

Felipe Baldino Moreira

**An experiment on conceptual design of
pervasive mobile games using quality
requirements**

Dissertação de Mestrado

Dissertation presented to the Programa de Pós-Graduação em
Informática of the Departamento de Informática, PUC-Rio as
partial fulfillment of the requirements for the degree of Mestre
em Informática.

Advisor: Prof. Bruno Feijó

Rio de Janeiro
July 2015



Felipe Baldino Moreira

**An experiment on conceptual design of
pervasive mobile games using quality
requirements**

Dissertation presented to the Programa de Pós-Graduação em
Informática, of the Departamento de Informática do Centro
Técnico Científico da PUC-Rio, as partial fulfillment of the
requirements for the degree of Mestre.

Prof. Bruno Feijó

Advisor

Departamento de Informática — PUC-Rio

Prof. Monica Maria Ferreira da Costa

Departamento de Informática – PUC-Rio

Prof. Luis Paulo Santos Valente

UFF

Prof. José Eugenio Leal

Coordinator of the Centro Técnico Científico — PUC-Rio

Rio de Janeiro, July 10th, 2015

All rights reserved.

Felipe Baldino Moreira

Graduated from Universidade Federal Fluminense (Rio de Janeiro, Brazil) in Computer Science. He is a Certified Project Manager from PMI since 2007. He Worked at Bradesco Seguros from 2001 to 2011 and at Accenture from 2011 up to 2013 managing projects. Also he worked as a professor at Plinio Leite University in 2002 (Computer Architecture) and at Universidade Veiga de Almeida from 1999 to 2002, where he taught web development. He has expertise in project management, image processing, technology development, and Business Intelligence.

Bibliographic data

Moreira, Felipe Baldino

An experiment on conceptual design of pervasive mobile games using quality requirements / Felipe Baldino Moreira; advisor: Bruno Feijó. — 2015.

88 f. : il. (color.); 30 cm

Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, Departamento de Informática, 2015.

Inclui bibliografia.

1. Informática – Teses. 2. Jogos pervasivos móveis. 3. Smartphones. 4. Reconhecimento de imagem. 5. Visão Computacional. 6. Realidade misturada. 7. Jogo de realidade híbrida. I. Feijó, Bruno. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

To my wife and son

Acknowledgement

To my advisor Prof. Bruno Feijó for the support, everyday kindness and the encouragement to pursue this work.

To the CNPq and PUC-Rio, for the financial support, without which this work would not have been realized.

To all my colleagues that helped to create this project.

Luis Valtente for its pioneer work on quality requirements related to pervasiveness, and hour and hours of conversations about this.

To Omar Koury, who helped me a lot with the conceptual design of the game that I present in this dissertation

To Oi futuro Project for financial support

To ICAD/Vision Lab for suport and motivation.

To all my friends that have inspired and supported me to complete my graduate studies.

To all my colleagues at PUC-Rio,who had me loved this place.

To the Informatics Department staff for the constant help.

To my wife without her support, none of this would have happend.

To my son, that bravely supported my absence.

To everyone who have not hindered.

Abstract

Moreira, Felipe Baldino; Feijó, Bruno (Advisor). **An experiment on conceptual design of pervasive mobile games using quality requirements.** Rio de Janeiro, 2015. 88p. MSc. Dissertation — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Pervasive games is an emerging game genre that mixes up mobile devices (such as smartphones), virtual worlds, and gameplay based on the real world, creating a mixed-reality game. This recent area lacks literature about conceptual design and quality requirements related to pervasiveness – the ultimate and elusive quality that differentiate pervasive mobile games from traditional digital games. In the present work, we discuss the development of a pervasive mobile game using quality requirements related to pervasiveness. Also, we consider those requirements in the entire game project (e.g., design, production, and post-production stages), focusing on the analysis, implementation, and gameplay. We expect that our results could help in improving the current state of design guidelines to develop pervasive mobile games.

Keywords

Pervasive Game; Smartphones; Image Recognition; Computer Vision; Mixed Reality Game; Hybrid Reality Game.

Resumo

Moreira, Felipe Baldino; Feijó, Bruno. **Um experimento no design conceitual de jogos pervasivos móveis usando requisitos de qualidade**. Rio de Janeiro, 2015. 88p. Dissertação de Mestrado — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Jogos pervasivos móveis são um tipo novo de jogo que mistura aparelhos móveis (como telefones celulares), mundos virtuais e atividades que acontecem no mundo real. Nessa área recente, a literatura sobre projeto conceitual e requisitos de qualidade sobre pervasividade são escassas. A pervasividade é a qualidade (difícil de ser definida objetivamente) que difere os jogos pervasivos móveis dos jogos digitais tradicionais. Neste trabalho, nós discutimos o desenvolvimento de um jogo pervasivo móvel usando requisitos de qualidade relacionados à pervasividade. Nós consideramos esses requisitos durante todo o processo de desenvolvimento (i.e., as etapas de projeto, produção e pós-produção), focando na análise, implementação e gameplay. Nós esperamos que os nossos resultados possam ajudar a melhorar guias de desenvolvimento relacionadas a jogos pervasivos móveis.

Palavras-chave

Jogos pervasivos móveis; Smartphones; Reconhecimento de imagem; Visão Computacional; Realidade misturada; Jogo de realidade híbrida.

Contents

1	Introduction	12
1.1	Motivations	14
1.2	Objectives and contribution	15
1.3	The Structure of the dissertation	15
2	Pervasive mobile games	16
2.1	What pervasive games are	16
2.2	Pervasive game classification	18
2.3	Some final remarks on pervasive games	20
2.4	Pervasive mobile games	21
3	Quality requirements for pervasive mobile games	22
3.1	Pervasive features as quality requirements	22
3.2	Descriptions of the quality requirements	24
3.3	Ingress game analysis	26
4	The proposed game	33
4.1	Game concept	33
4.2	Game mechanics	35
4.3	Game design	37
4.4	Computer Vision	43
4.5	Implementation	48
4.6	System architecture and modeling	48
4.7	Results	59
5	Closure	61
5.1	Conclusions	61
5.2	FUTURE WORK	63
6	Bibliography	64
A	Android application class diagram	70
B	Ingress evaluation	72
B.1	Device Independence(DI)	72
B.2	Uncertainty Handling Policy(UHP)	72
B.3	Local Space Redefinition (LSR)	72
B.4	Game Object Tangibility (GOT)	73
B.5	Checklist for Game Pacing(GP)	74
B.6	Involving Non-players(INP)	74
B.7	Usability (USA)	75
B.8	Daily Life Interleaving(DLI)	75
B.9	Game autonomy(GA):	76
B.10	Mobility(MOB)	76
B.11	Cross-mediality(CM)	77

B.12	Persistency(PER)	77
B.13	Social communication(SC)	78
B.14	Conformance to Physical and Social Settings(CPS)	79
B.15	Connectivity (CON):	80
B.16	Game content adaptability(GCA)	80
C	Pervasive game checklist	82
C.1	Device Independence(DI)	82
C.2	Uncertainty Handling Policy(UHP)	82
C.3	Local Space Redefinition (LSR)	82
C.4	Game Object Tangibility (GOT)	83
C.5	Checklist for Game Pacing(GP)	83
C.6	Involving Non-players(INP)	84
C.7	Usability (USA)	84
C.8	Daily Life Interleaving(DLI)	84
C.9	Game autonomy(GA):	85
C.10	Mobility(MOB)	85
C.11	Cross-mediality(CM)	86
C.12	Persistency(PER)	86
C.13	Social communication(SC)	86
C.14	Conformance to Physical and Social Settings(CPS)	87
C.15	Connectivity (CON):	87
C.16	Game content adaptability(GCA)	88

List of figures

1.1	The pervasive game Ingress.	13
1.2	The pervasive game “Uncle Roy All Around You”. Online players (a) collaborating with street players (b) to find Uncle Roy	13
3.1	Pervasive game features(gray) and perspectives(orange) proposed by Valente (Valente, 2011).	23
3.2	Ingress	27
3.3	Handling with uncertainty	28
3.4	Ingress Portal Acquirement	29
3.5	Ingress physical world and virtual world portals	29
3.6	Ingress communities	31
3.7	Ingress social communications screens	32
4.1	The trash can - the main portal to the game	35
4.2	Moving forward into the game	36
4.3	Site - player authentication	36
4.4	Using the mobile application	37
4.5	Interactive trash cans	38
4.6	Image detection fluxogram	44
4.7	Main Detection Code	47
4.8	Overall Application Architecture	49
4.9	Application Activity Diagram	50
4.10	Sequence Diagram of synchronization process between the Android Application and game website	51
4.11	Sequence Diagram of Android Authentication Process used in the fair use doctrine obtained from (Google.2015.f)	51
4.12	Application layout view	53
4.13	Sequence Diagram of site Authentication Process	54
4.14	Sequence Diagram of Game Master entering new missions	54
4.15	IR LED emitter and LED receiver circuit	56
4.16	Android database. The shaded rectangle (yellow) is the part that is implemented in the prototype.	57
4.17	Web database	58
4.18	Arduino circuit installed in the trash can cover	59
A.1	Android game class diagram	71

List of tables

3.1	Pervasive game features	22
4.1	OpenCV Methods and correlation with image tracking activity	44
4.2	Arduino Bill of Materials	55
B.1	Device Independence(DI)	72
B.2	Uncertainty Handling Policy(UHP)	72
B.3	Local Space Redefinition (LSR)	73
B.4	Game Object Tangibility (GOT)	74
B.5	Checklist for Game Pacing(GP)	74
B.6	Involving Non-players(INP)	75
B.7	Usability (USA)	75
B.8	Daily Life Interleaving(DLI)	76
B.9	Game autonomy(GA)	76
B.10	Mobility(MOB)	77
B.11	Cross-mediality(CM)	77
B.12	Persistency(PER)	78
B.13	Social communication(SC)	79
B.14	Conformance to Physical and Social Settings(CPS)	80
B.15	Connectivity (CON):	80
B.16	Game content adaptability(GCA)	81
C.1	Device Independence(DI)	82
C.2	Uncertainty Handling Policy(UHP)	82
C.3	Local Space Redefinition (LSR)	83
C.4	Game Object Tangibility (GOT)	83
C.5	Checklist for Game Pacing(GP)	84
C.6	Involving Non-players(INP)	84
C.7	Usability (USA)	84
C.8	Daily Life Interleaving(DLI)	85
C.9	Game autonomy(GA)	85
C.10	Mobility(MOB)	86
C.11	Cross-mediality(CM)	86
C.12	Persistency(PER)	86
C.13	Social communication(SC)	87
C.14	Conformance to Physical and Social Settings(CPS)	87
C.15	Connectivity (CON):	88
C.16	Game content adaptability(GCA)	88

1

Introduction

Pervasive games is a new form of digital entertainment where the gaming experience is extended out in the real world, mixing up the virtual and real worlds. Ingress (Google.2015.e) is an recent example of pervasive mobile game, where two factions (identified by the colors green and blue) fight for the control of public areas in cities worldwide (Figure 1.1). In Ingress, the players try to link especial public places, like monuments and landmarks (called "portals"), in order to create virtual triangular areas (called "control fields"). Progress in this game is measured by the number of "mind units" captured via those "control fields", which represents the number of people controlled by each faction. Players must be physically near a "portal" on the map to capture it by using a mobile device. Players are rewarded with AP (Access Points) for actions within the game. Another example of pervasive mobile game is "Uncle Roy All Around You" (Flintham et al., 2003), where two groups of players(online and street) collaborate with each other to find a fictional character called Uncle Roy.

While street players physically set out on a journey, online players login on a website that tracks the street players moving around on a virtual 3D world and shows virtual clues that can help the street players. Online players can guide or misguide the street players by sending text messages to them, and street players can send voice messages to online players (Figure 1.2).

In this game, a support team is required to solve network issues, guarantees players safety and occasionally communicate with players. At some point in the game, the players are asked one at a time to enter in a limousine, where an actor playing Uncle Roy, asks him if he can keep a commitment of helping a strange person for one year. The player accepts and the game ends.

In theory, this game investigates some of the social changes brought about by ubiquitous mobile devices, persistent access to a network and location aware technologies.

Pervasive games have many definitions and are far from a consensus. The main explanation for this fact is because pervasiveness is a difficult concept to define. The scenario becomes more complicated when we move towards pervasive mobile games. The reader is referred to the works by Valente et al. (Valente et al., 2013) and Valente (Valente, 2011) for finding an interesting discussion on this matter. In this dissertation we use the definition of pervasive

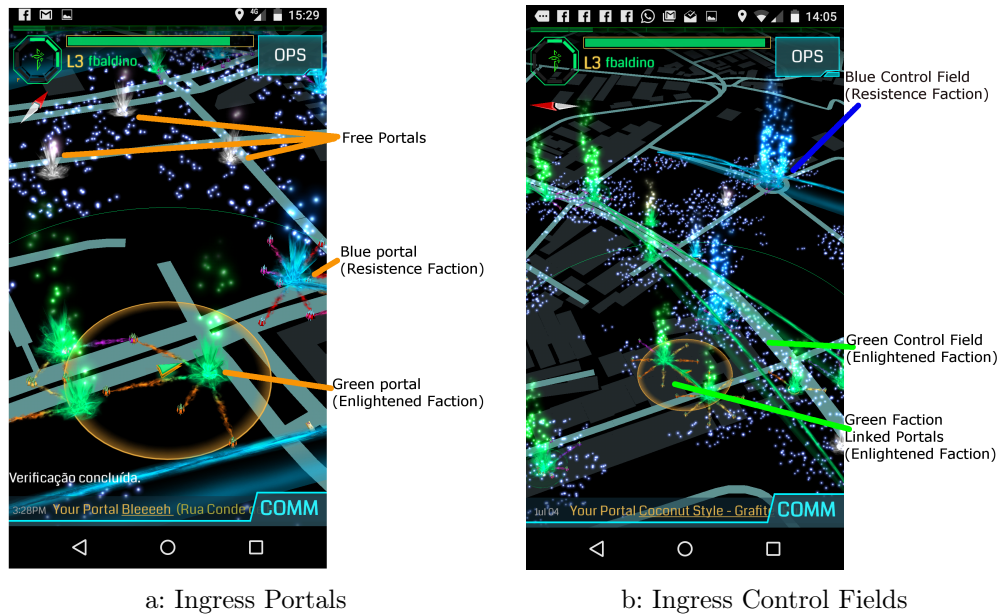


Figure 1.1: The pervasive game Ingress.



Figure 1.2: The pervasive game “Uncle Roy All Around You”. Online players (a) collaborating with street players (b) to find Uncle Roy

mobile games coined by Luis Valente (Valente et al., 2013) , who defined these games as "context-aware games that uses mobile devices". In fact, this dissertation is heavily based on the results by Luis Valente, some of them classified as unpublished works because they are still being processed by international publishers by the time the present dissertation was written.

In this dissertation, we are interested in using general requirements to design high quality pervasive mobile games. Furthermore, we are interested in experimenting with conceptual design of pervasive mobile games, gathering notes on the complete production pipeline of this type of game. Our starting point is the suggestion made by Luiz Valente (Personal Communications)¹ to use his pervasive mobile game features and perspectives defined in the above-mentioned works (Valente et al., 2013; Valente, 2011) as quality requirements related to pervasiveness. This suggestion made by him (personal communica-

tion, 2014)¹ is part of a paper he is currently writing. In this dissertation, we are using these requirements.

1.1 Motivations

The main motivations for the present work are the impressive performance of the game industry, and the attractiveness of the new field of pervasive mobile game design. In the following text we summarize the central aspects that captured the author's attention with respect to games.

In the USA, the game industry was a \$2.6 billion dollars industry in 1996, and in 2010 this wealth grown up to \$20.77 billion dollars. The worldwide game industry had a revenue of \$78,8 million dollars in 2012, and it is expected to have a \$111 million dollar this year, according to (Gartner, 2013, 4Q13). This industry employed more than 120,000 people in 2012 just in US, with an average compensation of \$90,000 per year, according to the 2012 Entertainment Software Association report (Siwek, 2010). Mobile games are the fastest-growing segment of the market, with revenue increasing from \$9.3 billion in 2012 to \$22 Billion this year, according to a recent report (Gartner, 2013, 4Q13). In this scenario, pervasive mobile games seems to represent one of the next new niches to come, because they represent an exciting new way of integrating the digital virtual world with the real physical world.

Games are complexes pieces of software that push computers to their limits. By pushing the hardware limits, they also push software, algorithms, software engineering, computer graphics, artificial intelligence, project management, and almost all computer science disciplines. Because of this, computer games has been a fruitful territory for computer science researchers. Games studies are even more comprehensive than the areas of computer science, because they have their roots in arts and design and many other fields. In fact, game studies, also called gaming theory², is an interdisciplinary field that deals with game design, players, and their role in society and culture.

¹Personal communications are valid references and their authorial rights must be respected. In the academic world, citing personal communications follows the recommendations by APA (American Psychological Association), where this type of reference is very common. Personal communications include letters, memos, personal interviews, telephone conversations, project meetings, emails, messages from discussion lists and electronic bulletin boards. They are cited only in the text (i.e. they are not included in the reference list), because they are not recoverable as an archived material.

²This term is different from "game theory" (a specific field of economics and mathematics that deals with models of conflict and cooperation between intelligent rational decision-making agents). However, current research on game studies incorporates elements of "game theory" .

1.2

Objectives and contribution

The objective of this dissertation is to develop a pervasive game using the quality requirements derived from the works by Luis Valente (Valente et al., 2013; Valente, 2011) (Luis Valente, personal communications, 2014) and observing the main difficulties that emerge in this process. This is an experimental work, in the sense that we are more interested in observing the experiment than in proposing solutions. The development questions treated in this dissertation may help game designers and game engineers to explore more possibilities for pervasive mobile games.

We refrain from presenting a section on related works because we are not proposing a new technique or model. Moreover we are not comparing our prototype with other similar games.

