UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

FLOW IN MOVEMENT INTERACTION GAMES

THESIS TO OBTAIN THE DEGREE OF:
MAESTRO EN INGENIERÍA (COMPUTACIÓN)

BY:
JAVIER RODRIGO DÍAZ ESPINOSA

Supervisor: DR. JUAN PABLO ROMERO MARES
IIMAS

MÉXICO, D.F. AUGUST 2013

To my family.
For all their unconditional love and support.
Thank you for your patience...

Contents

Acknowledgements

I would like to thank first the closest members of my family. My mother, Tere Espinosa who has been my source of encouragement and inspiration. This is one of the many rewards product of your hard work and sacrifices. Thank you so much.

My little brother, Daniel, for all the things we have lived together and all those we will still have ahead (always together Dani!) You have my unconditional support.

To my grandmother Nati, my second mother. To my uncles Javier and Aaron, for always being there for us.

To Samantha, for all your love, encouragement and help. You will always have a special place in my heart.

To all my friends.

A special thank you to Dr. Pablo Romero for his guidance, continuous help and support throughout this project.

I would also like to thank Victor Hugo Franco Serrano for his help with the visual design of the game, the 3D models and textures used in the game.

To CONACyT for the aid of the scholarship.

To the UNAM for giving so much to us and to this beautiful country.

Abstract

In this thesis we describe an empirical investigation into the nature of challenge present in games with movement-based interaction within the context of Flow. Although the importance of challenge in order to produce good experiences for players has been investigated before, with the introduction of new devices capable of detecting the player's movements, physical aspects could have a major impact on the player's experience because they have to use more of their body than just their hands like in traditional interfaces. Therefore challenges could be composite in the sense that they could comprise physical and intellectual elements.

This study employed an experimental paradigm to investigate three aspects of the composite challenges: whether they are balanced (no element dominates over the other), whether the cognitive and physical parts are well integrated (closely related with the activity at hand) and the impact of new challenges that might not be very intuitive to players in promoting a player experience of high quality. The results obtained suggest that balanced and integrated challenges are more likely to promote high quality experiences and that properly motivating a new challenge to the players has a positive impact to its potential to promote them as well.

Ensuring a balance between the difficulty of the game and the skills of the player is considered as the most important condition for promoting flow states. For that reason, we devised an adaptive system based on a neural network with the objective of maintaining such balance. The results show that the proposed adaptive system is capable of maintaining an adequate balance, making the game challenging but not discouragingly hard.

Chapter 1

OVERVIEW

1.1 Introduction

Computer science is the scientific and practical approach to computation and its applications. One of its most challenging disciplines is video game programing, because it combines concepts of computer graphics, networking, human computer interaction and artificial intelligence into real time applications. One of the main goals of video games is to entertain through intrinsic motivation, gamers play video games just for the experience of playing without an expectation of external rewards. Flow is a feeling of complete and energized focus in an activity, with a high level of enjoyment and fulfillment[13], the concept of intrinsic motivation has been strongly related to this theory (also known as Optimal Experience). Therefore, Flow has been proposed as one of the main reasons of why people play video games.

An important condition of the flow state is that the challenges of the task are commensurate with the person's level of skill. However those challenges could be composite in the sense that they could comprise physical as well as intellectual elements. The relation between Flow and video games has been studied before in computing, but the physical aspects have been strongly ignored. With the introduction of devices capable of detecting the player's movements to the gaming industry, the physical interaction required to play could be of major importance to the player's experience.

The Kinect sensor is a motion sensing input device developed by Microsoft. It enables users to control and interact with the Xbox 360 through a natural user interface using gestures and spoken commands. Kinect competes with the Wii Remote Plus and PlayStation Move for the Wii and PlayStation 3 home consoles, respectively. The Kinect sensor claimed the Guinness World Record of being the "fastest selling consumer electronics device" after selling a total of 8 million units in its first 60 days[46].

The document is structured as follows: In the following sections we give a broad description of the work performed and its dissemination. In Chapter 2 we will cover the literature review of the most important concepts related to this work. In Chapter 3 we will introduce formally the research question, the formulated hypotheses and the way they were addressed. We will also detail the firsts experiments conducted to characterize the challenges and test the performance of the proposed adaptive system. In Chapter 4 we will detail The Energos Game, a movement based interaction game developed by the author as a result of the analysis of the studies presented in Chapter 3. In Chapter 5 we present the details of the evaluations of The Energos Game and finally, in Chapter 6 we present the conclusions, improvements and future work.

1.2 Work Developed

The work reported describes an empirical investigation into the nature of challenge present in games with movement-based interaction within the context of flow. Its main objective was to characterize the challenges presented in movement interaction games so that they produce good experiences for the players. Although the importance of challenges, has been investigated before, most computing studies of this concept have considered challenge as cognitive mainly. With the introduction of new devices capable of detecting the player's movements, the physical aspects have an impact in how players perceive the difficulty of the game because they have to use more of their body than just their hands like in traditional interfaces. Considering the physical aspects, challenges could be composite in the sense that they could comprise physical and intellectual elements.

In Section 3.2 we introduce formally the hypotheses put forward in this work, those hypotheses are related with the following aspects of the challenges present in games with movement based interaction:

1. Balance of the intellectual and physical parts of the challenges. This balance refers to the fact that neither the physical nor the intellectual part dominate over the other in order to successfully play the game. When using traditional interfaces (mouse and keyboard or joystick) the challenges are mainly intellectual, but when considering movement based interaction the physical part of the challenges could be as important as the intellectual part.

2. Appropriate integration of the elements of the challenges. This is related with how closely related are the challenges with the activity.

3. Balance between the perceived difficulty of the game and the player's own perceived skills. We implemented an adaptive system using neural networks for maintaining such balance.

4. Introduction and motivation of unfamiliar challenges. Within a particular genre of video games there are recurrent challenges that are associated with it and players are familiar with them in the degree of their own experience within that genre. Unfamiliar challenges are challenges new to the players.

The results of the conducted studies suggest that:

1. A balance between the intellectual and physical parts of the game is essential to promote flow states.

2. When the elements of the challenges are closely integrated and both are significant to the gameplay, the game is more likely to promote flow states.

3. Maintaining a balance between the perceived difficulty of the game and the own player's perceived skills can be achieved dynamically through the proposed adaptive system.

4. Motivating an unfamiliar challenge through a narrative that justifies the introduction of the challenge has a positive impact to the promotion of episodes of flow.

The results obtained from these studies can promote a significant progress in the area of user experience and to the theory of optimal experience. To the best of our knowledge, this is one of the first studies that consider challenge not as a monolithic concept but as a composite one. Furthermore, this research will contribute to the accumulation of practical knowledge that can create game environments with a high quality user experience in a systematic manner and can thus have a significant importance in the field of applied research and commercial application designers.

1.3 Dissemination

The work presented in this thesis has been presented in the following congresses and seminars:

1. As a poster in Fun and Games 2012 celebrated in Tolouse France in September of 2012 (http://fng2012.org/. Accessed 1 June 2013).

2. As a work in progress paper in MexIHC-2012 celebrated in Mexico City in October of 2012 (http://www.mexihc.org/2012/. Accessed 1 June 2013).

3. As a talk in the “Seminario de Inteligencia Artificial” (Artificial Intelligence Seminar) organized by the DCC (Computer Science Department) of IIMAS (http://turing.iimas.unam.mx/ seminarioia/index.html. Accessed 31 March 2013).

4. Submited paper to Games and Culture (Indexed Journal) eISSN: 1555-4139; ISSN: 1555-4120.

5. As a exhibition module in Universum (Museo de las Ciencias).

   Keeping in mind the requirement of an ecologically valid environment we coordinated efforts with Universum (Museo de las Ciencias), a popular science museum located in Mexico City so that the final evaluation could be presented in the museum's facilities, even though this required much more work than evaluating the game in a laboratory.

Chapter 2

FLOW IN GAMING

In this chapter we will discuss the concept of Flow and how it has been integrated with video games considering traditional interfaces (mouse and keyboard or joypad). In these computing studies the challenges of the game have been considered mainly cognitive. However those challenges could be composite in the sense that they could comprise physical as well as intellectual elements and with the introduction of movement based interaction games the physical factor could have a major importance in the player's experience. This is one of the main propositions of the Embodied View of Flow proposed by Romero and Calvillo-Gámez[58], so it will be discussed here as well (Section 2.1.2).

During the evolution of games, there have been recurrent challenges that are associated with certain genres of video games. However, movement based interaction could introduce new challenges to the gaming scene. Considering the potential of natural interfaces, we believe that these new challenges could be either intuitive (when they resemble real life movements) or completely new to the players. The expertise of the player with the challenges present in the game could be of major importance in the way he/she perceives their difficulty and this would have repercussions on the player's experience (Section 2.2).

An important condition to achieve a Flow state is that the challenges of the task are commensurate with the person's level of skill. For that reason, we implemented an adaptive system that regulates the difficulty of the game depending on the player's performance. As it uses neural networks, in the last part of this chapter we will provide the background of Neural Networks and the benefits of using them (Section 2.3).

2.1 Flow

The Flow concept was introduced by Mihaly Csikszentmihalyi in 1975 with the book "Beyond boredom and anxiety"[18] with the objective of explaining happiness. In "Flow: The Psychology of Optimal Experience"[19], Csikszentmihalyi defines Flow as "a state of concentration so focused that it amounts to absolute absorption in an activity" and describes how these "optimal experiences" interrelates with happiness. The term Flow comes from a metaphor of a water current carrying the person along during the development of the activity[19]. The Flow experience is also commonly known as optimal experience or being "in the zone". Some other descriptions of a Flow state are:

> Flow represents the feeling of a complete and energized focus in an activity, with a high level of enjoyment and fulfillment[23].
>
> Flow is an extreme experience combining elements of skill, progress, challenge and focus to produce an optimal experience[16].

According to the studies of Csikszentmihalyi[19] and Csikszentmihalyi, Abuhamdeh and Nakamura[21], Flow has the following major components:

1. A challenging task that can be completed and requires skill. An activity that is goal-directed, bound by rules, requires attention and certain skills. According to Chen[14], Flow can emerge from any kind of activity.
2. Clear goals for the task.
3. Immediate feedback provided to the person.
4. Balance between the perceived challenges of the activity and the skills of the participant exists.
5. Concentration on the task at hand. The activity requires a high level of concentration. Only a select range of information can be allowed into awareness.
6. Deep but effortless involvement.
7. Sense of control over the actions. Lacking the sense of worry about losing control or the sense of exercising control in difficult situations.
8. Concern for self disappears during the activity, but after the Flow experience is over emerges stronger.
9. The sense of time duration is altered.

The combination of these elements cause a sense of deep enjoyment so rewarding that people feel that expending a great deal of energy is worthwhile simply to be able to feel it[19].

Csikszentmihalyi, Abuhamdeh and Nakamura[21] identifies 2), 3) and 4) as the conditions for an activity to promote Flow states and the other components as the characteristics product of a Flow state.

During the Flow experience people typically feel at the peak of their abilities, movements seem to occur by themselves, and both sense of time and worries seem to disappear [19] [37].

According to Csikszentmihalyi[19], an optimal experience is autotelic or self-contained. This is an experience that is performed not with the expectation of some future benefit, but simply because the doing itself is the reward.

In a Flow experience the individual participates in an active interaction with the environment that involves the execution of skill-related behaviors[45]. However, there has to be a balance between the perceived challenges of the task at hand and the own perceived skill of the participant[21].

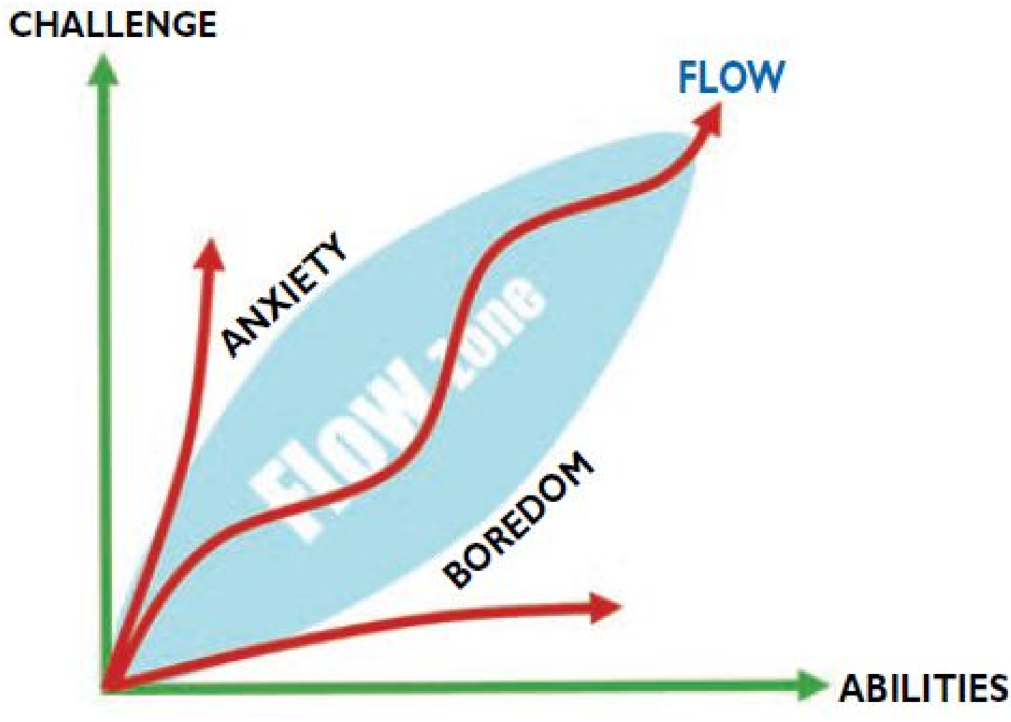

Figure 2.1: Flow zone factors. Taken from Chen (2007) [14]

If the perceived difficulty of the challenges is beyond the ability, the activity generates anxiety. But if the difficulty of the challenges fails to meet the ability of the participant, he/she will lose interest and eventually become bored (See figure 2.1). Keller[44] recognize the balance of skills and task demands as the most important condition for promoting Flow experiences.

2.1.1 Flow and Video Games

Sales of computer and video games in the U.S. have grown from $7.0 billion in 2005 to $10.5 billion in 2009[62]. Also, unit sales of computer and video games have risen from 226.3 millions units in 2005 to more than 273 million units in 2009 [62].

The video game industry generated nearly $25 billion in revenue in 2011. Sales of game software and content, including games made for consoles, portable gaming devices and PCs, as well as digital full game downloads, downloadable content and social games, accounted for approximately $16.6 billion of that total[9].

Looking at those numbers, we can think of video games as one of the most successful application of computing. But why?

The description of a player experience when totally immersed in a video game resembles the description of Flow states. Players lose track of time and forget external pressures. Gamers value video games based on whether those games can provide Flow experiences[40].

