FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Procedural Content Generation for Location-based Games

Ana Margarida Cardoso Carraca

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: António Coelho (PhD)

July 25, 2014

Procedural Content Generation for Location-based Games

Ana Margarida Cardoso Carraca

Mestrado Integrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

Chair: Jorge Barbosa

External Examiner: Leonel Morgado

Supervisor: António Coelho

July 25, 2014

Abstract

Location-based Games are games where the physical position of the player has influence on the gameplay. Most of these games are currently developed with a specific location and a narrative in mind, so as to provide the user with a more immersive experience. Thus, Procedural Content Generation has potential to be of great value to this type of games, as they can be extended to arbitrary locations in a less labor-intensive fashion.

The main goal of this dissertation is to create a prototype of a tool that is able to generate a model of the current location based on geographical data. This content should be created according to OpenStreetMap data, obtained every time the application is loaded. Non-player characters containing embedded knowledge of the location are used to interact with the player, and to present information about nearby sights in an interactive way. Procedural Content Generation techniques will be used to generate the components, and a possible impact of the use of this tool for tourism purposes was studied.

This study, based on the input of a group of volunteers, proved that the developed prototype is usable for the locations used in the tests. A comparison between different types of controllers was also done, revealing that there is no relevant difference between the first and the third-person perspectives.

Resumo

Os Jogos Baseados na Localização possibilitam o desenvolvimento de narrativas e experiências em locais específicos. Como tal, estão muitas vezes limitados pela área geográfica e para uma narrativa para a qual foram desenvolvidos, para que possam criar experiências mais imersivas para o utilizador. Através de técnicas de Geração Procedimental de Conteúdos é possível adaptar as mecânicas de jogo definidas a outros locais.

O principal objetivo desta dissertação é criar de um protótipo de uma ferramenta que permita a geração de um modelo da localização atual do utilizador, através da utilização de informação geográfica. A informação sobre a localização, obtida do OpenStreetMap é integrada em agentes, sendo utilizadas técnicas de Geração Procedimental de Conteúdos para gerar os componentes. Estes agentes contêm, assim, informação sobre Pontos de Interesse numa área próxima dos mesmos. A usabilidade e um possível impacto do uso da aplicação em turismo são estudados, provando que o protótipo desenvolvido é usável para as localizações escolhidas.

Este estudo, baseado em dados de um grupo de voluntários, provou que o protótipo é aceitável a nível de usabilidade para as localizações estudadas. Uma comparação entre diferentes controladores foi feita, revelando não haver uma diferença relevante entre as perspetivas em primeira e terceira pessoa.

Acknowledgements

First, I'd like to thank the Serious Games Institute, for this opportunity and for all the support and help provided. Special thanks to those that helped me by taking part in the testing sessions.

I'd also like to thank my supervisor at FEUP, Prof. António Coelho, for his help and input through de duration of the project.

Special thanks to Francisco, for his unconditional support and patience.

To my family, for understanding how important this was for me and for their support.

Finally, I'd like to thank my friends back in Portugal and everyone that I've met in Coventry that made me really enjoy my time here.

Ana Carraca

”My work is a game,
a very serious game.”

M. C. Escher

Contents

List of Figures

LIST OF FIGURES

List of Tables

LIST OF TABLES

Abbreviations

AI	Artificial Intelligence
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
CGA	Computer Graphics Architecture
cgNEAT	content-generating Neuro Evolution of Augmenting Topologies
EA	Evolutionary Algorithms
FPS	First-person Shooter
GA	Genetic Algorithm
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobile Communications
GUI	Graphical User Interface
J2ME	Java 2 Platform, Micro Edition
JSON	JavaScript Object Notation
LBG	Location-Based Game
MMORG	Massively Multi-player Online Role-playing Games
NPC	Non-player Character
OSM	Open Street Maps
PCG	Procedural Content Generation
POI	Point Of Interest
PRNG	Pseudo-random Number Generator
SGI	Serious Games Institute
SMS	Short Message Service
SUS	System Usability Scale
XML	Extensible Markup Language
2D	Two-dimensional Space
3D	Three-dimensional Space

Chapter 1

Introduction

In the first chapter of the dissertation, the context and motivation for the development of the project are going to be presented. The work done is also going to be explained, in the context of the institute where it was developed. In the end of the chapter, there is a brief description of the structure of this document.

1.1 Context

With the development of computer technologies, game platforms have also evolved, and the idea of being sat in front of a desktop in order to play a game is no longer the only that is accepted. These original desktops have been, in part, replaced by smaller and more portable devices, like mobile phones or tablets, that give the user the freedom to move. This way, and with the use of previous experiments (for example Geocahing),a new set of environments for gameplay activities, and the concept of location-based games, have been created [JBFT07].

These games are normally developed to fit the requirements of a specific place, so the location where games are played has influence on the gameplay, creating a more immersive experience. This way, they need to be re-factored so they can be used in a different location. Procedural Content Generation has the potential to find a solution to this problem, extending this type of games in an easy way to different locations.

This project was developed at the Serious Games Institute (SGI). This Coventry University institute was set up to follow the rapid changes in game technologies, computer modelling simulations and digital media, that are now used for non-leisure contexts. The Research and Development Group at the SGI provides applied research support for the development of games in the areas of health, education and environment, but also virtual world technologies. Their research is focused on comparing traditional and game-based learning approaches and to find new uses for game technologies, linking the research and development community in the UK and abroad [Mag13].

1.2 Goals

The main goal of this project is to develop a way to procedurally generate content, based on the geographic information of the location.

In order to achieve that, it is necessary to study and develop a method that allows to adapt location-based experiences to different urban locations. These experiences should be based on geo-specific information. This information to be integrated should include terrain mapping, for the generation of the game environment, and external information about urban environments, including points of interest, streets or other information.

By the end of the development, it would be possible to use Procedural Content Generation methodologies to generate game scenarios (including the road pattern of the location and some other relevant details) for Location-based Games, and have a possible impact of these games in tourism. This should make possible the adaptation of games to different cities, and still maintaining the goals and mechanics of the game design, as well as its gameplay.

1.3 Structure

Besides the introduction, this dissertation contains 5 more chapters. In Chapter 2, the related work that has previously been developed is described, and related projects and techniques are presented. In Chapter 3, the proposed solution is explained, extending the problem presented on the introduction. In Chapter 4, the technical details of the implementation of the project are presented. In Chapter 5 the tests are described and the correspondent results are presented. The Chapter 6 presents the final conclusions and future directions that can be added to the project.

Chapter 2

Related Work

This chapter is going to present the state of the art of the two most relevant topics of the project: Procedural Content Generation and Location-based Games. It is also going to present some related work in the topic of Mobile Tourism, as the location mechanics used in Location-based Games can be of great use in Tourism applications.

2.1 Procedural Content Generation

Computer games are, more than ever and still increasingly, present in people's lives. Games like FarmVille, Sims, World of Warcraft are played by millions of people around the world [HMVI13].

The development of a complex game is a task that requires a significant amount of time and effort, to produce content such as characters, maps, objects or terrains [CBPD11]. The quantity and quality of that content have an important influence on the gameplay.

The development of real worlds is a particularly demanding task, as it requires a high level of details that, unlike fantasy settings, should be conceived according to the real world information [SC10].

The current trend on computer games is to be able to create larger and richer worlds, providing abundant and interesting contents for the players to explore [Rud09]. However, this is a task that implies a lot of resources and time consumption, both for game developers and designers. This way, such tasks call for large modelling teams, long development times and, consequently, great production costs, with a big impact on the project costs and budget. So, the current biggest challenge is to find a way to improve the level of detail of the contents in a game and, at the same time, lower the costs implied on its creation.

Procedural techniques were born from this need, to describe complex shapes or animations that are too difficult to specify explicitly. The use of these techniques in computer graphics is now widespread [MCPS96].

2.1.1 Concept

Referred to as Procedural Content Generation, the employment of generative computational algorithms is used to create content. This content can be more easily adapted as it is not fixed like content developed by hand. This way, it is possible to reduce the involvement of a designer and the time needed to develop the same amount of content [SPD11].

However, this method can be complex and unintuitive to use for some designers, as it is hard to control and can be difficult to integrate the final results in the virtual world [STdKB11]. This is due to its random nature and the absence of integrated solutions, being sometimes not so used as manual content creation [CAR+10].

Some of the benefits can be summarized in the next items [Hal08]:

- **User-generated content** — A regular user may not have the required artistic skills to produce game-quality content in a small amount of time. With a procedural system, this can be done through some intuitive inputs, in a quick and easy way for anyone.

- **Programmer-generated content** — If a team is not strong in terms of arts, this shouldn't be a problem and quality content can still be produced by programmers, using procedural methods, only with the supervision of an artist. This way, unlimited content can be produced without being directly from the artist.

- **Productivity** — In order to produce the same output, a procedural system requires fewer inputs, and less work from the designer, increasing its productivity.

- **Data compression** — This can be useful to send content over a network or to be sure content can fit in a limited storage. To take advantage of this compression, the algorithmic work should be done after the distribution of the program (on-the-fly generation).

Despite all the benefits, there are some aspects that can fail with the use of PCG for game development [Coo07]:

- **Setting goals** — When a game is dynamically generated, it can be to easy to understand the pattern of the goals of the game, and players will just miss the fact that goals are supposed to give the player an opportunity to learn something or get something from the achievement of it.

- **Social factors** — The procedural concept can fail on the creation of conversation, voice acting or plot, if these concepts are not close to the real human behaviour.

- **Density** — The fact that procedural content generation has the ability to generate infinite levels, it does not mean that games should support infinite levels. This way, the game can lose its purpose, and become repetitive and lacking creativity.

- **Programmer-artist** — It is necessary someone with both design and programming skills to make procedural content work and look good at the same time, and good programmers with design or art skills can be rare.

Despite this, the use of procedural methods is becoming more frequent due to the interesting results and the reduction of costs [SC10].

2.1.2 Methods

There is a range of different methods to generate content, and each method can fit different purposes and generate different types of content. According to [HMVI13], methods for PCG can be subdivided in different classes:

- Pseudo-Random Number Generators;
- Generative Grammars;
- Image Filtering;
- Spatial Algorithms;
- Modelling and Simulation of Complex Systems;
- Artificial Intelligence.

Some state-of-the-art methods and some possible uses are going to be presented in the next sections.

2.1.2.1 Pseudo-random Number Generation

A Pseudo-random Number Generator (PRNG) also known as a Deterministic Random Bit Generator, is an algorithm that creates a sequence of bits that is determined from an initial value (seed). Although the output is not perfectly random, it appears to be, as it is statistically indistinguishable from random values [BBB+12]. This method was one of the first approaches of PCG on games, and it is also one of the simplest [HMVI13].

PRNGs are commonly used for texture generation as a noise generator, and sound generation [Far07]. For road generation, this method can be useful to calculate a random parameter for deviance to enable some relevant noise at each step of the creation [KM07].

To generate the universe in the game Elite (1984, and its sequel from 2014), this method is used (Figure 2.1). The sequence of bits that was generated by the algorithm is used to determine the position of the stars. The games Sangband and Unangband also use techniques based on PRNGs to generate mazes and rooms [HMVI13].

Figure 2.1: Screenshot from the game Elite [Bra12]

2.1.2.2 Evolutionary algorithms

Evolutionary Algorithms (EA) are able to locate a diverse range of optimal results in a complex function, and that way they are a great choice for creating some types of content.

One use of these algorithms is for the creation of mazes and puzzles for opponent games [Ash10]. One good example of EA are Genetic Algorithms.

- **Genetic Algorithms** — A Genetic Algorithm (GA) is a search heuristic, in the field of Artificial Intelligence, that was inspired on the method of natural selection present in nature [Mel96]. It is included in the group of EA and it is composed by a series of chromosomes, that typically take the form of bit strings.

  A GA involves three types of operators:

  - **Selection** — Selects chromosomes in the population for reproduction (the fitter the chromosome, the higher the probability to be selected to reproduce);
  - **Crossover** — Chooses a locus, and exchanges the sub-sequences before and after that locus between chromosomes to create two offsprings;
  - **Mutation** — Flips the bits in a chromosome randomly.

  The GA requires a Fitness Function that assigns a score to each chromosome in the current population, depending on how well that chromosome solves the problem.

  Textures can be generated by mutating symbolic expressions [Sim91]. It can be applied, for example, an equation that calculates a color for each pixel coordinate, and then be applied vector transformations, noise generators or image processing techniques.

  GA can also be used to generate vegetation [Sim91], indoor maps [TYSB11] and terrains [FVC12].

2.1.2.3 Fractals

Fractals are shapes that contain a large degree of self-similarity, and this means they usually contain smaller copies of themselves [KM06]. This way, they possess infinite detail, and the closer they are observed, the more detail they can reveal [PW99]. Fractals are defined by a recursive algorithm, and the more recursion, the more detailed the shape will be.

There are a lot of different contents in nature that can be created through the use of Fractal algorithms, like plants (ferns are a very intuitive way to observer fractals) and trees (the Figure 2.2 is an example), rivers and snow flakes [Opp86].

In order to create a random terrain, a Fractal algorithm can be applied [CAR+10], and it has the advantage of being able to be computed at any desired resolution [FVC12].

2.1.2.4 Grammars

- **Lindenmayer Systems** — Also known as L-systems, it is a system for parallel string rewriting mechanisms, based on a set of production rules, where each string consists on a set of modules that are interpreted as commands (the parameters for these commands are stored in the modules). They use formal grammars to describe how components are altered, and are defined by a set of variables, a set of constants, a start state of the system and a set of production rules [MHSN10].

  When writing a complex rule system, there is a large number of parameters and conditions that need to be implemented to the L-system [PM01].

  These systems are capable of describing grammars that can generate building models, urban features and vegetation [MPB05].

  For the creation of vegetation, it is possible to generate plants that interact with the environment, and have features like the development of tree branches limited by collisions or the competition between trees to obtain more water in the soil or more access to sun light [MP96].

  Another example of the use of L-systems is XL3D, a modelling system that creates virtual environments. These environments are generated using Geospatial L-systems, that are an extension of the L-systems incorporating the ability to perceive the spatial relations between urban elements and between them and the environment, known as geospatial awareness. Virtual environments created by this system can be highly detailed, including a high level of visual fidelity [CBSF06].

- **Shape Grammar** — A Shape Grammar is a set of shape rules, that are applied in order to generate a set of designs. This application has, normally, two main purposes:

  – Design tools to generate varieties of design languages;

  – Behave as design analysis tools, that can be used both to analyse existing designs (to better understand them) and to generate shape rules that produce the designs.

Figure 2.2: Tree generated by a Fractal algorithm [Opp86]

That is why shape grammars are useful both to decompose existing designs and create new designs [MIBG13]. In the architecture field, structures of buildings have many repeating shapes and patterns, and those can be described using rules [CAR+10].

The definition of a shape grammar should include a finite set of shapes (Σ), a finite set of shape rules (R), a finite set of initial shapes (I) and a finite set of final shapes (F).

It starts with an initial axiom shape and rules are iteratively applied to it, replacing selected shapes with others. A rule has a shape on the left (labelled), called a predecessor, and one or more shapes and commands on the right, called successors. Commands are macros that can generate shapes or new commands [Pat12].

Shape grammars have become widely used in the generation of building façades, as it can be observed by its commercial release [TSL+11].

A CGA shape grammar is a grammar, for the procedural modelling of buildings, to obtain large scale city models with high visual quality and high level of detail. The user can specify interactions between the entities of the hierarchical descriptions due to the context sensitive shape rules. In the figure 2.3, it is possible to see a building composed by volumetric primitives (cubes and roofs) that was created using only 6 rules [PE06].

The CityEngine has been developed as a procedural approach to model cities based on an adapted model of L-systems, being used to create large-scale road patterns and generate buildings [PM01]. In 2006, this evolved into a framework that enables users to reconstruct architectural designs, using shape grammar rules as inputs. This system is able to process environments of any size, being the data represented as a GIS format [HKS08].

- **Context-free Grammar** — Context-free Grammars can represent Context-free Languages, another method that is used for PCG. A context-free grammar, G, is defined by the ordered tuple in 2.1 [HMU00].

$$G = (V, \Sigma, R, S) \tag{2.1}$$

In this tuple,
V is a finite set of variables,

Figure 2.3: Building generated by a Shape Grammar [PE06]

Σ is a finite set of terminals ,
R is a finite relation and
S is the start symbol or variable.

Grammars use rewrite rules to generate the strings in the language. In this context, what the grammar has to generate are strings that correspond to blocks of the content to generate. The language generated by a grammar is the set of strings, containing only terminal symbols, that can be generated.

An example of a possible application of this method, is on the generation of mazes or dungeons. In this case, a grammar is constructed with objects like `opponent`, `health`, `key` and `door` as terminal symbols, and non-terminal symbols are also added to create larger subsets. In the Figure 2.4, there is a representation of a level generated by the system represented by [Ada02]:

```
start opponent opponent JUNCTION
          (JUNCTION (switch door health bonus)
                    opponent opponent opponent health bonus bonus)
health health bonus end
```

Another example of the application of Context-free Languages, is on the creation of vegetation (Figure 2.5) and textures [Ada02]. It can be also applied for the creation of sound [Far07].

2.1.2.5 Tiling

Tiling is a basic procedural technique, that is used by creating small sections of 2D graphics that could be repeated on screen and assembled together, to create the virtual world. It can also be used in the form of multi-texturing to create highly detailed and varied textures from layers of base textures.

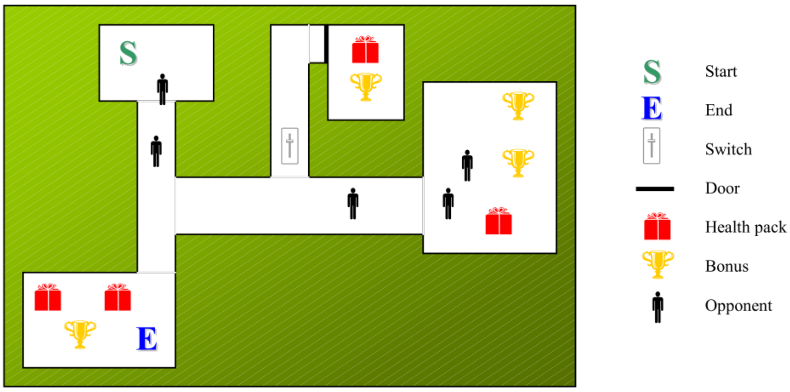


Figure 2.4: A possible geometric description of the level [Ada02]

The creation of new materials results from the combination of a set of detailed textures, colour maps and blending maps. Using this technique, terrain can be procedurally textured by the application of several layers of detailed textures, and vast areas can be textured in detail (which is not possible using a single high resolution texture).