1.3

The Structure of the dissertation

This dissertation is structured as follows.

Chapter 2 discusses the basic concepts that we use throughout this work. Chapter 3 presents the 16 quality requirements that we use to analyse an existing pervasive mobile game (Ingress) and to develop our own game. Chapter 4 presents the creative process we used to develop our game, as well as the game itself. Chapter 5 presents our results, future works, and lessons learned. Appendix A contains the class diagram of our prototype. Appendix B presents our complete analysis of the pervasive mobile game Ingress, considering the quality requirements presented in Chapter 3. Appendix C presents more details about the quality requirements proposed in Chapter 3, as well as their corresponding checklists.

2

Pervasive mobile games

In this chapter, firstly we provide definitions for several terms related to "games" that we use in this dissertation. Then we compare pervasive games with traditional digital games. Finally we present a short discussion on pervasive mobile games.

2.1

What pervasive games are

In this section, firstly we will define game and other general terms used in this work. Also we will compare pervasive games with traditional ones.

2.1.1

Definition of game

The dutch philosopher Johan Huizinga (Huizinga, 1938), back in 1938, proposed seminal academic definitions of "game" and "play". Latter that work was republished in English in 1949, and republished again in 1971. He described "play" as follows:

" ... a free activity standing quite consciously outside "ordinary" life as being "not serious", but at the same time absorbing the player intensely and utterly. It is an activity connected with no material interest, and no profit can be gained by it. It proceeds within its own proper boundaries of time and space according to fixed rules and in an orderly manner. It promotes the formation of social groupings which tend to surround themselves with secrecy and to stress their difference from the common world by disguise or other means." (Huizinga, 1938)

This definition is quite broad, so we use a more specific and recent definition:

"A game is a rule-based formal system with a variable and quantifiable outcome, where different outcomes are assigned different values, the player exerts effort in order to influence the outcome, the player feels attached to the outcome, and the consequences of the activity are optional and negotiable." (Juul, 2003)

2.1.2

The elements of a game

Any game has a set of rules, a boundary separating the real world from the amazing and playful world, and the game play. This boundary was referred by Huizinga (Huizinga, 1938) as a sort of contract he called "magic circle". The use of this term referring digital games was first made by (Salen and Zimmerman, 2004) and defined as – a sort of membrane that encloses the virtual world. The concept of pervasive games, as emphasized in the next sections, blurs and breaks this circle (Montola et al., 2009; Montola, 2012)

The set of rules in game defines what players are allowed to do and what players are not allowed to do. The rules also define the playing field and other elements related to the gameplay. For instance, in a soccer match the game starts with eleven players on each team and the main objective is to score goals without physically harming the others. Players are expected to adhere to all rules, such as not being allowed to touch the ball with their hands (unless the player is a goalkeeper). The rules also define penalties for aggressions and unfair plays. The set of game rules creates a barrier separating the playful and the real ordinary life. This is a part of the Magic Circle Contract proposed by Huizinga (Huizinga, 1938; Montola et al., 2009) that acts like a contract that all players voluntarily agree to respect. This contract defines where and when the game will take place, who will be the players, and the set of rules of the game. This implicit agreement allows a different interpretation of several acts as game activities that have no effect outside the game. Gameplay is the set of events that applies the game rule system and defines the magic circle. It starts with the first interaction of the player and the game system and ends when the player quits the game, obtains victory, loses the competition, does something that unfairly disqualifies the player, or the end game condition is reached. Game plays events should transform all the above situations into an enjoyable experience for all players involved in the game.

In our view a game should be politically correct and we will assume the following definition of political correctness: "... is a term that refers to enforced language, ideas, or policies that address perceived discrimination against political, social or economical groups ("protected classes"). These groups most prominently include those defined by gender, race, religion, ethnicity, sexual orientation, age and disability" (Wikipedia.b, 2014).

2.1.3

Pervasiveness

Pervasiveness as defined in the (Your Dictionary, 2015) is: - "The state or quality of being present in all parts of a particular thing or place." so, pervasive game is a game that is spreaded around as defined above by Montola (Montola et al., 2009) :

- "A pervasive game is a game that has one or more salient features that expand the contractual magic circle of play spatially, temporally, or socially" (Montola et al., 2009).

Comparing traditional digital games and pervasive games is a difficult task, because the definition of a pervasive game is broad and quite unspecific, so we should focus on the characteristics that are unique to pervasive games. According to (Montola et al., 2009), the expansions of the contractual magic circle are the following ones:

- Spatial expansion: The game defines a playing area where gameplay takes place, and inside that area the game rules apply and shall be respected (Montola et al., 2009). A pervasive game takes place in the real world and changes the significance of ordinary places and objects to fit the game objectives and scenario. The game mixes up the real and virtual worlds.
- Temporal expansion: A game has a time to start and end (and all players agree with this). In pervasive games, not all games have a predetermined time to end, and by the time the player enter the game, the game could be running for several days or weeks. For instance, in some games (like the game killer (Wikipedia, 2014a)), which run for several years in various campi in the 80's), each match may take several weeks and the players may sleep, eat, and go to parties while playing the game. The concept of game session is somewhat blurred in pervasive games.
- Social expansion: This expansion blurs the definition of players and non-players allowing the involvement of non-players. People that do not even known about the game tend to be involved, and theirs participation range from spectatorship to full participation (Montola et al., 2009).

2.2

Pervasive game classification

In this section we illustrate some types of pervasive game. It is not our aim to create or define all pervasive game types, but just to give a glimpse to the reader about some examples of pervasive games. We list some types of pervasive mobile games above:

- Location-based games(or location-enabled game) are games that use player location to evolve in-game. In those games, the player must have some mechanism to report his/her location such as GPS or any other location technology. Examples are Ingress (Google.2015.d), BotFighters(Sotamaa, 2002) and Insectopia(Peitz et al., 2007).
- Treasure hunts are the oldest genre of pervasive games, and the most well established. In this type of game players need to find certain objects in an unlimited game space (McGonigal, 2011). Some examples are letter boxing (Letterboxing North America), Shelby Logan’s run (Wikipedia.2014.d; Seattletimes, 2008) and geocaching (Farman, 2009; Schlatter and Hurd, 2005).
- Assassination games: This type of pervasive game emerged in the 1960’s when students at universities started to play games inspired by the film *la decima vittima* (Wikipedia.c, 2014). The classic examples of this type of game not even use digital elements, that is, they are games entirely in the real world, such as: Gotcha, KAOS (Killing as organized sport), Juggernaut, Battle Royal, Paranoia, Killer, Elimination, or Circle of Death (Wikipedia, 2014a). In this type of game, players try to eliminate each other using mock weapons in an effort to become the last surviving player. Cruel 2B Kind (McGonigal, 2011; McGonigal and Bogost) is an experimental alternate reality game that is also an “assassination” game, in which the players do not know who else may be playing and they slay other players with compliments and other kind phrases. .
- Pervasive LARPs: LARP (Live Action Role-Playing), also called theater style, are pervasive games where the participants physically act out as a character in an environment that has at least partially been propped to look like the story setting (Montola et al., 2009). A classic example of a LARP book is *Vampire: the Masquerade* (Dansky and Brucato, 1998) . *Prosopopeia Bardo* is a Pervasive LARP game created by the IPERG Group (Iperg, 2006) that uses surveillance and mobile communications equipment to create a LARP experience in the streets of Stockholm. In a conventional LARP game there is a clear distinction between the real world and the fictional world. In a pervasive LARP, these worlds are mixed.
- Alternate Reality Games (ARGs): The ARG interest group of IGDA (Martin et al., 2006) defines an ARG as follows:
 "Alternate Reality Games take the substance of everyday life and weave it into narratives that layer additional meaning, depth, and interaction

upon the real world. The contents of these narratives constantly intersect with actuality, but play fast and loose with fact, sometimes departing entirely from the actual or grossly warping it – yet remain inescapably interwoven. Twenty-four hours a day, seven days a week, everyone in the country can access these narratives through every available medium at home, in the office, on the phones; in words, images, and sounds. Modern society contains many managed narratives relating to everything, from celebrity marriages to brands and to political parties, which are constantly disseminated through all media for our perusal - but ARGs turn these into interactive games. Generally, the enabling condition to(sic) is technology, with the(sic) internet and modern cheap communication making such interactivity affordable for the game developers. It's the kind of thing that societies have been doing for thousands of years, but more so. Much more so." One example of this type of game is the Ingress game (see Chapter 3 and appendix B).

2.3

Some final remarks on pervasive games

As McGonigal (McGonigal, 2006) says, pervasive games may provide "infinite affordances", in the sense that players, when solving in-game puzzle, can use any property/objects in their environment to conduct infinite variations of play moves. Usually, in games that support infinite affordances, the players are free to choose their own goals. According to Montola et al. (2009): "the combination of infinite affordances and unpredictable environment leads into surprising coincidences and occurrences. These occurrences often lead to intensive and fun game experiences, which have not been planned by any designer or participant". Those unpredictable actions not planned by the game designer are recognized by (Montola et al., 2009) as "emergent gameplay"

Some other final remarks are:

- When involving non-players, the game design must consider that they have not accepted the magic circle contract, they may not know about the game, and they may not like to get involved.
- Pervasiveness is not strictly a function of rules and game design, but playing styles can also make a substantial difference (Montola et al., 2009).

2.4

Pervasive mobile games

Pervasive Mobile Games are pervasive games that use mobile devices, such as smartphones and smart objects to perform activities during the gameplay. In the present dissertation, we consider mobile pervasive games those games that exhibit the following characteristics (Valente et al., 2013):

- Context-awareness: means the game is able to adapt the gameplay according to the current environment conditions (i.e., the context). This involves sensing the environment and using this information as source for game content or to provide customized experiences (Schilit et al., 1994). Examples of information that can be sensed are lighting, humidity, magnetic fields, WIFI networks, Bluetooth devices, and location (e.g., GPS).
- Access remote data on the go: games that use networking to access remote information.
- Multiplayer: there are more than one player playing the game each time.

More specifically, in the present work, we will use the following devices:

- Smartphones: are the current enabler of pervasive mobile games as these devices are ubiquitous and have a lot of embedded sensors. For example, some sensors commonly found in smartphones are accelerometer, compass, gyroscope, location (GPS), WIFI, Bluetooth, and cameras.
- Smart Objects: These objects are computing devices equipped with sensors and networking capabilities. These devices also might have actuators (Valente et al., 2013) (personal communications, 2014). In this work, the smart objects are represented by Arduino microcontrollers. We chose to use Arduino in this work because they are low cost devices, are easy to use, and can be customized with several different sensors. For example, in our experiments we used light sensors to detect some player actions.

3

Quality requirements for pervasive mobile games

As previously mentioned, the objective of this dissertation is to develop a pervasive mobile game using the quality requirements defined in (Valente et al., 2013) (Valente, 2011) (personal communications, 2014). This chapter presents these requirements that we used to guide the development of the proposed game.

3.1

Pervasive features as quality requirements

Our starting point is the works by Luis Valente (Valente et al., 2013; Valente, 2011), where he proposes pervasive game features and perspectives as guidelines for developing pervasive mobile games. Later on he suggests to use those features and perspectives as quality requirements (personal communication, 2014). These features are listed on Table 3.1 and shown in Figure 3.1 how they are related.

No.	Feature	Abbreviation
1	Device Independence	DI
2	Uncertainty Handling Policy	UHP
3	Local Space Redefinition	LSR
4	Game Object Tangibility	GOT
5	Game Pacing	GP
6	Involving Non-players	INP
7	Usability	USA
8	Daily Life Interleaving	DLI
9	Game Autonomy	GA
10	Mobility	MOB
11	Cross-mediality	CM
12	Persistency	Per
13	Social Communication	SC
14	Conformance to Physical and Social Settings	CPS
15	Connectivity	Con
16	Game Content Adaptability	GCA

Table 3.1: Pervasive game features

Furthermore, Valente (Valente et al., 2013) propose checklists that help the designer to identify these qualities in existing games or to introduce the qualities in new games. Appendix C shows the checklist.

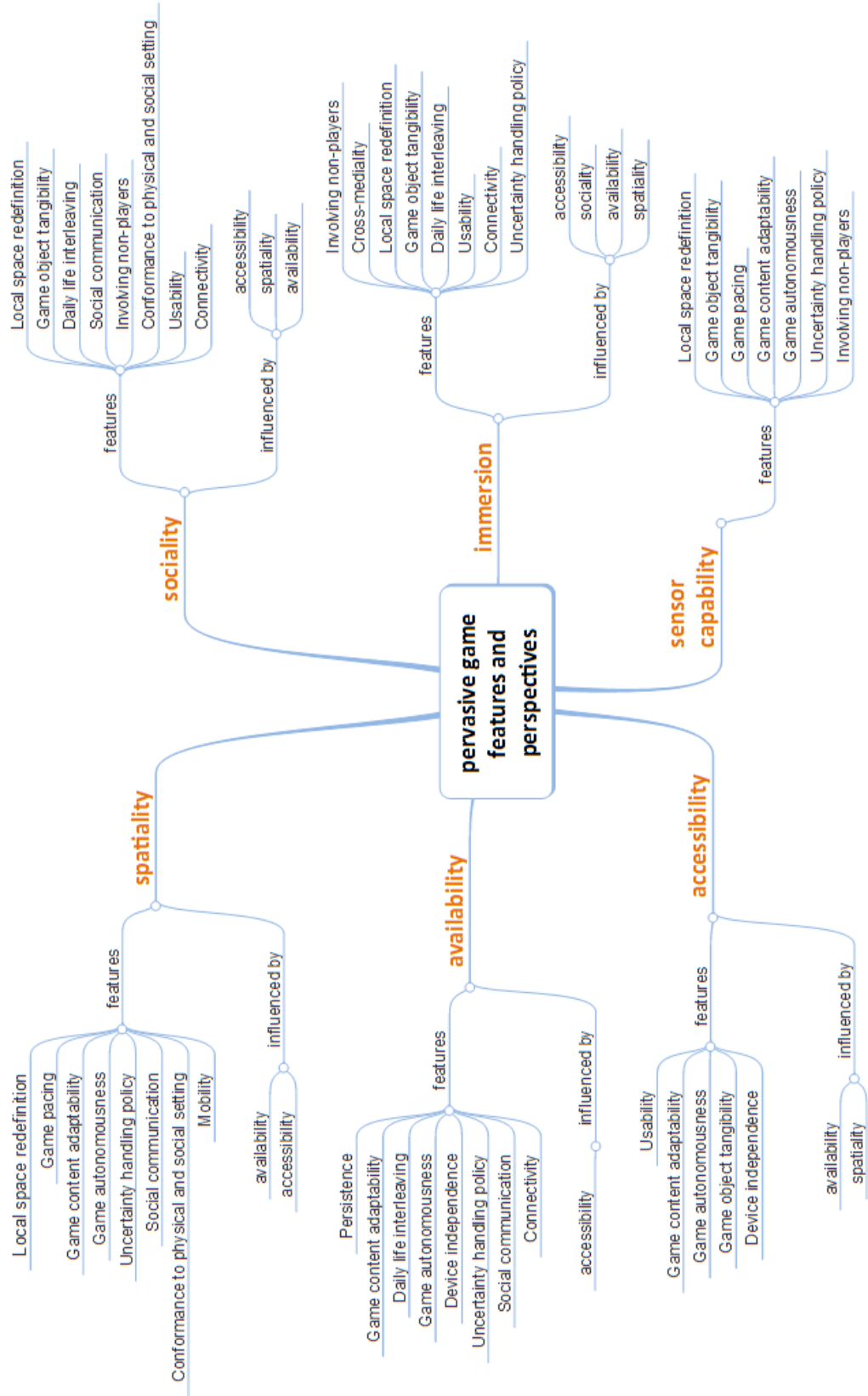


Figure 3.1: Pervasive game features(gray) and perspectives(orange) proposed by Valente (Valente, 2011).

3.2

Descriptions of the quality requirements

In the rest of this section, we briefly present a short description of each feature that we consider quality requirements according to the suggestion made by Luis Valente (personal communication, 2014).

1. Device Independence(DI): Relates to the possibility of playing the game in multiple platforms.
2. Uncertainty Handling Policy(UHP): Relates to how the game handles noticeable boundaries, breaks, or gaps among technology components, due to inherent technology limitations in precision, accuracy, availability, and other uncertainties.

Some authors (Bell et al., 2006; Benford et al., 2006) have classified the dealing with this types of uncertainty in 4 ways: Remove: Design the game that no technology limitation appears sometimes is not possible. Hide: In this approach the limitation appears in the game, but is corrected and anticipated so the players does not notice. Manage: The game has a series of contingency ways to work and as it is programmed to work in several ways, it shows the player the impression that it was designed to work the way the user sees it, but it may be in a contingency mode and the player isn't aware of it. Reveal: With this approach the users is informed about the issue and asked to solve it.(i.e. reestablish network connection) Exploit: These issues are used as a design principle and are inside the game core mechanics as proposed by (Benford et al., 2003).

3. Local Space Redefinition(LSR): Relates to how the game is able to change the meaning of the places where the game happens, including integrate physical spaces (and their elements) as game content.
4. Game Object Tangibility(GOT): Relates to using the mobile phone to implement tangible object metaphors.
5. Game Pacing(GP): Relates to how the pacing of game activities affects the behavior of technology, and vice-versa (e.g., sensor technologies may not work well in a high-paced game).