Common characteristics of video games are[61]:

- Concrete goals and manageable rules.
- They provide action that can be manually or automatically adjusted to our capabilities.
- They provide clear feed-back in terms of running scores, collections of artifacts, or progress reports.
- They have abundant visual and aural information that helps screen out distraction and facilitate concentration.

Flow is widely accepted to be one of the fundamental reasons of why people play games[22]. Sweetser P. and Wyeth P.[65] developed the GameFlow model, that is an adaptation of the concept of flow developed by Csikszentmihalyi to video games. It is used to measure enjoyment in games. Consists of eight core elements - concentration, challenge, skills, control, clear goals, feedback, immersion, and social - related to Csikszentmihalyi's[19] elements of flow (see Table 2.1).

In summary, Sweetser[65] identifies the following main characteristics:

1. The game must keep the player's concentration through a high work-load.
2. The tasks of the game must be sufficiently challenging to be enjoyable and the player must be skilled enough to undertake them.

Games Literature	Flow
The Game	A task that can be completed
Concentration	Ability to concentrate on the task
Challenge and Player Skills	Perceived skills should match challenges and both must exceed a certain threshold
Control	Allowed to exercise a sense of control over actions
Clear Goals	The task has clear goals
Feedback	The task provides immediate feedback
Immersion	Deep but effortless involvement, reduced concern for self and sense of time
Social Interaction	Not Available

Table 2.1: Mapping the Elements from Games Literature to the Elements of Flow, taken from Sweetser (2005)[65]

3. The tasks must have clear goals so that the player can complete them.
4. The player must receive feedback on progress toward completing the tasks.
5. If the player is sufficiently skilled and the tasks have clear goals and feedback, then he or she will feel a sense of control over the task.
6. The resulting feeling for the player is total immersion or absorption in the game.
7. Which causes them to lose awareness of everyday life, concern for themselves, and alters their sense of time.

Challenge is consistently identified as the most important aspect of good game design. Games should be sufficiently challenging, match the player's skill level, vary the level of difficulty, and keep an appropriate pace[65]. Games should be designed to have a level of challenge that is appropriate and not discouragingly hard or boringly easy[65].

Different games privilege abilities in these two categories: cognitive abilities and psycho motor abilities [43]. With the use of movement-based interfaces the players are able to use active body movements to control the game and with the introduction of devices able to capture the movement of the players, there has been a huge development in games that uses this kind of interaction. However, as this a new research area, there are not many studies on how the tangible aspect of the physical world affect immersion in Human Computer Interaction.

A natural user interface (NUI), or Natural Interface refers to a user interface that is:

1. Effectively invisible to its users, or becomes invisible with successive learned interactions.

2. Intuitive because resembles interaction with the physical world.

We use our body to interact with the world and modify our environment, so moving our bodies to communicate comes natural to us.

Categorizing the movement-based interfaces by the amount of movement they require, we can identify three sets[54]:

1. Minuscule movements required (eye movements interfaces).

2. Moderate arm movements required (used in virtual environments).

3. Significant physical activity, also known as exertion interfaces (found mainly in entertainment and games context).

Bianchi-Berthouze et al.[10] found that body movements appear not only to increase the players' level of engagement but also to modify the way they get engaged. For the players, gaming was no longer just a question of challenge; it was about the experience itself[10].

2.1.2 Embodied View of Flow

In computing studies of Flow ([31],[39] [67]), the task has been considered mainly cognitive. However, when considering movement-based interaction, challenges could be composite in the sense that they could comprise physical as well as intellectual elements[59]. Romero and Calvillo-Gamez (2012)[59] consider balanced challenges (intellectual and physical) of central relevance for applications trying to promote Flow states.

Romero and Calvillo-Gamez[58] propose a view of Flow based on phenomenology and embodiment taking as a main reference the work of Dourish[28]. This embodied view of Flow has four central points: 1) The importance of attention. 2) The importance of context. 3) In Flow with or though the system. 4) The role of the body in the interaction with the system. We will focus on 1) and 4) because of their immediate relevance to the work presented in this thesis.

2.1.2.1 The Importance of Attention

Flow has been defined as a state of deep concentration that is perceived as effortless[58]. The concept of effortless attention is the central part of our definition of Flow and it is the characteristic that we will use to measure whether or not people experience Flow states. Romero and Calvillo-Gamez[59] make the distinction of captive attention (when attention is captured by external stimuli without spending any effort) and when people choose to maintain attention on specific stimuli, by spending some effort to keep the focus. Normally, the subjective attentional effort in a task is proportional to the demands of the task, until there comes a point in which no increase in effort is possible (see Figure 2.2a). But there are occasions in which, at some point during the execution of the task, one is concentrated so thoroughly that suddenly attention seems effortless. Then, a state of effortless attention can be accomplished after a period of effortfull attention (see Figure 2.2b). Another important point mentioned by Romero and Calvillo-Gamez[59] is that this struggle is required, because otherwise, the mind tend to wander aimlessly preventing focus and control.

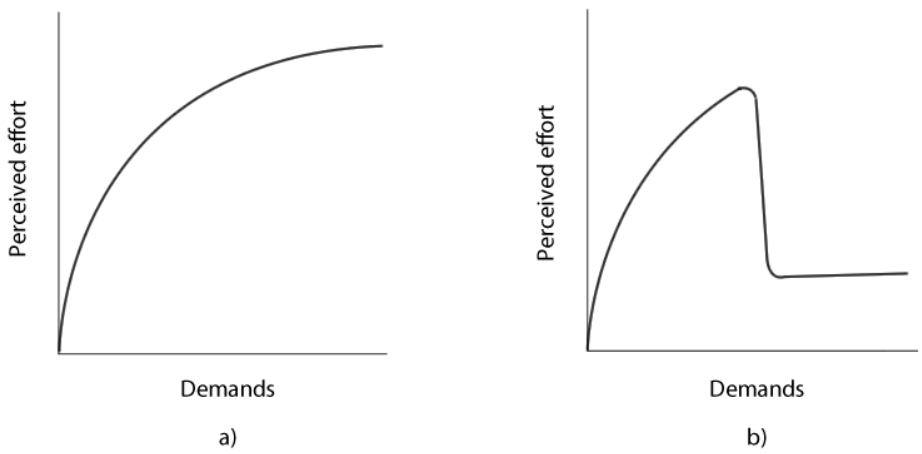

Figure 2.2: Effort vs. demands in a) effortful and b) effortless attention. Taken from Bruya (2010)[11]

2.1.2.2 The Role of the Body and Composite Challenges

The body plays a central role in this view of Flow. Although Human-computer interaction has always consisted of embodied action (traditionally of small movements of the hands on the keyboard and mouse), the engagement of the body can be more significant to the user experience. Therefore, the challenges of the task have to be analyzed to understand how they engage the physical and intellectual parts of the person.

Romero and Calvillo-Gamez[59] also define the concept of composite challenges. Studies of Flow generally measure challenges and skills using unidimensional scales, yet challenges may require several skills from the user, so unidimensional scales for measuring challenges may not be the best option. According to Ellis et al. [29], challenges might be composite multidimensional constructs in the sense that they are associated with the cognitive, physical and emotional parts of the person.

Within the context of a traditional form of interaction (using mouse and keyboard or gamepads), challenges are unbalanced as they are frequently dominated by cognitive aspects. Within movement-based interaction however, balanced challenges might have a central relevance in promoting Flow. Finneran and Zhang[31] have raised the question of whether achieving a Flow state depends on focusing all of the cognitive, physical and emotional parts of the person on the same activity. So besides balancing the relevance of challenge elements, another important aspect could be whether those elements are only tangentially related or closely integrated to the activity.

2.2 Familiar Challenges in Games

Video games are mostly categorized into genres by their gameplay, independently of their game-world content or their visual characteristics. Those genres are defined by a set of gameplay challenges[6]. For example, a Real Time Strategy game will be considered as such if it meets certain gameplay challenges regardless of whether it takes place in an imaginary future or a specific time of past history. Familiar challenges is a term we use to refer to challenges that are associated with a genre. Players expect those challenges when they relate a game to a genre. In Table 2.2 we detail a list of the most common video game genres with their general description and some examples of popular games that fit within that genre. However, this will not be a comprehensive table, the video games industry is always changing and evolving so it could become obsolete in a short period of time. Its objective is to give a broad perspective of the most common genres of video games and some of its characteristics and known challenges.

2.2.1 Action Genre

The action genre is one of the most important genres nowadays. It was the most popular game genre in 2011, accounting for 19 percent of all games sold during that year[30]. Action genre was one of the earliest types of video games. They require quick reflexes, accuracy and careful timing to overcome or conquer obstacles. Most of the arcade games are action games. The goal of most action games is to overcome enemies without being destroyed. It also has a real-time game play with an emphasis on combat. Within the action genre, we can identify sub-genres with different characteristics, Table 2.3 mention them in more detail.

2.2.2 Familiar Challenges in Movement-Based Games

At the time of writing, there were 3 important devices for detecting the player's movements, the Wii Remote[53], the PlayStation Move and the Kinect sensor. The Wii Remote and the PlayStation Move operate in a similar way, the player uses a handheld motion controller wand and the console monitors the controller's position and orientation to detect the player's movements. The Kinect sensor on the other hand, features an RGB camera, depth sensor and multi-array micro-

Genre	Characteristics	Common Challenges	Games
Adventure	Game play without reflex challenges or action. It involves strong story elements.	Exploration, puzzle-solving, collecting items and interaction with people or the environment.	Myst, The Longest Journey, Monkey Island.
Action	They require quick reflexes, accuracy and careful timing to overcome or conquer obstacles. The goal is to overcome enemies without being destroyed. Real-time game play with an emphasis on combat.	Controlling a character or avatar, navigate through levels, collecting objects and abilities, avoiding and battling enemies with various attacks.	Super Mario Bros, Halo, Quake, Street Fighter
Action-Adventure	Hybrid of the Action and Adventure genres. Genre has become increasingly popular.	They combine the fast and real-time game play of Action games with story elements and puzzles of Adventure games.	Uncharted, God of War, Legend of Zelda: Ocarina of Time.
Role Playing Games	Draw their game play from traditional role-playing games. The player is in the role of one or more characters who specialize in specific skill sets while progressing through a predetermined storyline.	Exploration, puzzle-solving, collecting items and interaction with people or the environment.	Star Wars: Knights of the Old Republic, The Elder Scrolls series, Final Fantasy series.
Simulation	Simulation games try to accurately re-create real-world experiences. For that reason, they should follow real-world laws as much as possible	Operating vehicles, building or managing cities, civilizations, armies etc, compete in sports or other events.	Flight Simulator X, Sim City, Madden NFL
Strategy	Game play typically involves managing a limited set of resources to achieve a goal. Most strategy games are militaristic, involving combative play and competition.	Managing resources, constructing buildings and units, movement and combat, multitasking between all mentioned challenges and being under constant pressure.	Civilization series, Starcraft series, Age of Empires series.

Table 2.2: Common Genres of video games and their well known challenges.

phone which provide full-body 3D motion capture, facial recognition and voice recognition capabilities. These two kinds of movement detection provide advantages and disadvantages. For example, the use of a handheld controller proves to be much more accurate, but it doesn't allow full body recognition. However, the kinect sensor is much more sensitive to changes in illumination and the motion tracking is less precise but the controller-free concept is undeniably appealing.

We chose the kinect sensor for the development of our game because it allows full body recognition and also was very popular at that time so we will focus

Action Sub-Genre	Characteristic Challenges	Examples
Platformers	Traveling between platforms traditionally by jumping. Running and climbing ladders and ledges. They also frequently borrow elements from other sub-genres like fighting and shooting.	Super Mario Bros[76], Donkey Kong[69], Space Panic[74].
Shooters	Can be divided into first-person shooters (FPS), third-person shooters (3PS), rail-shooters and classic shooters, depending on the camera perspective and movement. Focus on combat involving a reticle to aim projectile weapons to destroy enemies or objects and focus on survival or high scores	Quake[73], Halo[71], Gears of War[70]
Fighting	Focus on one-on-one combat between two characters. Linking long chains of button presses on the controller (combos) to attack. Pressing accurately a combination of buttons to make a counter attack.	Street Fighter[75], King of Fighters[72], Dead or Alive[68].

Table 2.3: Familiar challenges and examples of sub-genres of the Action Genre

on Kinect games. However all three are good options for the development of movement interaction games.

Although the introduction of exertion interfaces for gaming is relatively new, we can identify recurrent challenges and game-plays for movement interaction games. Particularly for games that use the Kinect sensor, the most popular games could be categorized inside the action genre because they require quick reflexes, accuracy and careful timing. Kinect games mostly focus on the next game-plays:

1. Generic Action. This includes navigating trough the world, avoiding obstacles, fighting enemies or doing assorted challenges depending on the game.

2. Dancing. Game play oriented around the player's interactions and synchronization with a predefined choreography of a song.

3. Shooting. Involves using the hands to control the shooting reticles, it also includes gestures that implements different mechanics in a particular game.

4. Racing. Use gestures to control a car and race against other drivers.

5. Sports. Control the avatar with movements of the body, resembling how the sports are played.

Considering the compatibility to integrate challenges of the general action category with the shooters category and the popularity of shooters that use traditional interaction, we will focus only on category 1 and 3.

2.2.2.1 Generic Action

Many games that fit inside this category are arcade games and mini games with assorted challenges. However, common elements are: map the movements of the player to an avatar to avoid obstacles, adopt a certain posture, put the hands or feet in a certain position to achieve something (for example in Kinect Adventures: 20,000 Leaks, the player needs to cover cracks and holes while being inside a glass cube underwater using his or her limbs and head) and use predefined gestures to do an specific move inside the game (for example, make a rotated figure-8 motion with a hand to reflect laser fire in Kinect Star Wars). An example of a commercial game that requires a lot of physical challenges from the player is explained in the following section.

2.2.2.2 Kinect Adventures: Reflex Ridge

Figure 2.3: Reflex Ridge minigame from Kinect Adventures.

Kinect Adventures is an Xbox 360 video game launched in 2010, that consists of several mini games that utilize the kinect motion camera to copy the player's full body movements and project them onto the in-game avatar [15].

One of the mini games is Reflex Ridge, an on-rails obstacle course, in which one or two players (in split screen) race over platforms through roller coaster-like tracks. The objective of the game is to finish the course in the quickest time, while also collecting as many Adventure pins as possible. To collect pins, the players must gather the ones scattered along the road, while also avoiding multiple obstacles by jumping, stepping to the right or to the left, and ducking [63] (See Figures 2.3 and 2.4). Each time a player gets hit by an obstacle his platform will slow down. Jumping in place makes the platform move faster along its rail, also grabbing onto handles allow the player to launch themselves forward to gain more speed. The game is timed, with extra time left over at the end of the course added to the Adventure pin total [48].

Figure 2.4: Playing movements in Reflex Ridge. Taken from the owner's manual of Kinect Adventures.