This systems can provide several advantages, as vast and detailed landscapes and terrains can be created from small sets of texture tiles. These maps can be easily used for online gaming, such as massively multi-player on-line role-playing games (MMORPG) and other online applications where game resources are shared. Requirements concerning storage and memory are minimized, and so it is possible to store and render worlds of vast dimensions in real-time.

Tiling is one of the most basic procedural techniques, and has been applied in the development of many classic games including Sonic and Mario [KM06].

2.1.2.6 Grid Layout and Geometric Primitives

This is an approach for the development of cities that have a road network with a grid layout, and combining simple geometric primitives, buildings can then be placed generating a city in real-time.

The roads of the city are created with the pattern of an uniform grid, where the size of each block is constant (can be adjusted globally). Building generation uses the location of the building, in form of a coordinate in the grid, as a seed for its generation. This way, the appearance (height, width and number of floors) of the building is determined by this seed. Buildings can be extended over more than one grid block, to create a more disjointed road network and give the city a more realistic appearance.

The geometry of the buildings is generated using the concept of combining geometric primitives to form the sections of the building. Each section is built using a different floor plan.

An example of techniques applied to generate cities are demonstrated in a virtual city named Undiscovered City (Figure 2.6), being a proof of concept that it is possible to run in real-time

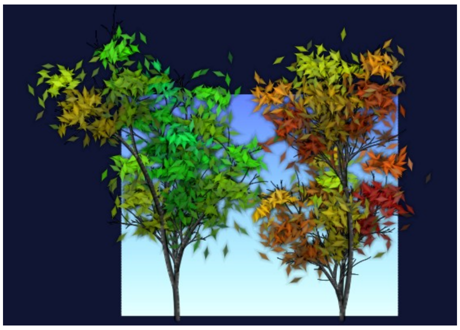


Figure 2.5: Tree generated using a Context-free Language [Len08]

at interactive frame rates. This system applies some optimisations, like rendering only the visible buildings (easily detected because of the regular grid network) and implementing a building cache [GPSL03b].

2.1.2.7 Perlin Noise

Perlin Noise is a noise function, initially created to help creating more natural textures to be used in the film Tron (1992) [KM06].

Noise is generated by using a pseudo-random value generator, which is then interpolated into coherent noise [Rud09]. This noise composes layers together using different ratios to build rounded fractals used for generating heavily eroded terrains [Dav03].

This technique, and more recent adaptations, is very popular and it is often included by default in most graphic software and video games APIs.

A prototype of a 3D landscape with trees and clouds, generated based on Perlin Noise, demonstrates a terrain procedurally generated, that can be freely explored on the horizontal plane in real time because it expands around the user's point of view [GPSL03a].

The generation of terrains using Perlin Noise is based on the creation of a height map purely procedurally generated [CBPD11].

2.1.3 Use of Procedural Content Generation in Games

PCG has been used in a wide range of games during the last decades. Some of these games are presented in this section.

- **Rogue (1980)** — *Rogue* is an ASCII graphic role-playing game, whose major innovation was its ability to generate dungeon levels without a limit. The levels are generated by dividing dungeons into solid rock and empty space, and distributing monsters and loot in the empty space. Although the levels were not so complex as they could be if they were hand-designed, the innovation was the ability to generate infinite levels without human supervision [CM06]. *Rogue* would definitely be a different game without the use of PCG, as it wouldn't be able to surprise the player more than on the first play-through [SGOgW11].

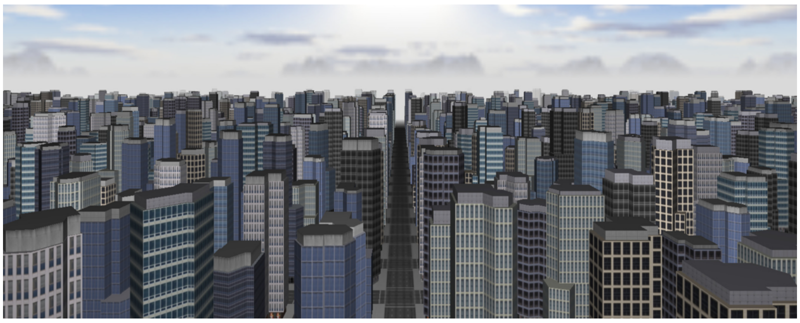


Figure 2.6: Undiscovered City [GPSL03b]

- **Elite (1984)** — As a space exploration game, *Elite* was one of the earliest games to generate a full world (including planetary positions, names, politics and general descriptions [BL09]) procedurally, using PRNG [HMVI13] and a good example of the compactness obtained by this method [FVC12]. *Elite* was composed by eight galaxies, each containing 256 stars, providing an expansing environment, stored in only a few kilobytes [LCL11].

- **Civilization (1991)** — *Civilization* is a series of strategy games where the world map is generated by simple methods, like seeding islands in the middle of the ocean that grow in random directions [TYSB11]. However, PCG is not the core of the experience for most of the players, and it is possible to play using human-authored maps [SGOgW11].

- **Diablo (1996)** — *Diablo* has a similar level generation from the one used in *Rogue*: the infinite creation of dungeons giving the player many more rewarding hours of game-play than fixed human-designed levels [CM06].

  Although the dungeons were generated, they were found to be very similar. This was due to the fact that some restrictions and constraints need to be applied so all dungeons fit a certain size and shape and that they should always be possible for the player to complete [Ada02].

- **.kkrieger (2004)** — The game *.kkrieger* is an interesting example of PCG, as all its content is stored only in 97 280 B [LCL11] (four orders of magnitude less than a similar game) [HMVI13]. It is a 3D FPS having demoscene-like[1] goals, that uses procedural techniques to generate textures, meshes and sounds that are used to create a complex, immersive game.

- **Dwarf Fortress (2006)** — *Dwarf Fortress* is a complex game that has been in development since 2002. It has more than one game mode, but when someone talks about the game, it is usually about the simulation mode, in which the player has the task to build a dwarf city.

  The complexity of the simulation, that includes complex interaction among hundreds of different objects placed in a procedurally-generated game world [HS02], allows for some

[1]Demoscene - computer art subculture that produces demos that consist on visual presentations running in real-time, showing programming, artistic and musical skills.

wonderful emergent game play as various game elements collide in interesting and challenging ways [Tys12].

The content that is procedurally generated includes outdoor maps and the ecosystem [HMVI13].

- **Left 4 Dead (2008)** — *Left 4 Dead* (and its sequel *Left 4 Dead 2*) uses a director program than handles the content of the game. This program places the enemies in different positions and amounts based upon the current situation of the player and also status, skills and location [ZM12], generating scenarios based on the analysed stress level of the player [HMVI13]. There are also physical gestures that are mapped to the game, giving more interest and more replay value. In the sequel *Left 4 Dead 2*, the geometry and content of the levels also evolve according to the gameplay [LCL11].

- **Spore (2008)** — *Spore* is a game that asks the player to follow and control the development of a certain specie from its beginning (microscopic organism) until it is an intelligent and social creature that is able to explore the galaxy. Planets (visual and characteristics) are generated procedurally [HMVI13], by the generation of its terrain [AN07].

  Players are able to create their own creature and procedural methods are then applied to generate the animation of the variation of the creatures [LCL11].

- **Borderlands (2009)** — This game is a cooperative online FPS that uses PCG to create a big number of weapons (around 17.750.000 according to [Rob09]) through the use of a single parameter vector and the combination of different properties [SGOgW11]. Although many of the weapons are not useful, exploring these items is an important part of the gameplay [TYSB11].

- **Galactic Arms Race (2010)** — *Galactic Arms Race* is a game built on search-based PCG where players pilot a space ship through the sectors of a galaxy and fight enemies to get new particle system weapons [HGS09]. These weapons are built using the cgNEAT algorithm and are represented as vectors of real values, that are interpreted as connection topologies and weights for neural networks. The fitness of a weapon is calculated depending on the time that the weapon is being used by the players relative to how long the weapon sits unused on the weapon cache. This function is interesting, because players can indicate their preference without needing to know the mechanics of the game [TYSB11].

- **Minecraft (2011)** — *Minecraft* is a recent game with "primitive" graphics that uses PCG both for re-playability but also exploration. The player is provided with a vast world to explore and the formation of the world influences players' strategy [SOGWWF12].

  A three-dimensional world is procedurally generated before the player is spawned on the game and the game is fired up for the first time. The world is then presented as a collection of meter-square blocks, from tree leaves to the clouds floating overhead. This procedurally generated world leads players into exploration mode, walking around the world to find the highest peaks and the deepest caverns [Dun11].

2.2 Location-based Games

Mobile phones and tablets are sometimes treated as just another platform for which games can be developed and published. The possibilities that these devices provide, via their ability to keep connected and easily located even while moving, are sometimes ignored.

The possibility to extend the virtual world of a traditional game to the real world, allowing the user to play games that have information of their physical location, provides the ability to interact in both worlds [RMC06].

2.2.1 Introduction

A Location-based Game (LBG) is a game that uses the physical position of the player (or any other location) as a mean of input or to generate or access location-based information, usually via a GPS sensor module [Jac11]. Due to the requirements of mobility, these games are almost exclusive for mobile platforms. LBGs allow to learn more about how people interact with the environment they are in, as well as to understand mental models and locations associated with a given environment [BMMU11].

2.2.2 Location Methods

Different location methods provide different levels of performance and capabilities. This should enable the highest precision possible with the smaller delay.

Some methods for the location of the mobile device are [RMC06]:

- **GPS** — GPS was originally developed by the US military and it is based on 24 satellites that orbit the earth. A GPS receiver can calculate its position if at least three satellites are visible, using Time of Arrival methods. Its accuracy can vary between two and ten meters, but higher resolutions can be achieved.

  There are some external factors that can influence the result: rounding errors, clock inaccuracies, multipath, atmospheric effects, indoor location, etc. More than three satellites can be used to find a more precise location.

  Assisted GPS systems can overcome some of these problems by using a cellular connection to transmit remotely-collected satellite navigation data. The use of this method can reduce acquisition time and be able to provide indoor accuracy to within 50m.

- **Implied Location Solutions** — In these solutions, mobile devices can interact with objects or systems that have a known location. Examples of this are Bluetooth and WLAN, and the triangulation method can also be applied.

- **Mobile Phone Network** — This method can locate the position of the mobile device by locating the Base Transceiver Station that the device is using. However, the location is really inaccurate as it can be anywhere between 2 and 20 kilometres from the Station.

2.2.3 Issues

There are some issues that can be found in most of the LBGs [JC11], due to hard to predict problems that can occur. These issues normally fit in one of these categories:

- **Game design issues** — Regarding game design issues, it is important to consider where and how the game is going to be played, to keep the player safe. LBGs expose the player to the real world and it can involve some type of unpredictability, that can be reduced if the player's behaviour is limited due to the gameplay.

- **Hardware limitations** — In LBGs some hardware is used and it is important to have it into account. An example are the location methods, like GPS or data connections, that can fail and the game should still be playable without using them, but instead use an alternative, sacrificing to a minimum the gameplay experience.

- **Location-related information availability and suitability** — LBGs often use information such as maps, weather, or some other types of location-related information to make the game unique and location-based. Always use remotely stored content that can be updated via web services, in order to guarantee that the information is always up to date and available to anyone independent of the device's storage. However, it should be taken in consideration the fact that the information can be inaccessible (for any reason) and the game should still be possible to play.

- **Player's fitness and pace** — As the location of the player has influence in the gameplay, his movement around the real world is, in most of the games, used as an input. There are games where the player's speed has also influence. This means that some games can be too hard or even impossible to play by an unfit player, providing an unfair gaming experience. To solve this, the game should be able to take the player's pace into account and balancing its difficulty in real-time, so it is not too easy or too difficult. Another possible aspect that games do not have in mind normally, is the need to stop to catch their breath. At these moments, the game should automatically pause or, at least, slow down significantly, giving the player the chance to gather their strength, and not losing the game due to the pause.

- **Player's data protection** — There are some LBGs that cache the user's location information for future uses or for statistical purposes. The storage of this data should be done remotely, as it is a safer place considering how easy it is to lose a mobile phone. As a security measure, a username and password should always be asked before it is possible to access any data.

  Always alert the player about the risks, so that they are informed about these issues.

None of these issues is an impossible problem to solve, so it is important that they are always kept in mind during the development of a LBG.

2.2.4 Examples

Location mechanics have been used in the development of games for a while, and some examples are described next.

- **Geocaching (2000)** — With more than a million registered users [Jac11], *Geocaching* is one of the most popular location-based games in the world. A simple game that consists on a series of "caches" (a container of some kind that holds a "treasure") positioned anywhere in the world and described on the website[2]. The goal of the game is to find the most "caches", exploring the area where the cache is, and solving riddles in order to find the coordinates of the exact position [JC11].

- **Botfighters (2001)** — *Botfighters* is a GSM-cell positioning SMS-shooter game [Wal07] that recreates a paintball game. It is a search and destroy combat game, regarded as the first location-based game to see commercial use [JC11].

  In the first version of the game, a player should send a SMS to find the location of another player and if the player is nearby, he could shoot the other player's robot by sending another SMS. The use of SMS and a location method with small accuracy creates a delay in the game and there was almost no interaction between players. So, a new version of the game, *Botfighters 2*, was released. This version uses a J2ME client and GPRS connectivity in order to upgrade and incorporate collaborative gameplay, trading weapons and bonus material, AI-controlled opponents and character upgrading [RMC06].

- **Mogi (2003)** — *Mogi* is a cell phone and web-based item collection and trading game where the actual position of a player corresponds to the position in the game world [Wal07]. The goal of the game is to obtain as many points as possible, by collecting treasure items randomly spreaded over a virtual map of Tokyo. Although the gameplay is singleplayer, players can trade objects with others to complete their collection and get an higher score [RMC06].

- **Pac-Manhattan (2004)** — *Pac-Manhattan* is an urban game that uses the New York City grid to recreate the videogame Pac-Man. A player, dressed as Pac-Man, runs around the Manhattan area and tries to collect all of the virtual "dots" like the original game. Other four players, dressed as ghosts, attempt to catch Pac-Man before all of the dots are collected. Besides these 5 players, there are 5 more players that are one-of-the-field players' controllers, and only these controllers have the knowledge of how the game is developing and will guide his respective field player [BBC+09].

- **Fugitive (2006)** — *Fugitive* is a multiplayer game, played on the University of British Columbia campus using mobile Tablet PCs. The goal is to find and capture a hidden object called "The Fugitive" on a digital map of the campus, communicating among the 3-person team. The map shows the position of the player providing visual clues that signal if The Fugitive is near.

[2] http://www.geocaching.com/

The game includes two parts: a catch phase, where the players move around to find the invisible and stationary fugitive forming a triangle around it; then there is a chase phase where the players need to chase the now moving object [JBFT07].

- **SCOOT (2007)** — *SCOOT* is a game that integrates web, mobile devices and public displays as tools to guide groups through public places, using mixed reality. Players are supposed to search hidden objects in public places, being sent clues via SMS, challenging them to play mini games in both worlds in order to reveal the dynamics of the place and how to go on in the game [PM07].

- **CityExplorer (2008)** — *CityExplorer* is a game whose main purpose is to produce geospatial data that can be useful for non-gaming applications.

  The game is inspired by the board game *Carcassonne*, that always starts with a single tile of a fragmented and hidden game board. Players draw a new line and lay it down to extend the land of the game, placing one of their markers on the tile that had just dropped. Once all tiles are laid down, the final scoring takes place. The player with the highest score wins.

  For *CityExplorer* the idea is the same, but the board is replaced by the real world. In order to win *CityEngine*, the player needs to set up as many markers as possible in a citywide game area. The game area is divided into squares, where the setting of markers is allowed, but only on predefined categories. The player who has the most markers in a square claims the domination and is assigned credits at the end of the game [MMS+08].

- **Free All Monsters! (2011)** — *Free All Monsters!* is a game who incorporates user generated content, to encourage creativity. The goal of the game is that participants explore the city and visit specific locations that host strategically placed monsters. They should "free" the monsters that are in close enough proximity. Players are provided with a "Monster Spotter's guide" and if they want to generate their own monsters for inclusion in the game, they can do so on the website [LCW11].

2.3 Mobile Tourism

Technological developments during the last decades, specially the ones on information and communication technologies, have changed different sectors of the economy. This includes the tourism sector, bringing changes in marketing and advertising, but also destination planning [AB11]. This easy access to information led to an easier collection of data about destinations, POI and travel plans in general [BMS+12].

Tourists want to be able to access personalized and updated information any time and anywhere, using any type of media [SGP+]. In order to provide this type of service customization, systems need to be highly adaptive, but also intelligent, mobile and advisory. This can apply both to the preparation of the vacation but also during the vacation itself. These type of applications, usually including location-based mechanisms, are called Mobile Tourism Guides. There are a series of Mobile Tourism Guides that have already been implemented, and are presented next.

2.3.1 Mobile Tourism Guides

Some Mobile Tourism Guides are going to be presented next, based on the evaluation made in [SGP+].

- **CATIS** — *Context-Aware Tourist Information System* (CATIS) is a web-based service that provides the user with information about his location for the current time, according to the speed, direction and personal preferences [PBHS03].
- **COMPASS** — *COntext-aware Mobile Personal ASSistant* (COMPASS) is a context-aware recommendation system that uses external map services through proprietary interfaces and also third party interfaces that provide content such as museums and restaurants information. Context properties (weather, traffic, ...) can be incorporated via web services [SPK+04].
- **CRUMPET** — *Creation of User-friendly Mobile Services Personalized for Tourism* (CRUMPET) provides the user with information or recommendations about restaurants, tours or tourist attractions, including pro-active tips if the user is near an interesting sight. It uses a map to show the user location and also the position of nearby POI [PLM+].
- **Cyberguide** — *Cyberguide* is a mobile tour guide, that assists the user in a tour to the Georgia Institute of Technology, giving information about the demos in display there [AAH+96].
- **DeepMap** — *DeepMap* is both a mobile guide and a planning tool, aiming for tourists of the city of Heidelberg. It generates personal tours, considering personal interests and needs, cultural background, type of transportation and other circumstances (as traffic, weather or financial resources) [MZ].
- **GeoNotes** — *GeoNotes* is a system that tries to minimize the difference between real and digital world, allowing the users to create information. It is a location-based system that allows tourists to retrieve notes about their current location, that were introduced by other tourists [PEFS02].
- **GUIDE** — *GUIDE* provides the user with information about sights, maps and the ability to create a tour for the Lancaster city [CDMF00].
- **Gulliver's Genie** — Gulliver's Genie is a prototype focusing on delivery content, based on the user location, in a proactive way. Based on the tourist location, a map with the current position and orientation and the nearest POIs are shown. When the user reaches an attraction, a presentation is displayed on the screen [OO04].
- **LOL@** — *Local Location assistant* (LOL@) is an application including tourist information about the city of Vienna. It features predefined tours, information about POIs, a routing functionality and multimodal interaction based on a map [AKM+02] [PKK01].