6. Involving Non-players(INP): Relates to integrating non-players in the game. For example, non-players may be seduced to become players and the game may use information collected from non-players as game content.
7. Usability(USA): Relates to traditional usability issues from focused on mobile devices and specific aspects of pervasive mobile games.
8. Daily Life Interleaving(DLI):Relates to how the pervasive game is able to become diffuse through daily life, enabling a player to integrate gaming sessions with other non-game activities.
9. Game autonomy(GA): Relates to how the game is able to function as an independent entity without requiring setup and supervision while game sessions happen.
10. Mobility(MOB):Relates to mobile computing aspects (using wireless connections on mobile devices) and the size of the game area, which may require players to move through far distances to complete game activities.
11. Cross-mediality(CM):Relates to cross-media aspects in game such as multiple games interfaces in different (heterogeneous) types of devices, such as mobile phones, desktop PCs, and custom hardware.
12. Persistency(PER): Relates to maintaining game state to be accessed through different game sessions over time.
13. Social communication(SC): Relates to how the game is able to foster social communication, acting as a medium for people to communicate in several scopes (i.e., co-located, distributed, or both).
14. Conformance to Physical and Social Settings(CPS): Relates to ethical and privacy concerns, conforming to social conventions, safety concerns (e.g., protecting players from harm while moving around), and adequacy

to physical settings (e.g., relying on audio feedback on naturally noisy places).

15. Connectivity(CON):Relates to networking architectures, infrastructure, and requirements, in several scopes (i.e., global, local, or both).
16. Game content adaptability(GCA):Relates to context-awareness aspects, such as how a games is able to adapt the gameplay regardless of the environment conditions, while keeping the same (or planned) functionality. The checklist associated to each quality requirement above mentioned can be found in Appendix C.

3.3

Ingress game analisys

In this section we analyze the popular Google game named Ingress, using the 16 quality requirements presented in Section 3.1. We conducted this analysis using the checklists associated to each quality requirement Appendix B presents the full analysis in table format, and Appendix C presents the complete checklist.

3.3.1

What is Ingress

Ingress(Google.2015.e) is an Augmented Reality Massive Multiplayer Online Role Playing Game(ARMMORG). This game can be downloaded from the Google store or Apple store. Figure 3.2 illustrates some screenshots of this game.

3.3.2

Ingress game story

The story is that Project Niantic has uncovered the existence of "Exotic Matter" or XM for short. XM is both matter and energy that flows around Landmarks (public places with sculptures, paints or other tourist interesting places). Each place where XM flows is a portal in the game. XM is important for the humankind and we can control humans by controlling it. Players need to choose between two factions, the Enlightened (Green Faction) that wants to harness XM to uplift the entire humankind and the Resistance(Blue Faction) that tries to protect humankind from the XM. Before entering the game, new players choose a faction that cannot be changed later. Progress in this

game is measured by the number of "mind units" captured by creating areas called "control fields". These fields are created by linking tree portals forming a triangle, without crossing any existing link. The goal is to claim all portals to your faction.

3.3.3 Quality requirements

In this section we analyse Ingress according to the quality requirements presented in Chapter 3.

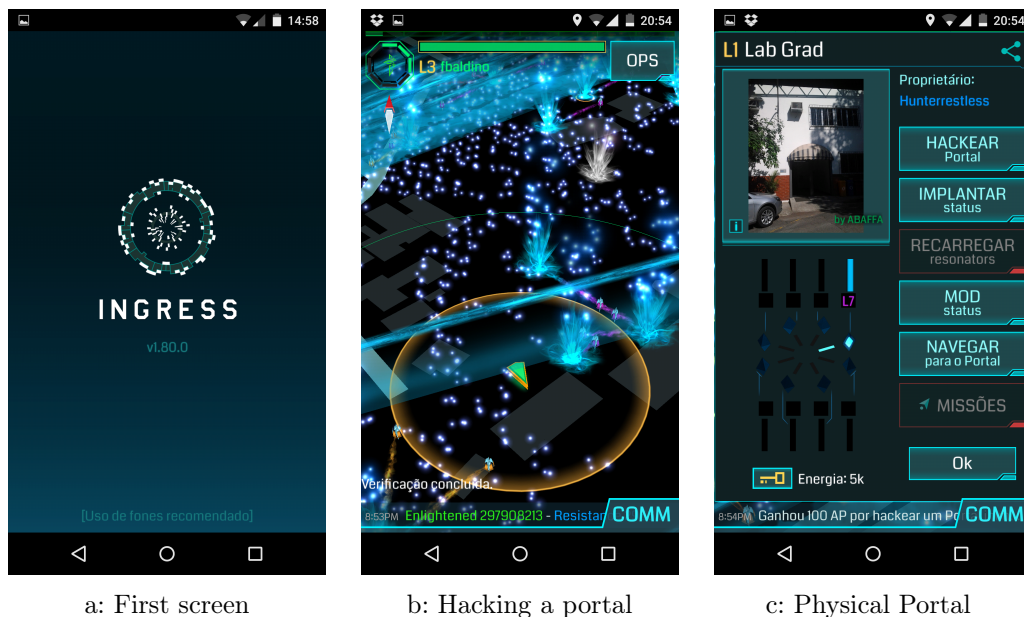
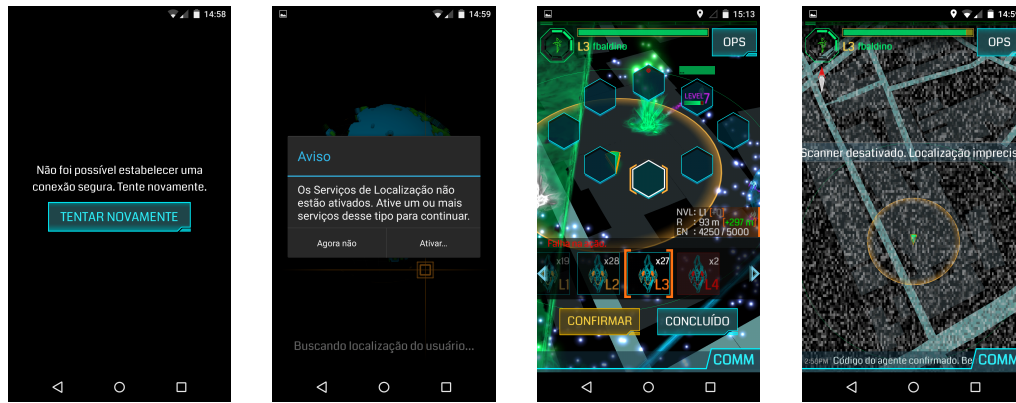


Figure 3.2: Ingress

The results are presented below:

1. Device Independence (DI) - Ingress is available for mobile phones or tablets using Android and iOS (since July 2014)(DI1).
2. Uncertainty Handling Policy(UHP) - Ingress only use the reveal strategy to all sensors and network it uses (UHP1). Any unavailability halts the game until the resource or sensor information needed is available (UHP2). The error messages can be observed in Figure 3.3 and its subfigures: revealing connection error 3.3a, revealing the need of location 3.3b, revealing the absence of internet with an error message Figure 3.3c, revealing that GPS is not available Figure 3.3d



a: Connection Error b: Request to Use Location c: Without Internet d: Without GPS

Figure 3.3: Handling with uncertainty

3. Local Space Redefinition (LSR) - The game integrates physical places by using landmarks(LSR1), such as pictures, sculptures or any other physical world landmark inputed as a game portal by the game administration (LSR2,LSR4). Those landmarks are virtual world portals where XM can be collected(LSR3). The game calls attention to existing landmarks without adding new ones(LSR6). Figure 3.2c shows a physical world place where a portal is located. If a player finds some landmarks that are not a game portal, he/she can send a request to create a portal on that landmark. Figures 3.5a and 3.5c show real world portals and Figure 3.5b shows virtual portals.
4. Game Object Tangibility (GOT) - The smartphone is used only as an access terminal(GOT1).
5. Game Pacing(GP) - The technology used is compatible with the game pacing(GP1), although sometime the player needs to wait the GPS to report it's new location or the network to finish some ingame activity stalling the user until the operation is finished. Hacking a portal witch has two modes, one fast that takes just one click, another more time consuming that takes about 1 minute(GP3) and consists of repeating several patterns as displayed in Figure 3.4b and verified in Figure 3.4c

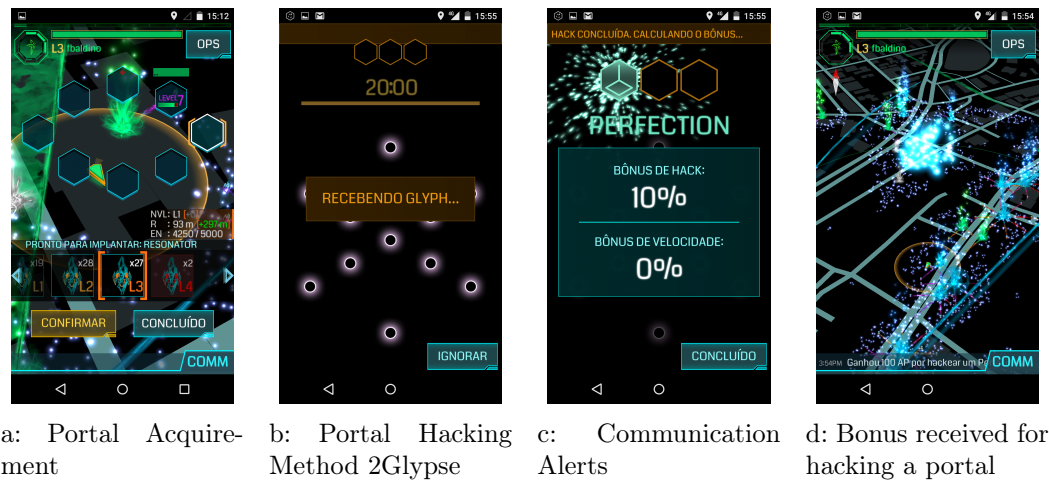


Figure 3.4: Ingress Portal Acquisition

6. Involving Non-players(INP) - The game does not involve non-players(INP1), but it uses the landmarks(INP3) created by others as shown in Figures 3.5a and 3.5c . Game portals are viewed as shown in Figure 3.5b.

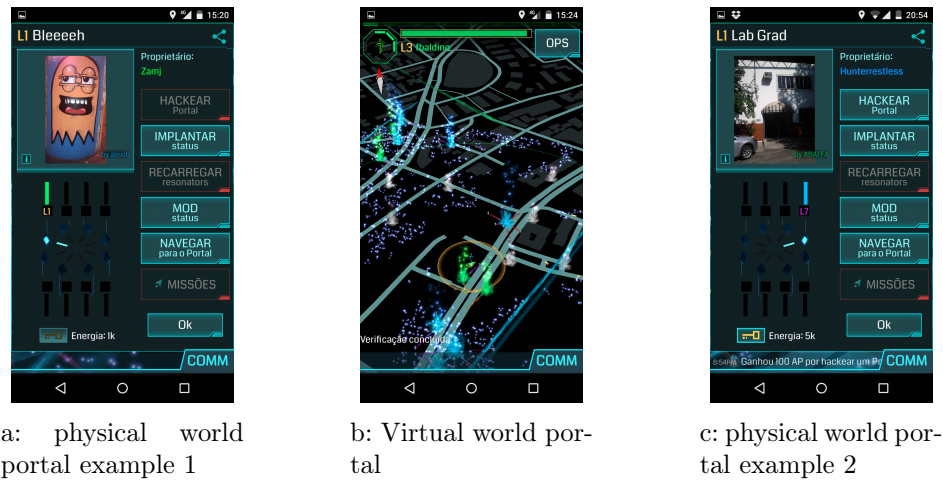


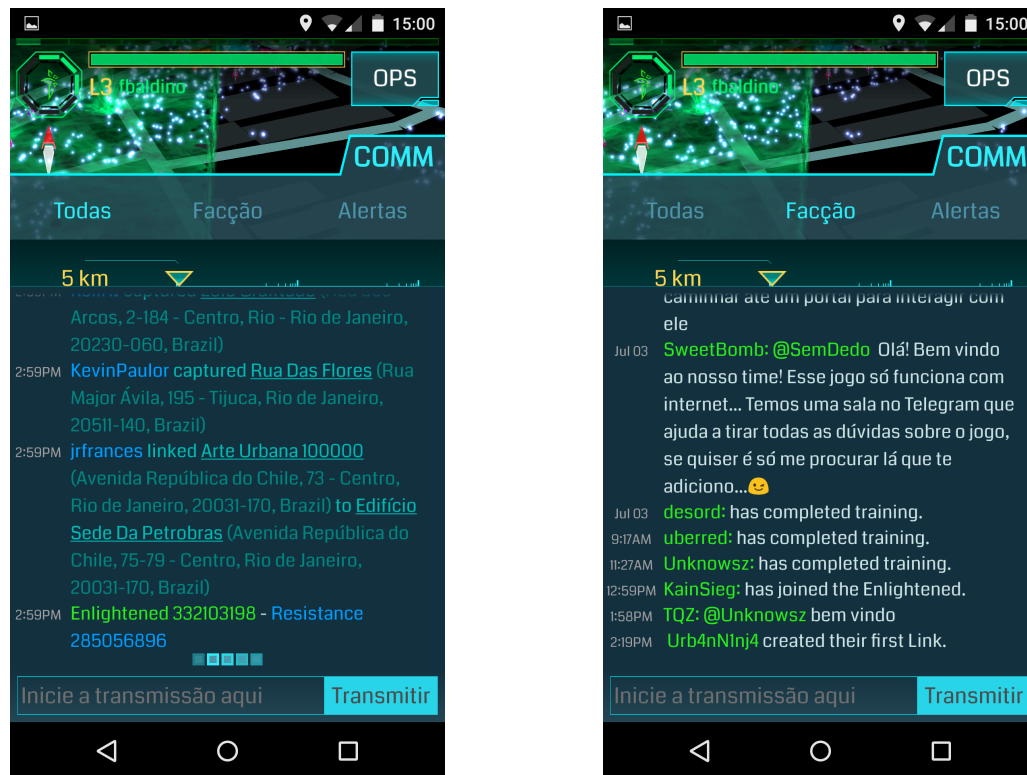
Figure 3.5: Ingress physical world and virtual world portals

7. Usability(USA) - Although it is not expected the player to focus full time attention on the device screen(USA1), most players do this. It is very easy to become absorbed by the game and forget that you are in a dangerous zone, or in the middle of the streets, specially if you use headphones.
8. Daily Life Interleaving(DLI) - The game provides a persistent game world (DLI1) and all game activities are designed to be interrupted(DLI2). Games activities do not need the players long time commitment, although

this is stimulated. The game also calls the users by e-mail and game app messages(DLI6), both can be seen in Figure 3.7d. This game also permits game play without time restriction, this means, a player can go to a dangerous place late at night to get a portal for his/her team(DLI5).

9. Game Autonomy(GA) - This game is bounded to specific local places (GA1) and the only preparation needed is to import the game portals database to the game server by the supporting staff.
10. Mobility(MOB) - It requires internet connection during the game-play(MOB1), and also requires players to move to specific locations(MOB2). The game need network, and it also needs the portals created. The portals are spreaded all over the world and the position of the players are reported to the game server, allowing the server to know where exactly the player is.
11. Cross-Mediality(CM) - Ingress has no crossmediality(CM1,CM2,CM3).
12. Persistency (Per) - The game maintain a persistent game world(PER1) and supports three levels of inside game communication: global,faction and alert list (PER2).
13. Social Communication(SC) - Ingress provides Global social communication using it's in game comm channel(Figure 3.7a), local communication using the Faction(Figure 3.7b) and alert channel (Figure 3.7b). Players also receive outside game communication(Figure 3.7d) (SC1). The game transforms the players relationship by improving commitment and teamwork(SC2). The game also stimulate players to form groups to hang around hacking or capturing portals (SC3). Landmarks receive more attention from players, making the players discover places that others may ignore, linking players to the community(SC4). Players can submit portal requests, and those requests are a form of emergent gameplay(SC5).

Figure 3.6 shows the in game calling to participate in active communities.



a: Social communities call from game part 1

b: Social communities call from game part 2

Figure 3.6: Ingress communities

14. Conformance to Physical and Social Settings (CPS) - Privacy is handled by using just nicknames to identify players and automatically publish only in game progress data (CPS1). The game does not intended to disturb non-players(CPS2). It is also designed not to expose players to embarrassing situations, although it can happen(CPS3). The game conforms with social conventions and etiquette(CPS4). Game activities are adequated to the physical setting(CPS5).
15. Connectivity(Con) - This game needs global connectivity to the game servers to perform game actions. When the connection is lost no game action can be accomplished. Local connectivity is also used to inform the player that his portal is suffering an attack or has been destroyed(CON1). The desired place to play this game is full with portals and safe for players to stay (CON3).
16. Game Content Adaptability(GCA) - Slow network access will increase the time needed to perform in-game activities, also, GPS speed and quality will disrupt game experience making the user wait the sensor to report his/her new position(GCA1). It is required that players move from

portal to portal stimulating player movement in the real world(GCA2). Only portals that are close by are viewable to the player(GCA3). Players can play the game in various places without manual intervention(GCA4).



Figure 3.7: Ingress social communications screens

4

The proposed game

This chapter describes the game, how we connect this game to the quality requirements related to pervasiveness, the game architecture and its features. Also we give a glimpse of the technical background, through a brief discussion about the tools, technologies, and techniques that the game prototype uses. We named our game "Hide or Reveal".

4.1

Game concept

The game concept outlines the features of the game, the goals, and the overall gameplay. In this section we present a very short game concept document. We start with a very concise high concept statement, sort of one-paragraph pitch document. Afterwards we present premises, elements, and features.

4.1.1

High concept statement

A trash can summons you to a game that may change the future of the human kind. You are expected to solve riddles, conquer places and find hidden objects for your team. The player and the team that acquire more places and objects in the shortest time will have controll over everything.

4.1.2

Premise and story synopsis

You throw something in a trash can and, unexpectedly, it starts emitting a strange noise followed by some blinking lights. Then you take a closer look at the trash can and notice that there is a strange code written on it. You search Google for this code and find a site asking you for a color code, and you decide to enter the same color sequence displayed by the trashcan. The site asks for your help to accomplish several missions. You agree with this position and login into the site. Then the site explains that there are two teams fighting in a battle to conquer magic places marked with images. These teams are the Hackers and the Paladins – the former group wishes to publish and free the information containing in those magic places, and the later wishes to control the knowledge power hidden in those places. The magic places are

receptors of information sent by intelligent beings somewhere. Then you are asked to download an APP to accomplish the missions for the team you choose. You start the app in your mobile device and start hunting pictures around to conquer those places. Also you can attack players of the other team stealing points from them (“attack mode”).