2.2.2.3 Shooting

The shooting genre has been ported in a particular way to the exertion interfaces that use kinect, most games uses the hands to control the shooting reticle and to shoot (like in Child of Eden[66], The Gunslinger or Crimson Dragon). They also limit the movement of the player to a predefined path, so that they wouldn't wander freely, making them a Rail Shooter. Sometimes, the advancement through the path is automatic and other times the player can control it. An example of a worldwide popular shooter (during the 2011 year) for kinect is Child of Eden.

2.2.2.4 Child Of Eden

Child of Eden is a Rail Shooter created by Tetsuya Mizuguchi, which is the successor to his 10-year-old game Rez. It was released in 2011 for Xbox 360 and PlayStation 3 [27].

Child of Eden thrusts you in the center of a battle to save Project Lumi, a mission to reproduce a human personality inside Eden, the archive of all human memories. As the project nears completion, the archive is invaded by an unknown virus. The Players mission is to save Eden from the virus to restore hope and peace [66].

2.2.2.5 Gameplay

Levels consist of corridors with predefined paths where the virus is found, the player advances through the paths automatically. The virus presents itself as colorful shapes that the player must eliminate. Each of the player's shoots creates a melodic sound that merges with the background music, which is also closely linked to the graphics (See Figure 2.5) [26].

The game can be played with traditional controllers or using the kinect sensor, while using the Kinect, players must aim using their hands. With the right hand the player controls the Lock-on Laser reticle which can lock up to eight targets at a time, and by flicking the hand forward quickly the laser is released and destroy all the locked-on targets. Extra points are awarded if players successfully release the shot with eight targets locked and if they activate the beam in sync with the background music (if this is achieved, "Perfect" message appears on the screen, see Figure 2.5) [49] [66].

Using the left hand, the players activate a Tracer beam to shoot down enemy projectiles as well as certain enemies (who are always purple-colored). This tracer fires automatically, therefore the player only focus on aiming the reticle. Players can switch between weapons by clapping their hands [49].

Finally, players are able to activate Euphoria (a screen-clearing bomb, essentially) by raising both hands in the air a and release a burst of energy which has a powerful effect on all on-screen enemies [66].

Figure 2.5: Screenshot of Child of Eden.

However, the new affordances of games bring possibilities of the creation of new challenges. An example of a game that implements a new challenge is explained in the following section.

2.2.2.6 Body and Brain Connection: Pacman Mini game

Body and Brain Connection is a collection of puzzle mini-games that asks to solve mental problems, such as math questions to keep the user's brain active, but in order to answer the questions the player must perform various physical motions, thus reinforcing the mental answers by the motion controls [27]. The mini-games are split into math, memory, logic, reflex, and physical with several games for each category [57].

In particular, there was a mini-game about coordination that included a non intuitive unfamiliar challenge (See Figure 2.6 for a screenshot). The idea is similar to Pac-Man, but the way the game is executed is all about body and brain link. The screen is split into two: on the left is Pac-Man and the ghost characters, while on the right is the fruit [33]. By holding the cursor over Pac-Man with the left hand, players must guide the Pac-Man character away from the ghosts. At the same time, the player has to keep the right hand cursor over a moving piece of fruit for as long as possible to gain points [32]. The player has to focus on both objects in order to earn the highest score possible.

This mini game is an example of a challenge that is new and therefore not very well known to players. It can require a big amount of concentration and also to focus on two different things at the same time, which can be quite hard.

Figure 2.6: Minigame from Body and Brain Connection with strong coordination elements.

During this section, we have presented the concept of familiar challenges in games. They are challenges that are associated with a genre of game because of their recurrence. Players are familiar to those challenges in the degree of their own experience with video games of a certain genre. Although the introduction of new ways to interact with the game (movement-based interaction) is relatively new, games are starting to show some recurrent challenges and players will get used to them in the degree in which they play kinect based games. Exertion interfaces have another advantage: if the gameplay and controls are planned and programmed to resemble real life movements, challenges could be familiar to us by analogy with the physical world. This is already happening, for example, mapping the exact movements of the player to an avatar in order to avoid obstacles (like in Kinect Adventures: Reflex Ridge) makes a connection between the player's body and the avatar. This connection could make the player use the same movements that he or she would use to avoid that particular obstacle in the physical world.

2.3 Adaptive Challenge in Gaming

In the Section 2.1 we have mentioned that the balance between the player's own perceived skills and the perceived level of challenge is essential to promote Flow states. One option to provide a commensurate level of challenge is for the game to have a mechanism capable of automatically making adjustments to its difficulty depending on the performance of the player. This would result in an adaptive system, whose concrete implementation options may vary from simple heuristics to AI-based approaches for complex games[8]. In most games, the level of difficulty can only be adjusted by choosing a multi-tier parameter that represent "easy","medium" and "hard" difficulty. This leaves little room for personalization options trough different combinations of parameters, because they mostly alter them either in a smooth linear fashion, or through steps represented by the levels of difficulty. The concept of Dynamic Difficulty Adjustment (DDA) has been introduced to games recently, where the individual abilities of each player are detected and then parameters, scenarios and behaviors of the video game are changed in real-time accordingly. However, the way each method of adjustment implemented varies a lot. We will use a combination of a decision making problem in the optimization context with supervised learning. The objective of the optimization research is to develop efficient techniques for finding maximum or minimum values of a function of one of several variables[7]. Usually, this function is known as objective function or cost function. Decision making always involves making a choice between several possible alternatives. For us, the alternatives are different combination of parameters that define the difficulty of a game. There are two categories of decision making problems:

1. The set of possible alternatives for the decision is a finite discrete set typically consisting of a small number of elements.

2. The number of possible alternatives is either infinite, or finite but very large, and the decision may be required to satisfy some restrictions and constraints.

The function that models the relation between difficulty and the player's performance could be very hard to determine, and also could be defined in a different way depending on the player. However, supervised learning do not need the function beforehand, because it will try to infer it from the training examples.

The machine learning task of inferring a function from labeled training examples is known as supervised learning[60]. Each training example consists of an input object and a desired output value. There is a supervised learning algorithm that after an analysis of the training examples, produce an inferred function that will be used to map unknown examples.

For our problem, we are changing the values of different parameters that impact directly on the difficulty of the game while trying to maximize both the performance of the player and the difficulty of the game. We discuss this in more detail in Section 3.3.3. We used neural networks to try to solve this complex problem. Other techniques used in supervised learning include Bayesian Statistics, genetic algorithms, decision three learning, etc, however, for the time being, we will focus on neural networks because those were used in the development of the adaptive module on our system (see section 2.3.2 for the benefits of using neural networks).

2.3.1 Artificial Neural Networks

Most of the theory presented in this section is taken from Haykin (1999)[36].

The brain is a highly complex, nonlinear and parallel information system. Computational neurobiologists have constructed very elaborate computer models of neurons in order to run detailed simulations of particular circuits in the brain. An artificial neural network, commonly referred to as "neural network", is an information processing paradigm inspired by the way biological nervous systems process information. However, in the Computer Science environment, we are more interested in the general properties of neural networks and not so much in how they are actually "implemented" in the brain. Therefore, abstract neurons that are much simpler can be used.

Haykin [36] defines a neural network as a massively parallel distributed processor made up of simple processing units, which has a natural propensity for storing experimental knowledge and making it available for use.

These neural networks resembles the brain in two aspects:

1. Knowledge is acquired by the network from its environment through a learning process.

2. Interneuron connection strengths, known as synaptic weights are used to store the acquired knowledge.

The learning process is implemented by a learning algorithm. Traditionally, the goal of this process is to modify the synaptic weights of the network to attain a desired design objective. But also, it is possible for a neural network to modify its own topology.

2.3.2 Benefits of using Neural Networks

- Generalization. Capability of the neural network to produce reasonable outputs for inputs not encountered during the learning phase. This is a very important point for us, because we want to predict the player's performance with several combinations of parameter's values to determine one that could fit the player's skills.

- Nonlinearity. If the mechanism responsible of generating the input signal is inherently nonlinear, this property is highly important. Luckily, an artificial neural network can be linear or nonlinear.

- Input-Output Mapping. The neural network learns from the examples by constructing an input-output mapping for the problem at hand. Supervised learning involves modification of the synaptic weights of the neural network by using a set of training examples. Each example consists of a unique input signal and a corresponding desired response. Then an example picked from the set is presented to the network and the synaptic weights are modified to minimize the difference between the desired response and the actual response of the network produced by the input signal. The training of the network is repeated for many examples in the set until the network reaches a steady state where there are no further significant changes in the synaptic weights.

- Adaptivity. Neural networks have a built-in capability to adapt their synaptic weights to changes in the surrounding environment. Particularly, a neural network trained to operate in a specific environment can be easily retrained to deal with minor changes in the operating environmental conditions.

- Neurobiological Analogy. The design of a neural network is motivated by analogy with the brain, which is a living proof that fault tolerant parallel processing is not only physically possible but also fast and powerful.

2.3.3 What is a Neuron?

A neuron is an information-processing unit that is fundamental to the operation of a neural network. There are four basic elements of the neuron (see fig 2.7):

1. Set of synapses or connecting links, each one characterized by a weight or strength of its own. A signal x_j at the input of synapse j connected to the neuron k is multiplied by the synaptic weight w_{kj}. The first subscript of the synaptic weight refers to the neuron in question and the second refers to the input end of the synapse to which the weight refers. The synaptic weight of an artificial neuron may lie in a range that includes negatives as well as positives values.

2. An adder for summing the input signals weighted by the respective synapses of the neuron (linear combiner).

3. An activation function for limiting the amplitude of the output of a neuron. Typically is the closed unit interval $[0, 1]$ or alternatively $[-1, 1]$

In Figure 2.7 there is also an externally applied bias, denoted by b_k. The bias b_k increases or lowers the net input of the activation function, depending

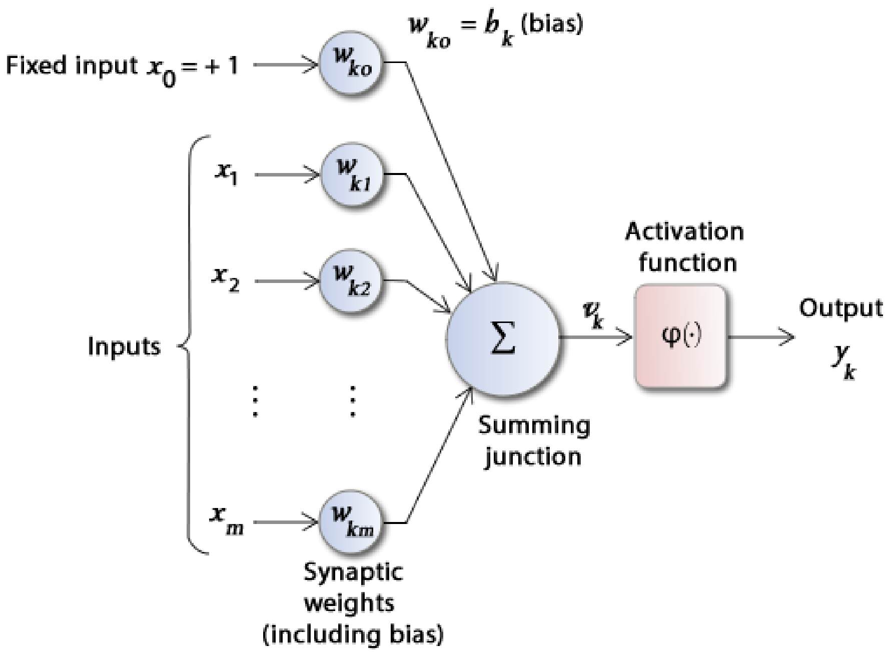

Figure 2.7: Nonlinear Model of a neuron.

on whether is positive or negative, respectively. This bias is implemented as an input signal (x_0) fixed at +1 and a synaptic weight equal to the bias b_k.

Using the bias b_k applies an affine transformation to the output u_k of the linear combiner shown in figure 2.7, as shown by:

$$v_k = u_k + b_k \tag{2.1}$$

Depending on whether the bias is positive or negative the relationship between the induced local field or activation potential of neuron k and the linear combined output u_k is modified as shown in Figure 2.8

The next equations describe the neuron k in mathematical therms:

$$u_k = \sum_{j=0}^{m} w_{kj} x_j \tag{2.2}$$

and

$$y_k = \varphi(v_k) = \varphi(u_k + b_k) \tag{2.3}$$

where $x_1, x_2, ..., x_m$ are the input signals; $w_{k1}, w_{k2}, ..., w_{km}$ are the synaptic weights of neuron k; u_k is the linear combiner output due to the input signals; b_k is the bias; $\varphi()$ is the activation function; and y_k is the output signal of the neuron.

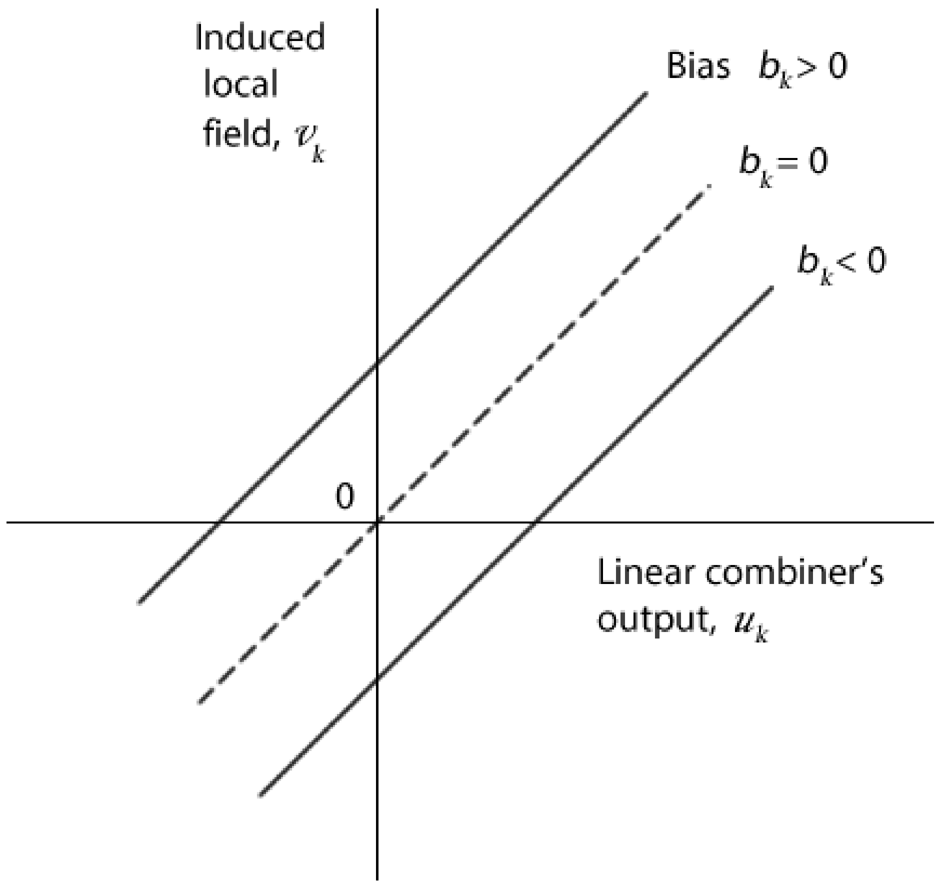

Figure 2.8: Affine transformation produced by the presence of a bias; note that $v_k = b_k$ at $u_k = 0$.