- **MobiDENK** — *MobiDENK* provides navigation support and information about POIs. The data about existing monuments, showed on a map, can be stored on the device or dynamically loaded from a server. Information about historical buildings can be presented in a multimedia form view [KB04].
- **PinPoint** — *PinPoint* is an existing framework to develop web applications based on external architecture, implementing a prototype of a web-based mobile tourist guide [Rot].
- **Sightseeing4U** — *Sightseeing4U* is a prototype of a tourism guide, providing generic components to create media content [BHH$^+$04].
- **TIP** — The *Tourist Information Provider* (TIP) provides the users with information about their current and nearby sights, matching the user preferences. Based on the user information stored in a database and the user current location, the system creates a profile for the recommendations [HZB05].
- **Tourist Guide** — *Tourist Guide* is a tourist guide developed for the the Adelaide city. Based on the user position, it displays detailed information about nearby POIs and public utilities. Can be operated in Map, Guide or Attraction mode, showing textual and multimedia information about a sight [SHT03].
- **TravelPlot Porto** — *TravelPlot Porto* consists on an interactive way to visit the city of Porto, trying to find a hidden treasure, in order to explore Porto's history and monuments [FA12]. To achieve that, it tells a story about a tourist on a mission to save Port Wine, divided into nine chapters and including 42 different locations of the city.

2.4 Conclusion

In this chapter, the concepts of Procedural Content Generation and Location-based Games were presented, featuring examples of applications and techniques applied. Both Procedural Content Generation and Location-based Games are not completely new areas, and they have a considerable amount of previous work.

Different applications, with different goals were described, and the advantages and possible impacts of both topics were clearly presented. It was understood that there is no technique of PCG that overcomes the others, as they are strongly related to the content they are going to generate.

Although there is work done in Mobile Tourism Guides, most of these guides present the information as a simple map using internal content. There is still work that can be done to improve this perspective.

The main goal of this dissertation is not to find an innovative aspect on one of the areas, but it is instead to conjugate both areas and apply Procedural Content Generation in Location-based Games, something that is not already done, and that can bring new approaches for the development of this type of games. The product that is going to be created should have a focus on the tourism area.

Chapter 3

Procedural Content Generation for Location-based Games

In this chapter, a detailed description of the problem in focus in this dissertation is going to be presented, as well as the hypothesis to be tested for the respective problem. For the stated problem, the Methodological Approach for the development of the project will be presented.

3.1 Problem Context

The evolution of mobile devices, as smartphones and tablets, has made them cheaper and easily available, being a real alternative to regular mobile phones.

With this evolution, it's easier and more affordable than ever to play Location-based Games in an immersive way. As already explained in the last chapter, Location-based Games are a type of games in which the gameplay evolves with the movement of the player in a real setting.

When this type of games have a serious purpose and they are not a simple "treasure hunt" in random locations, they can be strongly related to a narrative developed both in the real and the virtual world. This can have a huge potential for learning purposes, but several other benefits like interaction between players, investigation of objects, active search for information, development of new skills or engagement in different activities.

When the games are based on a narrative, this can be an important tool to add value to the experience. The process of narration combines different components such as actions and events, and gives them a logical meaning and a sequence, creating a coherent whole. This way, LBGs that feature a narrative can include simulators, situated language learning, educational action learning or museum games [Avo12].

With the influence that the narrative can have in a LBG in mind, it's simple to understand that these games are developed for a specific location (city, museum, etc.) putting all the development effort on it. When the game is finished and it needs to be adapted for another location, the game environment as well as the game mechanics need to be adapted and this can be a complex job.

3.2 Proposed Solution

The solution proposed by this project, in order to solve the previously presented problem (games being developed for a specific location), consists on creating a tool to adapt the game environment and mechanics to different urban places. This methodology includes Procedural Content Generation techniques to automate the procedure of creating content for the game.

This should be embodied as a mobile application, that uses geographic information in order to do this, so the changes can be automatically done based on this information.

The tool should be based on the existence of Points of Interest, obtained from the collected information, that are spread around the location for which the tool is developed. The information of these mentioned Points of Interest is embedded in Non-player Characters that are located on the environment, and that interact with the player through a text dialogue.

Focusing on creating a better experience in tourism activities, this approach for the solution should create a game that includes mechanics to immerse the user in the location where the game is played. The agents, embodied as NPCs, can create challenges for the user to complete, in order to obtain a better knowledge of the location and a more interactive experience. These agents can be located in strategic positions, for example near important sights, and create a specific challenge for the given place. A score or level system also add a goal and a motivation to the experience.

After the development of a prototype, it is going to be tested with a group of volunteers, to be possible to get real results from the use of the developed product. This testing phase should be used to validate the usability of two versions of the prototype, as well as to obtain qualitative input on the current state of the tool but also future value that can be added to the project later.

The development of the application is described in detail in the chapter 4, a more extensive explanation on the testing methodology on 5 and the final conclusion obtained with the project in chapter 6.

3.3 Expected Contributions

This project expects to contribute by creating a prototype of the previously presented tool with a possible impact in future games and other applications, used, for example, for tourism purposes.

The main areas that the project can be able to contribute to are Location-based Games and Serious Games applied to tourism, but also techniques to use geographical information to procedurally generate content.

The main expected contributions, concerning the project to be developed and the state-of-the-art projects that have already been studied, are:

- Create a better experience for LBGs, by adapting the mechanics and environments to the location where the game is going to be played;

- Allow users to explore POIs in a more immersive way;
- Use an interactive way to present information, through the use of characters with embedded knowledge;
- Create a prototype tool, where future value can be easily added (for example include gamification properties or richer environments, through the inclusion of real models or textures).

In a research perspective, some questions are also expected to have an answer by the end of the project:

- Is it an improvement to use character embedded information in comparison with the usual ways to present knowledge? Does this improve information retention?
- Is the tool acceptable in terms of usability?
- Which controller is preferred by the users?

3.4 Technological Decisions

There are several possible tools that can be picked for the development and deployment of this prototype, including the mentioned features.

The tools used and the reasons that make them the chosen ones are presented next.

3.4.1 Deployment Platform

Android[1] is a Linux based operating system, used mostly in mobile devices, as smartphones and tablets.

The Android platform was chosen for two main reasons. First, regarding the context of the project, a mobile platform is the target and according to [RG14], Android is the operating system used in most of the devices worldwide, allowing the project to reach a larger audience, as it can be seen on the Figure 3.1.

3.4.2 Game Engine

Unity[2] is a game development ecosystem, with powerful rendering capabilities, rapid workflow and an easy multiplatform publishing. Its ease of learning [Pat14] and supporting communities are also an important factor to have in mind.

Unity's features allow an easy access to the GPS position obtained by the Android device, as well as an easy way to save and parse external files.

Also, this technology is already dominated by the developer, and combined with the other reasons, make it the chosen game engine.

[1] http://www.android.com/
[2] http://unity3d.com

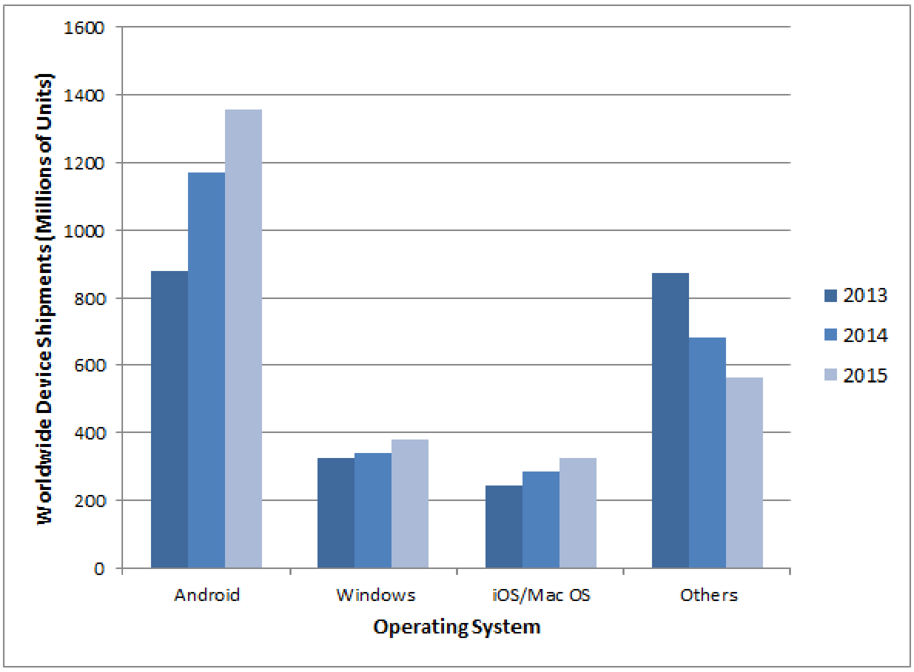


Figure 3.1: Worldwide Device Shipments by Operating System[RG14]

3.4.3 Geographic Information

To obtain the geographic information of the place where the game is going to be played there is the need to use some external system.

There are three main dominant players when choosing a free-to-access web-based map, that includes information about places [CJWM10]:

- **OpenStreetMap** — OpenStreetMap[3] (OMS) is a University College London's project, that consists on a set of map data that is free to use and edit, following the peer production model that was created by Wikipedia [HW08].
- **Google Places** — Google Places[4] is an extension of Google Maps[5] that can be used to find information about a place through a wide range of categories. These places are updated by owner-verified listings and user-moderated contributions [Goo13]. This way, Google Places API is integrated into Google Maps API as a library.
- **Bing Maps** — The Bing Maps REST Service[6] is an API that provides an interface to perform actions with the Bing maps [Bin14]. The Locations API is included in theses Services and is used to get information from a certain location.

[3] http://www.openstreetmap.org
[4] https://developers.google.com/places
[5] http://maps.google.com
[6] http://msdn.microsoft.com/en-us/library/ff701713.aspx

The accuracy of OpenStreetMap in a certain area is strongly dependent on the number of volunteers mapping that area and the location of the area. This represents a significant problem to the improvement of the accuracy and coverage of this system.

A study made by [CJWM10] comparing the coverage and accuracy of Google Maps and Bing Maps in Ireland found out that both are quite similar between them. Both these options have access limitations in terms of API requests, and that's why, despite the accuracy concerns, OpenStreetMap is the best choice.

3.4.4 Data Format

Depending on the Geographic Information System that is chosen, data can be returned in Extensible Markup Language (XML) or JavaScript Object Notation (JSON).

Although both languages are quite similar in terms of features, XML was chosen because it provides a better readability and it is easily parsed using Unity's Mono XML support.

3.5 Conclusion

The correct definition of the problem is fundamental to identify the requirements and necessities of the project. This information was clearly described in this chapter, starting with the description of the problem and then presenting the explanation of the possible solution that this project aims to create as an hypothesis.

The Technological Decisions that were taken were presented, with the aim to pick the best tools, so this project can be successfully developed.

Chapter 4

Implementation

This chapter is going to present the details of the design and implementation of the prototype developed for this project, according to the choices explained on the previous chapter, and the methodology followed during the development. It consists on a mobile application, that covers the aspects detailed in the Chapter 3.

4.1 Introduction

For the development of this project, a prototype with the main features of the solution presented in the previous chapter is developed. This prototype of the solution consists on an Android application, developed with Unity, that includes three main features:

- A procedurally generated environment for the scene, based on the location of the player or on the coordinates that were inserted;
- The inclusion of Non-player Characters that contain information about the nearby sights and Points of Interest;
- The possibility to move on the generated environment using controllers in the First and Third-person perspectives, including location-based mechanics.

Although the location mechanisms are not always used, generating content for a given location is the main idea, and even when using non location-based controllers the content of the scene is still procedurally generated according to the given coordinates.

The Android application connects with OpenStreetMap and Wikipedia through different APIs, according to the type of pretended data:

- **OpenStreetMap**
  - OpenStreetMap API — Nodes and ways that create the map;

- Overpass API — Points of Interest;
- Static Map API — Map texture.

- **Wikipedia**
  - Wikipedia API — Points of Interest around location;
  - DBpedia API — Description of the Point of Interest.

To connect with this APIs, the system does a series of HTTP requests. When the data is obtained, it is stored in local files on the mobile device and used to build the environment.

This procedure is done every time the application is loaded, and no data is recorded from the previous time. This ensures that the location is updated every time the application starts running.

4.2 Geographical Data Sources

Geographic Information is obtained from different OSM APIs, and the complete procedure is explained in detail next. Information regarding specific POIs is obtained from Wikipedia APIs.

4.2.1 Geographical Information

The OSM data can be obtained using its API[1]. The API should be used as seen is 4.1,

```
http://api.openstreetmap.org/api/0.6/map?bbox=LEFT,BOTTOM,RIGHT,TOP
```

Listing 4.1: Query format to obtain OSM data

where `LEFT` is the left most longitude coordinate, `BOTTOM` is the bottom most latitude coordinate, `RIGHT` is the right most longitude coordinate and `TOP` is the top most latitude coordinate of the required bounding box.

This will return an OpenStreetMap's file, containing all the nodes and ways in the limits described by the arguments of the bounding box. The data is presented in XML format. An example of a node and a way (composed by a list of nodes) can be seen in the code in 4.2 and 4.3.

```
<node id="381106509" visible="true" version="3" changeset="17685326" timestamp="
    2013-09-05T13:50:10Z" user="wheelmap_android" uid="524500" lat="51.4961677" lon
    ="-0.1247435">
    <tag k="historic" v="memorial"/>
    <tag k="name" v="The Buxton Memorial"/>
    <tag k="wheelchair" v="yes"/>
</node>
```

Listing 4.2: Example of a node returned by the OSM API

[1] http://api.openstreetmap.org/api

```
<way id="4255600" visible="true" version="8" changeset="19963903" timestamp="
    2014-01-13T00:25:09Z" user="abc26324" uid="125259">
    <nd ref="108094"/>
    <nd ref="25632144"/>
    <nd ref="2620430619"/>
    <tag k="highway" v="tertiary"/>
    <tag k="lit" v="yes"/>
    <tag k="name" v="Great George Street"/>
    <tag k="postal_code" v="SW1"/>
    <tag k="sidewalk" v="both"/>
</way>
```

Listing 4.3: Example of a way returned by the OSM API

When parsing the XML, different types of nodes and ways can be found, containing tags for each type. Each tag has a key (`KEY`) and a value (`VALUE`), as seen in 4.4.

```
<tag k="KEY" v="VALUE"/>
```

Listing 4.4: Example of a tag

When parsing the nodes in the file, there are some special types of nodes, that add content to the model.

- Tree

```
<tag k="natural" v="tree"/>
```

- Post Box

```
<tag k="amenity" v="post_box"/>
```

- Phone Box

```
<tag k="amenity" v="telephone"/>
```

- Bench

```
<tag k="amenity" v="bench"/>
```

- Bus stop

```
<tag k="public_transport" v="stop_position"/>
<!-- or -->
<tag k="highway" v="bus_stop"/>
```

- Waste Basket

```
<tag k="amenity" v="waste_basket"/>
```

- Traffic Lights

```
<tag k="crossing" v="traffic_signals"/>
```

The same way as the nodes, ways also have associated tags that will create different components in the environment:

- Building

```
<tag k="building" v="*"/>
```

- Water

```
<tag k="natural" v="water"/>
<!-- or -->
<tag k="waterway" v="riverbank"/>
```

- Park

```
<tag k="leisure" v="park"/>
<!-- or -->
<tag k="leisure" v="garden"/>
<!-- or -->
<tag k="landuse" v="grass"/>
```

All of these ways give a list of nodes that combined create the correspondent way object. In the case that this way is a street, it is represented by a line that follows the path of the street. The ones with the building tag generate building objects. These objects are represented by a mesh that extrudes the nodes from the ground plane to an higher one, creating a 3D object.

Both the water and grass components are represented by a mesh in the ground plane, differing on the material that is used.

The triangulation of these meshes is done using the Ear Clipping Method [Ebe14], that subdivides a non-convex polygon in triangles. UV coordinates are added to the vertices of the triangles, to make possible the use of textures on the meshes' materials. Normals and collision meshes are calculated by Unity.

4.2.2 Points of Interest

In order to obtain points of interest from the OSM, the Overpass API [2] is used. This is a read-only API that returns data corresponding to customized queries.

As nodes and ways cannot be requested at once, two different queries should be made (4.5 and 4.6):

```
http://overpass-api.de/api/xapi?node[bbox=LEFT,BOTTOM,RIGHT,TOP][tourism=*]
```

Listing 4.5: Query format to obtain nodes with the tourist tag

```
http://overpass-api.de/api/xapi?way[bbox=LEFT,BOTTOM,RIGHT,TOP]
```

Listing 4.6: Query format to obtain ways

where `LEFT` is the left most longitude coordinate, `BOTTOM` is the bottom most latitude coordinate, `RIGHT` is the right most longitude coordinate and `TOP` is the top most latitude coordinate of the required bounding box.

To filter the results to tourism attractions only, the correspondent tag should be added on the end of the query, as seen in 4.5.

The result is returned in a OpenStreetMap's file format, containing nodes or ways, structured as the examples bellow (4.7 and 4.8).

```
<node id="1306187479" lat="51.4983281" lon="-0.1279305">
    <tag k="name" v="Little Dean's Yard"/>
    <tag k="tourism" v="attraction"/>
</node>
```

Listing 4.7: Example of a node returned by the Overpass API

[2] http://overpass-api.de/

```xml
<way id="123557149">
    <nd ref="1377413860"/>
    <nd ref="1377413892"/>
    <nd ref="1377413865"/>
    <nd ref="1377413921"/>
    <nd ref="1377413860"/>
    <tag k="addr:housename" v="Big Ben"/>
    <tag k="building:part" v="yes"/>
    <tag k="building:roof:shape" v="pyramidal"/>
    <tag k="height" v="96"/>
    <tag k="min_height" v="60"/>
    <tag k="name" v="Big Ben"/>
    <tag k="tourism" v="attraction"/>
</way>
```

Listing 4.8: Example of a way object returned by the Overpass API

An object is added to the location of the POI, in case this is a node, or to the location of the center of the POI, in the case of a way object. The name of the object consists on the name obtained from the API (with the key `name`) and the coordinates of the location the following way:

```
NameOfPOI|lat,lon
```

This way, the coordinates of a POI can be easily seen by a simple search.