4.1.3 Elements

There are five basic elements in this pervasive mobile game:

1. Smart Objects: Objects (interactive or not) that are equipped with sensors, networks, and effectors.
2. Trash can: The most important smart object in this game. It is the game entry point, which consists of a smart trash can that will display colored lights and make noises when is used.
3. Site: The place where the game content is discovered and explained to the user.
4. APP: The mobile application that permits players to perform the following actions: capturing places, attacking opponents, and inspecting progress.
5. Markers: Images that the player must search for, using his/her mobile device.

4.1.4 USP - Unique selling points

USP (Unique Selling Points) is about what makes the game stand out from its competitors. We identify the following points:

- Pervasive mobile game
- Puzzle game that promotes tangencial learn (players may self-learn by searching in game topics presented to them).
- Two modes of participation: individual and team.
- Dual team competition to maximize the challenge.
- Realtime game progress report

4.1.5

The X statement

The X statement is a term used by Eletronic Arts to identify the main feature of a game in a single and concise sentence that describes the essence of the game. It is used to inform and guide both the developing team and the marketing team. It is used to make design decisions. The X Statement of the proposed game is the following: “The game, knowledge to be hidden or reveal, opponents, and friends are in the real world everywhere, anytime”

4.2

Game mechanics

Game mechanics can be defined as systems of interactions between the player and the game. However, this concept may have many interpretations. The reader is referred to an interesting discussion about this term in (Marczewski, A., 1993). In the present work, game mechanics are procedures and rules that affects the play experience and make difference in achieving the goals of the game.

The game starts with a trash can available at a public space(Figure 4.1a). When someone throws something into it (Figure 4.1b), the trashcan emits the sound of a falling object followed by an explosion and displays a sequence of color lights (Figure 4.1c). This represents the encounter of the player with the game. A player who is less inquisitive may stay repeating this action for fun without moving forward into the real game. This is the first level of fun with pervasive games (something already explored in the Fun Theory Project (Volkswagen, 2009)).



a: Step1



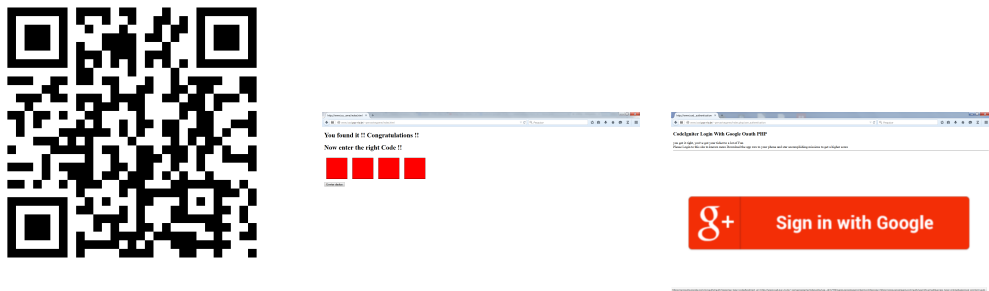
b: User Throws garbage in the bin



c: Trashcan makes sound and blink some lights

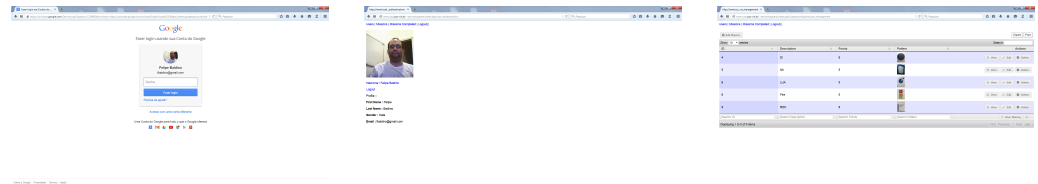
Figure 4.1: The trash can - the main portal to the game

Attached to this trash can, there is a QR code (or a short nonsense phrase) that allows players to search for the game and get into it.If the player uses the QR code (or searches Google for the nonsense phrase), he/she will be taken to the site of the game (Figure 4.2a). In our prototype, we used a QR code. When the user opens the main page of the site, it will display a color selection panel asking the same code that was displayed by the trash can (Figure 4.2b). Once the color code matches, the next screen congratulates the user and shows a button to login in on our prototype (Figure 4.2c). The login is made by a Google account, i.e. the user is requested to enter the authentication information in the Google site (Figure 4.3a). Once the Google login procedure is completed, a screen with user information is shown (Figure 4.3b) and the player has his/her game account created. After these procedures, the player can access the missions of the game (Figure 4.3c). Each mission consists of a brief description and an image to be found. Also this site will show the user’s accomplishment, his/her rank in the game, and a link to the mobile game app. The mobile application (an android app, in our prototype) syncs content with the server and asks the user for a nickname and a team (Figure 4.4a).



a: User notes something attached to the bin b: User following that information is taken to the game giving the color code site c: User gets into the game by

Figure 4.2: Moving forward into the game



a: Site requests Google account username and password b: User logs in c: User is able to see the list of missions and accomplishments

Figure 4.3: Site - player authentication

If the player wants to play a mission, he/she must select a mission from the mission list (Figure 4.4b) and capture the corresponding picture shown on the screen with the camera (Figure 4.4c). This image will be displayed inside the game zone, in a visible place. When the camera of the mobile device points to the image and recognizes it, the mission will be set as accomplished.

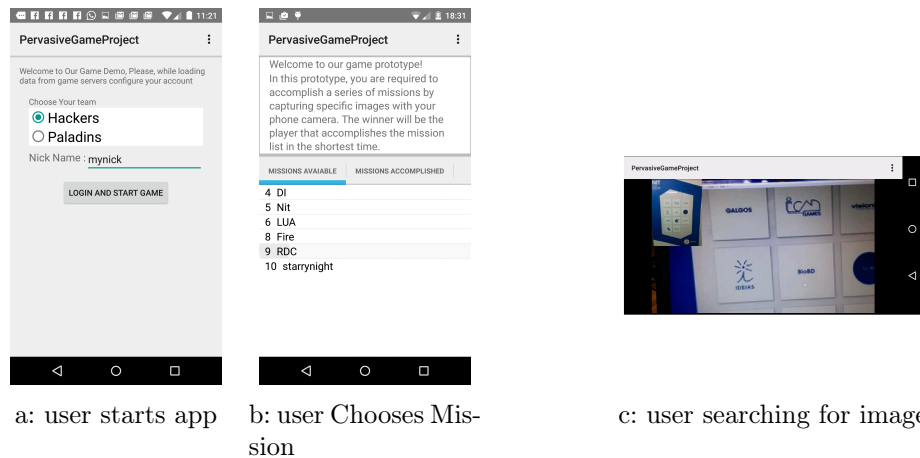


Figure 4.4: Using the mobile application

4.3 Game design

In this subsection we describe the game design process, the quality requirements and functional requirements.

4.3.1 The overall design process

We started the game design process inspired by "The world's deepest bin" project, which is part of the Fun Theory project (Volkswagen, 2009). "The world's deepest bin" project consists of a trash can (installed in a public space) that emits a sound of a falling object everytime something is thrown inside the trash can.

Also we were influenced by the Crowd Control TV Series (Crowd Control), which is an American reality documentary TV series on the National Geographic Channel. Crowd Control sets up a series of practical experiments to change social behavior, using devices with sensors, actuators, cameras, and speakers (e.g. a staircase looking like a piano makes more people prefer to climb it instead of using a moving staircase; a talking trash that influences the habit of people keep public spaces clean; or a musical highway where the tires passing over a special texture on the road, at the allowed speed limit, produces music as a reward for obeying the law). Our design team

started with four members including the author. This team made a series of creative sessions where brainstorm techniques and principles were used. Every member of the team were free to express their ideas for more surreal it appears at first no one was criticized. Before each creative session we discussed the non-functional requirements related to pervasiveness that needed to be improved. After each creative session we made a small analysis of viability of the suggestions presented, then we linked the new ideas with non-functional requirements related to Pervasiveness.

After watching some movies, and a lot of discussions, and inspired by the original trash can displayed in Figure 4.5a we made our own trash can to mimic the one from the Fun Theory project shown on section 4.3.1.



a: Trash can designed by Volkswagen (2009)



b: Trash can used in our game project

Figure 4.5: Interactive trash cans

After designing the entry point of the game, a number of questions arose. What to do next? How is it supposed to be fun? Pervasive games have a trend to mix the virtual and real worlds, but how could we do that? Then we decide to start researching on computer vision to answer some of those questions. We created a spike in this project to check the viability of using an Android device for image recognition. This resulted in an Android application that uses opencv (Opencv.org, 2014) to recognize images captured by the device camera. Also we started specifying the game mechanics like some other pervasive games that use smartphone cameras. The most important stage of the game design was the investigation about pervasive quality requirements proposed by (Valente et al., 2013). This stage is explained in the next section. We also planned an

"attack mode" allowing users to steal points of other players, which was not implemented in our prototype.

4.3.2

Pervasive quality requirements

In this section we analyse our game according to the quality requirements presented in Chapter 3. In fact, firstly we designed the game from these quality requirements and then we checked the features of our game. This was a cyclic process that ended in the final version of the game. The results are presented below.

1. Device Independence (DI) - The game is available only for Android devices (DI1).
2. Uncertainty Handling Policy(UHP) - There are three main sources of uncertainty in the current version of our game: networking, Bluetooth, and the camera.

We address networking uncertainties by designing an offline mode that operates transparently - the game runs in this mode every time networking is lost, without notifying the player and requiring further actions and decisions from the player. While in this mode, the game stores data locally and syncs them with the game server as soon as networking is reestablished.

Concerning the camera, a possible detrimental issue relates to the lighting conditions that a player faces when trying to capture an image. To solve this problem, we apply the "remove" strategy (Valente et al., 2013) by providing the proper lighting conditions in the places where the missions happen. This may include installing lighting infrastructure or carefully choosing places that already have adequate lighting conditions.

3. Local Space Redefinition (LSR) - This game may use images that already exist in physical places as game content (LSR1). For example one of the test missions was to take a picture of a fire extinguisher (LSR4). However, the game does not give meanings or roles to these physical elements (LSR2). The game activities may not affect the physical world (LSR6) if we use elements that already are part of these places.
4. Game Object Tangibility (GOT) - This quality does not apply to this game.
5. Game Pacing(GP) - this is a casual game and the game activities does not require fast responses from players (GP1,GP2). For example, the network

sync runs in the background and do not need immediate responses from players. Furthermore, all game activities are designed to permit casual play(GP3).

6. Involving Non-players(INP) - We designed the game to invite strangers and passers-by to play the game (INP1). The game entry point is the trash can.
7. Usability(USA) - The interaction of the game with the player is done only by the smartphone screen, and for safety reasons we do not wish the user to spend too much time focusing on the screen. So we use a strategy of short play sessions to limit that(USA1).
8. Daily Life Interleaving(DLI) - The game provides a persistent game world(DLI1) and all game activities are designed to be interrupted and resumed later without affecting player performance(DLI2) by using the concept of short play sessions (Peitz et al., 2007) (DLI3).
9. Game Autonomy(GA) -The game uses an interactive trash can that needs to be placed in a public place. In this regard, the game is dependent on this physical object and on the place where it is installed. This situation limits game autonomy(GA1). The game also needs images fixed in the game area where we want players to go. So, if there are no images in the area pre-session preparation will be necessary (GA2).
10. Mobility(MOB) - The game requires networking to sync content with the game server. Networking requirements can be reduced if we restrict game playing to an area with good WiFi coverage provided by us (Mob1). The game stimulates players to move in the real-world, as the players need to find the images. The game playing area is constrained to the places that contain the target images (Mob2). This is an important requirement as we want to be able to deploy our game in places such as universities and schools, where the game may serve as an interactive tool for people to get acquainted with these places. In the current version, the order of magnitude of the playing area is similar to a museum or school floor (Mob3).
11. Cross-Mediality(CM) - We have multiple devices with different roles, the trash can (where the user is invited to the game), can be used indefinitely by throwing objects in it to make it play the sound(CM1).Other examples are: The cellphone mode where the game really happens and missions are accomplished; and a query mode by consulting the information

on the site. We did not balanced the gameplay between the trashcan and the smartphone. We wish players to be able to have fun with the trashcan but no gameplay, letting the game play to happen just on the smartphone(CM2).

12. Persistency (Per) - We maintain a persistent game world(PER1).
13. Social Communication(SC) - No social communication was included in the game, but it will be nice to have team communication. This is an important topic for future work.
14. Conformance to Physical and Social Settings (CPS) - The game handles player privacy by identifying them through nicknames (CPS1). Regarding CPS2, the sound coming from the trash can might disturb people that are near it, although we believe that this is unlikely. Question CPS5 inspired us to create missions where players need to take pictures of objects that already exist in the playing area, whenever possible, as an attempt to not mess the physical place with attached pictures.
15. Connectivity(Con) - We need local connectivity to run this game: local network with Internet access or cellular data network. The most important use of networking is during the first initialization of the application(Con1). The rest of the time, the application will run well on cellular networks. If there is no connection after the initial setup the game can continue(Con2). The attack mode, witch was not implemented in our prototype, will need full connection.
16. Game Content Adaptability(GCA) - The game is designed to work with "static context" - we need to select and position images in the environment before game sessions take place (GCA1), which decreases accessibility - players need to go to the place where the game happens (GCA3). Also, if we want to change the physical place where the game happens, we may need to prepare the place to host the game. This may include updating the server with image content related to this new place (GCA4). Finally, in this game the context usage requires players to move while playing (GCA2).

4.3.3

Functional requirements

Our application has three layers: a mobile layer (which is android, in our prototype), a Web layer, and a Smart Object layer (which, in our prototype, is the trash can only). Each one of these layers has its own functional requirements, as described below.

1. Mobile Layer

- (a) Recognize images using an Android Camera.
- (b) Display the list of missions.
- (c) Authenticate the user.
- (d) Synchronize with the server.
- (e) Mark a mission as accomplished in the server and locally.
- (f) Display message when mission is accomplished.
- (g) When a mission is accomplished it cannot be accomplished again.
- (h) Implement Attack infrastructure by turn on Bluetooth radio and search devices (this feature was not implemented in our prototype).
- (i) Get Bluetooth devices that are near (this feature was not implemented in our prototype).
- (j) Request Nickname from the server sending Bluetooth MAC Addresses (this feature was not implemented in our prototype).
- (k) Receive push notifications for each attack attempt that is received or made (this feature was not implemented in our prototype).
- (l) Display a list of attacks (received and made) and show successful ones (this feature was not implemented in our prototype).

2. Web Layer

- (a) Should display a welcome message and ask for the color code.
- (b) When the color code is incorrect displays an error message.
- (c) If the color code entered is correct, congratulate user and asks for external site login.
- (d) Should implement for the administrator the ability to enter new missions.
- (e) When logged in, users can view the mission list and the missions accomplished.

- (f) Provide data in a nice format for the android app including missions, missions accomplished and images.
- (g) Generate push notifications to each device, when attacked.
- (h) Show the list of attacks attempts (made and received).

3. Smart Objects Layer

- (a) Every time something is thrown in the trash can it will play the sound of falling objects.
- (b) After playing the sound, the trashcan displays a sequence of 4 colored LEDs for some seconds.
- (c) The detection time should be fast enough to permit small object detections.
- (d) Sound should be loud enough to be heard in a noisy place.
- (e) Time between the fall of an object and the start of the sound should be less than 1 second.
- (f) Provide clear and loud sound.
- (g) Play high quality MP3 files.

Due to schedule restrictions in the dissertation planning, our prototype did not implement the following items 2g,2h,1h,1i,1j,1k,1l, and 3g.

4.4

Computer Vision

In this section we give a glimpse of computer vision and describe the key processes used in our prototype. We assume the following definition by Wu and Sun (2013):

“ Computer vision is the science that develops theoretical and algorithmic basis to automatically extract and analyze useful information about an object or scene from an observed image, image set or image sequence. (Gunasekaran, 1996; Sun, 2000; Sun and Brosnan, 2003; Zheng et al., 2006; Zheng et al., 2006b; Du and Sun, 2006). ”

The conjunction of growing processing power, Internet speed, and high quality camera summed with GPS and accelerometers makes a smartphone a suitable device for detect and recognize images displayed in the real world using contextual information. We use computer vision in our game to recognizing some images previously placed in the game field acting as a game elements.

In our prototype we implemented the tracking and validation of the image searched on a camera frame. Those processes are explained in the following sections.

4.4.1 Image Tracking

Locating a picture in another is a complex process, because the searched image may be in a different scale, rotated or only partially visible. Because of this complexity, we are going to break down this activity in a few smaller and simpler tasks as listed above.

