTP Requests and Paul Price Facebook solution [32]. Similar to the solutions presented to Facebook and Twitter, this implementation contains the following steps: (1) The internet Connection is validated every time that gamers do Google requests; (2) A HTTP GET calling a request authorization token to the API is done. The login prompt and generation of request token are only requested the first time that players use Google account in game; (3) default user browser is opened and he logins with Google account and accepts application permissions; (4) user is redirected to a web server page and success message is shown, meanwhile the authorization token obtained via login dialog is exchanged for a access token and a refresh token; (5) the desktop game polls the web server to check if access token for that user is already available; (6) The access token is used to do HTTP requests to Google APIs; (7) Access and refresh tokens are stored until they are valid or until the user decides to do logout. Contrary to Facebook and Twitter APIs, an additional access token verification step is done, because Google access tokens have a short live time and they need to be updated from time to time using a specific refresh token [39].

4.4 Web Server

The web server uses a basic ASP.NET MVC website and a MySQL database for the monitoring of players requests. In addition to the support of different HTTP request methods (e.g. GET, POST, PUT), the ASP.NET MVC application also includes a JSON library [40] to send and receive data information in this format.

The web server contains two different tiers: the user interface component (UI component), which is the part of the web server that include the views for the presentation of user interface layer to the players and the backend component, which is the part of the web server that provides the business and logic layer. The UI component provides the interface of the web browser instances in the desktop game client and also presents the basic information about the game and privacy policy. The backend component is the core of the web server and is responsible for ensuring the communication between the desktop game client and the other two components of the web server: the Database and the UI Component. Besides this, the backend component also include support for authentication and authorization of the desktop game client application in the social networks. Since this functionality is common to all social overlay tools the web server provides a simple and clear interface for each social network that can be used by all the modules of the social overlay.

The database is responsible for the persistence and monitoring of players' activity in social networks, for example, for obtain the statuses of social networks requests and for prevent duplicate social networks requests. This external database is also used as an information repository for the public players' information in social networks (name, gender, username and user ID).

5 Prototype Implementation

This prototype was developed in C# with Unity3D, additionally we used FFMPEG libraries for capturing system and a modified version of Facebook SDK C# for the connection with the Facebook. Our web application on the server is developed in ASP.NET and use MVC web application to ensure communication between the social overlay of the game client and the database and external social networks components.

The social overlay can be executed on Windows, Mac OSX and Linux, but we currently working in a compatibility version of the record feature for Mac OSX and Linux, because it requires different installations of FFMPEG libraries. Below is presented a compatibility table for each social overlay feature:

Table 2. Compatibility table of social overlay features.

Feature	Windows x86/x64	Mac OS X x86/x64	Linux x86/x64
UI components	✓	✓	✓
Capture Screenshots	✓	✓	✓
Record Videos	✓	✗	✗
Facebook Share (upload videos, screenshots and stories)	✓	✓	✓
Twitter Share (upload videos, screenshots and stories)	✓	✓	✓
YouTube Share (upload videos)	✓	✓	✓

As the prototype is in first stage only the essential requirements were fully implemented, these requirements will be presented in four sections: (1) Share Menu; (2) Record and Upload Videos; (3) Capture and Upload Screenshots; (4) Share Game Log.

5.1 Share Menu

To share videos, screenshots, gameplays and game-related stories, all the player needs is a Facebook, Twitter or Google account. The players have access to the following menu (Fig.5):

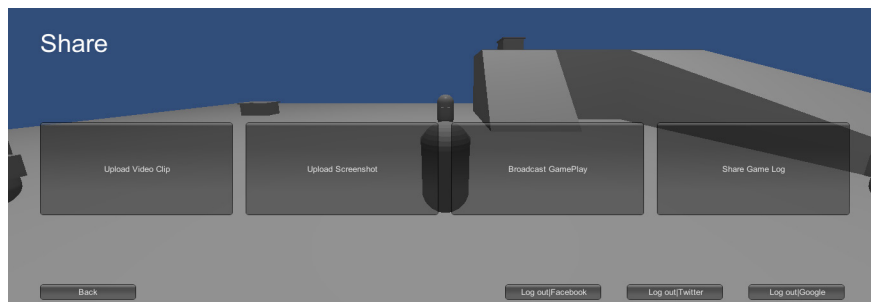


Fig. 5. Share Menu of APO' social overlay

This menu includes “Upload Video Clip”, “Upload Screenshot”, “Broadcast Gameplay” and “Share Game Log” options, but currently the option “Broadcast Gameplay” is not enabled.

5.2 Record and Upload Videos

The players can record short video clips (3s~15 minutes) while playing a game and upload them to social networks. To record players have to press and hold the record button (or record shortcut) for at least 1 second.