4.2.3 Map Texture

The map image used to texture the ground plane is obtained through the Staticmap API[3] that returns a Portable Network Graphics (.png) image, centred on a point with `LATITUDE` and `LONGITUDE` coordinates (4.9).

```
http://staticmap.openstreetmap.de/staticmap.php?center=LATITUDE,LONGITUDE&zoom=15&
    size=600x450&maptype=mapnik
```

Listing 4.9: Query format to obtain data from the Wikipedia

4.2.4 Information on the Points of Interest

To obtain information on a specific point of interest, a combination of two API's should be used.

First the Wikipedia API [4] is used to obtain the 5 nearest results around a coordinate, according to the query below (4.10), where `LATITUDE` and `LONGITUDE` correspond to the coordinates of the required POI.

[3] http://staticmap.openstreetmap.de/

[4] http://en.wikipedia.org/w/api.php

```
https://en.wikipedia.org/w/api.php?action=query&list=geosearch&gsradius=10000&
    format=xml&gscoord=LATITUDE|LONGITUDE
```

Listing 4.10: Query format to obtain data from the Wikipedia

This query returns the names of the five nearest POIs within the given distance (`gsradius`), in a XML file, in the format shown in 4.11.

```
<gs pageid="216091" ns="0" title="House of Commons" lat="51.4999" lon="-0.124667"
    dist="4.4" primary="" />
```

Listing 4.11: Example of a POI returned by the Wikipedia API

The distance from the name of the wanted POI and the results returned by Wikipedia, is calculated to see the result that is the closest to the pretended one.

With the title with the smallest distance, a request to DBpedia [5] is made (4.12), returning the 3 best results.

```
http://lookup.dbpedia.org/api/search.asmx/KeywordSearch?MaxHits=3&QueryString=NAME
```

Listing 4.12: Query format to obtain data from the DBpedia

From the given results, the difference to the initial name string should be calculated to check the best match.

To calculate the difference between the name of the POI and the title of the result of the page obtained, the Levenshtein Algorithm [Lev] is used to calculate the string distance.

4.3 Game Design

In this section, the application will be explained in detail in a gaming perspective.

When the application is opened, it is presented a menu containing several fields that allow the input of important information for the course of the game. With this menu, the coordinates of the location are defined, the corresponding requests to the different APIs are done, and the creation of the scenario is done according to them.

The user can move through the scenario according to the chosen controller. While moving, it is possible to interact with the different agents (NPCs) spread on the terrain. To interact with them, the user should click on the model of the agent. A dialogue will then be opened.

[5] http://dbpedia.org

4.4 Mechanics

In this section, the mechanics that are used on the application are presented. This includes the flow of the application, the generation of content to include on it, the possible Character Controllers types and the mechanics related to the Non-player Characters.

The basic concept consists on movement mechanics of the player combined with the different mechanics concerning game modes.

4.4.1 Progression

The application has three main components: the initial menu, the scene itself and the pause menu. These components are explained in detail below.

4.4.1.1 Initial Menu

Running the application will automatically open this menu, where it is possible to input all the necessary data to create the content for the model. The following options are presented on the menu:

- **Input coordinates or use GPS** — Allows to choose whether the coordinates of the center of the model, and the starting point of the controller, are a set of coordinates that are input or the GPS is used to determine the real position of the user. The coordinates are validated so they represent real values.

- **Agents** — Allows to choose the type of behaviour pretended for the NPCs: following or static.

- **Movement Input** — The type of the controller to be used is selected here, and it is possible to choose from the following options:

  - Location Based — This requires that the GPS is used, with a First or Third-person controller;
  - First-person Controller — This controller represents a first-person view and can use a Touchpad or a Joystick;
  - Third-person Controller — With a third-person perspective.

- **Points of Interest** — Allows the user to choose if the POIs to include should be only tourism ones, or if it should have no restrictions.

4.4.1.2 Game

Everything in this scene is generated at runtime, being empty by the time it's loaded. This includes all of the necessary controls to walk around the world.

With the escape key, the user is redirected to the pause menu.

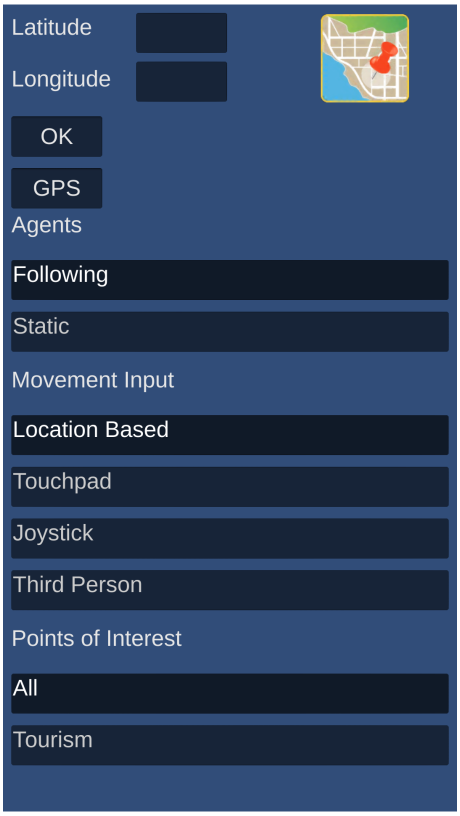


Figure 4.1: Screenshot from the initial menu of the application

4.4.1.3 Pause Menu

This menu allows the user to go back to the main menu, leave the application or go back to the game.

4.4.2 Content Generation

After parsing the files with the data, this should be used to procedurally generate the models at runtime. This allows the environment to be generated for any location, as long as information about that location is represented in the OSM.

The initial scene is empty, containing only the default ground plane, lighting and the different character controllers (that are activated or deactivated according to the choice of the user).

First, the data obtained from the OSM is used to generate the streets, that are represented as lines, and the buildings, as seen in the Figure 4.3. Having the initial list of vertices that are part of the base, the buildings are generated according to the following strategy:

1. Extrude the list of vertices to an higher lever, increasing the value of y by the high of the building;

2. Triangulate the faces of the building;

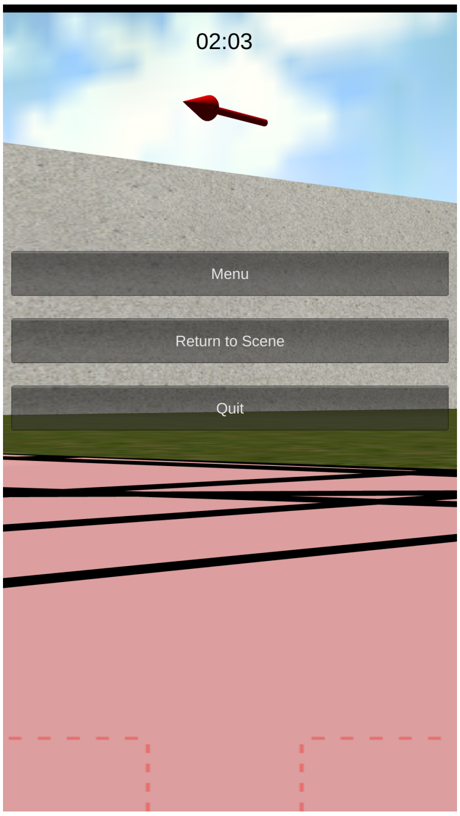


Figure 4.2: Screenshot from the pause menu of the application

3. Check the order in which the vertices are:
   (a) if clock-wise, reverse the list;
4. Calculate the UV coordinates for each vertex;
5. Recalculate the normals;
6. Recalculate the bounds of the object.

This data is also used to generate water and grass components, corresponding to lakes, rivers, parks and grass. An example of a grass mesh can be seen on Figure 4.3.

These meshes are created according to the following steps, having an initial list of the vertices of the mesh:

1. Triangulate the resulting polygon using Ear Clipping Algorithm [Ebe14];
2. Calculate the UV coordinate for each vertex;
3. Recalculate the normals;
4. Recalculate the bounds of the object.

Figure 4.3: Example of a part of the model including buildings, grass, a phone box, a NPC and trees

An example of the triangulation of a grass mesh can be seen on the Figure 4.4

Finally, the NPCs are added, according to the required strategy, to its location. They can be represented as male or female characters, with the same probabilities. The models that are used can be seen on the Figure 4.5.

Information from OSM is also used to add models of the following:

- Tree;
- Post Box;
- Phone Box;
- Bench;
- Bus Stop;
- Waste Basket;
- Traffic Lights.

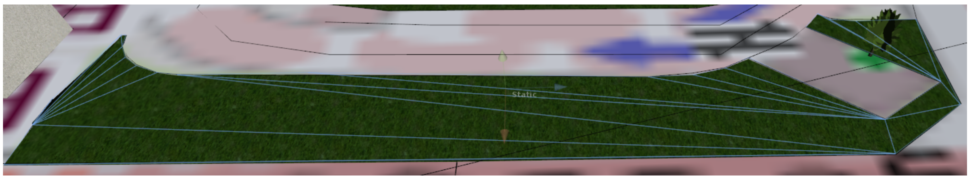


Figure 4.4: Triangulation of a grass mesh

These models were obtained from the Sketchup 3D Warehouse[6] and then simplified as much as possible to reduce the rendering requirements without loosing quality. Examples of a Phone Box (4.6a), Traffic Lights (4.6b) and Trees, Benches and a Waste Basket (4.6c) can be seen on the Figure 4.6.

After the data is obtained and parsed from the Overpass API, the objects corresponding to the POI are instantiated on the right positions.

Then the image of the map is requested and loaded, being scaled to the plane according to the value of the `Latitude` it represents, as seen below:

```
localScale = (BoundingBoxHorizontalDelta*ScaleFactor, 1, BoundingBoxVerticalDelta*
    ScaleFactor*cos(Latitude))
```

4.4.3 Character Controller

The user can interact with his character using two different types of controllers, presented bellow. The movement of the characters is dependent on the controller. The different types of controllers determine the different game modes.

Despite the type of controller, the input can be made using GPS or the typical controller input. This means that the movement can be done according to the user location or the input done by the player in the controllers present on the screen.

The location-based mechanics use the GPS and it keeps updating the position of the player. It has a control to change the rotation of the camera, so the player can look around.

The character moves at a velocity of about 5km/h, as it is the average value for a regular human adult (age 30-35) according to [RG05].

4.4.3.1 First-person Controller

The First-person Controller creates a sense of engagement with the space, as the perspective is closer to the optical perspective. It is logically the most effective way the player can truly act on the elements of the world [Tay02].

It includes a control to move the player and other to rotate the camera. This controls can be represented on the screen as:

[6] https://3dwarehouse.sketchup.com/

Figure 4.5: Male and female model of the NPC

- Touchpads;
- Joysticks.

The left control, either Joystick or Touchpad, controls the movement of the player in the model, and the right one controls the look of the player by the rotation of the camera.

4.4.3.2 Third-person Controller

Third-person point of view represents a physical presence through an embodied character in the world, within the context of the game space [Tay02]. In order to move the player, the user should click on the location he wants the player to move to.

The camera has an algorithm [7] attached in order to deal with collisions with the buildings.

4.4.4 Non-player Characters

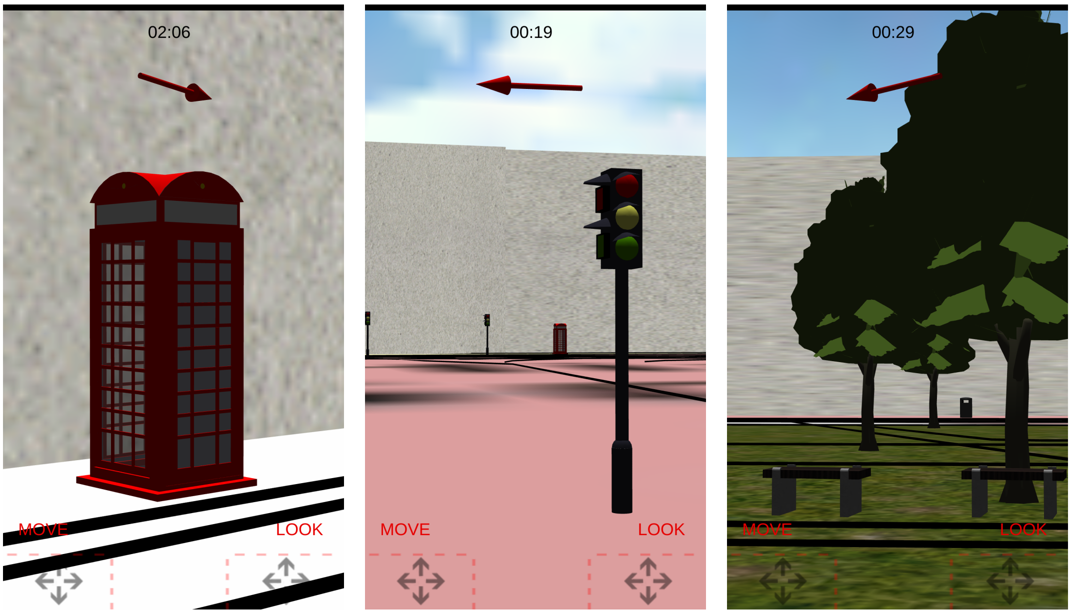
These characters represent the data that is presented by the system, through the knowledge they have embedded.

4.4.4.1 Movement

The NPCs can have two different types of behaviour:

- **Following** — This type of agents use the Unity NavMesh Agent approach[8], following the player at a given distance. With this behaviour, there is only one NPC in the model.
- **Static** — These approach consists on a given number of agents, spread around the model in random positions. These agents are static, not moving at any time.

[7] http://wiki.unity3d.com/index.php?title=SmoothFollowWithCameraBumper
[8] http://docs.unity3d.com/Documentation/Components/class-NavMeshAgent.html



(a) Phone Box (b) Traffic Lights (c) Trees and Benches

Figure 4.6: Pictures of different models

4.4.4.2 Dialogue

When a player clicks on a NPC, a dialogue window is opened, showing an informal speech. This window includes 4 different options, as seen on the Figure 4.7:

- **Which are the nearest Points of Interest?** — This option returns the 5 nearest (in a straight line) Points of Interest from the position of the camera.
- **Which are the attractions in less than 1km?** — Every POI in less than 1km in a straight line is going to be presented.
- **I would like to know more about:** — With this option is possible to choose from one of the 5 nearest POIs and obtain a small description of it.
- **Bye!** — Closes the dialogue window and returns to the main scene.

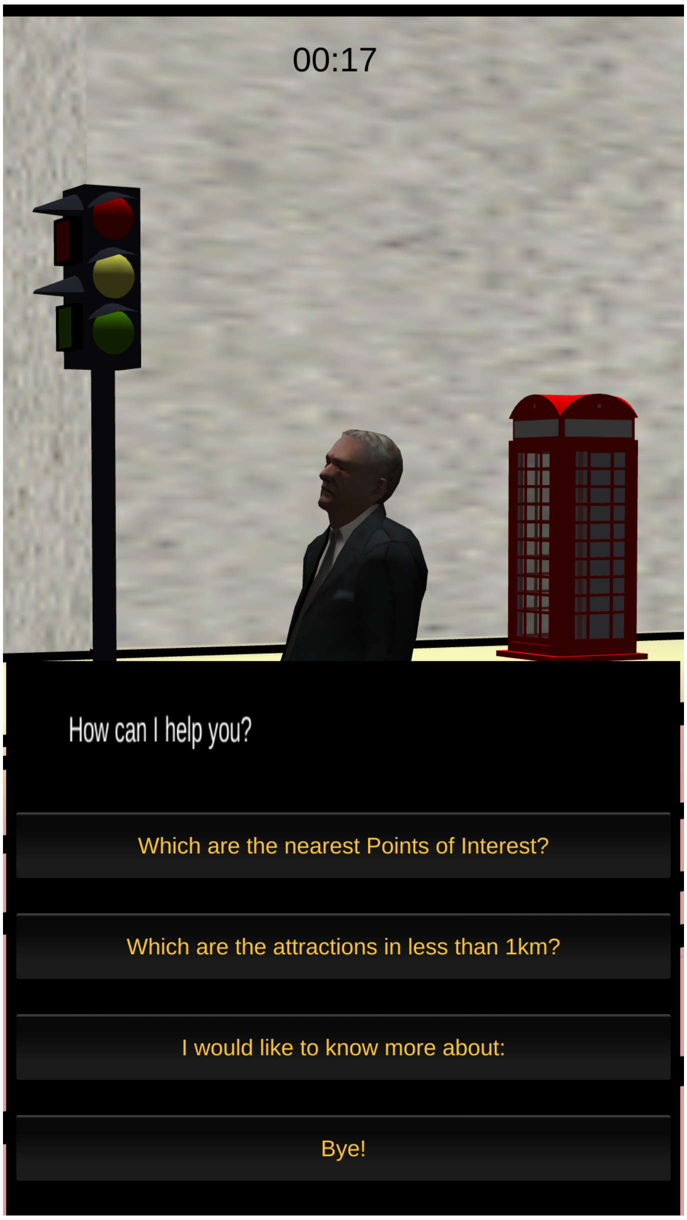


Figure 4.7: Screenshot of the initial dialogue

4.5 Conclusion

A correct definition of the problem, as well as all of the details explained on the methodological approach, in the previous chapter, was essential to identify the requirements of the application.

The technological decisions revealed to be the right ones, allowing the correct development of the tool.

More time would allow the development of a more detailed environment, and the inclusion of mechanisms to adapt game mechanics or dialogues to different locations. Also, better results would probably be achieved by the creation of a strategy that breaks the dependence with the different APIs and with data not correctly structured or defined (for example, the lack of tag attributes on OSM).

The solution was successfully implemented, filling the goals and requirements of the project.

Chapter 5

Testing and Results Analysis

There are some questions that need to be answered to prove the intent of this project. It aims to study whether character-embedded knowledge can be used to increase the retention of information and with this experiment, it will be tested the usability of the tool, comparing two different versions of it.