- Feature Detection
- Descriptor Extraction
- Find Matches
- Find the homography

All those processes are connected as shown on Figure 4.6. The computation of features and descriptors of the searched image (pattern image) is done only once, but this calculation on the camera image is done for each image frame. The relation of OpenCV Methods and this section is shown on Table 4.1. The main app code that perform the image identification is shown on figure 4.7

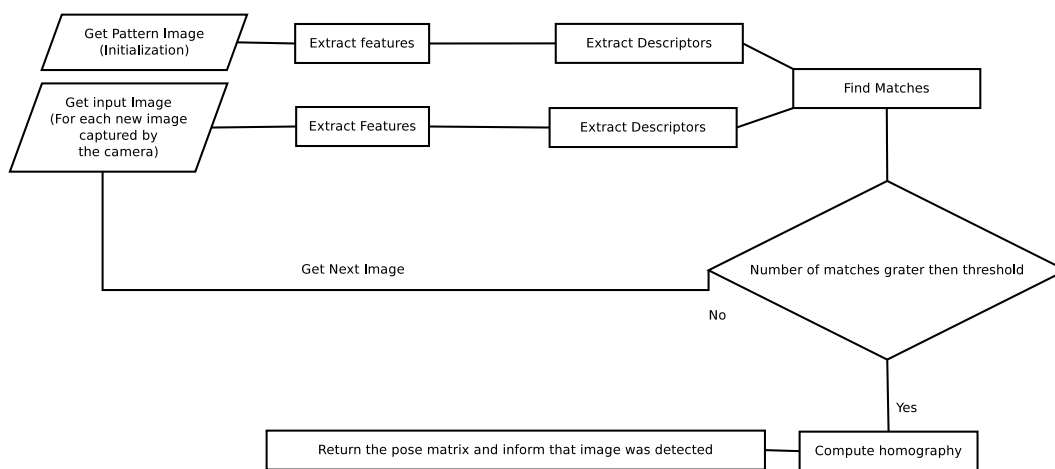


Figure 4.6: Image detection fluxogram

Section	Opencv 4 Android Method
4.4.2	FeatureDetector.create(FeatureDetector.STAR)
4.4.3	DescriptorExtractor.create(DescriptorExtractor.FREAK)
4.4.4	DescriptorMatcher.create(DescriptorMatcher. BRUTEFORCE_HAMMINGLUT)
4.4.5	Calib3dfindHomography(goodReferencePoints, goodScenePoints)

Table 4.1: OpenCV Methods and correlation with image tracking activity

4.4.2

Find Features

There is no exact definition of what a feature of an image is, but we are considering a feature as an interesting part of an image. These parts could be edges, corners or Blobs. For further information on this consult the paper by Schmid et al., 2000. . The first step in image identification is the feature extractor that is necessary, to discover what points are unique for an image. We use Star Feature Detector to perform this operation; it is implemented by (OpenCV4Android, 2014) also known as Center Surround Extrema or simply CenSureE (Agrawal et al., 2008) feature extractor. This feature extractor is scale invariant and relatively fast to be run on phones. The output of this stage is the list of features to be evaluated

4.4.3

Find Descriptors

We use the FREAK (Fast Retina Key Point)(Alahi et al., 2012) descriptor extractor. This descriptor extractor has a higher sensitivity in the center of the image, acquiring more descriptors in this region. It is rotation invariant and scale invariant.

4.4.4

Find Matches

Detecting the descriptors that corresponds to each other is a demanding computation task, and we use the Hamming Distance to identify the matches. The smaller that hamming distance, better will be the match, since Hamming Distance is the number of element changes needed to transform one sequence in another of the same size. We use the DescriptorMatcher.BRUTEFORCE_-HAMMINGLUT implementation of OpenCV to find the matches between the two set of descriptors, one from the pattern image and another from the camera acquired image. This descriptor matcher uses the hamming distance to determine witch descriptors from the first set corresponds to a descriptor in the second set.

4.4.5

Find the homography

Once we have the points correlation between the two images, we can calculate the POSE (position and orientation matrix) of the pattern image with respect to the camera generated image. with this matrix we can insert content in the camera feed by correctly projecting the object into the scene.

4.4.6 Code

Our code is based on the book “ Android application programming with OpenCV ”(Howse, 2013). The most important part of the computer vision code used in our App is listed in Figure 4.7. The methods apply listed in lines number 1 to 9 is called for each frame, and it is responsible to convert image to grayscale on line 2, generate the feature keypoints list on line 3 Extract the descriptor in line 4, calculate the matches in line 5. Method findPose in line 10 first get the minimum and maximum distance of the target (lines 11 to 26). From line 27 to line 35, it checks if the tracked object is in the scene. From line 36 to 46 it searches for good key point matches, from line 47 to 49 it ends the process if not found at least four good key points. From line 50 to 54 it calculates the homography, on line 55 it calculates the transformation of the reference corners and generate a candidate corners. on line 56 and 57 it checks if the transformation generated a convex contour. If not there no image was found. From line 67 to 92 it calculates the camera projection matrix.

```

1 public void apply(final Mat src, final Mat dst) {
2     Imgproc.cvtColor(src, mGraySrc, Imgproc.COLOR_RGBA2GRAY);
3     mFeatureDetector.detect(mGraySrc, mSceneKeypoints);
4     mDescriptorExtractor.compute(mGraySrc, mSceneKeypoints, mSceneDescriptors);
5     mDescriptorMatcher.match(mSceneDescriptors, mReferenceDescriptors, mMatches);
6     findPose();
7     draw(src, dst);
8 }
9
10 private void findPose() {
11     List<DMatch> matchesList = mMatches.toList();
12     if (matchesList.size() < 4) {
13         // There are too few matches to find the pose.
14         return;
15     }
16     List<KeyPoint> referenceKeypointsList = mReferenceKeypoints.toList();
17     List<KeyPoint> sceneKeypointsList = mSceneKeypoints.toList();
18     double maxDist = 0.0;
19     double minDist = Double.MAX_VALUE;
20     // Calculate the max and min distances between keypoints.
21     for(DMatch match : matchesList) {
22         double dist = match.distance;
23         if (dist < minDist) {
24             minDist = dist;
25         }
26         if (dist > maxDist) {
27             maxDist = dist;
28         }
29     }
30     // The thresholds for minDist are chosen subjectively based on testing. The unit is not related to pixel distances.
31     // It is related to the number of failed tests for similarity between the matched descriptors.
32     // The target is lost but maybe it is still close. Keep using any previously found pose.
33     if (minDist > 50.0) { // The target is completely lost.
34         mTargetFound = false;
35         return;
36     }
37     else if (minDist > 25.0) {
38         return;
39     }
40
41     // Identify "good" keypoints based on match distance.
42     List<Point> goodReferencePointsList = new ArrayList<Point>();
43     ArrayList<Point> goodScenePointsList = new ArrayList<Point>();
44     double maxGoodMatchDist = 1.75 * minDist;
45     for(DMatch match : matchesList) {
46         if (match.distance < maxGoodMatchDist) {
47             goodReferencePointsList.add(referenceKeypointsList.get(match.trainIdx).pt);
48             goodScenePointsList.add(sceneKeypointsList.get(match.queryIdx).pt);
49         }
50     }
51
52     if (goodReferencePointsList.size() < 4 ||
53         goodScenePointsList.size() < 4) { // There are too few good points to find the pose.
54         return;
55     }
56     MatOfPoint2f goodReferencePoints = new MatOfPoint2f();
57     goodReferencePoints.fromList(goodReferencePointsList);
58     MatOfPoint2f goodScenePoints = new MatOfPoint2f();
59     goodScenePoints.fromList(goodScenePointsList);
60     Mat homography = Calib3d.findHomography(goodReferencePoints, goodScenePoints);
61     Core.perspectiveTransform(mReferenceCorners, mCandidateSceneCorners, homography);
62     mCandidateSceneCorners.convertTo(mIntSceneCorners, CvType.CV_32S);
63     if (!Imgproc.isContourConvex(mIntSceneCorners)) {
64         return;
65     }
66
67     double[] sceneCorner0 = mCandidateSceneCorners.get(0, 0);
68     double[] sceneCorner1 = mCandidateSceneCorners.get(1, 0);
69     double[] sceneCorner2 = mCandidateSceneCorners.get(2, 0);
70     double[] sceneCorner3 = mCandidateSceneCorners.get(3, 0);
71     mSceneCorners2D.fromArray(new Point(sceneCorner0[0], sceneCorner0[1]),
72         new Point(sceneCorner1[0], sceneCorner1[1]), new Point(sceneCorner2[0], sceneCorner2[1]),
73         new Point(sceneCorner3[0], sceneCorner3[1]));
74     MatOfDouble projection = mCameraProjectionAdapter.getProjectionCV();
75     Calib3d.solvePnP(mReferenceCorners3D, mSceneCorners2D,
76         projection, mDistCoeffs, mRVec, mTVec);
77
78     // Positive y is up in OpenGL, down in OpenCV. Positive z is backward in OpenGL, forward in OpenCV.
79     // Positive angles are counter-clockwise in OpenGL, clockwise in OpenCV.
80     // Thus, x angles are negated but y and z angles are double-negated (that is, unchanged).
81     // Meanwhile, y and z positions are negated.
82     double[] rVecArray = mRVec.toArray();
83     rVecArray[0] *= -1.0; // negate x angle
84     mRVec.fromArray(rVecArray);
85     Calib3d.Rodrigues(mRVec, mRotation);
86     double[] tVecArray = mTVec.toArray();
87     // OpenCV's matrix format is transposed, relative to OpenGL's matrix format.
88     mGLPose[0]=(float)mRotation.get(0, 0)[0]; mGLPose[1]=(float)mRotation.get(0, 1)[0];
89     mGLPose[2]=(float)mRotation.get(0, 2)[0]; mGLPose[3]=0f;
90     mGLPose[4]=(float)mRotation.get(1, 0)[0]; mGLPose[5]=(float)mRotation.get(1, 1)[0];
91     mGLPose[6]=(float)mRotation.get(1, 2)[0]; mGLPose[7]=0f;
92     mGLPose[8]=(float)mRotation.get(2, 0)[0]; mGLPose[9]=(float)mRotation.get(2, 1)[0];
93     mGLPose[10]=(float)mRotation.get(2, 2)[0]; mGLPose[11]=0f;
94     mGLPose[12]=(float)tVecArray[0];
95     mGLPose[13]= -(float)tVecArray[1]; // negate y position
96     mGLPose[14]= -(float)tVecArray[2]; // negate z position
97     mGLPose[15] = 1f;
98     mTargetFound = true;
99     Log.i(LOG_TAG, " Image Detected" );
100 }

```

Figure 4.7: Main Detection Code

4.5

Implementation

The Arduino module of the prototype is a simple version that uses, the Arduino to generate the sound with just one speaker and a single channel audio amplifier. We did a lot of experiments with various types of mp3shields and Arduino boards. Also we tried to use an LDR (light dependent resistor) to detect the light variance of a falling object, but this device was too slow. Therefore we tested an infrared version that worked fine. Our website was implemented with Xampp (XAMPP Installers and Downloads for Apache Friends), a PHP development environment, running on a windows machine. It is worth to mention that this part of the implementation took a lot of time to make it work on a linux server.

4.6

System architecture and modeling

In this section we discuss the system integration and application architecture. Firstly we introduce the integration between android, web and Google.

4.6.1

Global overview

Our Application uses four devices as shown in Figure 4.8.

The entire Activity Diagram is shown in Figure 4.9 the highlighted part is implemented in our prototype.

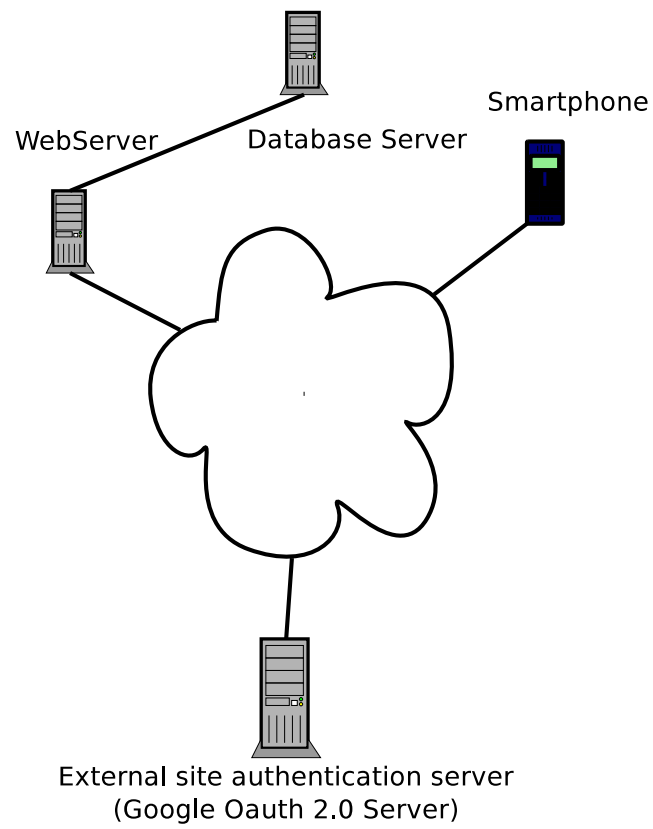


Figure 4.8: Overall Application Architecture

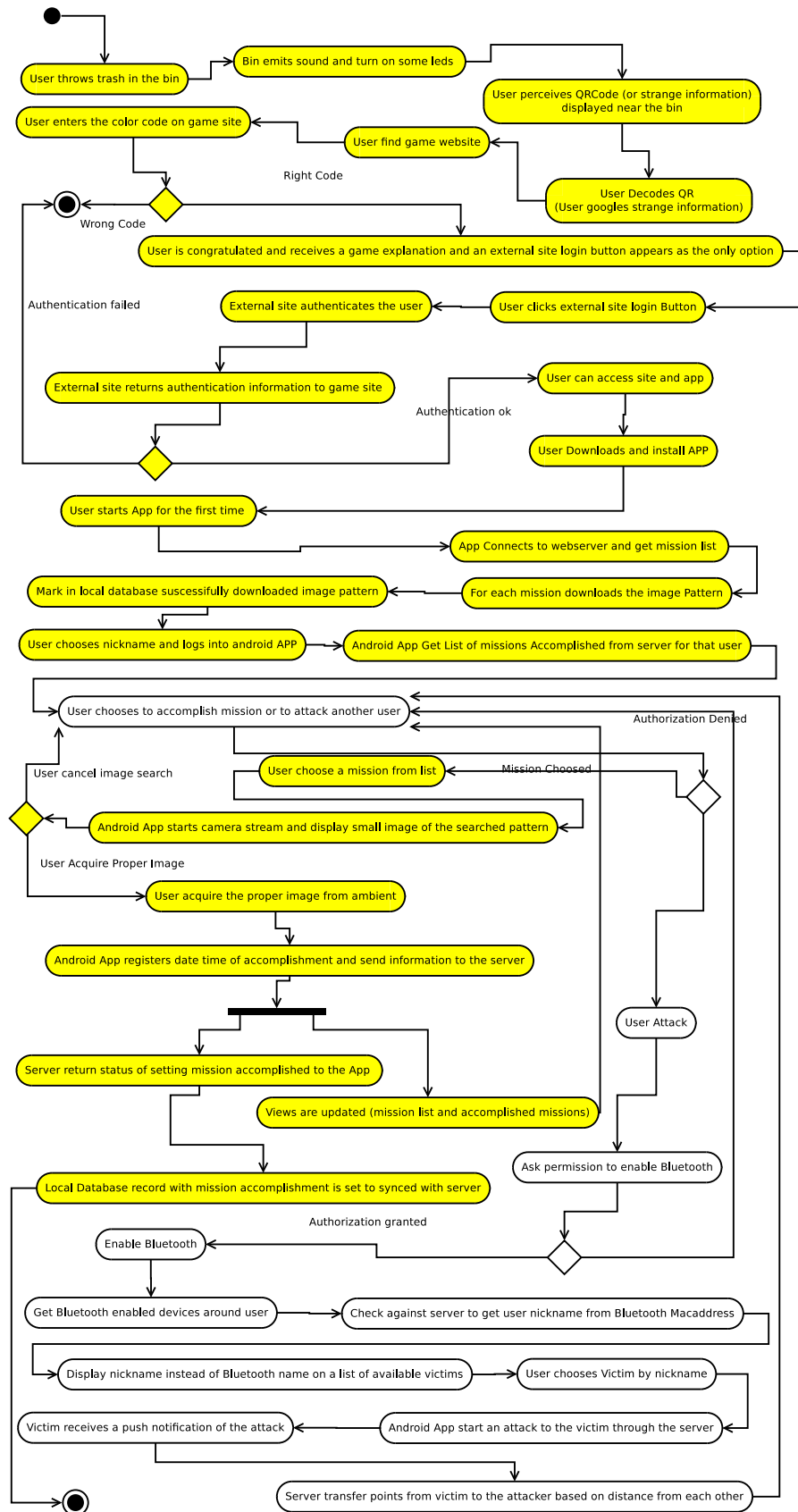


Figure 4.9: Application Activity Diagram

The Sequence Diagram of synchronization between the android application and the website is shown in Figure 4.10.

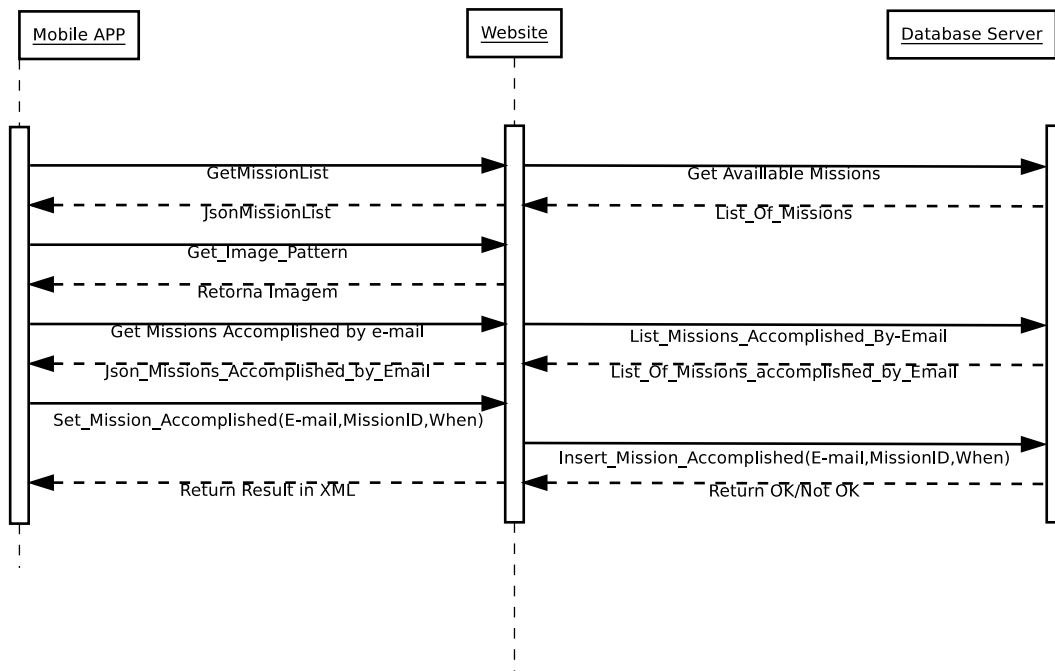


Figure 4.10: Sequence Diagram of synchronization process between the Android Application and game website

The sequence Diagram of the Google authentication process is shown in Figure 4.11. Although this diagram is specific to google, it shows the Open Authentication 2.0 (Oauth 2.0) process that is used also by other authentication services.