2.3.3.1 Activation Function

The activation function, denoted by $\varphi()$ defines the output neuron in terms of the induced local field v. There are different types of activation function (for example: Threshold Function or Piecewise-Linear Function), but the sigmoid function is by far the most common form of activation function used in the construction of artificial networks [36]. The logistic function is an example of a sigmoid function and is defined by:

$$f(x) = \frac{1}{1 + e^{-av}} \tag{2.4}$$

where a is the slope parameter of the sigmoid function, varying this parameter we obtain sigmoid functions of different slopes, as shown in Fig 2.9. One of the main advantages of this type of functions, is that it is differentiable. Differentiability is a requirement of an activation function when using the backpropagation algorithm as we will see later.

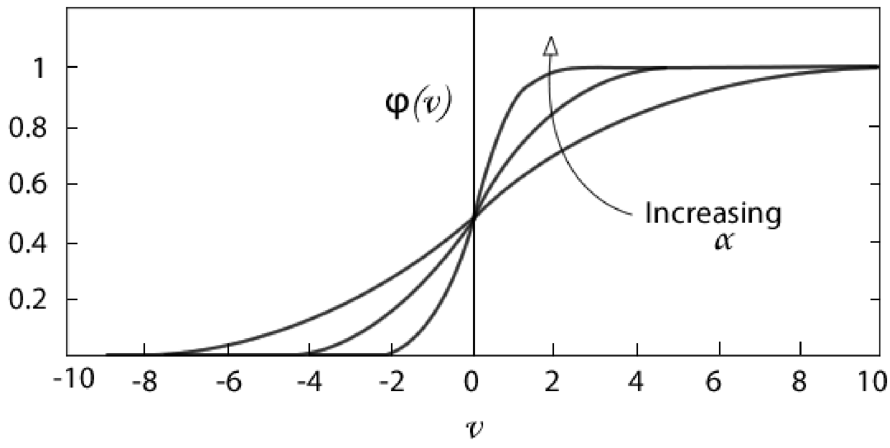

Figure 2.9: Sigmoid function for varying slope parameter a.

2.3.4 Multilayer Feedforward Perceptron Networks

Multilayer feedforward perceptron networks are one of the simplest but important class of neural networks. They have been applied successfully to solve some difficult and diverse problems by training them in a supervised manner with a highly popular algorithm known as the "Error Back-propagation Algorithm". They consist typically of a set of sensory units (source neurons) that constitute the *input layer*, one or more *hidden layers* of computation neurons, and an *output layer* of computation neurons. Figure 2.10 shows the architectural graph of a multilayer perceptron network with two hidden layers and an output layer.

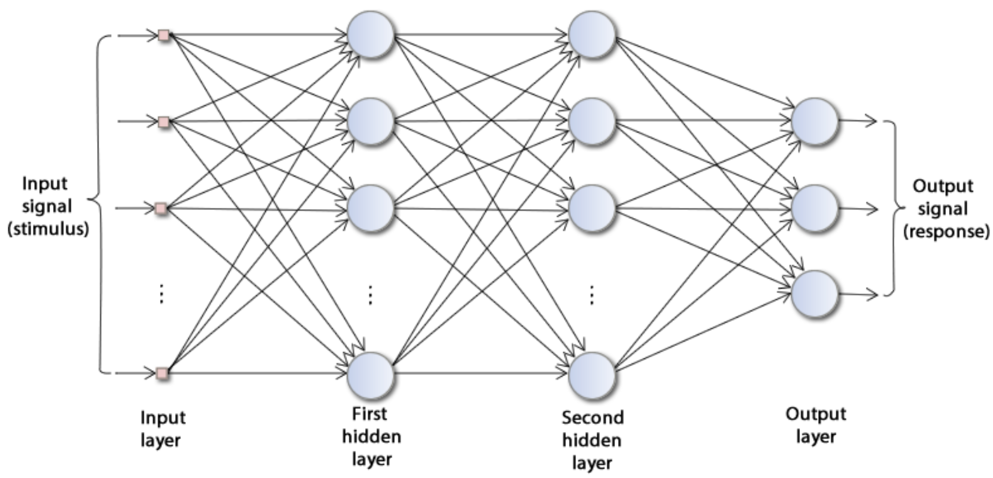

Figure 2.10: Architectural graph of a multilayer perceptron with two hidden layers.

A multilayer perceptron has three distinctive characteristics:

1. The model of each neuron in the network includes a nonlinear activation function (activation function). It is important to emphasize that the nonlinearity should be smooth (i.e. differentiable everywhere). As we mentioned before, a commonly used form of nonlinearity is the logistic function (see eq. 2.4). The presence of nonlinearities is important because otherwise the input-output relation of the network could be reduced to that of a single-layer perceptron, also, the use of the logistic function is biologically motivated, since it attempts to account for the refractory phase of real neurons.

2. One or more layers of hidden neurons that are not part of the input or output of the network. These hidden neurons enable the network to learn complex tasks by extracting progressively more meaningful features from the input patterns (vectors).

3. The network exhibits a high degree of connectivity, determined by the synapses of the network. The network shown in Figure 2.10 is fully connected. This means that a neuron in any layer of the network is connected to all the neurons in the previous layer.

It is through the combination of these characteristics together with the ability to learn from experience through training (supervised learning) that the multilayer perceptron derives its computing power. The development of the back-propagation algorithm represents a landmark in neural networks in that it provides a computationally efficient method for the training of multilayer perceptrons.

2.3.4.1 Error Back-propagation

Error back-propagation learning consists of two passes through the different layers of the network: a forward pass and a backward pass. In the forward pass, an activity pattern (input vector) is applied to the sensory nodes of the network, and its effect propagates through the network layer by layer. Finally, a set of outputs is produced as the actual response of the network. During the forward pass the synaptic weights of the networks are all fixed. During the backward pass, on the other hand, the synaptic weights are all adjusted in accordance with an error-correction rule. Specifically, the actual response of the network is subtracted from a desired (target) response to produce an error signal. This error signal is then propagated backward through the network, against the direction of synaptic connections, hence the name "error back-propagation". Now we will present the detailed steps for this algorithm:

1. Initialization: Pick the synaptic weights and thresholds from a uniform distribution (random values, often small in the range of (0-1)).

2. Presentations of Training Examples: Present the network with the complete set of training examples (commonly known as an “epoch”). For each example in the set, perform the sequence of forward and backward computations described under points 3 and 4, respectively.

3. Forward Computation: Let a training example in the epoch be denoted by $(x(n), d(n))$, with the input vector $x(n)$ applied to the input layer and the desired response vector $d(n)$ presented to the output layer. Compute the induced local fields and function signals of the network by proceeding forward through the network, layer by layer. The induced local field $v_j^l(n)$ for neuron j in layer l is:

$$v_j^l(n) = \sum_{i=0}^{m} w_{ji}(n) y_i^{l-1}(n) \tag{2.5}$$

where $y_i^{l-1}(n)$ is the output signal of neuron i in the previous layer $l-1$ at iteration n and $w_{ji}(n)$ is the synaptic weight of neuron j in layer l that is fed from neuron i in layer $l-1$. If neuron j is in the first hidden layer (i.e. $l = 1$), we set the value of $y_j^0(n)$ (output neuron j in previous layer) to the input vector $x(n)$, expressed in mathematical terms:

$$y_j^0(n) = x_j(n) \tag{2.6}$$

If the neuron j is in the output layer (i.e. $l = L$ L : depth of the network), set the value of $y_j^L(n) = o_j(n)$ Then we compute the error signal:

$$e_j(n) = d_j(n) - o_j(n) \tag{2.7}$$

where $d_j(n)$ is the jth element of the desired response vector $d(n)$

4. Backward Computation: Compute the δs (i.e. local gradients) of the network. If neuron j is in the output layer L:

$$\delta_j^l(n) = e_j^L(n) \varphi'(v_j^L(n)) \tag{2.8}$$

Or if the neuron j is in hidden layer l:

$$\delta_j^l(n) = \varphi'(v_j^l(n)) \sum_k \delta_k^{l+1}(n) w_{kj}^(l + 1)(n) \tag{2.9}$$

Where the prime in $\varphi'()$ denotes differentiation with respect to the argument. Then adjust the synaptic weights of the network in layer l accordingly to the generalized delta rule:

$$w_{ji}^l(n+1) = w_{ji}^l(n) + \alpha[w_{ji}^l(n-1)] + \eta \delta_j^l(n) y_i^(l-1)(n) \tag{2.10}$$

Where η is the learning-rate parameter and α is the momentum constant.

5. Iteration: Iterate the forward and backward computations under points 3 and 4 by presenting new epochs of training examples to the network until the stopping criterion is met. There are several measures used to decide when to stop training, we will use a combination between Maximum Epochs Reached and Generalization Set Mean Squared Error (MSE).

MSE This is the average of the sum of the squared errors ($desiredactual$) for each pattern in the generalization set.

$$MSE = \frac{\sum_{i=0}^{n}(desiredValue_i - actualValue_i))^2}{numberOfPatternsInSet} \tag{2.11}$$

Maximum Epochs Reached The algorithm will stop once a fixed number of epochs have elapsed.

This chapter has presented an overview of the literature considering Flow in gaming, embodied interaction, familiar challenges and adaptive challenge in gaming. All these concepts are necessary to understand the hypotheses that will be introduced in the following chapter.

Chapter 3

CHALLENGES IN AN EMBODIED VIEW OF FLOW AND THE ADAPTATION SYSTEM

In this chapter we will detail the hypotheses that were tested and the way they were addressed. We devised an experiment using an off-the-shelf commercial game with movement based interaction that investigated the importance of balance and integration in composite challenges.

We will also detail the idea behind the proposed adaptive system and discuss the experiment that tested its performance.

3.1 Research Question

Although the importance of challenge in order to produce good experiences for the players, has been investigated before, most computing studies of this concept have considered challenge as cognitive mainly. With the introduction of new devices capable of detecting the player's movements, the physical factor has an impact in how players perceive the difficulty of the game because they have to use more of their body than just their hands (like in traditional interfaces). When considering physical aspects, challenges could be composite in the sense that they could comprise physical and intellectual elements.

In this work, we investigated: the importance of balance in the composite challenges in promoting flow episodes; the way a loosely integration of the composite challenges impact on the promotion of flow episodes and on the perception of the players; and how unfamiliar challenges and motivated unfamiliar challenges affect the experience of players in the context of flow. The results obtained can contribute to answer the following question:

What are the characteristics that the challenges presented in a movement interaction game must have in order to promote states of effortless attention or flow?

3.2 Hypotheses

Now we will detail each one of the hypotheses proposed.

3.2.1 Appropriate Integration of Challenges

When the integration of the elements of a challenge is done without much consideration on how each element (physical and intellectual) relates to the gameplay and how players perceive them, we say it was integrated in a loosely way. We hypothesize that a tightly integration of challenges (each element is closely integrated to the gameplay and also relevant to successfully play the game) will be more likely to promote flow than a loose integration.

3.2.2 The Difficulty-Skills Balance

Maintaining a balance between the perceived difficulty of the game and the own player's perceived skills has been recognized as the most important condition to promote flow states. Our hypothesis is that an adaptive system with an optimization approach could maintain such balance.

3.2.3 Combination of Challenges

The experiment conducted by Romero and Calvillo-Gamez[59] used a challenge of kinesthetic memory with two different versions, a balanced and an unbalanced version and their results suggests that the balance in challenges is essential to promote states of effortless attention. Kinesthetic memory is a challenge that covers both the intellectual and physical parts, because it requires the player to first remember a sequence of movements and then use his/her body to imitate the posture. However, not all challenges can cover both parts, there are specific challenges that demand either cognitive abilities or physical interaction. We hypothesize that a challenge that favors the intellectual part could be integrated with another that favors physical interaction in order to balance the composite challenges.

3.2.4 Familiar and Unfamiliar Challenges

There are recurrent challenges for each genre of video games and players are familiar to them in the degree of their own experience with that particular genre. With the introduction of movement based interaction to the video games industry, games bring possibilities of new challenges. Considering the potential of natural interfaces, those new challenges could either be completely new to players or they could resemble real life movements and be intuitive to players. The hypothesis put forward is that when the gameplay involves non intuitive unfamiliar challenges, they could be properly motivated so they would be more likely to promote optimal experiences in players.

3.2.5 Addressing the Hypotheses

In this work we will try to answer the first general question by testing each one of the hypotheses mentioned. Particularly:

1. For the first hypothesis we investigated the ability to promote flow in a balanced but loosely integrated game and how this loose integration affects player experience.

2. We devised an adaptive system that uses a neural network, with the objective of balancing the difficulty of the game with the player's performance, while trying to maximize both.

3. For the third hypothesis we took a combination of challenges, a physical challenge (avoiding obstacles by dodging, ducking and jumping as presented in Kinect Adventures: Reflex Ridge) and a cognitive challenge (coordination, as presented in the PacMan minigame from Body and Brain Connection).

4. We integrated the challenges described in 3) with the game play of a popular game (Child of Eden) for the kinect that implements familiar challenges to many players (it is a shooter, one of the most well-known and popular subgenre of video games), like creating a strategy in how to attack the enemies, controlling the shooting reticles and avoiding damage from the enemies. We compared this version with an unbalanced version of the game (by taking away the coordination element) and finally we introduced a narrative that justifies the coordination element and gives background to the game in order to investigate if this has an impact on the experience of the players. This could help to characterize how non intuitive unfamiliar challenges could be introduced in order to promote flow states in movement interaction games.

3.3 Preliminary Studies

The hypotheses were evaluated empirically. To evaluate the first one, an experiment using an off-the-shelf commercial game was devised. For the evaluation of the second hypothesis a second experiment took place, we developed specifically a game prototype with the adaptive system proposed. Finally, for the third and fourth hypothesis a movement interaction game with a balanced and tight integration of the composite challenges was developed (reported in Chapters 4 and 5). This game had also the improved adaptive module result of testing the second hypothesis. All this process took place incrementally. The first steps were:

- Determining if games based on natural interaction could promote flow states.
- Starting an initial approach to the characterization of the challenges involved in games with natural interaction.
- Proposing a solution to the problem of the balance between the difficulty of the game and the player's skills.

For the first experiment participants compared two types of gaming sessions, one with an unbalanced and the other with a balanced challenge. The gaming session with the unbalanced challenge employed a physical challenge mainly and the other employed a challenge with physical and cognitive elements, however the cognitive aspect was loosely integrated.

For the second experiment we implemented a prototype that implemented an adaptive module using neural networks. Here participants were asked to play this prototype and then they compared subjectively the difficulty of the game with their own perceived skills. We also kept a record of their score. The other objective of this prototype was to learn about the possibilities and limitations of a developed application using the Kinect Sensor.