5.1 Usability test

The aim of this test is to verify whether the application can be accepted in terms of usability or not. Usability can be considered as the easy of use of an object (in this case a software system), and the capacity to learn how to use it.

These testing sessions are going to compare the First and Third Person controllers, in order to check which is the best option to include in the tool, and if both are accepted in terms of usability.

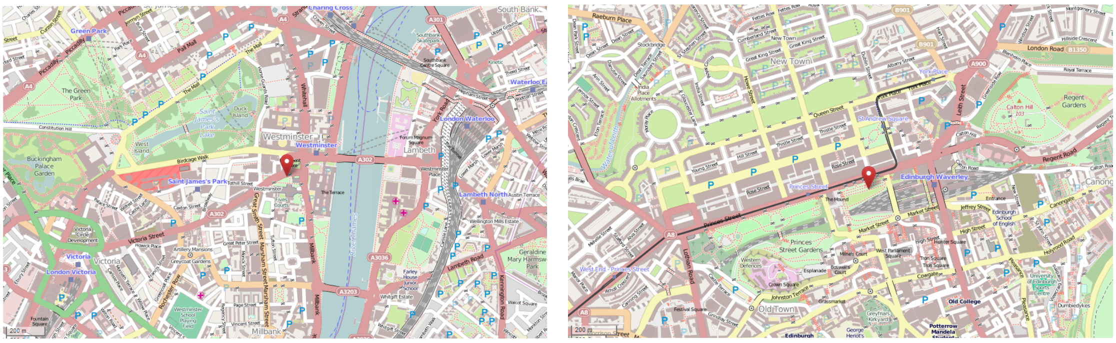
5.1.1 Methodology

The usability of the tool is going to be tested with a group of users using two different versions of the prototype for a short period of time, and filling a survey with questions regarding the usability, the use of the tool and the user background.

The two versions of the tool differ in the sense that one uses a Third-person Controller and the other a First-person equivalent. The order in which both versions are used is random, using a random number generator to determine the first version to be tested. The GPS is not used, as the version used does not feature the movement of the player.

According to [Vis13], the most visited cities in the United Kingdom during the last years are London and Edinburgh. A central location of these two cities is used in the tests, with coordinates `(51.5°, -0.1275°)` (Figure 5.1a) and `(55.952°, -3.1935°)` (Figure 5.1b) respectively.

The application runs for 3 minutes, containing 20 agents randomly positioned on it, returning to the initial menu after that period of time.



(a) London (b) Edinburgh

Figure 5.1: Test locations

The usability will be studied using an existing usability scale. Some considered scales are:

- **System Usability Scale (SUS)** — A 10 item questionnaire, with 5 response options going from "Strongly Disagree" to "Strongly Agree". The questions focus on topics as if the user would like to use the system frequently and how easy it was, giving a percentage value [Sau11].

- **Software Usability Measurement Inventory (SUMI)** — Consists on 50 statements to which the user has to reply "Agree", "Don't Know" or "Disagree". Answers are then transformed into a Global subscale, and five additional subscales called "Efficiency", "Affect", "Helpfulness", "Controllability" and "Learnability" [Kir12].

- **Questionnaire for User Interaction Satisfaction (QUIS)** — Related with the technology (system capabilities, screen factors and learning factors) with focus on selecting appropriate audiences [NS].

Because it is an easy to use and already proved scale according to [Bro], the System Usability Scale is going to be used. This system was also picked because it returns a percentage value that can be easily studied.

The SUS should be used after the respondent has had an opportunity to use the system, but before any discussion. Participants should be asked to record their immediate response, rather than thinking about items for a long time.

If users feel that they cannot respond to a particular item, the centre point (3) of the scale should be marked.

The testing session takes approximately 13 minutes, according to the following sequence of events:

1. Introduction to the tool and the experiment (1 min);

2. General questions regarding the participant background (1 min);

3. Testing the first version (3 min);

4. Questions regarding the first experiment (1.5 min);
5. Testing the second version (3 min);
6. Questions regarding the second experiment (1.5 min);
7. Final general questions (2 min).

According to the order of events previously shown, a survey is used to obtain some information. This information should feature the background of the user (in order to find a correlation about it and the answers given), the use of the application and, finally, the general opinion of the user.

The initial and more general questions regarding the participant background, include topics to verify whether:

- The user is a regular game player;
- He/she is a touch screen devices user;
- The participant travels frequently;
- The participant uses maps applications while travelling.

After both experiments, the participant is going to be asked to answer, besides the questions from the SUS, some questions regarding the experiment that he/she just finished. These questions include:

- If the user is able to recognize the location of the test and, in the case he/she was, where it is and if he has already been there;
- Usability questions based on the System Usability Scale;
- How can the system be improved.

By the end of the test, some final questions are then made to check:

- Which of the versions did the participant enjoyed the most;
- If he/she thinks that the order of the experiment had any influence on his/her preference;
- What future value could the tool have;
- Gamification possibilities;
- If there is some value that can be added to the system in the future.

5.1.2 Resourcing

The testing sessions use the same device, a Samsung Galaxy S4, with the following specifications:

- **Model** — GT-I9505
- **Operating System** — Android 4.4.2
- **Screen Resolution** — 1920 x 1080

The survey is filled online, during the duration of the experiment.
The tests consisted on two phases:

- **Pre-testing** — 6 users try the application and give feedback on how to improve it;
- **Testing** — 10 users test the improved version of the tool, concerning the feedback given on the previous phase.

As there is not a specific target group for the tool, users do not need to fill any pre-requisites.

5.2 Results

5.2.1 Pre-testing

This testing phase aims to get some feedback from the users. This feedback includes both topics related to the application itself and how to improve it for the posterior phases, but also regarding the testing session (structure, duration and relevance of the survey questions).

5.2.1.1 Participants

In this phase, data was recorded from 6 participants. From those participants, 5 (83%) are regular gamers. All of the participants use at least one type of touch screen devices and travel at least once a year, using maps applications when doing that. This information can be seen bellow on the Table 5.1.

5.2.1.2 Usability

The usability value according to the SUS, in a percentage scale, is calculated according to:

- For odd-numbered items: subtract one from the user response;
- For even-numbered items: subtract the user responses from 5;
- Add up the converted responses for each user and multiply that total by 2.5.

Do you play games regularly?		
Yes	5	83.3%
No	1	16.7%
What kind of touch screen devices do you use?		
Smartphone	6	100.0%
Tablet	3	50.0%
I don't use any	0	0%
Other	0	0%
How frequently do you travel?		
More than once a month	3	50.0%
Once a month	0	0%
A few times a year	2	33.3%
Once a year	1	16.7%
Less than once a year	0	0%
I don't travel	0	0%
Do you use any maps application when travelling?		
Yes	6	100.0%
No	0	0%

Table 5.1: Users demographic information from the Pre-testing phase

To achieve that, the following formula from the SUS is used:

$$\left(\sum_{i=1}^{5}\left((x_{2i-1}-1)+(5-x_{2i})\right)\right)\times 2.5$$

where x_N corresponds to the value of the question N of the SUS.

The values on the Table 5.2 represent the results of the usability values for each user, and can also be seen on the table of the Figure 5.2.

The distribution of the results from this phase can be seen on the graph of the Figure 5.3.

5.2.1.3 Participant Comments

When asked about ideas to add value to this system, most of the comments were positive, with concrete ideas to improve the system.

First-person Controller

Regarding the First-person Controller, the ideas were based on how to locate the NPCs easily and how to make the GUI more intuitive.

> "Something to point you towards the nearest point of interest."

> "The GUI needs improvement like the control buttons are not intuitive. The NPCs can not be located easily. Some sort of a map would be necessary."

	Order	Controller	Usability
1	1	First-person	85.0%
	2	Third-person	70.0%
2	1	Third-person	75.0%
	2	First-person	75.0%
3	1	First-person	60.0%
	2	Third-person	45.0%
4	1	First-person	87.5%
	2	Third-person	87.5%
5	1	First-person	72.5%
	2	Third-person	12.5%
6	1	Third-person	60.0%
	2	First-person	65.0%

Table 5.2: Usability results from the Pre-testing phase

Third-person Controller

The proposed improvements on the Third-person Controller version mentioned decreasing the value of the high of the camera. It was also mentioned the inclusion of instructions on the goal, by the improvement of the GUI controls.

An easier way to locate NPCs was also mentioned.

> ”A more recognisable environment would help with navigation, a lower camera angle might help too.”

> ”An option for camera movement independent to character movement.”

> ”Control buttons are difficult to use. Some sort of instructions of what is the goal of the game would be good. Cant locate the NPCs easily.”

> ”Controls felt very inconsistent, a GUI element for controls would help. Camera view makes it difficult to see what is around you. Could not find out where I was, only found 1 landmark. Buildings need colliders to stop player from walking through.”

General

The general idea to add value to the tool was to improve the content on it. Both the quantity and quality of the models included in the environment were mentioned, in order to create a less abstract environment.

> ”More content to populate the areas would help with immersion”

> ”As previously mentioned , the game is a bit too abstract to seem really helpful to me. So any work on that would seem worthwhile to me.”

> ”Higher resolution environments, models and map could help provide a more detailed and accurate observation of your surroundings. Could be used for road travel planning, city walks etc.”

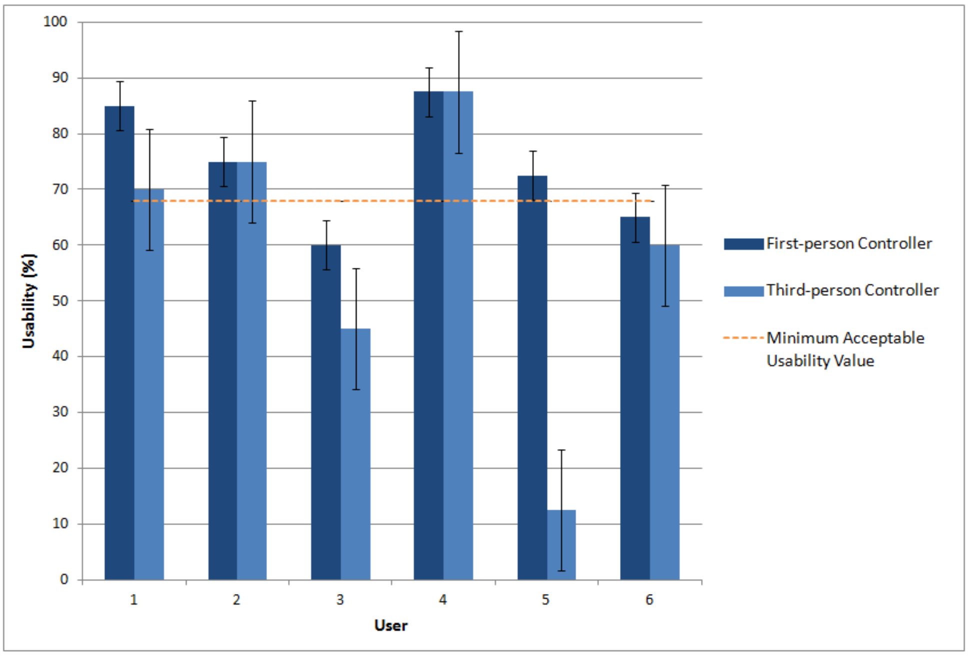


Figure 5.2: Usability results from the Pre-testing phase

5.2.1.4 Conclusion

Although the value of usability of the First-person Controller is above average (68%, according to [Sau11]), the value for the Third-person Controller is bellow, as it can be seen on the Table 5.3.

First-person Controller	Third-person Controller
74.2%	58.3%

Table 5.3: Mean Usability from the Pre-testing phase

Only one user was able to recognize one of the locations (Table 5.4). This is probably due to the fact that the users were trying to look for the map texture on the ground, and not trying to use the knowledge embedded on the NPCs.

	Yes		No	
First-person Controller	1	16.7%	5	83.3%
Third-person Controller	0	0.0%	6	100.0%

Table 5.4: Location recognition from the Pre-testing phase

All of the users preferred the version that was given the highest usability, being the First-person Controller chosen by everyone that claims to be a regular gamer, and the Third-person by the other.

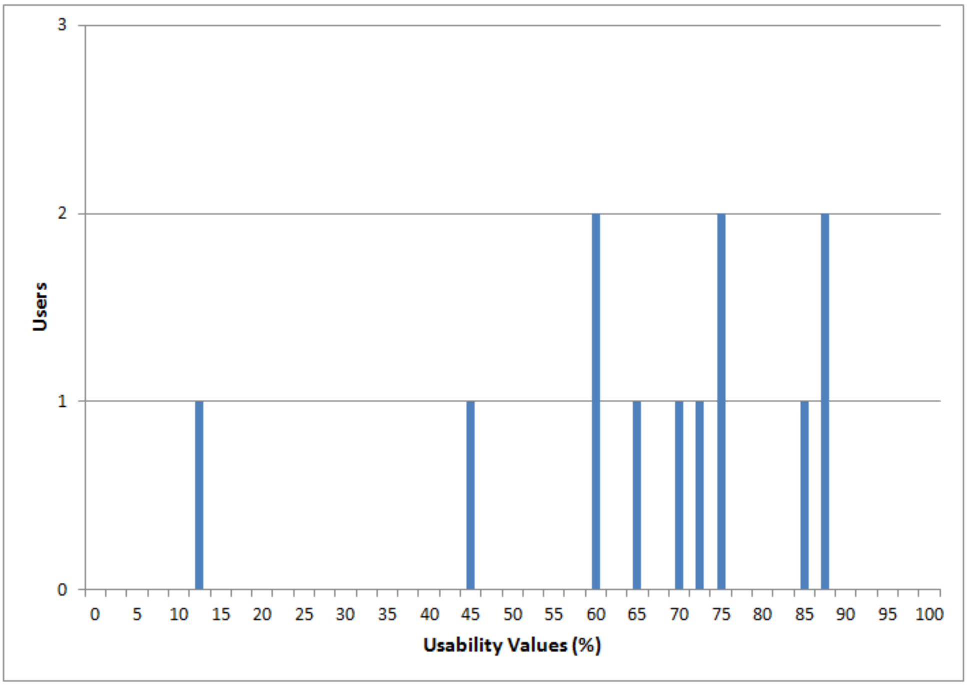


Figure 5.3: Distribution of the usability results from the Pre-testing phase

The favourite version of the majority of the users is the one with First-person perspective, as seen on Table 5.5.

	First-person Controller		Third-person Controller	
Regular Gamer	5	100.0%	0	0.0%
Non-regular Gamer	0	0%	1	100.0%

Table 5.5: Favourite version from the Pre-testing phase

After the end of this phase, some improvements need to be done, including:

- An arrow always pointing to the nearest NPC;
- GUI instructions regarding controls presented after the game is loaded, instead of the inclusion of the instructions on the initial menu;
- Instructions giving the hint to click on the NPCs;
- Changes on the position of the camera on the Third-person version.

5.2.2 Testing

This phase includes the improvements mentioned on the previous section. There were no changes regarding the testing methodology, as the participants mentioned the decisions were appropriate for the aim of the test.

5.2.2.1 Participants

In the Testing phase, data was recorded from 10 participants. From the 10, 8 (80%) are regular gamers. All of the participants use at least one type of touch screen devices, mainly smartphones and tablets. Regarding the frequency of travelling, most of them (90%) travel at least once a year, using maps applications when doing that. The detailed results from this section of the survey can be seen bellow (Table 5.6).

Do you play games regularly?		
Yes	8	80.0%
No	2	20.0%
What kind of touch screen devices do you use?		
Smartphone	9	90.0%
Tablet	6	60.0%
I don't use any	0	0.0%
Other	1	10.0%
How frequently do you travel?		
More than once a month	1	10.0%
Once a month	3	30.0%
A few times a year	5	50.0%
Once a year	0	0.0%
Less than once a year	1	10.0%
I don't travel	0	0.0%
Do you use any maps application when travelling?		
Yes	8	80.0%
No	2	20.0%

Table 5.6: Users background from the Testing phase

5.2.2.2 Usability

As already mentioned on the last section, the usability is calculated according to the following formula:

$$\left(\sum_{i=1}^{5}\Big((x_{2i-1}-1)+(5-x_{2i})\Big)\right)\times 2.5$$

where x_N corresponds to the value of the question N of the SUS scale.

The results of the usability values for each user can be seen on the Table 5.7. The results are also represented on the Figure 5.4.

The distribution of the usability results from the Testing phase can be seen on the Figure 5.5.

	Order	Controller	Usability
7	1	Third-person	57.5%
	2	First-person	37.5%
8	1	Third-person	65.0%
	2	First-person	70.0%
9	1	First-person	67.5%
	2	Third-person	45.0%
10	1	First-person	82.5%
	2	Third-person	85.0%
11	1	Third-person	85.0%
	2	First-person	77.5%
12	1	Third-person	80.0%
	2	First-person	90.0%
13	1	First-person	67.5%
	2	Third-person	75.0%
14	1	First-person	85.0%
	2	Third-person	85.0%
15	1	First-person	75.0%
	2	Third-person	75.0%
15	1	First-person	70.0%
	2	Third-person	72.5%

Table 5.7: Results of each user from the Testing phase

5.2.2.3 Participant Comments

In the question about how to improve the tool, most ideas are quite valid and can be easily implemented in the future.

First-person Controller

In more than one comment, users mentioned to be confused by the red pointer. Although the goal of the pointer was just to give the directions to the nearest NPC, disappearing when the user was close to it, this seems to have confused users.

> ”The system is quite user-friendly. However, the red pointer is quite difficult to track e.g. sometime it immediately changes the direction and sometime the pointer disappears.”

Although a reset option can make sense in the prototype used for the tests, in a version with location mechanisms this is not relevant, as the user cannot reset his location. A top-view map option was also recommended.

> ”Have a reset option. Have an overview top level map view. Improve the view aspect I could not see where I was going. Better quality imagery would be helpful.”