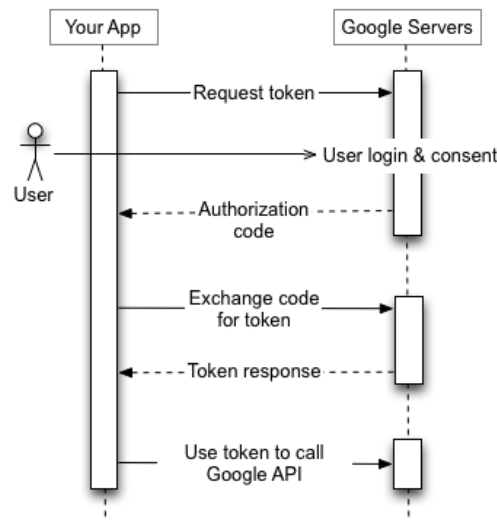


Figure 4.11: Sequence Diagram of Android Authentication Process used in the fair use doctrine obtained from (Google.2015.f)

4.6.2

Frameworks used in the prototype

In this section we discuss the frameworks used to develop the game:

1. Mobile Layer

- Android SDK (Google.2015.b)
- Android OpenCV (OpenCV4Android, 2014)
- GooglePlayServices (Google.2015.a)
- Java-json.jar (Json Foundation, 2015)
- Android sliding tab basic (Google.2015.d)

2. Web Layer

- CodeIgniter (British Columbia Institute of Technology, 2015)
- GroceryCrud (Skoumbourdis, 2015)
- Google API PHP Client(Google.2015.c)
- JQuery (jQuery Foundation, 2015)
- Codeigniter Rest Server(Chris Kacerguis, 2014)

3. Smart Objects Layer

- Arduino IDE

Mobile Layer

In the mobile layer, we created an application and a library. The application is responsible for managing the user interface, generate user feedback, and call the library for specialized image processing. The library handles the camera and all computer vision process by calling the proper OpenCV for android library instaled on smartphone. OpenCV for Android is designed as a standalone library APK that can be installed from Google Play store. The use of OpenCV requires that OpenCV SDK be included as an Android library project. The reader is refered to the guide and reference documentation on Google APIs for Android, which can be found in (Google.2015.a)

The library uses OpenCV to acquire images from the camera, get the proper camera parameters and calculate a transformation matrix to verify the presence of a predefined image in the image acquired by the camera.

This process consists of the following tasks: get camera sensor information; convert the acquired image to black and white; find corners; find points of interests; check the interest points of the pattern against the ones found in the camera image; generate a transformation matrix; and validate that at least a minimum set of points of interest matches between both images.

The main application is our pervasive game project interface. This application uses layout based on tab(see an example in (Google.2015.d)), such as the layout shwon in Figure 4.12

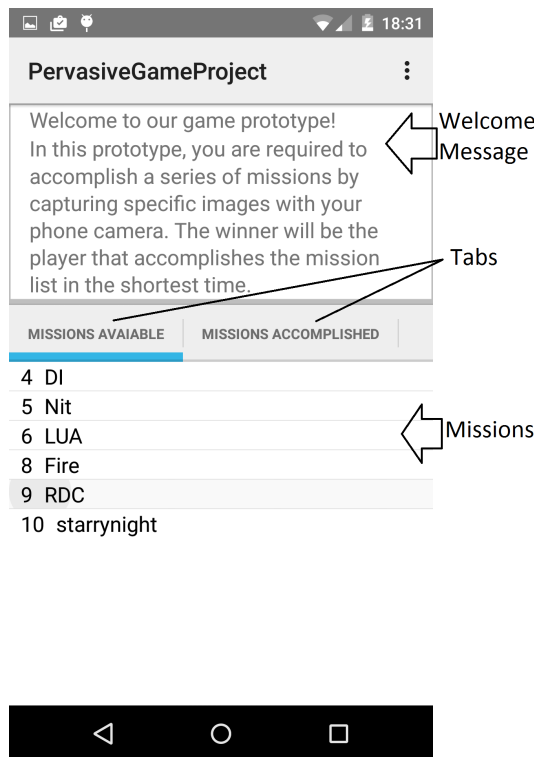


Figure 4.12: Application layout view

The main application uses a splash screen for loading the first data from server. After the user login it syncs the app with the sever that contains the accomplished missions and sends the user Bluetooth mac. After this task, it displays the list of missions to be accomplished. The attack mode, was not implemented in our prototype. The accomplish mission mode is the game safe mode, here the user will accomplish the missions by choosing a mission and pointing his/her phone to the image. When the app detects the image it will set the mission accomplished and send the data of accomplishment to the server. A user that never enters the attack mode will never be attacked. The attack mode enables Bluetooth and syncs (with the server) all users in attack mode around him/her, listing them by nicknames. When the user chooses to attack, he/she will receive a number of points that is proportional to the distance between the two players. An attacker can only attack a single player each time. The attacked player is always notified of the attack.

Web Layer

We created our website using PHP as the server-side language using CodeIgniter (British Columbia Institute of Technology, 2015) and (Skoum-

bourdis, 2015) frameworks. We decided to use Google authentication for user login, and to do this we needed the Google authentication Framework (Google.2015.c). We also needed (Chris Kacerguis, 2014) to generate data in the proper format to communicate with the mobile app. The sequence diagram of section 4.13 shows the user authentication process.

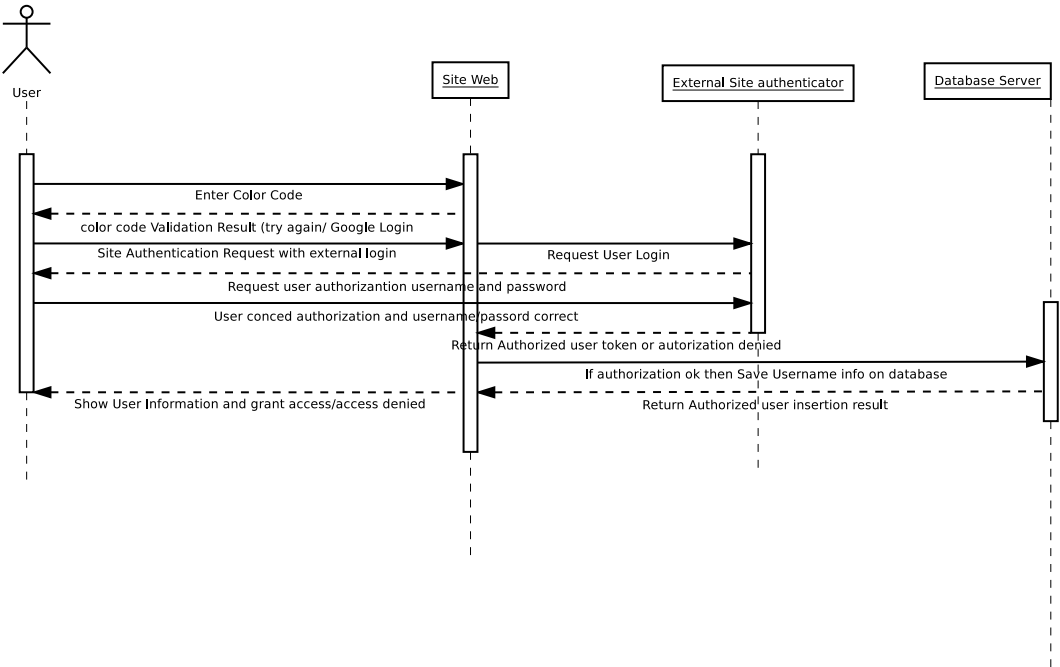


Figure 4.13: Sequence Diagram of site Authentication Process

The sequence of activities that permits the entry of new missions is presented in Figure 4.14. This sequence helps the process of populating the game missions.

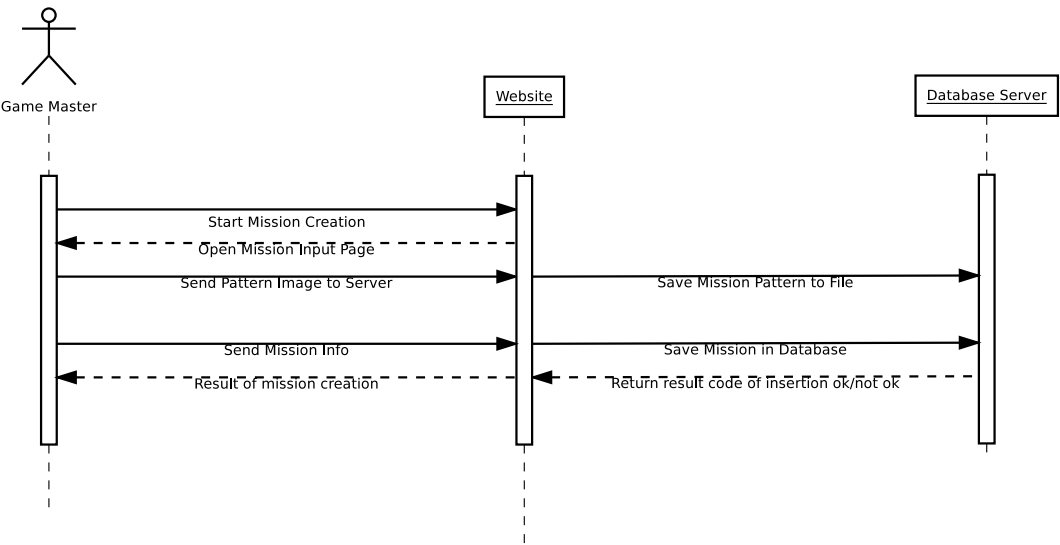


Figure 4.14: Sequence Diagram of Game Master entering new missions

Smart Objects Layer

The interactive mechanism of the trash can uses two types of IR (infra-red) LED: an IR LED emitter, commonly used in remote controllers, and an IR LED receiver used in TV sets and other devices that are controlled remotely. We use the power from the Arduino board and built the circuit shown in Figure 4.15 to connect the LED receiver and emitter to Arduino.

The micro controller starts by reading the level of the LED Detector. witch is a value between 0 and 1000. When the Arduino detects a decrease in this reading, it assumes that something passed between the receiver and the emitter and start playing the sound that was programmed. We used an Arduino Mega 2560 and an mp3 shield with microSD card slot. The entire bill of materials is listed on table 4.2.

Quantity	Item Description
1	Trash can
1	Arduino Mega 2560
1	Arduino MP3 Shield
1	Audio Amplifier
1	USB Power Pack
1	infra-red emitter Led
1	infra-red Receiver Led
1	High Bright Blue Led
1	High Bright Green Led
2	High Bright Red Led
3	100 Ω Resistor
1	220 Ω Resistor
1	2k2 Ω Resistor
1	500 Ω Resistor
2	8 Ω Speaker
1	Universal Board
1	Weld
1	Pack of Dupont Cables
1	Pack of Dupont Male Conectors
1	Hot Glue

Table 4.2: Arduino Bill of Materials

When the sound ends we switch a set of lights on. We use four high bright LEDs with fixed colors using the same circuit of 4.15a but with adifferent resistor. Because those lights are high brightness LEDs and consume more energy, they need an 100 ohms resistor and the Vcc, because they are connected to the digital output pins of the Arduino board.

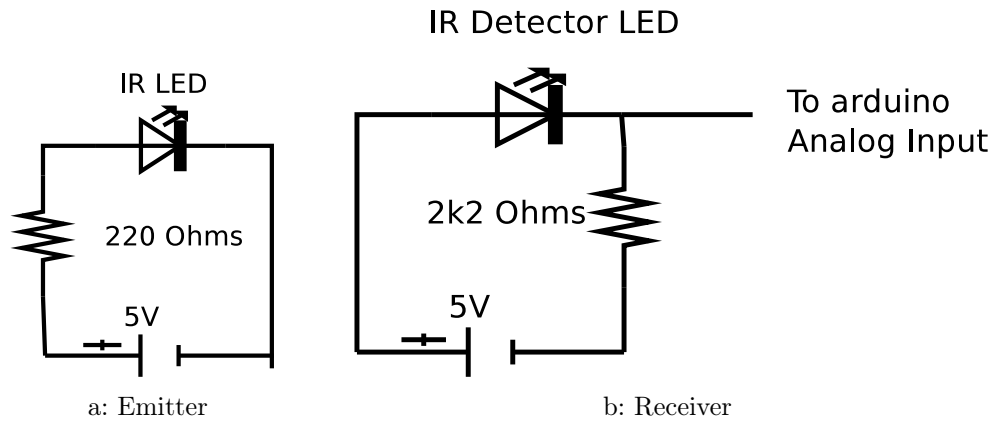


Figure 4.15: IR LED emitter and LED receiver circuit

4.6.3 Android

We implemented the application using android SDK. We use a local database as shown in Figure 4.16. This database was designed for the complete version of the game; but in our prototype, we are only using the tables marked in yellow. This part called `tb_mission` deals with the accomplished missions and the missions to be accomplished. This table has two special fields, one to inform if the pattern image was successfully downloaded (called `mission-patternDownloaded`) and the other that informs if the server has received the mission accomplishment (called `missionaccomplishedsynced`). Table `tb_last-sync` is intended to be used to provide synchronization from time to time by saving the date and time of the last successful complete sync. `tb_bluetooth` lists the Bluetooth MAC addresses discovered by the Bluetooth search process. `tblocaluser` saves the user's login information and `tb_bluetoothmac` holds the Bluetooth MAC address and nicknames received from the server. Table `tb_attacks_received` holds the attacks suffered by the user and `tb_attack_attempts` holds the logged user attack attempts. Although we have implemented the Bluetooth search process, we decided not to use it in our prototype, because of the complexity of the attack.

4.6.4 Website

Our website was developed to introduce the game to the user and prevent unauthorized use of the application. The requirement that the user enters the color code displayed by the trash can is our security check. We create the Web database shown in Figure 4.17, where the tables marked in yellow are the ones used by our prototype. We are not using the table `tb_attack` because



Figure 4.16: Android database. The shaded rectangle (yellow) is the part that is implemented in the prototype.

we do not implemented the attack mode in the prototype. Table `tb_mission_` - accomplished holds the information about the accomplished missions and table `tb_mission` holds the current mission list. The table `tb_usr` holds the user's personal data and the field `USR_DISPLAY_NAME` is the user's nick name. (We are not using the fields of latitude and logitude because we did not implemented the location awareness in this prototype.