To upload a video clip (Fig. 6) the player needs to: (1) Select a video; (2) Select a social network (Facebook, YouTube or Twitter or all the three); (3) (Optional step) Enter a comment; (4) (Optional step) Change video title; (5) Click in the share button.

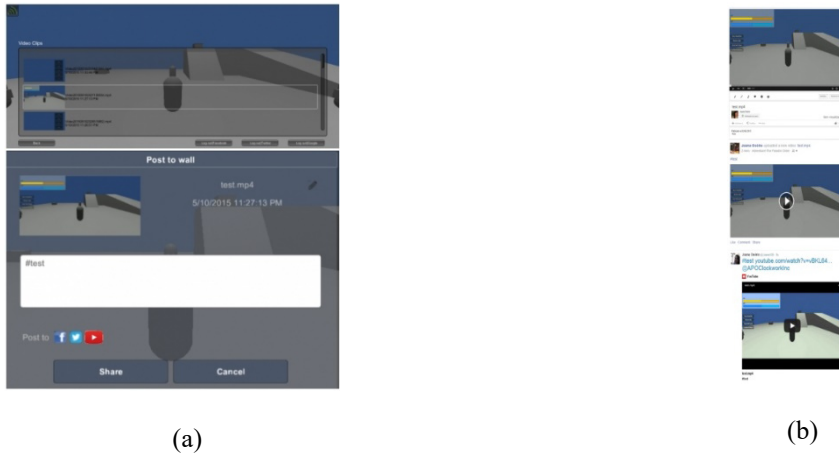


Fig. 6. Upload an APO' video to Facebook, Twitter and YouTube using the social overlay: (a) select a video to upload to social networks; (b) samples of posted videos in the social networks.

In the case of Twitter, as this platform does not currently support the uploading of videos in desktop platforms, users are asked to upload their videos to their youtube account and share the link to Twitter (Fig. 7). This option can be saved for next requests.

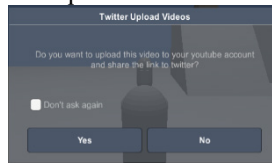


Fig. 7. User prompt to request permission for upload video to players' YouTube accounts and share the link in Twitter.

5.3 Capture and Upload Screenshots

The players can take screenshots while playing a game and upload them to social networks. To take a screenshot players have to press and hold Screenshot button (or screenshot shortcut) for at least 1 second.

To upload a screenshot (Fig. 8) the player needs to: (1) Select a screenshot; (2) Select a social network (Facebook, Twitter or both); (3) (Optional step) Enter a comment; (4) Click in the share button.

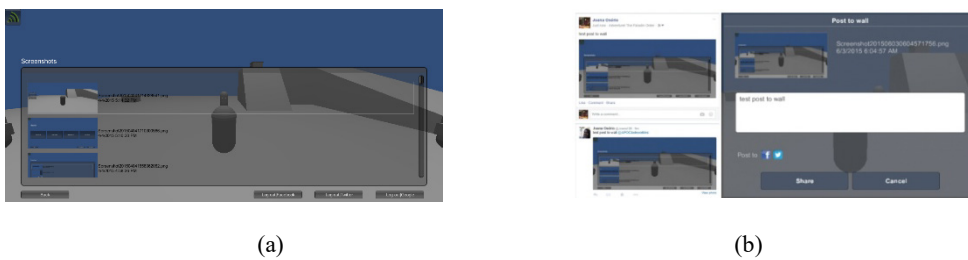


Fig. 8. Upload an APO' screenshot to Facebook and Twitter using the social overlay: (a) Select a screenshot to upload to social networks; (b) Share the selected screenshot to Facebook and Twitter with message "test post to wall".

5.4 Share Game Log

Players can share information about trophies and game events on social networks. Game-related stories are captured and stored by the desktop game application automatically and then they are read by the social overlay game log tool. By default, social overlay game log tool already captures some common stories of the games such as "started playing a game" event. To share a story (Fig. 9) the user needs to: (1) Select a story; (2) Select a social network (e.g. Facebook, Twitter); (3) (Optional step) Enter a comment; (4) Click in the share button.



Fig. 9. Upload story to Facebook and Twitter using the social overlay: (a) Share windows of game log option after select “Started playing” story; (b) “Started playing” story in Twitter and Facebook.

6 Alpha testing

The prototype of the social overlay based on the presented implementation, has undergone a preliminary testing phase that we called alpha testing. During this phase the social overlay among with Adventure! The Paladin Order (APO) game were evaluated with a survey using Google Forms, by a small number of players aged between 19 and 25 that are members of the APO’s test team. The testers received a link with game documentation and instructions. The testing was focused on the user needs and workflow desires for this kind of system more than on usability and graphics, and the users executed the tests by themselves without direct or indirect supervision. After testing they filled up the Google Form. Although the main purpose of the alpha testing was to identify the main limitations of our overlay, this testing also gave some insights to product awareness and future work.