3.3.1 Composite Challenges in a Loosely Integrated Activity

For our first experiment we used the Kinect Adventures: Reflex Ridge mini-game. In this game the challenge is mainly physical so we integrated a cognitive challenge (a mental arithmetic operation while playing the game). However, this integration was done in a loosely way, because the integration of the cognitive challenge was not related to the gameplay.

3.3.1.1 Experimental Design

The study comprised two parts, the first had a between subjects and the second a within subjects design. In the first part the independent variable was the type of composite challenge (a balanced or an unbalanced session), while the dependent variables were the players' change in state and their gaming experience (both of them self-reported using a questionnaire). In the second part the independent variable was the order of sessions (balanced first and unbalanced second or the other way round), while the dependent variable was the comparison of gaming experience (again self-reported with the use of a questionnaire). With this design, the player's experience was assessed from two perspectives:

1. As an implicit comparison of the balanced and unbalanced versions. Here we were interested in determine if the game promoted flow states, and if this was the case, determine whether the balanced or unbalanced were responsible for this effect.

2. As an explicit comparison of the balanced and unbalanced versions. Here we were interested in determine, according to the participants' opinion, which condition was more conducive to higher levels of concentration and to lower levels of effort.

The explicit comparison, by focusing on a specific aspect of the flow state, could corroborate the results of the implicit comparison.

3.3.1.2 Procedure

In line with the experimental design the procedure of the study comprised two parts. In the first part participants were asked to fill in a questionnaire to record their state, then they played the game (either the unbalanced or balanced version, randomly assigned) and filled in the state questionnaire again plus a playing experience questionnaire. The motivation behind recording the players' state twice was to assess their change in state as a result of playing the game. The second part took place immediately after that. Here, participants were asked to play the game again but this time with the alternative version (balanced if the first one was unbalanced and vice versa). Finally they were asked to fill in a questionnaire that explicitly compared their playing experience over the two conditions. In both parts the gaming session lasted for about 10 minutes.

3.3.1.3 Participants and Materials

There were 38 participants, 12 female (32%) and 26 male (68%). Some of the participants were undergraduate students at UNAM but the majority of them were not university students. Their academic level ranged from secondary school to university and their ages from 15 to 47 years. Most of them did not have any experience playing Kinect games nor the Reflex Ridge mini game.

The experiment compared two types of gaming sessions, one with an unbalanced and the other with a balanced challenge. The unbalanced version consisted in playing the game as intended, while the balanced version included a request to play the game and simultaneously count the number of obstacles that were successfully avoided. Keeping a count of the avoided obstacles did not have a real impact on the game, so the integration of the cognitive challenge is considered loosely.

The unbalanced version comprised a challenge that was mainly physical, while the balanced version comprised a challenge in which the physical and cognitive elements had a more similar relevance, although those elements were loosely integrated. The questionnaires employed in the experiment contained questions commonly included in flow studies[13][20][31][45]. There were three questionnaires, the first one tried to ascertain the emotional state of participants, the second their playing experience and the third explicitly compared the playing experience over the two conditions. All of the questionnaires items had a Likert scale. The state questionnaire comprised items to assess the player's level of alertness, attention, awareness and happiness among other mood characteristics. The playing experience questionnaire comprised items to ascertain whether the player had experienced flow. Crucially, this questionnaire tried to assess whether the player's experience had been one of effortless or effortful attention by asking them *how much did you concentrate?* and *how hard was it to concentrate?* Additionally, the questionnaire also assessed the player's involvement, merging of action and awareness, level of excitement, self-consciousness, enjoyment and the player's perception of the overall level of challenge of the game. Finally the comparison questionnaire was similar to the playing experience one but instead of asking for a single playing experience it asked the player to compare the two playing sessions (asking for example *in which of the sessions did you concentrate the most?* and *in which of the sessions was it harder to concentrate?*).

3.3.1.4 Results

The results of the study can be divided into those related to the change in players' state, those associated with their playing experience and those related to their comparison of the game versions. Almost all of the data did not comply with normality assumptions so the analysis was non-parametric mostly (the only exception was the analysis of the level of challenge).

The analysis for the players' state compared their change in state (the difference between before and after playing the game) for the two experimental conditions. All of the items registered a positive change (the after scores were consistently higher than the before scores), but there was no significant difference between the conditions for any of the items. This result suggests that although the players perceived an improvement in their state (they were more alert, more aware, more active, more energetic, more excited, more happy and more sociable after playing the game), the perceived magnitude of the improvement was similar for the two conditions. Regarding playing experience, according to Csikszentmihalyi and Nakamura (2010)[20], in a model assuming effortless attention the level of concentration should be inversely correlated with concentration effort and self-consciousness and directly correlated with involvement, merging of action and awareness, level of excitement and intrinsic motivation. Considering the data for both experimental conditions, we found this to be the case. The scores for the level of concentration (those associated with the question *how much did you concentrate?*) indeed correlated inversely with those of concentration effort (those associated with the question *how hard was it to concentrate*) and with self-consciousness; and directly with involvement, merging of action and awareness, level of excitement and intrinsic motivation (see Table 3.1). When we investigated the correlations between level of concentration and concentration effort for each of the conditions (unbalanced and balanced sessions), we found the correlation to be significant only in the balanced case ($r(18) = -.609, p < .05$). This suggests that the balanced condition was responsible for the effect and therefore for the promotion of effortless attention in the game.

It is also worth noting that the item of the questionnaire related with the experience of losing awareness of everyday life while playing the game was not part of the significant correlations. Additionally, the perceived level of challenge was not significantly different neither between both conditions nor from the commensurate level value (the middle value of the Likert scale), which means that participants overall did not think that the game was either too difficult or too easy. Regarding the explicit comparison of playing experience, Figures 3.1 and 3.2 illustrate the comparisons for concentration level and concentration effort (associated with the questions *in which of the sessions did you concentrate the most?* and *in which of the sessions was it harder to concentrate?*, respectively). In both cases, the graphs represent the histograms of the 9 point likert-scale questionnaire items (which were normalized to the range -1 to 1, with the middle point zero meaning no preference). In the case of concentration level (Figure 3.1), there were significant differences between the conditions, this is, there were

		Concentration
Effort	Correlation	-0.519*
	Sig.	0.001
Involvement	Correlation Sig.	0.516*
	Sig.	0.001
Merging of Action and Awareness	Correlation Sig.	0.321*
	Sig.	0.049
Excitement	Correlation Sig.	0.536*
	Sig.	0.001
Self-Consciousness	Correlation Sig.	-0.444*
	Sig.	0.005
Intrinsic Motivation	Correlation Sig.	-0.423*
	Sig.	0.008

Table 3.1: Spearman Correlations for flow indicators. N=38, correlation is significant (2-tailed) at the 0.05 level (*)

differences depending on which session was played first ($z(-3.192), p < .05$). The left hand side of the graph represents the balanced-first condition while the right hand side shows the unbalanced-first. In both cases, the upper part of the figure is for the balanced preference while the lower part is for the unbalanced. According to the graph, those who started with the unbalanced version thought they had concentrated the most with the other version (the balanced one), while those who started with the balanced version had a slight preference for the unbalanced one. This result seems to indicate that there were learning effects, participants thought they concentrated the most in the second session. Regarding the concentration effort (Figure 3.2), there were no significant differences between the conditions. According to the graph, both of them thought it was more difficult to concentrate with the balanced version of the game.

3.3.1.5 Discussion

The balanced challenges experiment evaluated the importance of balance in composite challenges by comparing a balanced with an unbalanced activity in terms of their potential to promote states of effortless attention or flow. The balanced activity was loosely integrated in the sense that some of the aspects or elements of the challenge it comprised were only loosely related with it. Keeping a count of a score, although challenging from a cognitive perspective, was not part of the game itself and perhaps could be considered as extraneous to the task by the participants. The main result of the study suggests that the balanced activity

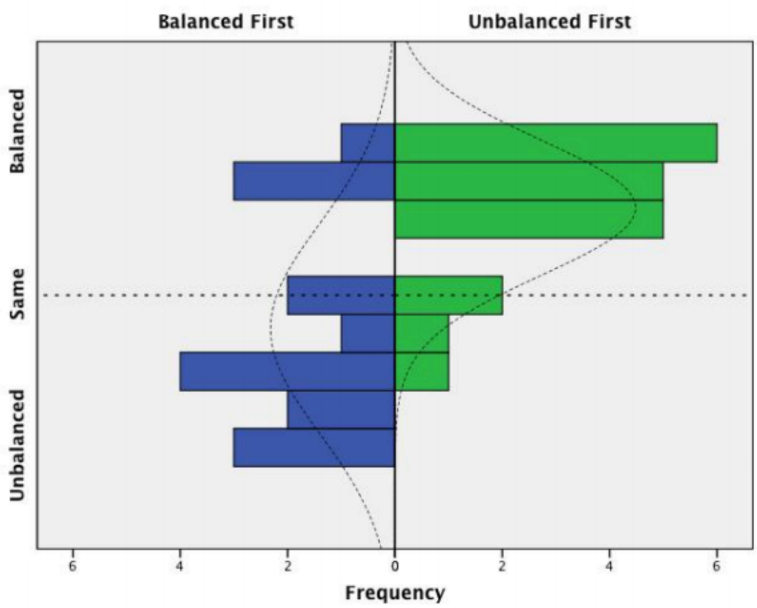

Figure 3.1: Comparison for the level of concentration between the balanced and unbalanced versions.

promoted states of flow but it was not clear which activity promoted the highest levels of concentration more frequently.

The results for the player's state suggest that overall players considered that their state improved after playing the game, this is, that they were more alert, aware, active, energetic, excited, happy and sociable after playing the game. However, there was no difference associated with the game versions; in other words, the balanced and unbalanced activities had similar outcomes regarding the players' state. This suggests that, at least in this case, the relative importance of the elements of challenge was not a relevant aspect when determining the change of state for a player.

The results for playing experience suggest that, although overall the data for both conditions indicated that episodes of high concentration tended to be associated with states of flow, the balanced activity seemed to be mainly responsible for this effect. In other words, keeping a mental count of the score tended to promote episodes of effortless attention. However, when comparing the games explicitly, participants considered that the session in which they had to keep a mental count also tended to promote the highest levels of concentration effort. This is not surprising if we consider that keeping a count of the score was extraneous to the game and therefore potentially distracting.

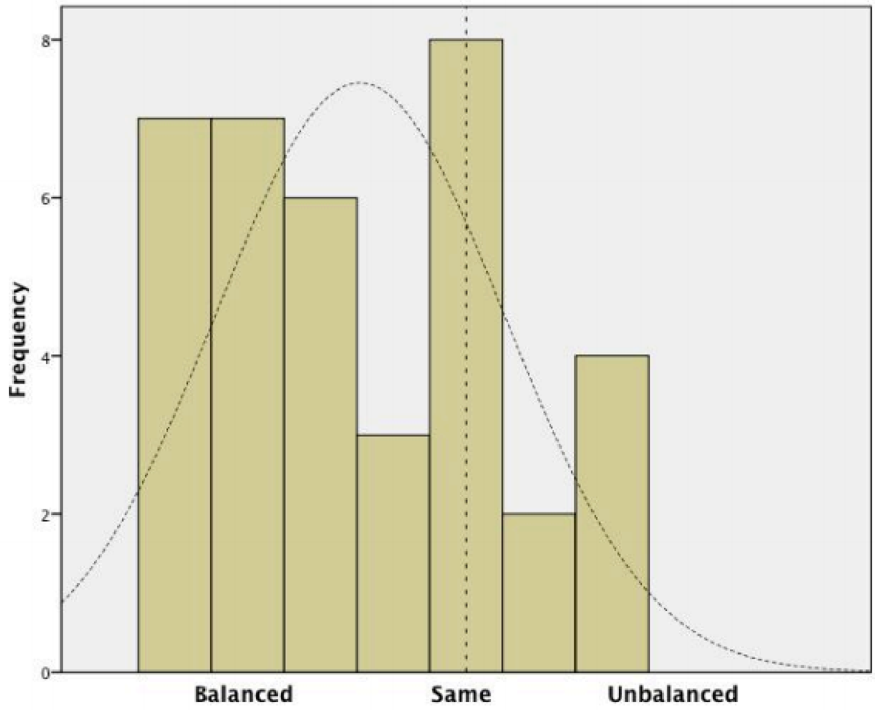

Figure 3.2: Comparison for the concentration effort between the balanced and unbalanced versions.

3.3.2 The Prototype and the Adaptation System

Regarding the second experiment, we decided to make a prototype based on Child of Eden that implemented an adaptive system based on a neural network. The generalization, nonlinearity and adaptivity advantages of the neural networks (see Section 2.3.2) are the main reasons to use this approach. Child of Eden was one of the most successful games using the kinect sensor of the 2011 year. It has been described in the following terms:

> *It's rare for me to play a game and feel nothing but happiness, but that's what happened when I stepped in to save Eden. In a way, Child of Eden touched my soul. Yes, that sounds cheesy, but it's true. This is a game everyone should experience, especially with Kinect[64].*

We also believe that Child of Eden is a good example of games with natural interaction capable of promoting flow states. As described in section 2.2.2.5, Child of Eden involves shooting various objects that come onto the screen, using the hands to control the reticles. The movement trough the world is automatic and restricted to a predefined path. This is in line with the appearance of the enemies and their movement, every level is based on a carefully planned script. Trying to implement that on our prototype would require a lot of planning and level design, so we had to simplify that design as we will see later.

The first decision that we had to make regarding the development environment was the library that we would use for the Kinect integration. At that time (October 2011) there were different options to do this, like OpenKinect [4], OpenNI [5] and Kinect for Windows SDK [3]. We decided to go with Kinect for Windows for the following reasons:

- Cost: Although it is closed source, it was free for non commercial use. And as we did not plan to commercialize the game, we could use it freely.

- It was developed by Microsoft just like the hardware, so they should know internal information about the device that the open source society must reverse engineer. They also invested a lot of money into this device (they destined an advertising budget for the launch of Kinect of US$500 million dollars[56]) so we thought that they will do what they felt necessary to keep their SDK up to par.

- End user environment: As our objective was to test the game in a controlled environment, we would not have to worry about compatibility problems in other computers. We would only need to make sure the game would work in our computer, so the multi platform advantage of the open source options was not of much use for us.

The Kinect for Windows SDK could be used with two programing languages, C++ and C#. Considering those programing languages, there were several options for the graphics library (from low level graphics library like OpenGL or DirectX to more high level libraries like OpenSceneGraph, XNA or Ogre), we chose C# and XNA because:

1. XNA was developed to make game programming easy and take out tedious low level coding[35] (compared to OpenGL or DirectX where tasks like rendering a model or loading a texture require a lot of work).

2. Code examples can be found on a vast number of sites[35]. In the XNA Creators Club Online Web Site we have access to a huge repository of code, models, textures and other samples.

3. There was already a project[25][24] that made an integration between XNA and the Official SDK for Kinect. The programing language common to XNA and Kinect for Windows is C#.