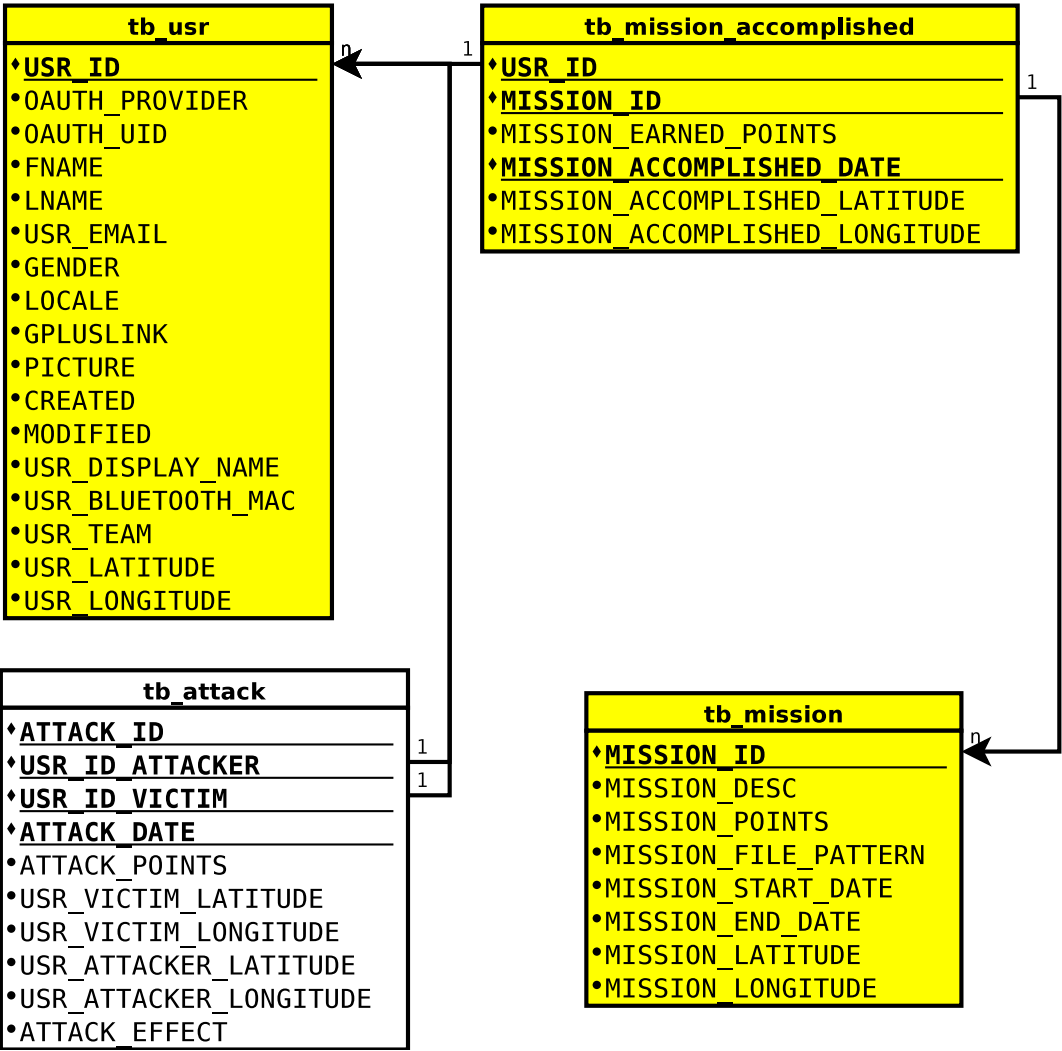


Figure 4.17: Web database

4.6.5
Arduino

We have implemented the Arduino prototype as described in 4.6.2, but we did not make the roles to fix it in the trash can because we were affraid of destroying the only trash can we had in the process of making the roles in the fiberglass trash can. Our implementation uses a LED board with eight multicolored LEDs instead of 4 LEDs, and we do not use an MP3 board. We use Arduino tone generator to produce the sound. The first prototype implementation mounted in the trash can is shown in Figure 4.18

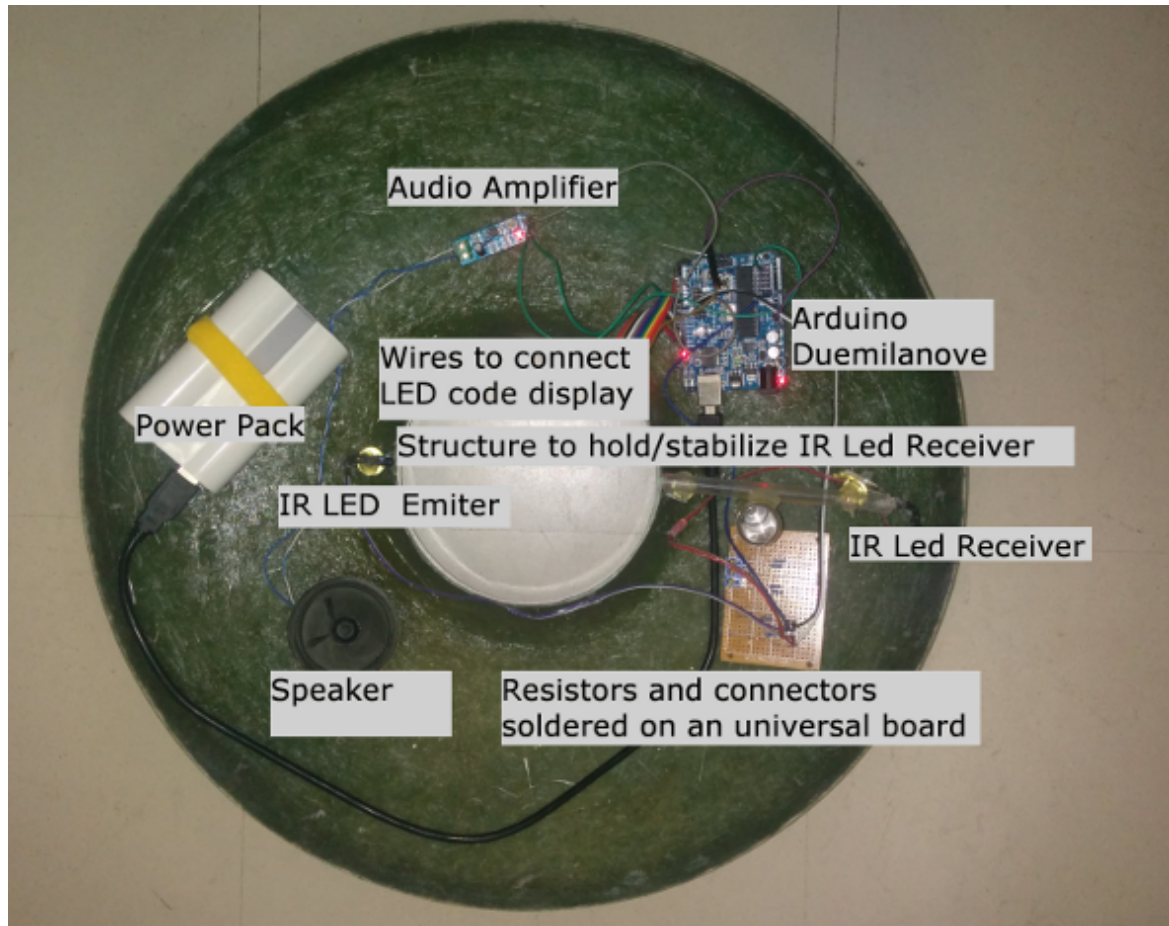


Figure 4.18: Arduino circuit installed in the trash can cover

4.7 Results

The trashcan was put in the author's living room for a weekend and his 5-year son started to put trash only in the noisy trash can. The objective of this test was to validate the circuit and detect abnormal functioning. Also we wanted to determine the battery life in normal use condition. The kid loved to take part in the experiment. The problem was to stop him when the test was over. We determined that the battery was not fully discharged after two days and there was no malfunction of the device. After this test, we started our game evaluation. We asked help from two students between 20 and 25 years old, from both sex. They were gamers and also had experience in developing games. Both testers were guided to perform the game activities and their opinion are listed below:

1. We run our tests with a QRCode, but asked the users about the use of a secret phrase stamped on the trash can may not be the best way. Normally persons do not look for strange phrases found everywhere on the internet.

2. The first information on the game site should be that it is a game.
3. In the game mobile application, the images should appear in different screens.
4. Sometimes the application recognizes an image that is not effectively captured.
5. When the game is over, it would be great to display a message of congratulations and the total consumed time.
6. Call the player's attention when he/she is near of a mission.
7. Promote competition to enhance the social communication outside the game by giving badges and bonuses.
8. The list of missions in the mobile app appears too small and the user may end up clicking in the wrong mission.
9. The sound of the trash can should be shorter.
10. The quality of the sound from the trash can could be better.
11. The Mission list should be ordered according to the users distance to the mission location.
12. A timer should be displayed while the missions are in progress.
13. A second chance of accomplishing the missions in a shorter time should be possible.

5 Closure

5.1 Conclusions

We aimed to produce an experimental work on the development of pervasive mobile games that could cover from game concept to post-production. In this dissertation, we hope to have enriched a little bit the scarce literature on the creation of pervasive mobile games, by documenting this process and allowing future pervasive game designers to access it. Also we hope that this work can give a small contribution to a better understand of the quality requirements related to pervasiveness and help improving the practical aspects of the innovative work done by Valente (Valente et al., 2013) (Valente, 2011).

More specifically, in this work, we explore the use of quality requirements found in the above-mentioned works by Luis Valente (Valente et al., 2013) (Valente, 2011) to help the conceptual design of a new game. Also we briefly report the entire process, from the conceptual design to the post-production stage. The techniques and processes used in our experiment can help future game designers and developers to produce higher quality games using those quality requirements. Although we did not gather the feedback of other game designers and iterate over a series of user experience and design evaluation sessions, this work contributes with more pragmatic knowledge about the pervasive game development process. Besides the items listed in section 4.7, we also identified several design points that could lead to a better, simpler and faster game creation of pervasive games. These points can accelerate the implementation of future prototypes. This is a question of utmost relevance. In fact, we observed in our experiments that the role of prototyping in the development of pervasive games is much more crucial than in other types of games. We did not perform validation tests to guarantee the soundness of this statement, but we could verify its importance during the experiments.

The use of the quality requirements related to pervasiveness during the design process helped us to project a game that adheres to the features of this game type. By revisiting and iterating over these quality requirements during the design process, we could improve the quality of our game. We showed the last game analysis in Section 4.3.2, although we have analysed the game several times during the design process. During the development work, we

clearly noticed that the proposed quality requirements worked as a first design guide. Also we noticed that those quality requirements are interconnected with each other as shown in Figure 3.1 and proposed by Valente (Valente et al., 2013; Valente, 2011).

We list below some points in the game design that could be improved:

- We have not designed a reward system. This would have improved the entire experience.
- The lack of a complete guide to help in the design process creates difficulties to balance the importance of each requirement. For example, we almost forgot the need of light to take the picture(UHP).
- We did not explored the quality requirements through an extensive number of design cycles. This could have improved the quality of the game and its playability.

Also we have some more technical recommendations, which are listed below:

- The use of a queue management will help a lot in managing the synchronization process.
- Dealing with multiple software stacks and completely different apps is a difficult and time consuming task. In the proposed architecture, each one of the three layers (mobile, web, and smart objects) uses completely different software stack, and to switch applications is very time consuming. It would have been a lot easier if we had used the same application stack, such as java in all layers.
- Arduino is difficult to use with audio. Although there are audio boards for Arduino, it would be more suitable to find a better solution, such as Raspberry PI (Raspberry Pi - Teach, Learn, and Make with). We recommend the Raspeberry Pi platform, because it will permit easier communication procedures involving the trash can, the android app, and the website.
- The sensors in the trash can must be fixed tighter, so they will not move when the trash can is moved or something bumps in it.
- Sound is low. The use of an audio amplifier and good speakers are mandatory.
- LED Bright is important, so to use high brightness LEDs is a must.
- Be aware of the power drained by the high brightness LEDs. If necessary, use a driver circuit to preserve the micro-controller.

- Code Igniter runs differently on Linux, because Linux is case sensitive and windows is not. So, when developing on Windows to deploy to linux, additional attention should be taken with the classes filenames.
- Grocery Crud needed to be adapted to run on Linux. It has some case filename issues.
- The constant deprecation of classes in android sdk generated a lot of rework.

5.2

FUTURE WORK

There are a lot of future works to recommend. Firstly we wish to implement this game using a more productive framework, like Unity. Unity has higher productivity and now has support for augmented reality. Also it permits the construction of a single mobile application that can be run on iOS and Android devices. Secondly, the current version of the website is too simple. So, we recommend to expand it or, perhaps, to transfer its functionality into the app. This is an important question to be investigate. Moreover, there are the following improvements to incorporate in the prototype:

- Implement this game for iOS and Windows phone. This can shed light in the problem of multiple platforms and its impact on the development process.
- Build the trash can according to the full specification.
- Implement the authentication process to work with Facebook and an independent login system with e-mail validation.
- Implement push notifications.
- Interact more with players and designers.
- Perform validation tests.
- Expand the methodology to include an immersion item, to verify the hability of the game to put the user in a trance state during the play, and how that interfears with others quality requirements.
- Use solar power to recharge the trash can battery.
- Use the trashcan to aplify the sound.
- Create a more complete guideline to help the creation of pervasive mobile games.

6

Bibliography

AGRAWAL, M.; KONOLIGE, K.; BLAS, M. R. **Censure: Center surround extremas for realtime feature detection and matching**. [S.l.]: Springer, 2008. 102–115 p.

ALAH, A.; ORTIZ, R.; VANDERGHEYNST, P. Freak: Fast retina keypoint. In IEEE. **Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on**. [S.l.], 2012. p. 510–517.

BELL, M. et al. Interweaving mobile games with everyday life. In **Proceedings of the SIGCHI Conference on Human Factors in Computing Systems**. [S.l.]: ACM, 2006. (CHI '06), p. 417–426. ISBN 1-59593-372-7.

BENFORD, S. et al. Coping with uncertainty in a location-based game. **IEEE Pervasive Computing**, vol. 2, no. 3, p. 34–41, 2003. ISSN 1536-1268.

BENFORD, S. et al. Can you see me now? **ACM Trans. Comput.-Hum. Interact.**, vol. 13, no. 1, p. 100–133, 2006. ISSN 1073-0516. Available from Internet: <<http://doi.acm.org/10.1145/1143518.1143522>>.

British Columbia Institute of Technology. **CodeIgniter Web Framework**. [S.l.]: British Columbia Institute of Technology, May 2015. Available in <http://www.codeigniter.com/> [Accessed 01 jully, 2015].

Chris Kacerguis. **CodeIgniter Rest Server**. 2014. Available in <https://github.com/chriskacerguis/codeigniter-restserver> [Accessed 01 jully, 2015]. Version 2.7.1.

CROWD Control. Available in <http://channel.nationalgeographic.com/crowd-control> [Accessed 25 april, 2014]. (Crowd Control).

DANSKY, R.; BRUCATO, P. **Vampire: The Masquerade**. Revised edition. [S.l.]: White Wolf Publishing, 1998. ISBN 978-1-56504-249-0.

DU, C.-J.; SUN, D.-W. Learning techniques used in computer vision for food quality evaluation: a review. **Journal of food engineering**, Elsevier, vol. 72, no. 1, p. 39–55, 2006.

FARMAN, J. Locative life: Geocaching, mobile gaming, and embodiment. **Digital Arts and Culture Conference**, vol. 1, p. 6, 2009.

FLINTHAM, M. et al. Uncle roy all around you: mixing games and theatre on the city streets. In **DIGRA Conf.** [S.l.]: Citeseer, 2003.

Gartner, 2013. **Forecast: Video Game Ecosystem, Worldwide, 4Q13.** 4Q13. Available in <http://www.gartner.com/newsroom/id/2614915> [Accessed 15 june, 2015].

Google.2015.a. **Overview of Google Play Services.** Available in <https://developers.google.com/android/guides/overview> [Accessed 28 june, 2015].

Google.2015.b. **Android SDK.** Available in <https://developer.android.com/sdk/index.html> [Accessed 18 june, 2015]. (Android SDK).

Google.2015.c. **API Client Library for PHP | Google Developers.** Available in <https://developers.google.com/api-client-library/php/> [Accessed 06 june, 2015].

Google.2015.d. **Android Sliding Tabs Sample Application.** Available in <http://developer.android.com/samples/SlidingTabsBasic/index.html> [Accessed 30 june, 2015].

Google.2015.d. Comercial, **Ingress.** Available in <https://www.ingress.com/> [Accessed 04 july, 2015].

Google.2015.e. Comercial, **Ingress.** Available in <https://www.ingress.com/> [Accessed 04 july, 2015].

Google.2015.f. **Using OAuth 2.0 for Web Server Applications.** Available in <https://developers.google.com/identity/protocols/OAuth2WebServer#callinganapi> [Accessed 30 june, 2015].

GUNASEKARAN, S. Computer vision technology for food quality assurance. **Trends in Food Science & Technology**, Elsevier, vol. 7, no. 8, p. 245–256, 1996.

HOWSE, J. **Android application programming with OpenCV.** [S.l.]: Packt Publishing Ltd, 2013. <https://www.packtpub.com/application-development/android-application-programming-opencv>.

HUIZINGA, J. Homo ludens: A study of the play-element in culture. **Buenos Aires: Emecé**, Beacon Press, -, p. –, 1938.

IPERG. **IPerG - Integrated Project of Pervasive Games.** 2006. Available in http://www.pervasive-gaming.org/iperg_games5.php/ [Accessed 03 july, 2015].

jQuery Foundation. **jQuery - JavaScript library**. [S.l.]: JQuery Foundation, 2015. Available in <https://jquery.com/> [Accessed 06 june, 2015].

Json Foundation. **JSON**. 2015. Available in <http://www.json.org/> [Accessed 1, june, 2015].

JUUL, J. The game, the player, the world: looking for a heart of gameness. In **DIGRA Conf**. [S.l.: s.n.], 2003.

LETTERBOXING North America. Available in <http://www.letterboxing.org/>, [Accessed 03 jully, 2015].

Marczewski, A. Gamasutra: Andrzej marczewski's blog - game mechanics and gamification. 1993.

MARTIN, A.; THOMPSON, B.; CHATFIELD, T. **Alternate reality games white paper**. [S.l.], 2006. -, - p. IGDA ARG SIG Wiki.

McGonigal, J. **Reality is broken: Why games make us better and how they can change the world**. [S.l.]: Penguin, 2011.

MCGONIGAL, J.; BOGOST, I. **Cruel 2 B Kind - a game of benevolent assassination**. Available in <http://www.cruelgame.com/> [Accessed 25 april, 2014].

MCGONIGAL, J. E. **This might be a game: ubiquitous play and performance at the turn of the twenty-first century**. Tese (phdthesis) — University of California, Berkeley, 2006.

MONTOLA, M. Social constructionism and ludology implications for the study of games. **Simulation & Gaming**, vol. 43, no. 3, p. 300–320, 2012. ISSN 1046-8781, 1552-826X.

MONTOLA, M.; STENROS, J.; WAERN, A. **Pervasive Games: Theory and Design**. [S.l.]: CRC Press, 2009. ISBN 978-0-12-374853-9.

OAUTH 2.0. Available in <http://oauth.net/2/> [Accessed 21 june, 2015].

OPENCV4ANDROID. **Open CV port for Android Version 2.4.10**. [S.l.]: Itseez, 10 2014. Available in http://docs.opencv.org/doc/tutorials/introduction/android_binary_package/04A_SDK.html [Accessed 14 june, 2015].

OPENCV.ORG. **Open Computer Vision**. [S.l.]: Itseez, 10 2014. Available in http://docs.opencv.org/doc/tutorials/introduction/android_binary_package/04A_SDK.html [Accessed 29 june, 2015].

PEITZ, J.; SAARENPÄÄ, H.; BJÖRK, S. Insectopia: Exploring pervasive games through technology already pervasively available. In **Proceedings of the International Conference on Advances in Computer Entertainment Technology**. [S.l.]: ACM, 2007. (ACE '07), p. 107–114. ISBN 978-1-59593-640-0.

RASPBERRY Pi - Teach, Learn, and Make with. Available in <https://www.raspberrypi.org/> [Accessed 21 june, 2015].