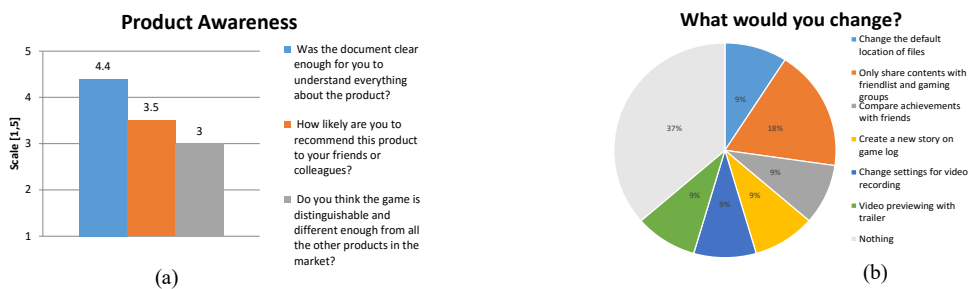


Fig. 10. APO’ test team opinions: (a) Product awareness (scale of 1 to 5); (b) Product problems and limitations (open answer).

Regarding the product awareness (Fig. 10 (a)) testers found that the documentation and overall of the social overlay tools work well, however they did not consider our product distinguishable enough. We think that with the roadmap features planned we will manage to improve this aspect.

The major part of testers (37%, Fig. 10 (b)) did not report any limitations or problems of the features that are currently implemented, but some testers (18%, Fig. 10 (b)) do not like to post directly to their public social network wall and would rather post only to their friend list or gaming groups. A minor part of them (9%, Fig. 10 (b)) also want to change the default path where videos and screenshots are stored, compare achievements with friends and create personalized stories in game log. Finally, they request to change the settings for video recording, for example change the video frame rate (9%, Fig. 10 (b)) and to view a trailer of the video (9%, Fig. 10 (b)) instead of showing only a thumbnail. These limitations will be analysed and prioritized for future developments.

Our Clockwork Inc. team also suggested some roadmap features for the options of Share Menu that are already implemented, from which we highlight the following: (1) record the recent minutes of gameplay continuously and automatically recorded; (2) record the best moments of the game automatically; (3) capture screenshots of the best moments of the game automatically; (4) post to specific group/communities; (5) use private message to share screenshot; (6) use private message to share game log message. All these suggestions were classified by the testers with an average score of 3.7 or above in a scale of 1 to 5. Besides the roadmap features suggested by our team APO’s test team would like to have: (1) support for Stream;

(2) support for Skype; (3) support for specific game community such as Achievement hunter and Speedrunning community; (4) support for Reddit; (5) support for Vine.

7 Conclusion and Future Work

In this paper, we presented a prototype of a social overlay for desktop games that was tested and developed in the context of Adventure! The Paladin Order game, in order to provide social media resources for the personalization of gameplay experiences of the players.

This prototype is in its first stage (alpha version), therefore only the essential requirements were implemented and significant development is required to improve these requirements and implement the other three missing requirements: “Broadcast gameplay”, “Craft an item” and “Use passive party system”.

The next step of this work, during the beta phase implementation, is the evaluation and prioritization of the suggestions and problems reported by the test users. The beta testing phase will conduce to a bigger engagement of the APO’s test team in the game. All the suggestions, problems and bugs reported will be collected using online surveys and individual test sessions with screen sharing or alternatively with recorded videos, so we can observe and analyse the reactions and complains for each tester. In addition, regarding to the recording of videos and the sharing of videos on Twitter we need to improve some aspects related to the compatibility and usability, because currently players cannot record in Linux and Mac OSX systems and to share videos on Twitter the player’s Google account is needed. Although our service does not address anyone under the age of 13 (“Children”), our targets are players of different age groups, so we also have to take special care in the developing of the user interface which needs to be more player-friendly and predictable.

With this first alpha testing analysis we conclude that this social overlay will not only play an essential role in the integration of a social overlay for multiplatform desktop games (Windows, Linux and Mac OSX), but it will also contribute to the development of social media integration in games.

Acknowledgments. Our thanks go to Tiago Magalhães, creator of “Adventure! The Paladin Order” game and also for APO’s test team for all the help and dedication in the construction of this prototype. This work was supported by Instituto Superior de Engenharia do Porto (ISEP) and supervised by Professor Filipe Pacheco.