3.3.2.1 Designing the Gameplay

We designed the first prototype to implement the most basic shooting system of Child Of Eden (the tracer), we used spherical enemies in order to simplify the collision detection and a simple movement algorithm for the enemies. Also the view vector of the camera and the position of the player were fixed. We implemented waves of enemies that appear in front of the player and approach him/her with a specific speed (determined by the adaptive system as we will see later), when the wave finishes we check how many collided with the player to determine the damage received.

The objective of the game is to destroy as many enemies as possible, aiming with the hands while trying to receive as little damage as possible. There is a reticle for the right hand that is red and another for the left hand that is blue. There are two types of enemies, red and blue. Red enemies can only be destroyed by aiming with the red reticle and blue ones with the blue reticle. See fig. 3.3.

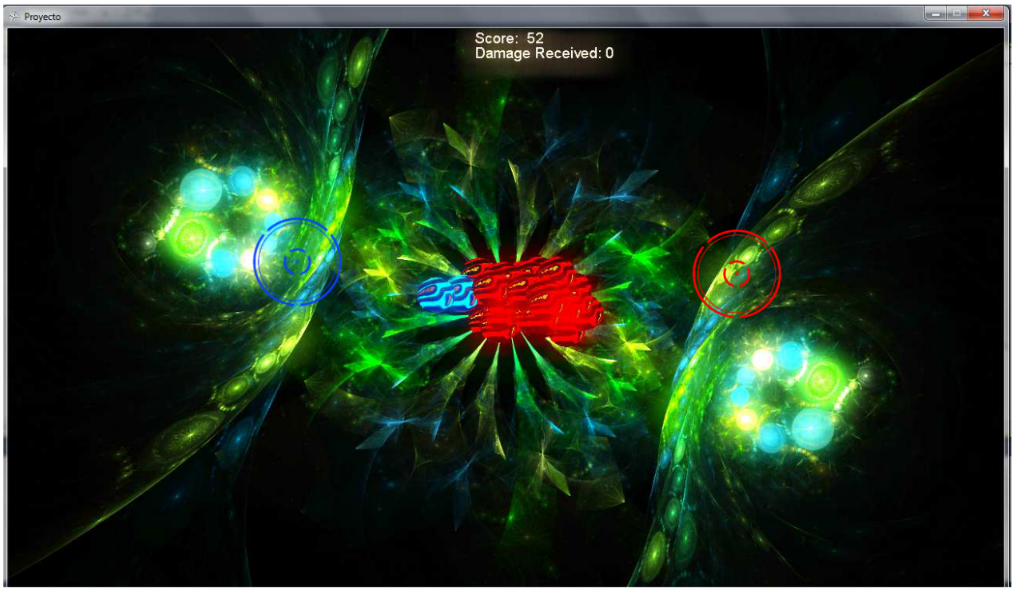

Figure 3.3: Screenshot for the first prototype.

3.3.3 The Adaptive Module

In order to promote flow states there has to be a balance between the perceived challenges of the task at hand and the player's skills (see Section 2.1). To fulfill this requirement we proposed an adaptive system based on a neural network in charge of regulating the difficulty of the game depending on the player's performance.

There are two components that are essential to an adaptive system[6]:

1. A performance evaluation
2. An adjustment mechanism

Determining the time and frequency to the adjustment phase is very important. Applying frequently the adjustments to the settings when having an ongoing play sessions may favor to approximate the optimal settings for a player, but if the time window is too short, there may not be too much information to learn. We implemented “waves” of enemies and the adjustment phase is applied after each round. There are existing projects which have focused on movement-based games with the purpose of rehabilitation or therapy that have identified three generic difficulty parameters:

1. Accuracy[34].
2. Speed[6] [12] [34].
3. Range of motion (or amplitude).

Matching the first two parameters with our game design, we can identify them with the size of the enemies and the speed at which they move. However, as we are not trying to impose a range of motion to the player for therapeutic goals, the range of motion should be determined just by the body size of each player. We also defined some other parameters that will impact on the difficulty of the game, so the final set of parameters is:

- Size of the Enemies.
- Speed of the Enemies.
- Probability for an enemy to be red.
- Total number of enemies in a wave.

Now that we have a set of parameters that define the difficulty of the game, the idea behind the use of the neural network is as follows. There will be waves of enemies that will take those parameters, after the wave has ended, we measure the performance of the player (the performance evaluation phase) and we train the neural network with that information. Then, for the adjustment phase, we use the neural network to calculate the performance to several (as much as we can) different combinations of patterns and we keep the best one, then we present it to the player and retrain the network with this new information.

We defined 2 formulas, one would evaluate the performance of the player by taking into account the number of enemies destroyed, the average time to destroy an enemy, the damage received, the speed of the enemies and the number of enemies on the screen (all values are normalized).

$$f_1 = NE * NS * (pCR(rED) - aTR) + ((1 - pCR)bED - aTB) - dR \quad (3.1)$$

Where:

- NE: Normalized value for the total number of enemies of a wave. 0 corresponds to 10 enemies and 1 to the max number of enemies in a wave (100).
- NS: Normalized value for the speed of the enemies.
- pCR: Probability for an enemy to be red.
- rED: Number of red enemies destroyed. Normalized respect to the total number of red enemies.
- aTR: Average time in destroying a red enemy. Calculated by counting all the time that took the player to destroy all red enemies divided by the total number of red enemies (if the player didn't destroy all red enemies then we take the time of the wave and divide it by the number of red enemies destroyed).
- bED: Number of blue enemies destroyed. Normalized respect to the total number of blue enemies.
- aTB: Same as aTR but for blue enemies.
- dR: Normalized value for the damage received from both types of enemies. A value of 0 corresponds to no damage and 1 corresponds to damage equal to the full hp of the player.

The other formula would evaluate a "fun" factor considering just the number of enemies destroyed without a penalization for a bad performance (damage received and time to destroy the enemies).

$$f_2 = pCR(rED) + (1 - pCR)bED \quad (3.2)$$

Then in order to evaluate the combination of parameters and the player's performance we used a weighted sum for the value of each formula.

$$0.7(f_1) + 0.3(f_2) \quad (3.3)$$

After that we defined a large range for the parameters, for example, in the case of the speed of the enemies, on the lower limit the enemies should move very slow and on the upper limit they move very fast. These parameters became the input neurons for our neural network and the value of the formulas the output neurons. We considered a supervised learning approach.

The values of the ranges are listed below:
Size Range: [1, 10]
Speed Range: [0.0000001, 0.3]
Total Enemies Range: [10, 100].

There was an initial training phase, that consisted of the following steps: (a) generate a random input pattern, present it to a human player for an amount of time and then calculate the values of the output neurons using the formulas described. (b) Use this data as a training vector for the neural network. (c) create another random pattern and repeated the process until the performance of the network was good enough (when the non playable combinations rarely occur).

After this first phase of training, the game was ready to be presented to the final players. Then the second phase of learning would take place. Before each wave, the neural network created 444 random input patterns (max number possible without noticing lag between frames) and calculated the expected values of the output neurons, then it presented the best pattern (the one with the biggest value in Equation 3.3) to the player. After a wave is over, the network was retrained with the actual data obtained from the player's performance. Then the cycle would be repeated constantly so it would keep adapting to changes in the player's performance.

Looking back at equation 3.1, we can see that its value cannot be larger than 1.0, and the only way to obtain 1.0 is to play with the maximum difficulty settings and get no damage, and take 0.0 seconds in destroying all the enemies which is impossible. We could aim to obtain the best combination of parameters that maximizes the weighted sum 3.3. However, when trying to promote flow is not the objective competence, but the participant's own perception of competence that matters[19]. This makes our job a little easier because we are not looking for the best combination of parameters, instead we are looking for one that is "good enough" so that the player believe him/her-self capable of fulfilling the objectives of the game. So the objective of the neural network is to learn the player performance to different patterns of the parameters and to extrapolate this performance to unknown combinations. We believe that if have enough patterns to extrapolate and we keep the best of them we could approximate to that ideal difficulty for the player.

For keeping things simple and considering that the adaptive module is not the central point of this thesis, we refer to the No Free Lunch Theorem of Optimization (NFLT). The interesting and surprising result of the NFLT for search algorithms tells that "on average all search algorithms perform no better than random search"[38]. If we cannot make some prior assumptions about the function we are working on, we cannot expect a priori to perform any better than random search and the risk we take is that it might actually perform worse[38]. Thats why we decided to generate the combinations of parameters in a random way. This is a point that could be improved in future work.

3.3.3.1 Neural Network Architecture

Now that we have defined the input neurons (size, speed, probability and total number) and the output neurons we need to determine the size of the hidden layer. We will be using the next heuristic formula:

$$(I+1)(H)+(H+1)O \approx S/3 \tag{3.4}$$

Where: I: Number of neurons of the input layer. H: Number of neurons of the hidden layer. O: Number of neurons of the output layer. S: Size of the training set. We consider this as 1000, more than enough for our purposes.

Doing the maths, we obtain that the hidden layer should have 25 neurons.

In summary, the architecture of the neural network is as follows: perceptron's based layers consisting of an input layer (4 neurons), a single hidden layer (25 neurons) and an output layer (2 neurons). We used the back-propagation algorithm as the learning method with a learning rate of 0.01 and momentum of 0.9 (both values must be inside the range of [0,1], heuristic results suggest that a combination of high momentum and low learning rate yield to a faster convergence). The activation function for the neurons is :

$$f(x) = \frac{1}{1+e^{-x}} \tag{3.5}$$

3.3.3.2 Programming the Adaptive Module

The adaptive module for the prototype consisted in three classes (see Figure 3.4). The NeuralNetwork class is an abstraction of the neural network itself. The NeuralNetworkTrainer class contains an instance of NeuralNetwork and implements the training procedure. The NeuralNetworkManager is the class in charge of the communication between the neural network and the game. It contains instances for NeuralNetwork and NeuralNetworkTrainer. For the full class diagrams of the three classes refer to Appendix A.

The game class also contains an instance of a class called EnemiesManager, this class receives the parameters calculated by the NeuralNetworkManager class (size, speed, probability and total number) and creates the lists of enemies (red and blue). It also keeps the count of time passed to determine aTR and aTB, the count of enemies destroyed (red and blue), controls the movement of each enemy and records how much damage the player received.

The whole prototype consisted of 15 classes to implement the necessary functionality (adaptation system, rendering of graphics, loading and writing files, etc.) with a total of 3046 code lines.

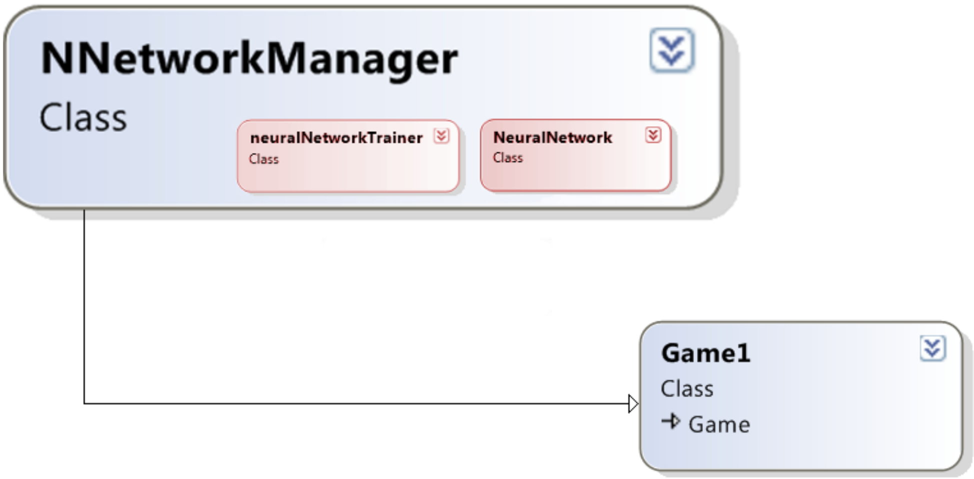

Figure 3.4: Structure of the Adaptive Module for the Prototype.

3.3.3.3 Procedure

We tested the first prototype in an empirical way using a between-subjects experimental design. Three participants were asked to play the game and then evaluated their experience related to their subjective opinion about the difficulty of the game. We also kept track of their performance according to equation 3.3 while playing the game.

3.3.3.4 Participants and Procedure

There were three participants, 2 female and 1 male. The three of them had different expertise in video games.

- Participant 1 (P1): She did not have much experience playing video games, neither with traditional interfaces (joystick or keyboard and mouse) nor movement-based games.
- Participant 2 (P2): She had medium experience with games that use traditional interfaces and also with games that use the kinect sensor.
- Participant 3 (P3): He had considerable experience playing games with traditional interfaces and medium experience with games that use the kinect sensor.

For this experiment, the 3 participants played 52 waves of enemies in the game. We kept track of the value of equation 3.3 and the weights of the neural network were saved in a csv file. The initial phase of the training was performed by the first author, who has a high amount of experience with both kinds of interfaces for video games (traditional and kinect). After the first phase concluded, the game was presented to P1, then to P2 and finally to P3. At that time, the game was coded to save the weight file of the neural network when the game session finished, so the starting point for P1, was the trained network for the first author, the starting point to P2 was the trained network for P1 and so on.

3.3.3.5 Results

In Table 3.2 we show the values of the performance evaluated with equation 3.3 for the three players and see figure 3.5 for a graph of all three of them.

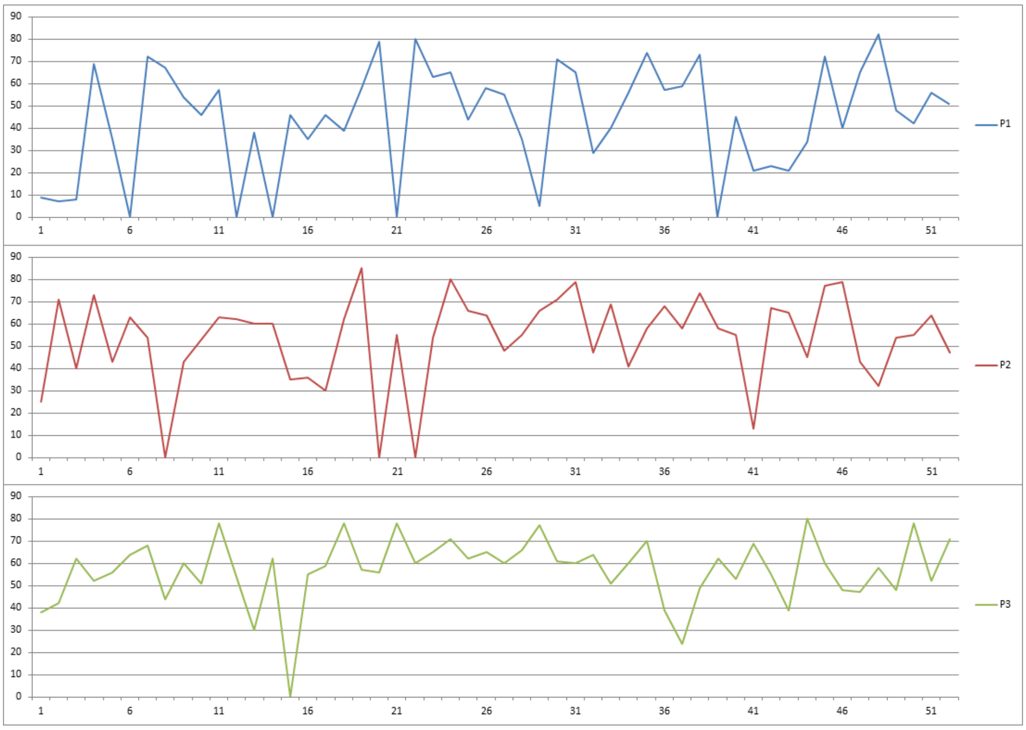

Figure 3.5: Graphics for the performance of the 3 players of the Prototype.