Third-person Controller

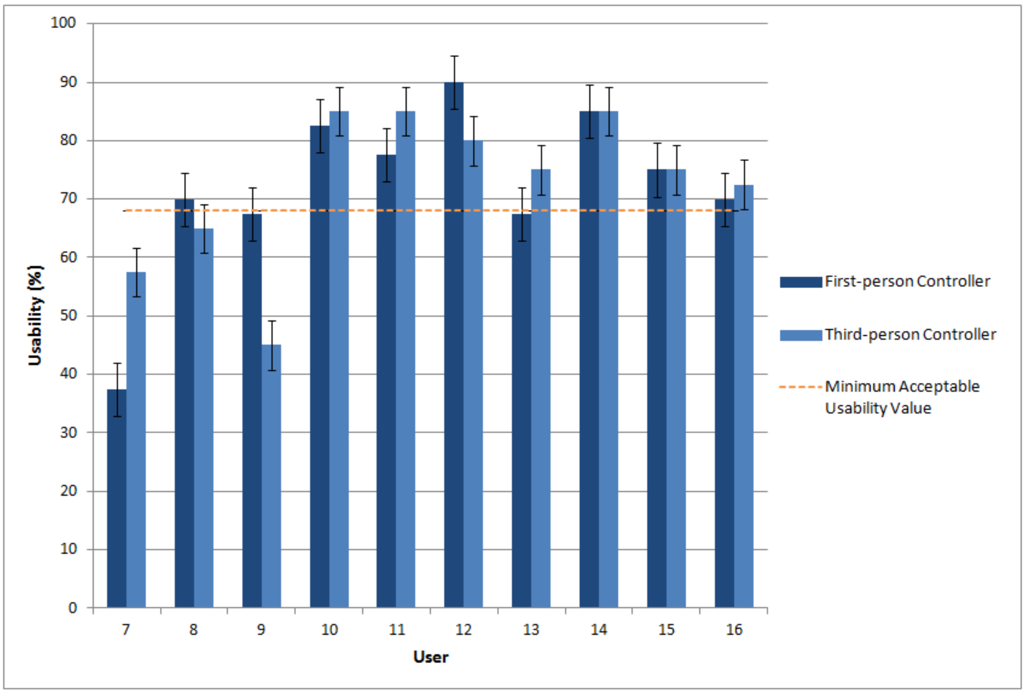


Figure 5.4: Usability results from the Testing phase

Regarding the Third-person perspective, the main improvements that were mentioned included make the controls more responsive when the touch was far from the position. The option to look around before walking was mentioned again.

> ”User controls seemed a bit unresponsive. Sometimes the character didn’t respond when I touched far away.”

> ”Similar to first person mode, the third person mode contains difficulties associated with pointers disappear and reappear at some point of the session. Further, the the character struck and irresponsive to finger inputs at some point. However, I find third person mode to be more comfortable than first person mode.”

> ”(...)Controls were slightly harder to use, I preferred being able to look around before I walked anywhere.”

General

When asked which are the possible improvements for the tool, independent of the version, the use of a better image and a top map option were mentioned.

> ”I think the maps need to be a bit clerer, I couldn’t easily tell what the lines were on the floor.”

The inclusion of Wikipedia links as a way to learn more about the different Points of Interest was also mentioned.

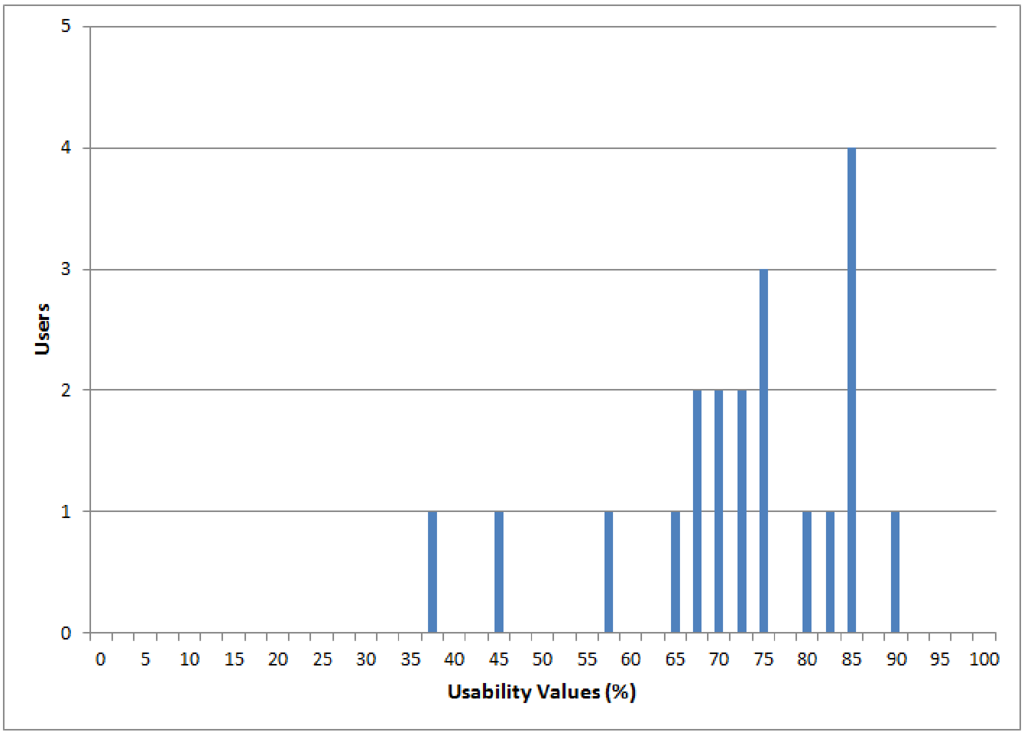


Figure 5.5: Distribution of the usability results from the Testing phase

”Better Imagery and maybe a map option. Would be nice if the information enabled links to wikipedia where you can then explore for more information.”

5.2.2.4 Conclusion

According to the previously done research by [Sau11], the minimum usability value for the results so an application can be accepted is 68%. As it can be seen on the Table 5.8, the average of the results from this phase of tests are acceptable for both versions.

First-person Controller	Third-person Controller
72.25%	72.5%

Table 5.8: Average Usability from the Testing phase

More people were able to recognize the location on this phase, as seen on the Table 5.9. This was probably due to the pointer combined to the initial GUI instructions, that made people find NPCs in an easier way.

	Yes		No	
First-person Controller	3	30.0%	7	70.0%
Third-person Controller	4	40.0%	6	60.0%

Table 5.9: Location recognition from the Testing phase

The favourite version is still the First-person, specially for Regular Gamers (Table 5.10). The favourite version for Non-regular Gamers is not conclusive (50%-50%).

	First-person Controller		Third-person Controller	
Regular Gamer	5	62.5%	3	37.5%
Non-regular Gamer	1	50.0%	1	50.0%

Table 5.10: Favourite version from the Testing phase

5.3 Results Analysis

5.3.1 Usability Questions

The difference between the average values of First and Third-person are not relevant, as t-testing showed no significant difference in the mean values (p=0.48, n=10). This is probably due to the fact that there is not clearly a favourite version, although the sample is too small for the results to be conclusive.

The distribution of the results, as seen on 5.6, and the small difference from the minimum acceptable value can be inconclusive too.

Based on the results obtained, participants that claim not to play games regularly tend to prefer the simplicity of the Third-person controller, compared to the controllers on the First-person equivalent. This version, although, tends to be the preferred for regular gamers, because it is more flexible (giving more walking freedom) and allows to look around before walking somewhere.

These results, however, could be different with a larger number of participants as the number of users saying they don't play games regularly was smaller than the ones that say they do play.

The different way to present instructions combined with the pointer should have influenced the fact that more testers were able to recognize the location of the scenario. These two things combined create a goal for the user, instead of the random walk around that was happening on the Pre-testing phase.

The average usability of the Third-person controller increased with the improvements made between the two phases. This made the two versions approved according to the System Usability Scale, although there was a small difference on the First-person values.

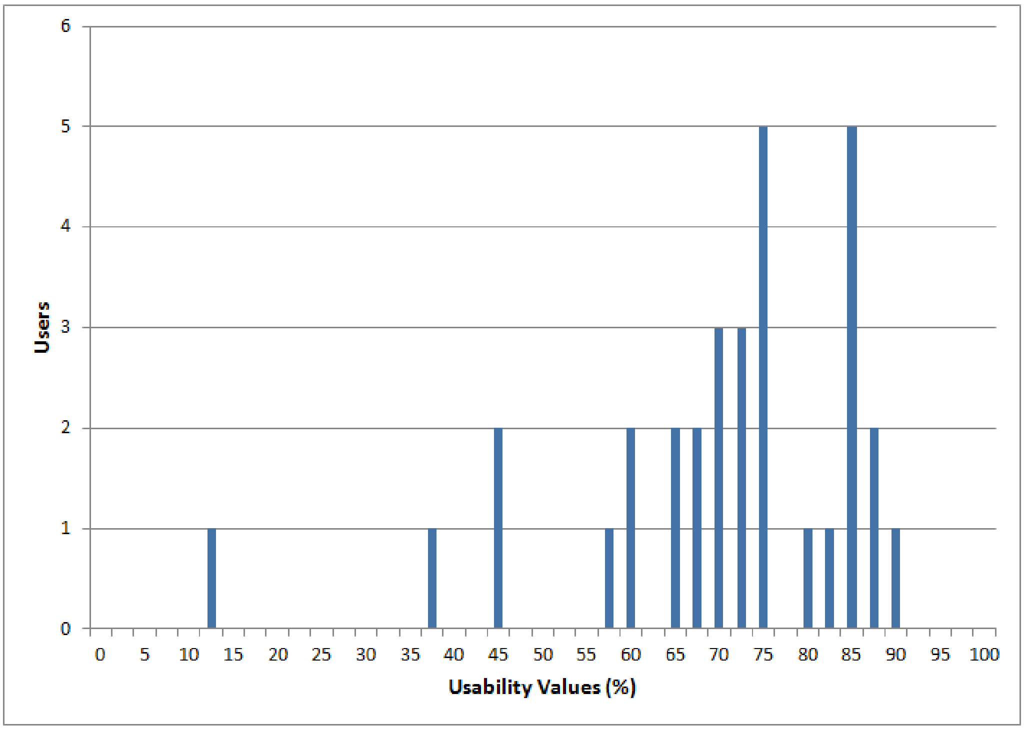


Figure 5.6: Distribution of the usability results from the Pre-testing and Testing phase

5.3.2 Qualitative questions

When asked if gamification properties could improve the general experiment of the application, merging the results from both phases, 2 users say that it wouldn't improve, 7 say it would and the other 7 don't know (Table 5.11).

	Yes		No		I don't know	
Do you think that gamification properties would improve the general experience of the application?	7	43.75%	2	12.5%	7	43.75%

Table 5.11: Gamification question results from the both phases

Although there was a clear relation, for each user, between the favourite version and the version with the highest usability value in the Pre-testing phase, on the second phase this relation was not that clear. From the 10 users, only 6 picked the version with the highest usability.

No one says that the order in which both versions were played could have had influence on the experiment, but 40% of the testers assume they don't know if that is a possibility, as can be seen on the Table 5.12.

5.4 Conclusion

After both testing phases described on this chapter, the version is completely tested.

	Yes		No		I don't know	
Do you think the order in which you've tried both versions had any influence on your answer to the last question?	0	0.0%	6	60.0%	4	40.0%

Table 5.12: Influence of the order results from the Testing phase

Some of the improvements worked well, as the users seem to know what they are supposed to do after reading the instructions (that were on the main menu in the initial version and then moved to a GUI message after the scene is loaded) and following the pointer. Despite that, participants were still looking to the map in order to find out the location instead of trying to find a NPC.

The number of participants of the two different phases can have influence on the final results. With a higher number of participants the results could have been slightly different, and the analysis of the results could be more complete, as different methodologies could be applied.

This experiment only tested the usability of a prototype of the tool, and not the final version of it, that includes location awareness. So, these tests only proved that the application is acceptable in terms of usability.

Some improvements can still be made on the tested version of the tool, and some possible ones can include:

- **Top-view map** — This can help players to have a different perspective from the location;
- **Controllers** — Both controllers need to be improved to create a more fluid movement;
- **Better map image quality** — Although this is not a priority, it can improve the amount of information provided, and help the player not to feel lost. This needs some research, as the limitations of the API do not provide better quality images from the ones used;
- **External links for more information** — Besides the small description that is provided for the different POIs, external links can be added to include more information;
- **More intuitive pointer** — The pointer can be more intuitive, as some of the decisions made during its implementation confused the users (for example, hide the pointer when the distance to the NPC was not relevant);
- **Goal** — Using gamification features would probably add a goal to the application and it would be easier for the user to know the decisions to take in a certain moment;
- **Models** — Adding more models would create a less abstract environment;
- **Textures** — Another way to make the model less abstract would be by adding different textures to the buildings, streets or any other components.

Chapter 6

Conclusion

This final chapter finalizes the work done, summarizing the dissertation.

It presents the contributions and conclusions after the project is finished, but also possible improvements and future work and lessons that were learned through its development, during all the phases of the project.

6.1 Final Remarks

Tourism and mobile devices are more popular than ever, due to the development of the information technologies. One of the ways of improve the tourist experience is due to specific games and mobile applications, typically developed to a specific location.

The contribution of this dissertation was to create an application that is able to create content procedurally, and that has been proved to be usable for the two studied locations.

The methodology created allows to generate a scenario based on the location of the player. Besides the direct application of the tool, this methodology can be used in future location-based applications, creating more immersive experiences for tourists.

6.2 Future Work

The work of this dissertation consisted on developing the application for a fixed period of time. Because of that, some ideas and possible improvements that showed up during its development couldn't be implemented. There were also some ideas that were given by the participants of the testing phases that would add value to the tool. Some examples of that could feature improvements in terms of:

- **Mechanics** — Some improvements can be added to improve already existing mechanics or to add new ones. Some possible examples are:

  - A top-view map;
  - Improved controllers;
  - External links for more information;
  - Possibility to chose the categories of the showed POI;
  - A more intuitive pointer to the nearest NPC.

- **Scenario** — Improvements on the scenario would also be important to create a more immersive environment. This can include:
  - Improved models;
  - Different textures;
  - Better map image quality.

- **Offline mode** — The tool is completely dependent on an Internet connection, in order to receive data. Although most users have a 3G connection and the possibility to use the application anywhere, this does not apply to everyone and, also, data is usually limited or including extra charges. One way to avoid this dependence would be by creating a way to store information prior to the use of the application.

- **Augmented Reality** — Augmented Reality improves the interaction between the user and the real world, providing a much more immersive experience. This could be one interesting improvement, in order to visualize specific attractions for example.

- **Sequence in the dialogue** — Adding a sequence to the dialogue would create a more fluid application, as it would allow to have a consistent plot and narrative.

- **Different languages** — As it aims to be a tourist application, and besides being English an universal language, a multilingual platform would reach a larger number of users. This could be achieved by changing the language of the interfaces and menus, and also from the results of the requested data, by changing the API requests when possible.

There are also improvements that could have been made on the methodology used for the process:

- **Testing** — A testing session with more participants could create more reliable and relevant results. This would allow the results not to be so dependent of a single user, but also the possibility to use different techniques and methodologies to analyse the data (for example ANOVA).

- **Location-based** — A bigger focus on the location-based version, and include these techniques on the testing methodology, would allow to fully prove the questions of this dissertation.

6.3 Lessons Learned

The development and testing of the application showed that a correct planning, and the ability to follow it, is very important, leading to better choices and a better final product.

One important detail that was missing and that was noticed only during the development of the prototype were some limitations of the APIs. This lead to a switch from Google Maps to OpenStreetMap and a poor image quality for the ground map.

The overall plan was successfully implemented, allowing the creation of a completely functional prototype of the application on time for testing it properly and having enough time to explain everything clearly on this paper.

References

[AAH+96] Gregory D Abowd, Christopher G Atkeson, Jason Hong, Sue Long, Rob Kooper, and Mike Pinkerton. Cyberguide: A Mobile Context-Aware Tour Guide. (April 1995):1–21, 1996.

[AB11] Gülfem Işıklar Alptekin and Gülçin Büyüközkan. An integrated case-based reasoning and MCDM system for Web based tourism destination planning. *Expert Systems with Applications*, 38(3):2125–2132, March 2011.

[Ada02] Author David Adams. Automatic Generation of Dungeons for Computer Games. 2002.

[AKM+02] Hermann Anegg, Harald Kunczier, Elke Michlmayr, Günther Pospischil, and Martina Umlauft. LoL @: Designing a location based UMTS application. 2002.

[AN07] Calvin Ashmore and Michael Nitsche. The Quest in a Generated World. In *Situated Play, Proceedings of DiGRA 2007 Conference*, pages 503–509, 2007.

[Ash10] Daniel Ashlock. Automatic generation of game elements via evolution. In *Proceedings of the 2010 IEEE Conference on Computational Intelligence and Games*, pages 289–296. IEEE, August 2010.

[Avo12] Nikolaos Avouris. A review of mobile location-based games for learning across physical and virtual spaces. 18, 2012.

[BBB+12] Elaine Barker, William Barker, William Burr, William Polk, and Miles Smid. Computer Security. (July):1–147, 2012.

[BBC+09] Amos Bloomberg, Kate Boelhauf, Dennis Crowley, Christopher Hall, Will Lee, Morekwe Molefe, Mike Olson, Megan Phalines, Mattia Romeo, Oli Stephensen, Pakorn Thienthong, and Peter Vigeant. Pac Manhattan. http://pacmanhattan.com/index.php, 2009.

[BHH+04] Cherif Branki, J Felix Hampe, Reiner Helfrich, Karl Kurbel, Gerhard Schwabe, Frank Teuteberg, Stefan Uellner, Rainer Unland, and Gerhard Wanner Editors. mobile MM4U – framework support for dynamic personalized multimedia content on mobile systems. 3:204–215, 2004.

[Bin14] Bing. Bing Maps REST Services. http://msdn.microsoft.com/en-us/library/ff701713.aspx, 2014.

[BL09] Matt Barton and Bill Loguidice. Gamasutra - The History of Elite: Space, the Endless Frontier. http://www.gamasutra.com/view/feature/3983/the_history_of_elite_space_the_.php, 2009.

[BMMU11] Lynne Baillie, Lee Morton, David C. Moffat, and Stephen Uzor. Capturing the response of players to a location-based game. *Personal and Ubiquitous Computing*, 15(1):13–24, June 2011.

[BMS+12] Montserrat Batet, Antonio Moreno, David Sánchez, David Isern, and Aïda Valls. Turist@: Agent-based personalised recommendation of tourist activities. *Expert Systems with Applications*, 39(8):7319–7329, June 2012.

[Bra12] David Braben. Kickstarter » Elite: Dangerous by Frontier Developments. https://www.kickstarter.com/projects/1461411552/elite-dangerous/, 2012.

[Bro] John Brooke. SUS - A quick and dirty usability scale.