References

1. Konert, J., Göbel, S., Steinmetz, R.: Video Game Personalization via Social Media Participation. In: Göbel, S. and Wiemeyer, J. (eds.) Games for Training, Education, Health and Sports. pp. 35–46. Springer International Publishing (2014).
2. Hayati, P., Potdar, V., Talevski, A., Firoozeh, N., Sarenche, S., Yeganeh, E.A.: Definition of spam 2.0: New spamming boom. In: 2010 4th IEEE International Conference on Digital Ecosystems and Technologies (DEST). pp. 580–584 (2010).
3. Clockwork Inc.: Adventure! The Paladin Order, <http://tdmei.azurewebsites.net/>.
4. Duggan, M., Ellison, N.B., Lampe, C., Am, Lenhart, a, Madden, M.: Demographics of Key Social Networking Platforms, <http://www.pewinternet.org/2015/01/09/demographics-of-key-social-networking-platforms-2/>.
5. Shin, D.-H., Shin, Y.-J.: Why do people play social network games? *Comput. Hum. Behav.* 27, 852–861 (2011).
6. zyngaAdmin: FarmVille, <https://company.zynga.com/games/farmville>.
7. Hof, R.: Social Games: From the Farm to the Mafia, <http://is.gd/PkAIZK>, (2010).
8. Lynley, M.: Electronic Arts’ The Sims Social hits 4.6 million daily players a week after launch, <http://is.gd/U8K8UZ>.
9. Andrew, W.: Half a billion people have installed “Candy Crush Saga,” <http://is.gd/H7qWtG>, (2013).
10. Fjællingsdal, K.: Let’s Graduate - A thematic analysis of the Let’s Play phenomenon. (2014).
11. Pires, K., Simon, G.: YouTube live and Twitch: a tour of user-generated live streaming systems. In: Proceedings of the 6th ACM Multimedia Systems Conference. pp. 225–230. ACM (2015).
12. Brunelle, W.: Social Media and Video Games, <http://is.gd/iNRL2K>.
13. Mack, C.: Facebook and Twitter Go Live on Xbox 360, <http://is.gd/Px19hr>, (2009).
14. Kelion, L.: PlayStation 4 console announced by Sony in New York, <http://www.bbc.com/news/technology-21526450>, (2013).
15. Takahashi, D.: Overwolf launches social overlay for online games, <http://is.gd/7Txi33>, (2011).
16. Echobit LLC: Evolve, <https://www.evolvehq.com/welcome>.

17. Kietzmann, J.H., Hermkens, K., McCarthy, I.P., Silvestre, B.S.: Social media? Get serious! Understanding the functional building blocks of social media. *Bus. Horiz.* 54, 241–251 (2011).
18. Loguidice, B., Loguidice, C.: *My Xbox: Xbox 360, Kinect, and Xbox LIVE*. Que Publishing (2012).
19. Raby, M.: Microsoft retires Facebook, Twitter Xbox apps, <http://is.gd/Fu4Vil>, (2012).
20. Overwolf: Overwolf supported games, <http://www.overwolf.com/supported-games/>.
21. Unity: What is Unity?, <https://unity3d.com/unity>.
22. K900 (Product Owner of Evolve): Looking for Evolve, the game from Turtle Rock? Sorry, but you're in the wrong place, <http://is.gd/samZVx>, (2015).
23. Hurk, A.M. van den: *Social Media Crisis Communications: Preparing for, Preventing, and Surviving a Public Relations #FAIL*. Que Publishing (2013).
24. Unity: Submitting your content to the Asset Store, <http://unity3d.com/asset-store/sell-assets>.
25. Schell, J.: *The Art of Game Design: A Book of Lenses*, Second Edition. A K Peters/CRC Press, Boca Raton (2014).
26. Unity Technologies: Application.CaptureScreenshot, <http://is.gd/Nx3NYM>.
27. DirectShowNet: DirectShowNet library, <http://directshownet.sourceforge.net/about.html>.
28. Tomar, S.: Converting Video Formats with FFmpeg. *Linux J.* 2006, 10– (2006).
29. Balanukhin, R.: Ruslan-B/FFmpeg.AutoGen, <http://is.gd/yR1ZFW>.
30. Facebook: Graph API Overview, <http://is.gd/RmaFCy>.
31. Hardt, D.: The OAuth 2.0 authorization framework. (2012).
32. Price, P.: Connecting to Facebook from your Unity desktop game, <http://www.ifc0nfig.com/accessing-facebook-unity-game-2/>.
33. Mono: Release Notes Mono 2.6, <http://is.gd/X3nm76>.
34. Facebook SDK for .NET, <http://csharpsdk.org>.
35. Twitter: REST APIs, <https://dev.twitter.com/rest/public>.
36. Hammer-Lahav, E.: *The oauth 1.0 protocol*. (2010).
37. Yang, Y.: Let's Tweet In Unity, <https://www.assetstore.unity3d.com/en/#!/content/536>.
38. Google: Google APIs Explorer, <https://developers.google.com/apis-explorer/#p/>.
39. Google: Using OAuth 2.0 for Web Server Applications offline access, <http://is.gd/umx0rg>.
40. Shrestha, P.: SimpleJson, <https://github.com/facebook-csharp-sdk/simple-json>.