Finally we asked to the participants to compare explicitly the difficulty of the game with their own perceived skills, with a 10 pint likert-scale (0: too low; 5: proportional; 10: too high). The Table 3.3 shows their answers:

3.3.3.6 Discussion

Overall, we believe that the system had a very acceptable performance as the 3 players felt that the game's difficulty was slightly above their own perceived skill

Wave Number	P1	P2	P3
1	9	25	38
2	7	71	42
3	8	40	62
4	69	73	52
5	35	43	56
6	0	63	64
7	72	54	68
8	67	0	44
9	54	43	60
10	46	53	51
11	57	63	78
12	0	62	54
13	38	60	30
14	0	60	62
15	46	35	0
16	35	36	55
17	46	30	59
18	39	62	78
19	58	85	57
20	79	0	56
21	0	55	78
22	80	0	60
23	63	54	65
24	65	80	71
25	44	66	62
26	58	64	65
27	55	48	60
28	35	55	66
29	5	66	77
30	71	71	61
31	65	79	60
32	29	47	64
33	40	69	51
34	56	41	60
35	74	58	70
36	57	68	39
37	59	58	24
38	73	74	49
39	0	58	62
40	45	55	53
41	21	13	69
42	23	67	55
43	21	65	39
44	34	45	80
45	72	77	60
46	40	79	48
47	65	43	47
48	82	32	58
49	48	54	48
50	42	55	78
51	56	64	52
52	51	47	71

Table 3.2: Performance of the players evaluated with equation 3.3.

	P1	P2	P3
The difficulty level of the game compared with the level of my own abilities was:	6	6	6

Table 3.3: Explicit comparison between the level of difficulty of the game and the participants' own perceived level of skill.

level (suggesting that they felt challenged by the game) despite having different players in terms of their gaming experience.

Considering that the neural network was first trained by a user with much more experience than P1, it was expected that her performance wouldn't be so good at the beginning. We can observe 5 drops to 0 of her score, 4 of them in the first 21 waves. However, considering the score of all the waves, P1 got a mean of 44.11. For P2, there were only 3 drops to 0 and her mean for the score was 53.17. It is important to note that the starting point for P2 was the trained network after the session of P1.

For P3, there was only 1 drop to 0 and his mean was 57.07. Again, the starting point for P3 was the trained network for P2.
This suggests that although the level of expertise has an impact on the score, this impact is minimized because the neural network finds some suitable parameters for the player so her/his score wont drop dramatically (in case that the player is not very skilled) or get to the top of the score (in case that the player is very skilled).

Regarding the explicit comparison between the game difficulty and their abilities, all players answered 6, suggesting that they felt challenged by the game, but they didn't consider it too difficult.

The drops of score to 0 were the result of a combination of a high speed and a small size of the enemies. For the first phase of training (the training with the first author), this kind of combination (which is not playable) was very common to find. However, with the training of the network their frequency of appearance dropped significantly. Also, comparing the 9 non-playable waves (for the 3 players) to the 147 playable waves we can say that the behavior of the system was acceptable. In addition, the duration of an unplayable wave ranged from 1 to 2 seconds (because of the high speed of movement of the enemies), which is a short period of time.

3.3.4 Improvements

Considering the first experiment, keeping a count of the score, although challenging from a cognitive perspective, was not part of the game itself and perhaps could be considered as extraneous to the task by the participants. The main result of the study suggests that the balanced activity promoted states of flow but it was not clear which activity promoted the highest levels of concentration more frequently. The main enhancement would be to integrate a cognitive challenge with a physical challenge to a gameplay, in which both parts are important to successfully play the game (tightly integrated).

Regarding the second experiment, the following improvements were implemented for the final game:

1. After presenting the methodology in the "Seminario de Inteligencia Artificial" (Artificial Intelligence Seminar) organized by the DCC (Computer Science Department) of IIMAS[42], we received feedback about the adaptive system. We realized that the way we were using the neural network was

not the most suitable, as the network was learning the value of a known formula. Neural networks are used when the structure of the formula is unknown or it makes too difficult to calculate its value. Clearly, our case was neither of these, so we decided to change our approach on that matter. Instead of assigning an explicit formula for the output neurons, we decided to assign directly the parameters of the game that we want to predict to the output neurons.

2. It would be best to have a fixed duration for a wave, instead of it depending on the speed of the enemies.

3. The first training should be done by a person (or persons) with a medium level of experience with movement interaction games and after the training is complete, this should be the starting point for all the people that will play the game.

These improvements were incorporated into The Energos Game, which was used to test the last two hypotheses presented in Section 3.2. The Energos Game will be introduced in the following chapter.

Chapter 4

THE ENERGOS GAME

The Energos Game is the result of the analysis of the studies presented in the previous chapter. It has a tight integration between the cognitive and physical challenges present in the game and also features the improved version of the adaptive system. In this chapter, we will detail its main features (main objectives, controls and associated feedback) and its development process.

The Energos Game is a movement based interaction game developed by the author with the objective of evaluating the last two hypothesis presented in Section 3.2. It uses the kinect sensor for detecting the player's movements and combines a physical challenge (dodging enemies using movements that resemble those used in real life) and a cognitive challenge (focus attention on two screens at the same time to coordinate what happens on each one) into a shooter game (one of the most popular sub-genre of video games). This creates a combination of familiar challenges (those associated with the shooter sub-genre) a non intuitive unfamiliar challenges (dividing attention on two screens and coordinate the movements on both of them at the same time).

There are three versions of the game, each one with the same objective: To achieve the highest score in a limited time (10 minutes). This is accomplished by destroying as many enemies as possible and receiving the least damage from them. In Figure 4.1 is shown the main menu of the game (it also includes a tutorial entry).

The different versions implement familiar challenges with or without the unfamiliar challenge, either with a justification for its introduction or without it. The evaluation of those versions will try to ascertain whether familiar challenges and properly motivated unfamiliar challenges are more likely to promote optimal experiences in players considering also the balance in the composite challenges.

Figure 4.1: Main Menu Screen of The Energos Game.

4.1 Requirements

From the adaptation of flow to games in an embodied view presented in section 2.1, we can identify the next requirements for a movement-based video game that could promote flow states:

1. A balance between the perceived challenges and the skills of the player.
2. Considering the cognitive, physical and affective parts in the design of the challenges to make a balanced version of the game.
3. Clear and compatible goals.
4. Provide immediate feedback about the performance of the player.

4.1.1 The Skill-Challenge Balance

As mentioned before in Section 3.3.3, we are addressing this problem from an optimization point of view. Considering that the results of the adaptive module from the prototype (see Section 3.3.3.5) were acceptable, we decided to use the same approach but also implementing the improvements mentioned in Section 3.3.4 for The Energos Game. We will detail these changes in Section 4.3.4

4.1.2 Balanced Composite Challenges

When designing the challenges of the game and how the player should face them, it is important to consider both physical and cognitive requirements of the challenges. The gameplay of The Energos Game was designed in this way (see Section 4.2). Also, considering the results of the first study (see Section 3.3.3.5) it was important that the integration between the cognitive challenge and the physical challenge would result on both parts being important to successfully play the game (a tightly integrated challenge).

4.1.3 Clear and Compatible Goals

Goals in the world of video games vary in two kinds[61]:

- Those that have an ultimate, attainable goal (e.g. finishing the story mode of an action-adventure game).
- Those that are played for high score (the goal is to beat your previous scores, or the scores from other players).

In both cases, the player is clear of what the objective is at the beginning of play. The goal of The Energos Game is to obtain the highest score. The mechanics to increase the score are detailed in section 4.2.1.

4.1.4 Immediate Feedback

According to Sweetser[65]:

1. Games should use scores to tell players where they stand and give positive feedback to encourage mastery of the game.

2. Players should always be able to identify their score and status in the game.

3. In-game interfaces and sound can be used to deliver necessary status feedback.

For this part we implemented two kinds of feedback in the game, visual and auditive. Specifically:

- The score of the player is always shown at the top of the screen and it is updated in real time (See Figure 4.2).

- When the player receives damage from any enemy, the screen turns red and there is a sound associated with this event (See Section 4.2.1.6).

- When the player is aiming and shooting an enemy, the cursor shows an animation and a sound is played to reinforce the feeling of control (See Section 4.2.1.5).

- When the player performs a movement to avoid an enemy (ducking, jumping, stepping right or left) there is a sound indicating that it was activated and an indicator in the user interface is turned on (See Section 4.2.1.3).

- We had a board with the list of the top three scores for each version visible to all the players, so they could know the highest scores before playing and adjust their game style or strategies during the play session.

In the section 4.2.1 we will detail the full controls and show how the feedback was supplied to the players.

4.2 Game Design and Gameplay

The methodology that we followed for the challenge design was:

1. Take a (world wide) popular game for the kinect sensor as a base for the game. Such game implements mechanics that are known by the players in the degree of their own experience with video games (familiar challenges).

2. Identify a challenge where the physical part is dominant in a commercial game.

3. Identify a challenge where the cognitive part is dominant, in another commercial game.

4. Integrate those two challenges into a new game play based on the first game.

Looking at commercial games for kinect of the 2011, we identified two challenges that demanded either physical interaction or cognitive abilities.

1. Controlling the avatar with the body like in the Reflex Ridge mini game of Kinect Adventures (see Section 2.2.2.2) favors a rich physical interaction because the player has to jump, step to the right or left, and duck in order to avoid obstacles. This kind of challenge resemble real life movements, so it could be very intuitive to new players.

2. Focusing attention on two screens at the same time (as presented in the PacMan minigame of Body and Brain Connection, see Section 2.2.2.6) demand a high level of attention and high cognitive abilities. This challenge is not usual in video games and therefore not very well known by players.

Considering the results of the first prototype, we started planning the characteristics of the new application. Here we list the first objectives we had in mind:

- A full body control of the application, instead of the limited control of the prototype, which used only the hands.
- Integrate the dual screen challenge of the PacMan minigame (see Section 2.2.2.6) to the game play.
- A more complex behavior of the enemies.
- Improved quality of the graphics.
- A redesigned gameplay focused on the composite challenges that would take into account the new movements of the player, the two screens challenge, the new behavior of the enemies and a new scoring system.

Analyzing the commercial games using a kinect sensor that were on the market in the spring of 2012, we noticed that those that had a navigation trough the virtual world almost never permitted to wander freely, there was always a predefined path for the player. We assumed that this is because it simplifies the design of the game, and also it is easier for the player to control it. As we said before, the movement in Child of Eden is automatic through a defined path, but as we wanted more physical interaction we took the controls of Kinect Adventures: Reflex Ridge for the movement through the virtual world, which is also restrained to a path. Looking back at the controls of Kinect Adventures: Reflex

Ridge, we can identify four important movements for avoiding obstacles: jump, crouch, move right and move left. We also decided that the player should control the movement through the path (when to advance, stop or go back). Then we integrated the challenge of the two screens by providing split screen for forward and backward views. The game requires the player to control each view with each hand because enemies will attack in both views.

Regarding the software development we decided to follow the same line of the prototype. We used XNA 4.0[50] as the graphics library, the Kinect for Windows 1.5 SDK[3] for the Kinect libraries and Visual C# 2010 Express from Microsoft as the IDE.
For the whole physics management (collisions, apply forces to objects, zero gravity environment, ray-cast methods, etc.) we used the free physics engine for XNA Jitter Physics[2].

For the first version we defined the following game play:

- The movement is restricted to a path, and the player can control the movement through it.

- The enemies will appear in front of the player, again as waves. The duration of the wave will be 30 seconds.

- The players will use their hands to move the reticles and aim. They will also have moves like in reflex ridge to dodge the attacking enemies.

- There will be only one type of enemy for this version and it could be attacked by either hand.

- The objective of the game is to achieve the highest score. We defined that the player wont die in the game, so the session is not interrupted. Instead, the damage received will reflect directly on the score. We defined the next scoring system: The player has an amount of "energy", if this amount is full, then all the damage caused to the enemies will add to the current score. If not, then the damage caused to the enemies will be used to refill the energy instead of scoring. The player's energy diminishes when he/she is hit by the enemies. In that way, if a player is receiving too much damage, even tough he/she may be destroying many enemies, he/she will not increase the score.

- The enemies will not be active at the beginning, it will be the decision of the player to activate them when shooting. That would favor the creation of different strategies. On the one hand, the player could activate many enemies to try to obtain more points but these enemies could overwhelm the player. And few activated enemies yield fewer points, but this is safer.

For the second version, the game play would be exactly the same, but with the following differences:

- The enemies should appear at the front and the back of the player, instead of just at the front.
- There are two types of enemies, just like in the prototype. One type could only be destroyed with the right hand and the other with the left hand.

The third version is exactly the same as the second one in terms of gameplay, but before playing, we introduced a narrative that tells the story of the game, gives context of the place in which the game develops and also has a justification for having to coordinate two screens at the same time.

We designed two neural network architectures, one for each version. But when we started testing the game, we decided to remove the differentiation of the enemies of the second and third version, because some people who tested the game said that it was very complex. The final shooting system resulted as follows: The reticle of the left hand can destroy any enemy that is on the left screen, regardless of color. And the same for the reticle for the right hand but on the right screen. In the end, we used the same architecture for the neural network in both versions. This architecture is explained in section 4.3.4.

4.2.1 How to Play

In Figure 4.2 we show the game play screen of the game for the One Screen and Dual Screen versions with the name of each element displayed. These elements are described next.

1. Score: Shows the score of the player, it is updated in real time.
2. Map: It is a representation of the world, it indicates the position of the player in real time.
3. Movement Indicator: Gives feedback to the players about the activation of the dodging movements (stepping right or left, jumping or ducking). For more details, see Section 4.2.1.3.
4. Direction Arrow: This is an auxiliary design to help the player to identify and orientate inside the world. These arrows are visible in the whole world and define the "front" and "back" direction of the path.
5. Avatar: This is the character that represents the player. The player's HP is represented by the orange colored part (see Figure ?? for the comparison of full and almost empty HP).