[CAR+10] Simon Cooper, Abdennour, El Rhalibi, Madjid Merabti, and Jon Wetherall. Procedural Content Generation and Level Design for Computer Games. In Daniela M Romano and David C Moffat, editors, *Artificial Intelligence and Simulation of Behaviour AISB convention 2010 AISB 2010 AIGames Symposium*, number April in Artificial Intelligence and Simulation of Behaviour (AISB) convention 2010 (AISB 2010), AI&Games Symposium, pages 35–40, 2010.

[CBPD11] Daniel Michelon De Carli, Fernando Bevilacqua, Cesar Tadeu Pozzer, and Marcos Cordeiro DOrnellas. A Survey of Procedural Content Generation Techniques Suitable to Game Development. *2011 Brazilian Symposium on Games and Digital Entertainment*, pages 26–35, November 2011.

[CBSF06] Antonio Coelho, Maximino Bessa, A Augusto Sousa, and F Nunes Ferreira. Expeditious Modelling of Virtual Urban Environments with Geospatial L-systems. 0(0):1–13, 2006.

[CDMF00] Keith Cheverst, Nigel Davies, Keith Mitchell, and Adrian Friday. Experiences of Developing and Deploying a Context- Aware Tourist Guide : The GUIDE Project. pages 20–31, 2000.

[CJWM10] Blazej Cipeluch, Ricky Jacob, Adam Winstanley, and Peter Mooney. Comparison of the accuracy of OpenStreetMap for Ireland with Google Maps and Bing Maps. 2010.

[CM06] Kate Compton and Michael Mateas. Procedural Level Design for Platform Games. *Proceedings of the Artificial Intelligence and Interactive Digital Entertainment International Conference AIIDE*, pages 109–111, 2006.

[Coo07] Daniel Cook. Lost Garden: "Content is Bad", 2007.

[Dav03] David Ebert. *Texturing & Modeling: A Procedural Approach*. 2003.

[Dun11] Sean Duncan. Minecraft, Beyond Construction and Survival. 2011.

[Ebe14] David Eberly. Triangulation by Ear Clipping. pages 1–13, 2014.

[FA12] Soraia Ferreira and Artur Pimenta Alves. Location based transmedia storytelling: The travelplot Porto experience design. 2012.

[Far07] Andy Farnell. An introduction to procedural audio and its application in computer games. (September):1–31, 2007.

[FVC12] Miguel Frade, Francisco Fernandez Vega, and Carlos Cotta. Automatic evolution of programs for procedural generation of terrains for video games. *Soft Computing*, 16(11):1893–1914, June 2012.

[Goo13] Google. Google Places API — Google Developers. https://developers.google.com/places/, 2013.

[GPSL03a] Stefan Greuter, Jeremy Parker, Nigel Stewart, and Geoff Leach. Real-time Procedural Generation of ‘ Pseudo Infinite ’ Cities. 2003.

[GPSL03b] Stefan Greuter, Jeremy Parker, Nigel Stewart, and Geoff Leach. Undiscovered Worlds – Towards a Framework for Real-Time Procedural World Generation. 2003.

[Hal08] Luke Halliwell. Procedural content generation. http://lukehalliwell.wordpress.com/2008/08/05/procedural-content-generation/, 2008.

[HGS09] EJ Hastings, RK Guha, and KO Stanley. Automatic content generation in the galactic arms race video game. *IEEE Transactions on . . .* , 1(4):1–19, 2009.

[HKS08] Jan Halatsch, Antje Kunze, and Gerhard Schmitt. Using shape grammars for master planning. In *Design Computing and Cognition’08*, volume 1, pages 655–673, 2008.

[HMU00] John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. Introduction to Automata Theory, Languages, and Computation, 2000.

[HMVI13] Mark Hendrikx, Sebastiaan Meijer, Joeri Van Der Velden, and Alexandru Iosup. Procedural content generation for games. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 9(1):1–22, February 2013.

[HS02] Erin J Hastings and Kenneth O Stanley. Interactive Genetic Engineering of Evolved Video Game Content. *Artificial Intelligence*, pages 5–8, 2002.

[HW08] Mordechai (Muki) Haklay and Patrick Weber. OpenStreetMap: User-Generated Street Maps. *IEEE Pervasive Computing*, 7(4):12–18, October 2008.

[HZB05] Annika Hinze, New Zealand, and George Buchanan. Context-awareness in Mobile Tourist Information Systems : Challenges for User Interaction. (1), 2005.

[Jac11] João Tiago Pinheiro Neto Jacob. A Mobile Location-Based Game Framework. 2011.

[JBFT07] Phillip Jeffrey, Mike Blackstock, Matthias Finke, and Anthony Tang. Chasing the Fugitive on Campus: Designing a Location-based Game for Collaborative Play. *Loading*, 1(May):1–16, 2007.

[JC11] João Tiago Pinheiro Neto Jacob and António Fernando Coelho. Issues in the Development of Location-Based Games. *International Journal of Computer Games Technology*, 2011:1–7, 2011.

[KB04] Jens Krösche and Jörg Baldzer. MobiDENK — Mobile Multimedia in Monument Conservation. pages 72–77, 2004.

[Kir12] Jurek Kirakowski. SUMI Questionnaire Homepage. http://sumi.ucc.ie/, 2012.

[KM06] George Kelly and Hugh McCabe. A survey of procedural techniques for city generation. *ITB Journal*, (14):87–130, 2006.

[KM07] George Kelly and Hugh McCabe. Citygen: An interactive system for procedural city generation. *Fifth International Conference on Game Design and Technology*, (GDTW '07):8–16, 2007.

[LCL11] Daniele Loiacono, Luigi Cardamone, and Pier Luca Lanzi. Automatic Track Generation for High-End Racing Games Using Evolutionary Computation. *IEEE Transactions on Computational Intelligence and AI in Games*, 3(3):245–259, September 2011.

[LCW11] Kate Lund, Paul Coulton, and Andrew Wilson. Free All Monsters ! A Context-aware Location Based Game. In *13th International Conference on Human Computer Interaction with Mobile Devices and Services MobileHCI*, pages 675–678. ACM, 2011.

[Len08] Mark Lentczner. *Community of Variation: Selections from the Context Free Art Gallery, 2005-2007*. 2008.

[Lev] Levenshtein. The Levenshtein-Algorithm. http://www.levenshtein.net/.

[Mag13] MAGELLAN – Part B. 2013.

[MCPS96] Stephen Forrest May, Wayne E. Carlson, Flip Phillips, and Ferdi Scheepers. AL: A language for procedural modeling and animation. Technical report, The Ohio State University, 1996.

[Mel96] Mitchell Melanie. An Introduction to Genetic Algorithms. 1996.

[MHSN10] Glenn Martin, Charles Hughes, Sae Schatz, and Denise Nicholson. The use of functional L-systems for scenario generation in serious games. In *Proceedings of the 2010 Workshop on Procedural Content Generation in Games*, page No.6. ACM, 2010.

[MIBG13] Kathryn E. Merrick, Amitay Isaacs, Michael Barlow, and Ning Gu. A shape grammar approach to computational creativity and procedural content generation in massively multiplayer online role playing games. *Entertainment Computing*, 4(2):115–130, April 2013.

[MMS+08] S. Matyas, Christian Matyas, Christoph Schlieder, Peter Kiefer, H. Mitarai, and M. Kamata. Designing location-based mobile games with a purpose: collecting geospatial data with CityExplorer. In *Proceedings of the 2008 International Conference on Advances in Computer Entertainment Technology*, pages 244–247. ACM, 2008.

[MP96] Radomír Měch and Przemyslaw Prusinkiewicz. Visual models of plants interacting with their environment. *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques - SIGGRAPH '96*, pages 397–410, 1996.

[MPB05] Jean-Eudes Marvie, Julien Perret, and Kadi Bouatouch. The FL-system: a functional L-system for procedural geometric modeling. *The Visual Computer*, 21(5):329–339, May 2005.

[MZ] Rainer Malaka and Alexander Zipf. Challenging IT research in the framework of a tourist information system. pages 1–11.

[NS] Kent L. Norman and Ben Shneiderman. Questionnaire for User Interaction Satisfaction. http://www.lap.umd.edu/quis/.

[OO04] M.J. O'Grady and G.M.P. O'Hare. Gulliver's Genie: agency, mobility, adaptivity. *Computers & Graphics*, 28(5):677–689, October 2004.

[Opp86] Peter E. Oppenheimer. Real time design and animation of fractal plants and trees. *ACM SIGGRAPH Computer Graphics*, 20(4):55–64, August 1986.

[Pat12] Gustavo Patow. User-Friendly Graph Editing for Procedural Modeling of Buildings, 2012.

[Pat14] Akekarat Pattrasitidech. Comparison and evaluation of 3D mobile game engines, 2014.

[PBHS03] Ariel Pashtan, Remy Blattler, Andi Heusser, and Peter Scheuermann. CATIS : A Context-Aware Tourist Information System. (June), 2003.

[PE06] M Pascal and Z Eth. Procedural Modeling of Buildings. 1(212):614–623, 2006.

[PEFS02] Per Persson, Fredrik Espinoza, Petra Fagerberg, and Anna Sandin. GeoNotes : A Location-based Information System for. pages 1–22, 2002.

[PKK01] Günther Pospischil, Harald Kunczier, and Alexander Kuchar. LoL @: a UMTS location based service. 2001.

[PLM$^+$] Stefan Poslad, Heimo Laamanen, Rainer Malaka, Achim Nick, Phil Buckle, and Alexander Zipf. CRUMPET: Creation of User-friendly Mobile Services Personalised for Tourism. pages 2–6.

[PM01] Yoav I H Parish and Pascal Müller. Procedural Modeling of Cities. 2001.

[PM07] D Polson and C Morgan. 'Now everything looks like a game': Mobile phones and location-based games. In *Mobility Conference 2007 - The 4th Int. Conf. Mobile Technology, Applications and Systems, Mobility 2007, Incorporating the 1st Int. Symp. Computer Human Interaction in Mobile Technology, IS-CHI 2007*, 4th International Conference on Mobile Technology, Applications and Systems, Mobility 2007, Incorporating the 1st International Symposium on Computer Human Interaction in Mobile Technology, IS-CHI 2007, pages 751–754, 2007.

[PW99] Aggregation Phenomena and Addison Wesley. Fractals. pages 94–113, 1999.

[RG05] Jessica Rose and James G. Gamble. *Human Walking*. 2005.

[RG14] Janessa Rivera and Laurence Goasduff. Worldwide Traditional PC, Tablet, Ultramobile and Mobile Phone Shipments Are On Pace to Grow 6.9 Percent in 2014. http://www.gartner.com/newsroom/id/2692318, 2014.

[RMC06] Omer Rashid, I A N Mullins, and Paul Coulton. Extending Cyberspace : Location Based Games Using Cellular Phones. *ACM Computers in Entertainment*, 4(1):1–18, 2006.

[Rob09] Andy Robinson. Gearbox interview. http://www.computerandvideogames.com/220328/interviews/gearbox-interview/, 2009.

[Rot] Jörg Roth. Context-aware Web Applications using the PinPoint Infrastructure.

[Rud09] Nicholas Eugene Rudzicz. *Arda: a framework for procedural video game content generation*. PhD thesis, McGill University, 2009.

[Sau11] Jeff Sauro. Measuring Usability with the System Usability Scale. https://www.measuringusability.com/sus.php, 2011.

[SC10] Pedro Brandão Silva and António Coelho. A Procedural Modeling Grammar for Virtual Urban Environment Creation. 2010.

[SGOgW11] Gillian Smith, Elaine Gan, Alexei Othenin-girard, and Jim Whitehead. PCG-based game design: enabling new play experiences through procedural content generation. *Star*, pages 5–8, 2011.

[SGP$^+$] W Schwinger, Ch Grün, B Pröll, W Retschitzegger, and A Schauerhuber. Context-awareness in Mobile Tourism Guides. pages 1–20.

[SHT03] Todd Simcock, Stephen Peter Hillenbrand, and Bruce H Thomas. Developing a Location Based Tourist Guide Application. 21(c), 2003.

[Sim91] Karl Sims. Artifical Evolution for Computer Graphics. http://www.karlsims.com/papers/siggraph91.html, 1991.

[SOGWWF12] Gillian Smith, Alexei Othenin-Girard, Jim Whitehead, and Noah Wardrip-Fruin. PCG-Based Game Design: Creating Endless Web. In *Proceedings of the International Conference on the Foundations of Digital Games - FDG '12*, page 188, New York, New York, USA, May 2012. ACM Press.

[SPD11] Nathan Sorenson, Philippe Pasquier, and Steve DiPaola. A Generic Approach to Challenge Modeling for the Procedural Creation of Video Game Levels. *IEEE Transactions on Computational Intelligence and AI in Games*, 3(3):229–244, September 2011.

[SPK$^+$04] Mark Van Setten, Stanislav Pokraev, Johan Koolwaaij, Telematica Instituut, and P O Box. Context-Aware Recommendations in the Mobile Tourist Application COMPASS. (August 2004):235–244, 2004.

[STdKB11] R.M. Smelik, T. Tutenel, K.J. de Kraker, and R. Bidarra. A declarative approach to procedural modeling of virtual worlds. *Computers & Graphics*, 35(2):352–363, April 2011.

[Tay02] Laurie N Taylor. Video Games: Perspective, Point-of-View, and Immersion. 2002.

[TSL$^+$11] Tim Tutenel, Ruben M. Smelik, Ricardo Lopes, Klaas Jan de Kraker, and Rafael Bidarra. Generating Consistent Buildings: A Semantic Approach for Integrating Procedural Techniques. *IEEE Transactions on Computational Intelligence and AI in Games*, 3(3):274–288, September 2011.

[Tys12] Peter Tyson. *Getting Strated with Dwarf Fortress*. 2012.

[TYSB11] Julian Togelius, Georgios N. Yannakakis, Kenneth O. Stanley, and Cameron Browne. Search-Based Procedural Content Generation: A Taxonomy and Survey. *IEEE Transactions on Computational Intelligence and AI in Games*, 3(3):172–186, September 2011.

[Vis13] VisitBritain. Top Towns Historical 2012, 2013.

[Wal07] Steffen P Walz. Pervasive Persuasive: A Rhetorical Design Approach to a Location-Based Spell-Casting Game for Tourists. In *Situated Play, Proceedings of DiGRA 2007 Conference*, pages 489–497, 2007.

[ZM12] Adeel Zafar and Hasan Mujtaba. Identifying Catastrophic Failures in Offline Level Generation for Mario. *2012 10th International Conference on Frontiers of Information Technology*, pages 62–67, December 2012.

REFERENCES

Appendix A

HTTP Request URLs

A.1 OpenStreetMap API

```
http://api.openstreetmap.org/api/0.6/map?bbox=
          [BOUNDING BOX MINIMUM LONGITUDE],
          [BOUNDING BOX MINIMUM LATITUDE],
          [BOUNDING BOX MAXIMUM LONGITUDE],
          [BOUNDING BOX MAXIMUM LATITUDE]
```

A.2 Overpass API

A.2.1 Way Objects

```
http://overpass-api.de/api/xapi?way[bbox=
          [BOUNDING BOX MINIMUM LONGITUDE],
          [BOUNDING BOX MINIMUM LATITUDE],
          [BOUNDING BOX MAXIMUM LONGITUDE],
          [BOUNDING BOX MAXIMUM LATITUDE] ]
```

A.2.2 Node Objects

```
http://overpass-api.de/api/xapi?node[bbox=
          [BOUNDING BOX MINIMUM LONGITUDE],
          [BOUNDING BOX MINIMUM LATITUDE],
          [BOUNDING BOX MAXIMUM LONGITUDE],
          [BOUNDING BOX MAXIMUM LATITUDE] ]
```

A.2.3 Tourism Way Objects

```
http://overpass-api.de/api/xapi?way[bbox=
          [BOUNDING BOX MINIMUM LONGITUDE],
```

```
          [BOUNDING BOX MINIMUM LATITUDE],
          [BOUNDING BOX MAXIMUM LONGITUDE],
          [BOUNDING BOX MAXIMUM LATITUDE] ][tourism=*]
```

A.2.4 Tourism Node Objects

```
http://overpass-api.de/api/xapi?node[bbox=
          [BOUNDING BOX MINIMUM LONGITUDE],
          [BOUNDING BOX MINIMUM LATITUDE],
          [BOUNDING BOX MAXIMUM LONGITUDE],
          [BOUNDING BOX MAXIMUM LATITUDE] ][tourism=*]
```

A.3 Texture API

```
http://staticmap.openstreetmap.de/staticmap.php?center=
          [CENTRE LATITUDE],[CENTRE LONGITUDE]&
          zoom=15&size=600x450&maptype=mapnik
```

A.4 Wikipedia API

```
https://en.wikipedia.org/w/api.php?action=query&list=geosearch&
          gsradius=10000&format=xml&
          gscoord=[LATITUDE]|[LONGITUDE]
```

A.5 DBpedia API

```
http://lookup.dbpedia.org/api/search.asmx/KeywordSearch?MaxHits=3&
          QueryString=[NAME STRING]
```

Appendix B

Interfaces

B.1 Initial Version

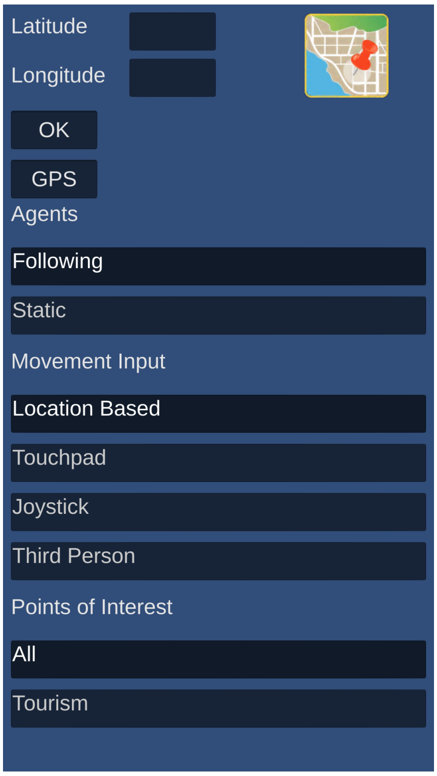
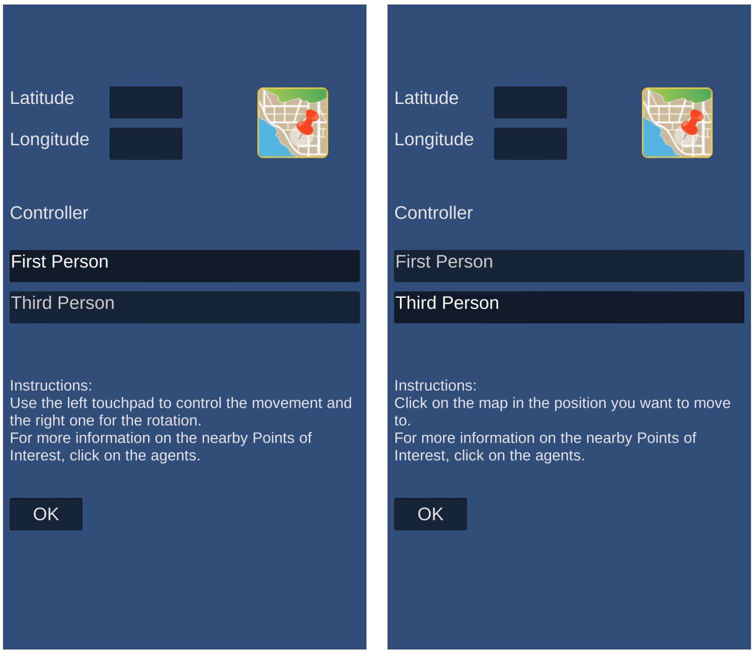