SALEN, K.; ZIMMERMAN, E. **Rules of Play: Game Design Fundamentals**. [S.l.]: MIT Press, 2004. ISBN 978-0-262-24045-1.

SCHILIT, B.; ADAMS, N.; WANT, R. Context-aware computing applications. In **First Workshop on Mobile Computing Systems and Applications, 1994. WMCSA 1994**. [S.l.: s.n.], 1994. p. 85–90.

SCHLATTER, B. E.; HURD, A. R. Geocaching: 21st-century hide-and-seek. **Journal of Physical Education, Recreation & Dance**, vol. 76, no. 7, p. 28–32, 2005.

SCHMID, C.; MOHR, R.; BAUCKHAGE, C. Evaluation of interest point detectors. **International Journal of Computer Vision**, vol. 37, no. 2, p. 151–172, 2000. ISSN 0920-5691, 1573-1405.

Seattletimes. **The Game**. September 2008. Available from Internet: http://seattletimes.nwsources.com/html/pacificnw/2008177548_pacificpendgame14.html.

SIWEK, S. **Video Games in the 21st Century: The 2010 Report**. [S.l.]: Economists Incorporated, 2010. Available in <https://developer.android.com/sdk/index.html> [Accessed 03 july, 2015].

SKOUMBOURDIS, J. **Auto PHP Codeigniter CRUD - Grocery CRUD**. 2015. Available in <http://www.grocerycrud.com/> [Accessed 01 july, 2015].

SOTAMAA, O. All the world's a botfighter stage: Notes on location-based multi-user gaming. In CITESEER. **CGDC Conf.** [S.l.], 2002.

SUN, D.-W. Inspecting pizza topping percentage and distribution by a computer vision method. **Journal of food engineering**, Elsevier, vol. 44, no. 4, p. 245–249, 2000.

SUN, D.-W.; BROSANAN, T. Pizza quality evaluation using computer vision—part 1: Pizza base and sauce spread. **Journal of Food Engineering**, Elsevier, vol. 57, no. 1, p. 81–89, 2003.

VALENTE, L.; FEIJÓ, B.; LEITE, J. C. Features and checklists to assist in pervasive mobile game development. In **SBC - Proceedings of SBGames**. [S.l.: s.n.], 2013. XII.

VALENTE, L. P. S. **A methodology for conceptual design of pervasive mobile games**. Tese (phdthesis) — PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO - PUC-RIO, 2011.

VOLKSWAGEN. **The Fun Theory Project**. 2009. Available in <http://www.thefuntheory.com/> [Accessed 29 june, 2015].

WIKIPEDIA. **Assassin (game)**. June 2014a. Available in <https://en.wikipedia.org/w/index.php?title=Assassin%28game%29&oldid=611324472> [Accessed 12, june, 2014]. Page Version ID: 611324472.

Wikipedia.2014.d. **The Game (treasure hunt)**. Available in https://en.wikipedia.org/w/index.php?title=The_Game_%28treasure_hunt%29&oldid=603338587 [Accessed 21, may, 2014]. Page Version ID: 603338587.

WIKIPEDIA.B. **Political correctness**. 2014. Available in http://en.wikipedia.org/w/index.php?title=Political_correctness&oldid=611111026 [Accessed 09 june, 2014]. Page Version ID: 611111026.

WIKIPEDIA.C. **The 10th Victim**. 2014. Available in http://en.wikipedia.org/w/index.php?title=The_10th_Victim&oldid=609979562 [Accessed 30 may, 2014]. Page Version ID: 609979562.

WU, D.; SUN, D.-W. Colour measurements by computer vision for food quality control—a review. **Trends in Food Science & Technology**, Elsevier, vol. 29, no. 1, p. 5–20, 2013.

XAMPP Installers and Downloads for Apache Friends. [S.l.]: "Apache Friends". Available in <https://www.apachefriends.org/index.html> [Accessed 04 jully, 2015].

Your Dictionary. **Pervasiveness dictionary definition | pervasiveness defined**. 2015. Available in <http://www.yourdictionary.com/pervasiveness> [Accessed 03 jully, 2015].

ZHENG, C.; SUN, D.-W.; ZHENG, L. Recent applications of image texture for evaluation of food qualities—a review. **Trends in Food Science & Technology**, Elsevier, vol. 17, no. 3, p. 113–128, 2006.

ZHENG, C.; SUN, D.-W.; ZHENG, L. Recent developments and applications of image features for food quality evaluation and inspection—a review. **Trends in Food Science & Technology**, Elsevier, vol. 17, no. 12, p. 642–655, 2006b.

A

Android application class diagram

In Figure A.1 we show all implemented classes and how they correlate to each other.



PUC-Rio - Certificação Digital Nº 1312375/CA

B

Ingress evaluation

In this appendix we will present the Ingress game analysed according to the checklist of non-functional quality requirements related to pervasiveness.

Game Name: Ingress

Sensors : A-GPS

Network : Wifi / gprs

Support staff : -

Others : -

B.1

Device Independence(DI)

No.	Question	Answer
DI1	Is it possible to play the "same" game on multiple mobile device platforms?	Yes, in the beginning just on Android, after July 2014 on iPhone.

Table B.1: Device Independence(DI)

B.2

Uncertainty Handling Policy(UHP)

No.	Question	Answer
UHP1	Which strategy does the game use to handle technology limitations?	The reveal strategy
UHP2	How does the game handle technology limitations?	The game only reveals connectivity and GPS status
UHP3	Does technology correctly acknowledge physical world limits?	N/A

Table B.2: Uncertainty Handling Policy(UHP)

B.3

Local Space Redefinition (LSR)

No.	Question	Answer
LSR1	Does the game integrate physical places (or their elements) in the game-play?	Yes, by using landmarks as portals.
LSR2	Does the game give roles to physical places (or their elements) in the game-play?	yes
LSR3	Does the game help players to have alternative views of the physical environment?	yes
LSR4	What physical resources does the game use as game elements?	Landmarks
LSR5	Does the game use “live characters” to reinforce the mixed-reality overlay?	No
LSR6	How do game activities affect the physical world?	The game does not make any visible marks in the real world. But it may put in evidence the landmarks used.

Table B.3: Local Space Redefinition (LSR)

B.4 Game Object Tangibility (GOT)

No.	Question	Answer
GOT1	Does the game specify a role for mobile phones in the game, or are they mere access terminals?	No
GOT2	How does the game transform mobile phones into game objects?	N/A
GOT3	Does the game use other physical objects (other than mobile phones) equipped with sensors or actuators as game objects?	No
GOT4	Does the game use mobile phones as multiple props?	No
GOT5	How does using tangible objects improve player socializing in the game?	N/A

Table B.4: Game Object Tangibility (GOT)

B.5 Checklist for Game Pacing(GP)

No.	Question	Answer
GP1	Is the technology the games uses compatible with the game pacing?	Yes
GP2	Do game activities adapt their pacing to accommodate technology limitations in this regard?	No
GP3	Does the pacing of game activities require the player to focus exclusively on the game? For how much time?	When attacking a portal, the game requires the player to focus on the screen for about 5 to 60 seconds.

Table B.5: Checklist for Game Pacing(GP)

B.6 Involving Non-players(INP)

No.	Question	Answer
INP1	Does the game involve non-players? How does it do it?	No
INP2	Does the game have activities where players need to find out who the other players are?	No
INP3	Does the game generate/use content that is based on other (non-player) people?	Yes, the city landmarks.
INP4	Does the game use actors for non-player characters?	No.

Table B.6: Involving Non-players(INP)

B.7**Usability (USA)**

No.	Question	Answer
Usa1	Do game activities require players to focus on the device screen too much?	Yes, but most times it is on places crowded with portals.
Usa2	Do game activities use various modalities to interact with the player?	No.

Table B.7: Usability (USA)

B.8**Daily Life Interleaving(DLI)**

No.	Question	Answer
DLI1	Does the game provide a persistent game world?	Yes
DLI2	Are game activities designed to be interruptible?	Yes
DLI3	Do game activities require long time commitment from players?	No
DLI4	Which approaches or techniques does the game apply to blend with daily life?	N/A
DLI5	Does the game provide equal opportunities for playing in any time?	Yes
DLI6	How does the game communicate game-related events to players?	Yes, by e-mail and message box on game app.

Table B.8: Daily Life Interleaving(DLI)

B.9**Game autonomy(GA):**

No.	Question	Answer
GA1	Is the game bound to specific places or local context?	Yes. Part of the game content is based on physical landmarks that players have to reach to complete game activities
GA2	Does the game require configuring the physical space for a game session?	No
GA3	Does the game require any kind of supervision when players are playing it?	No
GA4	Does the game require custom hardware, actors, or other kind of related resources?	No

Table B.9: Game autonomy(GA)

B.10**Mobility(MOB)**

No.	Question	Answer
MOB1	Does the game need to use networking while the players are moving? If yes, how does the game handle networking limitation issues?	Yes, the game does not work without network.
MOB2	Does the game depend on specific locations to be played? Are players required to move between specific locations?	Yes
MOB3	What is the order of magnitude of the potential game area? Does the game require players to walk long distances?	The entire world is the play area, and stimulate players to walk long distances.
MOB4	How does the game keep physical player locations consistent within the game world?	It uses the GPS reported location, even though far from the real location.

Table B.10: Mobility(MOB)

B.11 Cross-mediality(CM)

No.	Question	Answer
CM1	Does it make sense for the game-play to include multiple devices with different roles? Does the game allow different modes of participation?	No
CM2	Does the game balance the game experience for the various specialized devices?	N/A
CM3	When using devices for multiple platforms (<i>e.g.</i> desktop, mobile), do the game activities have compatible paces,so as not to break the game-play?	N/A

Table B.11: Cross-mediality(CM)

B.12 Persistency(PER)

No.	Question	Answer
PER1	Do game sessions depend on previous sessions or stored data?	Yes/online data
PER2	Does the game support internal social networks or communities?	Yes, each faction has its own channel that anyone from that faction can send messages to the team and receive team messages.

Table B.12: Persistency(PER)

B.13**Social communication(SC)**

No.	Question	Answer
SC1	How does the game use technology to provide means to improve communication among people?	By publishing player's accomplishment to all players, it foster comments and help requests for team and individuals.
SC2	How does the game transform the relationships among players?	Empowering commitment and team work.
SC3	Does the game stimulate players to approach/start interactions with other people?	It is stimulated to form groups to walk around and get some portals.
SC4	Does the game use technology to foster community/social networks forming?	Yes in a minor scale, by calling attention to the landmarks in the city where people normally pass by, but the players needs to stops for a while.
SC5	Does the game present emergent gameplay?	Yes, each portal is created to match a landmark, and others can be added as players suggestions.

Table B.13: Social communication(SC)

B.14**Conformance to Physical and Social Settings(CPS)**

No.	Question	Answer
CPS1	How does the game handle player privacy?	It uses nicknames to identify players and publish only in game progress data.
CPS2	Do game activities possibly disturb non-players?	Yes, but just by having more people than usual stand up in a place
CPS3	Do game activities expose players (or non-players) to embarrassing situations?	No, it is not designed to expose players to embarrassing situations, although it can happen
CPS4	Do game activities conform to local social conventions/etiquette?	Yes
CPS5	Are game activities adequate to the physical setting of the game?	Yes

Table B.14: Conformance to Physical and Social Settings(CPS)

B.15**Connectivity (CON):**

No.	Question	Answer
CON1	What are the connectivity requirements for the game? Global? Local? None?	Global - everyone is informed of what happened. Local- the player that lost the ownership of the portal is informed
CON2	How does the game handle uncertainties related to connectivity?	No
CON3	What is the desired space scope for game activities?	Public places with landmarks and security.

Table B.15: Connectivity (CON):

B.16**Game content adaptability(GCA)**

No.	Question	Answer
GCA1	How does technology availability affect the game content and the player experience?	Without network or gps , there is no game
GCA2	Does context usage require players to move more in the physical world?	Yes, you can only hack a portal if you are close to it.
GCA3	Does context usage affect accessibility?	Yes, the visible portals are near the player, and to hack a portal (acquire to your team) if you are in the range of the portal
GCA4	Are players able to play the game in various places without manual intervention/re-installation/re-adaptation/orchestration?	Yes.

Table B.16: Game content adaptability(GCA)

C

Pervasive game checklist

C.1

Device Independence(DI)

No.	Question	Answer
DI1	Is it possible to play the “same” game on multiple mobile device platforms?	

Table C.1: Device Independence(DI)

C.2

Uncertainty Handling Policy(UHP)

No.	Question	Answer
UHP1	Which strategy does the game use to handle technology limitations?	
UHP2	How does the game handle technology limitations?	
UHP3	Does technology correctly acknowledge physical world limits?	

Table C.2: Uncertainty Handling Policy(UHP)

C.3

Local Space Redefinition (LSR)

No.	Question	Answer
LSR1	Does the game integrate physical places (or their elements) in the game-play?	
LSR2	Does the game give roles to physical places (or their elements) in the game-play?	
LSR3	Does the game help players to have alternative views of the physical environment?	
LSR4	What physical resources does the game use as game elements?	
LSR5	Does the game use “live characters” to reinforce the mixed-reality overlay?	
LSR6	How do game activities affect the physical world?	

Table C.3: Local Space Redefinition (LSR)

C.4

Game Object Tangibility (GOT)

No.	Question	Answer
GOT1	Does the game specify a role for mobile phones in the game, or are they mere access terminals?	
GOT2	How does the game transform mobile phones into game objects?	
GOT3	Does the game use other physical objects (other than mobile phones) equipped with sensors or actuators as game objects?	
GOT4	Does the game use mobile phones as multiple props?	
GOT5	How does using tangible objects improve player socializing in the game?	

Table C.4: Game Object Tangibility (GOT)

C.5

Checklist for Game Pacing(GP)

No.	Question	Answer
GP1	Is the technology the games uses compatible with the game pacing?	
GP2	Do game activities adapt their pacing to accommodate technology limitations in this regard?	
GP3	Does the pacing of game activities require the player to focus exclusively on the game? For how much time?	

Table C.5: Checklist for Game Pacing(GP)

C.6**Involving Non-players(INP)**

No.	Question	Answer
INP1	Does the game involve non-players? How does it do it?	
INP2	Does the game have activities where players need to find out who the other players are?	
INP3	Does the game generate/use content that is based on other (non-player) people?	
INP4	Does the game use actors for non-player characters?	

Table C.6: Involving Non-players(INP)

C.7**Usability (USA)**

No.	Question	Answer
Usa1	Do game activities require players to focus on the device screen too much?	
Usa2	Do game activities use various modalities to interact with the player?	

Table C.7: Usability (USA)

C.8**Daily Life Interleaving(DLI)**

No.	Question	Answer
DLI1	Does the game provide a persistent game world?	
DLI2	Are game activities designed to be interruptible?	
DLI3	Do game activities require long time commitment from players?	
DLI4	Which approaches or techniques does the game apply to blend with daily life?	
DLI5	Does the game provide equal opportunities for playing in any time?	
DLI6	How does the game communicate game-related events to players?	

Table C.8: Daily Life Interleaving(DLI)

C.9**Game autonomy(GA):**

No.	Question	Answer
GA1	Is the game bound to specific places or local context?	
GA2	Does the game require configuring the physical space for a game session?	
GA3	Does the game require any kind of supervision when players are playing it?	
GA4	Does the game require custom hardware, actors, or other kind of related resources?	

Table C.9: Game autonomy(GA)

C.10**Mobility(MOB)**

No.	Question	Answer
MOB1	Does the game need to use networking while the players are moving? If yes, how does the game handle networking limitation issues?	
MOB2	Does the game depend on specific locations to be played? Are players required to move between specific locations?	
MOB3	What is the order of magnitude of the potential game area? Does the game require players to walk long distances?	
MOB4	How does the game keep physical player locations consistent within the game world?	

Table C.10: Mobility(MOB)

C.11 Cross-mediality(CM)

No.	Question	Answer
CM1	Does it make sense for the game-play to include multiple devices with different roles? Does the game allow different modes of participation?	
CM2	Does the game balance the game experience for the various specialized devices?	
CM3	When using devices for multiple platforms (<i>e.g.</i> desktop, mobile), do the game activities have compatible paces,so as not to break the game-play?	

Table C.11: Cross-mediality(CM)

C.12 Persistency(PER)

No.	Question	Answer
PER1	Do game sessions depend on previous sessions or stored data?	
PER2	Does the game support internal social networks or communities?	

Table C.12: Persistency(PER)

C.13 Social communication(SC)

No.	Question	Answer
SC1	How does the game use technology to provide means to improve communication among people?	
SC2	How does the game transform the relationships among players?	
SC3	Does the game stimulate players to approach/start interactions with other people?	
SC4	Does the game use technology to foster community/social networks forming?	
SC5	Does the game present emergent game-play?	

Table C.13: Social communication(SC)

C.14**Conformance to Physical and Social Settings(CPS)**

No.	Question	Answer
CPS1	How does the game handle player privacy?	
CPS2	Do game activities possibly disturb non-players?	
CPS3	Do game activities expose players (or non-players) to embarrassing situations?	
CPS4	Do game activities conform to local social conventions/etiquette?	
CPS5	Are game activities adequate to the physical setting of the game?	

Table C.14: Conformance to Physical and Social Settings(CPS)

C.15**Connectivity (CON):**

No.	Question	Answer
CON1	What are the connectivity requirements for the game? Global? Local? None?	
CON2	How does the game handle uncertainties related to connectivity?	
CON3	What is the desired space scope for game activities?	

Table C.15: Connectivity (CON):

C.16**Game content adaptability(GCA)**

No.	Question	Answer
GCA1	How does technology availability affect the game content and the player experience?	
GCA2	Does context usage require players to move more in the physical world?	
GCA3	Does context usage affect accessibility?	
GCA4	Are players able to play the game in various places without manual intervention/re-installation/re-adaptation/orchestration?	

Table C.16: Game content adaptability(GCA)