Figure B.1: Main menu of the application

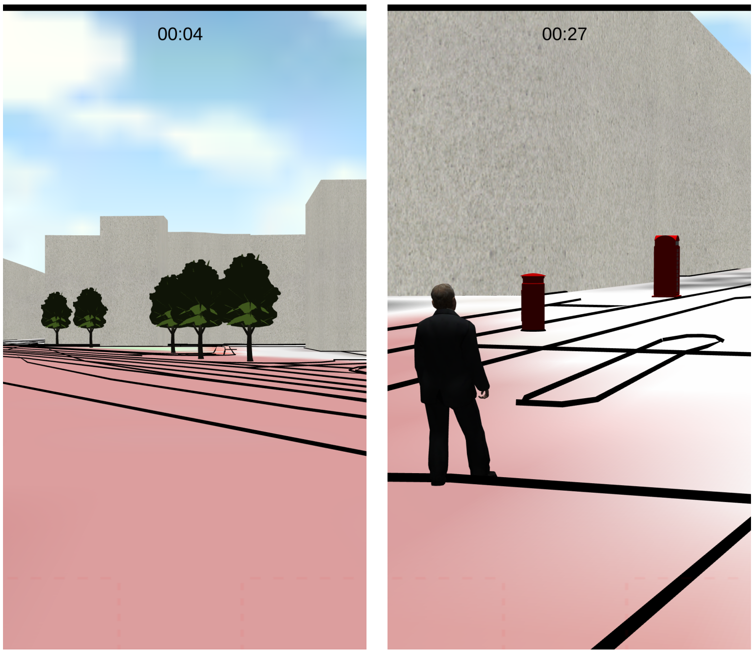
B.2 Test Versions

B.2.1 Pre-test



(a) First-person instructions (b) Third-person instructions

Figure B.2: Main menu of the Pre-testing phase



(a) Tree Model (b) Agent Model

Figure B.3: Main scene, showing some models in First-person view

(a) Phone Model

(b) Tree Model

Figure B.4: Screenshot from the main scene, showing some models in Third-person view

B.2.2 Test

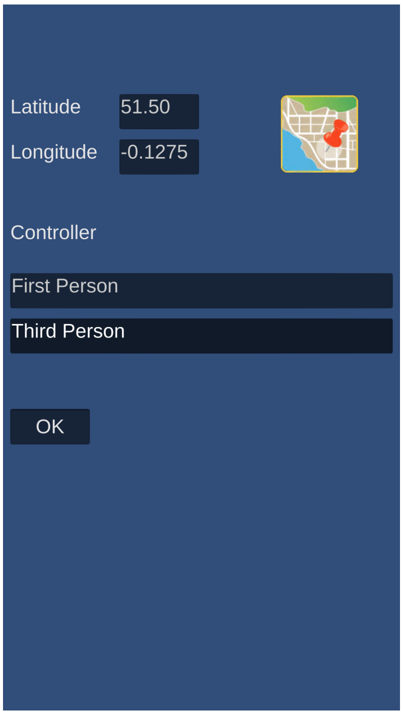


Figure B.5: Main menu of the Testing phase

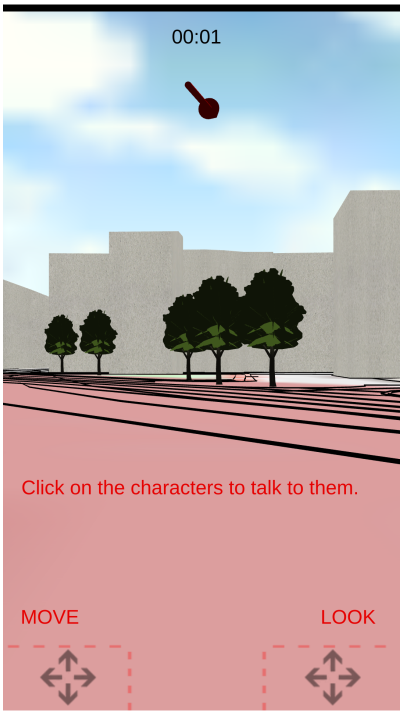


Figure B.6: Main scene, showing instructions for the First-person controller

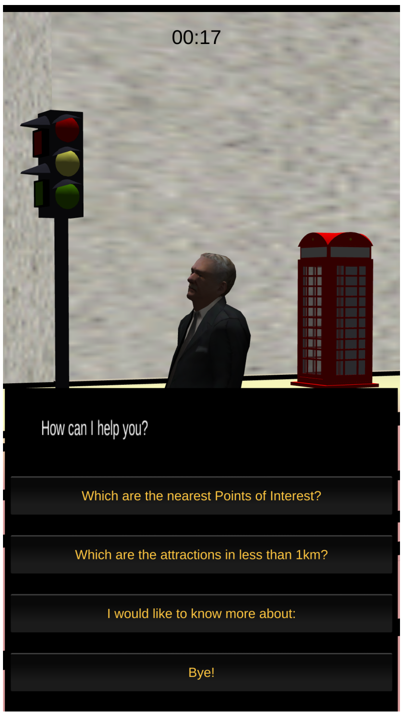
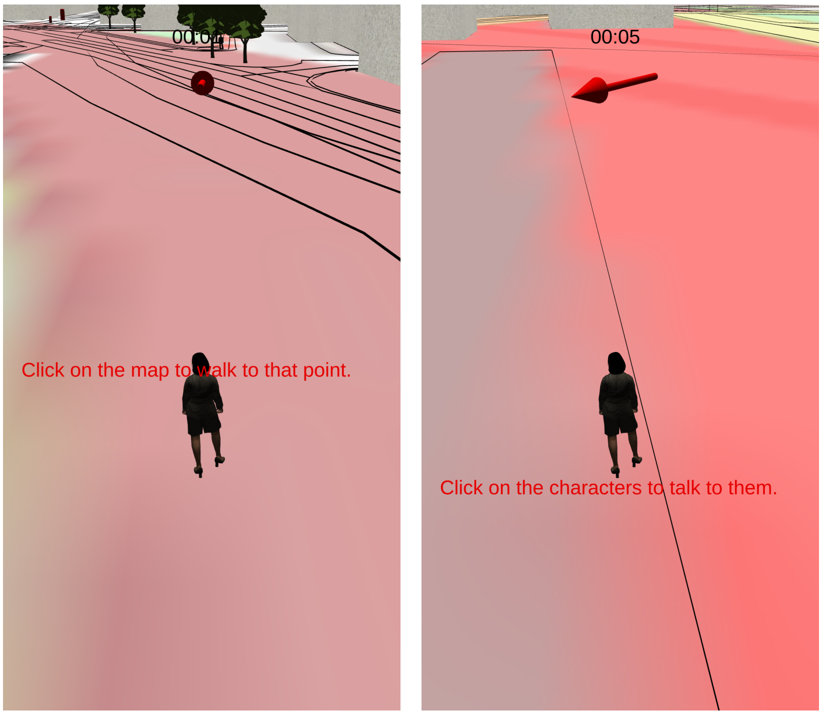


Figure B.7: Main scene, showing a dialogue with a NPC

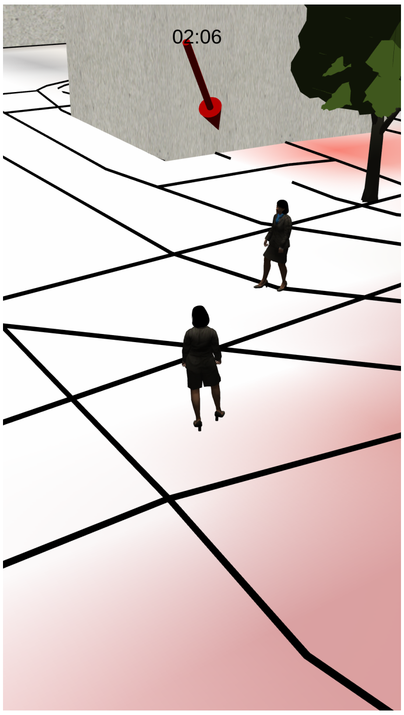


(a) Moving Instructions (b) Dialogue Instructions

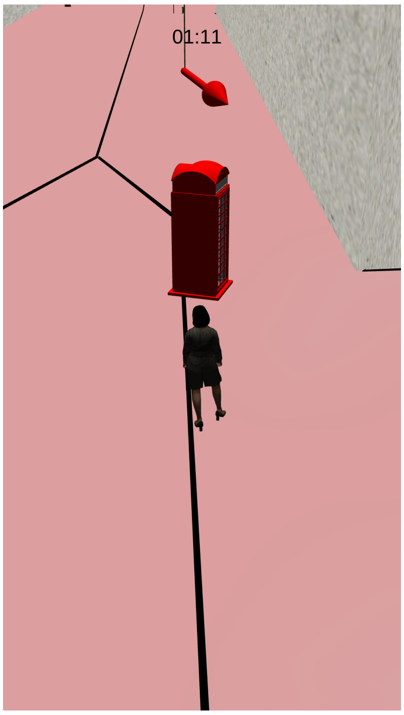
Figure B.8: Main scene, showing instructions for the Third-person controller

(a) Bench and Tree Models (b) Traffic Lights Model

Figure B.9: Main scene, showing some models in First-person view



(a) Agent Model



(b) Phone Model

Figure B.10: Main scene, showing some models in Third-person view

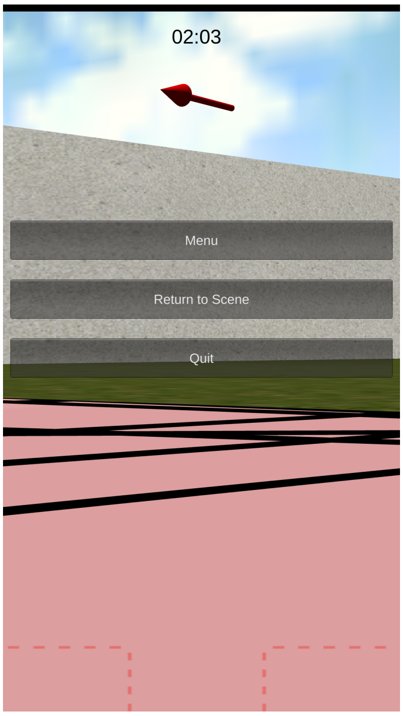


Figure B.11: Pause menu of the application

Appendix C

System Usability Scale

		Strongly Disagree				Strongly Agree
		1	2	3	4	5
1	I think that I would like to use this system frequently					
2	I found the system unnecessarily complex					
3	I thought the system was easy to use					
4	I think that I would need the support of a technical person to be able to use this system					
5	I found the various functions in this system were well integrated					
6	I thought there was too much inconsistency in this system					
7	I would imagine that most people would learn to use this system very quickly					
8	I found the system very cumbersome to use					
9	I felt very confident using the system					
10	I needed to learn a lot of things before I could get going with this system					

System Usability Scale

Appendix D

Test Questionaire

This experiment is part of a Master's degree final project and aims to study whether character-embedded knowledge can be used to increase the retention of information.

It will test the usability of the tool, comparing two different versions of it. It is completely anonymous, and no names or other information will be made public at any time.

D.1 General Questions

This section will try to help building your background.

Do you play games regularly? (required)

- Yes
- No

What kind of touch screen devices do you use? (required)

- Smartphone
- Tablet
- I don't use any
- Other:

How frequently do you travel? (required)
Consider travelling going to a different city from the one of your residence.

- More than once a month
- Once a month
- A few times a year

- Once a year
- Less than once a year
- I don't travel

Do you use any maps application when travelling? (required)

- Yes
- No

D.2 First Experiment

Input the following coordinates: Lat 51.50, Lon -0.1275
Choose the right type of controller.
Note that the application will return to the menu automatically after 3 minutes.

Which is the version you are testing first? (required)

- First-person Controller
- Third-person Controller

Please continue to the next page only after the experiment is finished.

D.3 Questions about the First Experiment

Could you recognize the real location of the scenario? (required)

- Yes
- No

If you could recognize the location, where is it?
If you were able to recognize the location, have you already been there?

- Yes
- No

Please rate your first experiment on the following points.
(required) The values go from 1 (strongly disagree) to 5 (strongly agree), and if you don't know how to respond to a particular item, mark the centre point of the scale (3).

- I think that I would like to use this system frequently

- I found the system unnecessarily complex
- I thought the system was easy to use
- I think that I would need the support of a technical person to be able to use this system
- I found the various functions in this system were well integrated
- I thought there was too much inconsistency in this system
- I would imagine that most people would learn to use this system very quickly
- I found the system very cumbersome to use
- I needed to learn a lot of things before I could get going with this system

Do you have any idea on how to improve this system?

D.4 Second Experiment

Input the following coordinates: Lat 55.952, Lon -3.1935
Choose the right type of controller.
Note that the application will return to the menu automatically after 3 minutes.

Which is the version you are testing now? (required)

- First-person Controller
- Third-person Controller

Please continue to the next page only after the experiment is finished.

D.5 Questions about the Second Experiment

Could you recognize the real location of the scenario? (required)

- Yes
- No

If you could recognize the location, where is it?
If you were able to recognize the location, have you already been there?

- Yes
- No

Please rate your second experiment on the following points. (required)
The values go from 1 (strongly disagree) to 5 (strongly agree), and if you don't know how to respond to a particular item, mark the centre point of the scale (3).

- I think that I would like to use this system frequently
- I found the system unnecessarily complex
- I thought the system was easy to use
- I think that I would need the support of a technical person to be able to use this system
- I found the various functions in this system were well integrated
- I thought there was too much inconsistency in this system
- I would imagine that most people would learn to use this system very quickly
- I found the system very cumbersome to use
- I needed to learn a lot of things before I could get going with this system

Do you have any idea on how to improve this system?

D.6 Final Questions

Which of the versions have you enjoyed the most? (required)

- First
- Second

Do you think the order in which you've tried both versions had any influence on your answer to the last question? (required)

- Yes
- No
- I don't know

Do you think that gamification properties would improve the general experience of the application? (required)

- Yes
- No
- I don't know

Do you think some value can be added to the tool in the future? If yes, what?

Appendix E

Usability Test Results

E.1 Pre-testing

	1	2	3	4	5	6
I think that I would like to use this system frequently	4	2	3	4	2	3
I found the system unnecessarily complex	1	1	3	2	1	2
I thought the system was easy to use	4	4	3	5	5	4
I think that I would need the support of a technical person to be able to use this system	1	1	2	1	1	2
I found the various functions in this system were well integrated	4	3	3	4	3	3
I thought there was too much inconsistency in this system	2	1	2	2	2	2
I would imagine that most people would learn to use this system very quickly	4	3	3	5	4	2
I found the system very cumbersome to use	2	1	2	2	4	2
I felt very confident using the system	5	3	2	5	4	4
I needed to learn a lot of things before I could get going with this system	1	1	1	1	1	2

Table E.1: Results from the First Person Controller version

	1	2	3	4	5	6
I think that I would like to use this system frequently	3	2	3	5	1	2
I found the system unnecessarily complex	3	1	3	1	2	2
I thought the system was easy to use	4	4	3	1	1	4
I think that I would need the support of a technical person to be able to use this system	1	1	3	1	5	2
I found the various functions in this system were well integrated	3	3	3	5	1	2
I thought there was too much inconsistency in this system	3	1	3	2	5	3
I would imagine that most people would learn to use this system very quickly	4	4	2	5	1	4
I found the system very cumbersome to use	2	1	3	1	5	3
I felt very confident using the system	4	2	2	5	1	4
I needed to learn a lot of things before I could get going with this system	1	1	3	1	3	2

Table E.2: Results from the Third Person Controller version

E.2 Testing

	7	8	9	10	11	12	13	14
I think that I would like to use this system frequently	1	2	4	4	2	3	3	4
I found the system unnecessarily complex	4	5	2	2	1	1	2	2
I thought the system was easy to use	1	5	3	4	4	4	4	5
I think that I would need the support of a technical person to be able to use this system	2	1	3	2	1	1	2	1
I found the various functions in this system were well integrated	1	3	3	4	4	5	3	4
I thought there was too much inconsistency in this system	4	1	2	2	1	1	1	1
I would imagine that most people would learn to use this system very quickly	1	5	3	4	2	4	4	4
I found the system very cumbersome to use	1	4	1	1	2	1	4	2
I felt very confident using the system	5	5	4	5	5	5	4	4
I needed to learn a lot of things before I could get going with this system	3	1	2	1	1	1	2	1

Table E.3: Usability questions from the First Person Controller

	7	8	9	10	11	12	13	14
I think that I would like to use this system frequently	2	2	3	4	4	2	3	3
I found the system unnecessarily complex	4	1	3	2	1	1	2	1
I thought the system was easy to use	5	5	4	5	4	3	4	5
I think that I would need the support of a technical person to be able to use this system	1	1	3	1	1	1	1	1
I found the various functions in this system were well integrated	2	2	2	4	5	4	4	4
I thought there was too much inconsistency in this system	1	1	4	2	2	1	2	2
I would imagine that most people would learn to use this system very quickly	3	4	3	4	3	4	4	4
I found the system very cumbersome to use	4	4	3	2	2	2	2	2
I felt very confident using the system	3	2	2	5	5	5	4	5
I needed to learn a lot of things before I could get going with this system	2	2	3	1	1	1	2	1

Table E.4: Usability questions from the Third Person Controller