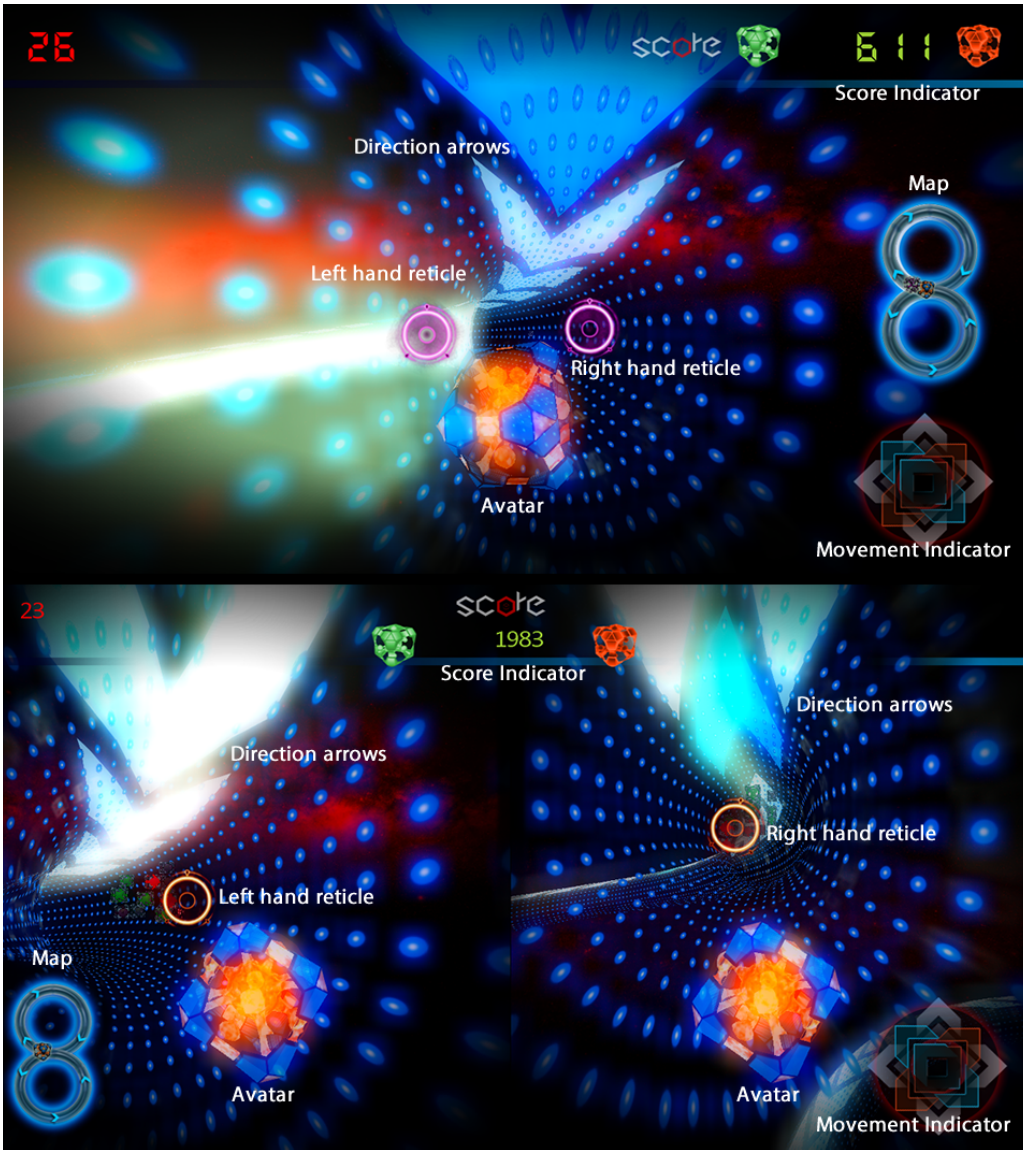

Figure 4.2: Screenshots of actual gameplay of The Energos Game for the One Screen and Dual Screen versions.

6. Right and Left Hand Reticles: The shooting reticle for the right and left hand.

Figure 4.3 shows the playing area of the game, it is a tapestry on the ground used to help the players locate the zones used to control the game (for the full explanation of the controls see Sections 4.2.1.2 and 4.2.1.3.

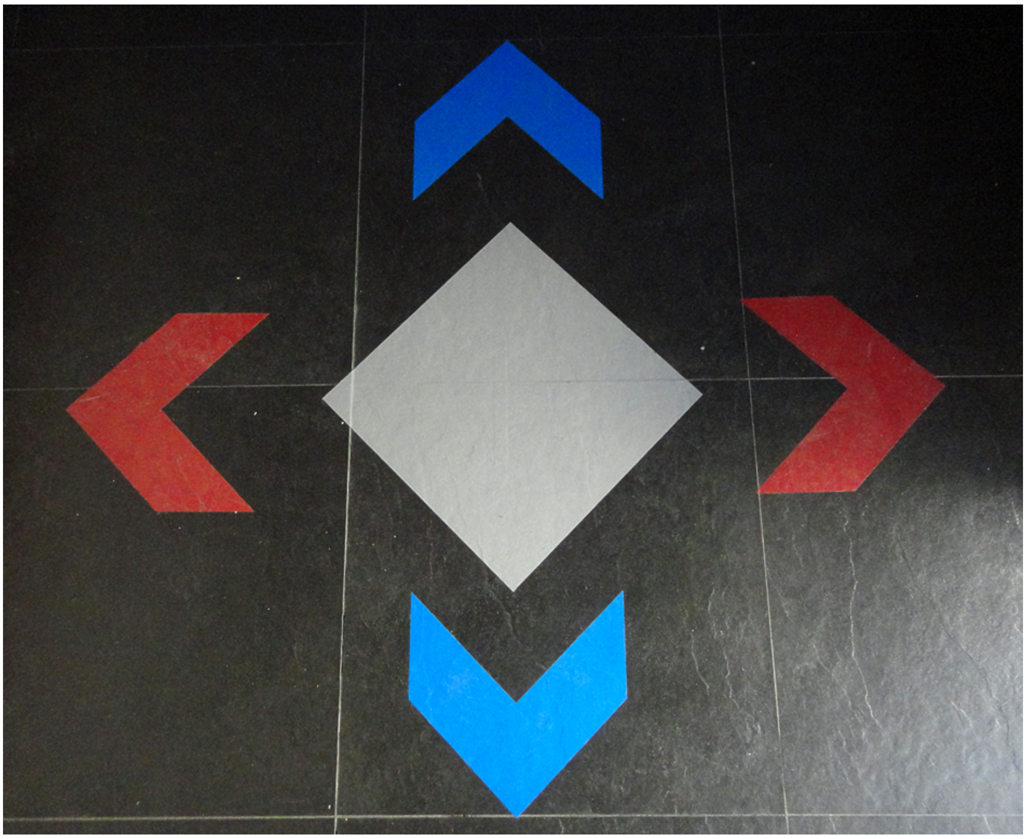

Figure 4.3: Tapestry of the game used to identify the zones where the player should stand to navigate trough the world path or dodge.

4.2.1.1 The Objective

The objective of the game is to compete for the largest score during the gaming session. As we have mentioned before, each session lasted for 10 minutes and there was a board with the top 3 scores for each version so that the player could compare her/his performance with that of the best players. As shown in Figure 4.1, there is a Tutorial entrance in the main menu. This was used to introduce the controls to the new players and the feedback mechanism. The following section explain each of the tutorial's steps.

4.2.1.2 Movement Trough the Path

The first part of the tutorial introduced the concept of moving through a path. Here, the player could either advance through the path, stand still or go back. The direction arrows shown in Figure 4.2 were introduced so that the player could always identify where is the "ahead" and "behind" direction of the path.

As shown in Figure 4.4 when the player steps on the front blue arrow of

Figure 4.4: Movements to advance (left image), stop (center image) and go back (right image) through the path.

the tapestry, the avatar advances through the path in the same direction of the direction arrows and when the player steps on the back blue arrow of the tapestry, it moves in the opposite direction. When the player stands at the center of the tapestry, the avatar stops its movement smoothly but almost instantly.

4.2.1.3 Avoiding Damage From Enemies

The objective of this section was to introduce the movements to avoid enemies. There are 4 movements to avoid enemies, steeping right or left, jumping or ducking. These movements are implemented in the same way as in Kinect Adventures Reflex Ridge (see Figure 2.4).

As shown in Figure 4.5, when one of these movements is performed by the player, there is feedback in the Movement Indicator about which movement is being activated. When the player stands on one of the red arrows (right of left) of the tapestry, the corresponding arrow in the Movement Indicator is colored red. If the player jumps or ducks the associated arrow is colored blue (up arrow for jumping and down arrow for ducking).

As will be discussed later (see Section 5), there were two evaluations of the game, first a qualitative evaluation of the system and then a quantitative evaluation. In the first evaluation (qualitative) the avatar's movements for evading enemies took as reference the predefined path, regardless of the camera orientation (For example, when the player stepped to the right, the avatar moved to the right of the front of the path and not the right of the camera). As a result

Figure 4.5: Steeping to the right and ducking to avoid enemies.

of the qualitative evaluation, we modified this control to take the reference from the orientation of the camera instead of the path, to make it more intuitive for players (we will discuss this in Section 5.1.4).

4.2.1.4 Controlling the Cursors and the Camera

When the kinect sensor detects a player, it obtains the player's body size to generate a projection matrix. This matrix was used to project the position of the player's hands to the 2D screen coordinates, so that the player could control the cursors (shooting reticles) directly by moving his/her hands. In order to turn the camera to the right, the cursor of the right hand should be taken to the right border of the screen, and to turn the camera to the left, the cursor of the left hand should be taken to the left border of the screen (see Figure 4.6).

4.2.1.5 Activating and Shooting Enemies

In order to activate an sleeping enemy (See Figure 4.7) the player has to shoot it. This was accomplished by laying any of the cursor for the One Screen Version and the corresponding cursor for the Dual Screen Version (left cursor for the left screen and the right cursor for the right screen) over an enemy. When the player is shooting an enemy, the cursor shows an animation and there is a particular sound associated with that and it takes a little less than 2 seconds of maintained fire to destroy an enemy. So the player has to coordinate and maintain her/his hands in order to destroy the enemies. The remaining hp (hit points) of an enemy can be determined by their color (see Figure 4.7). Also, the enemies' behavior

Figure 4.6: Turning the camera to the left.

depends on their remaining hp, if it is in the range of 100% to 50%, they will pursue and try to hit the player, if their HP is less than 50%, then they will flee from the player.

4.2.1.6 Receiving Damage

When an enemy is in contact with the player's avatar, the screen is colored red and there is also a sound when that happens (see Figure 4.8). The most effective way to escape when there are a few enemies is to destroy them. But if there are several enemies attacking the player it is better to perform one of the movements described in Section 4.2.1.3.

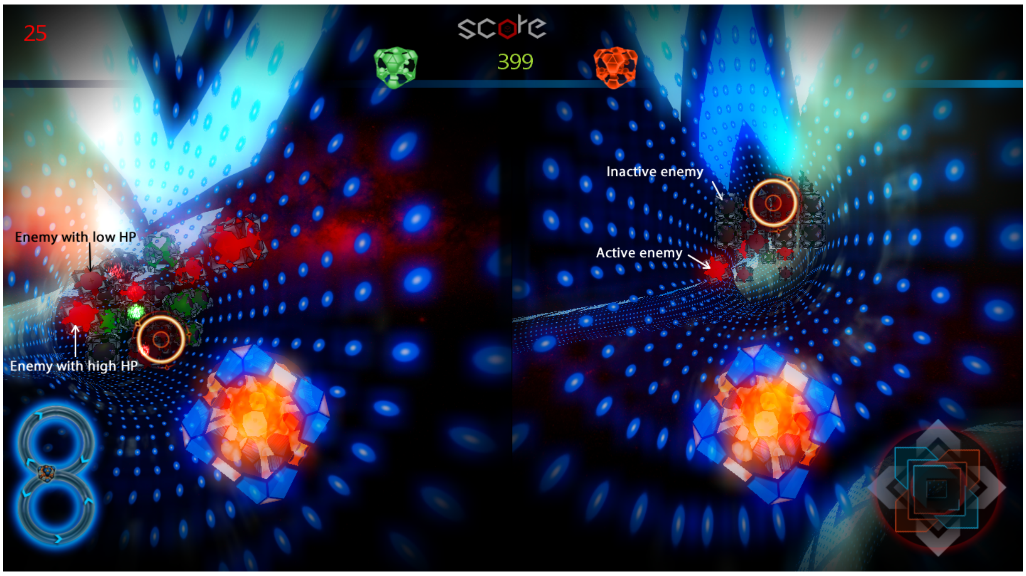

Figure 4.7: Comparison between inactive and active enemies and their level of hp.

Figure 4.8: The screen is colored red because the player is receiving damage.

4.3 General Architecture

Here we will detail the most important modules of The Energos game. We will mention their main functions and classes and their number of code lines.

As shown in Figure 4.9, The Energos Game consists of 3 main modules:

1. Screen Manager Module

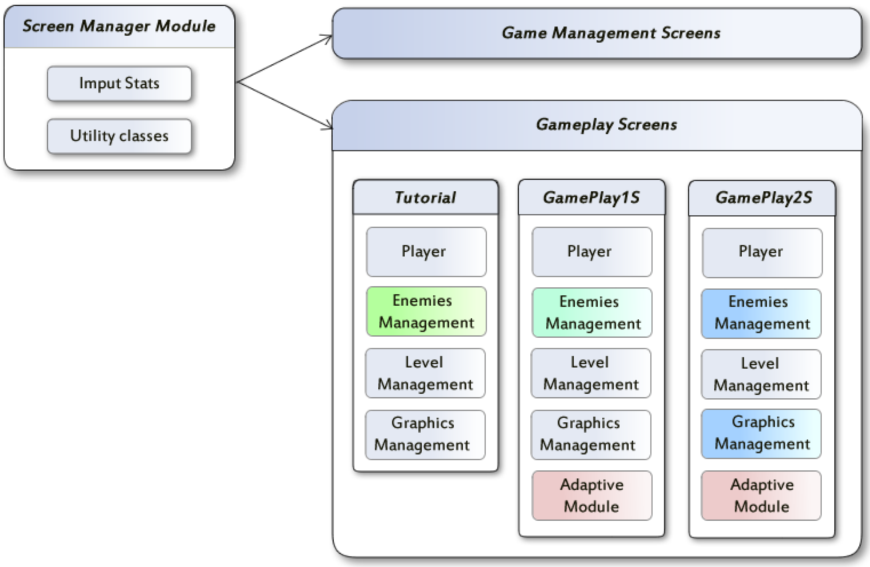

Figure 4.9: Diagram of the different modules that form The Energos Game.

2. Game Management Module

3. Game Play Screens Module

4.3.1 Screen Manager Module

We used the Game State Management Sample [51], which shows how to manage the transitions among menus and gameplay states.

The ScreenManager class is a reusable component that maintains a stack of one or more GameScreen instances. Each screen class (including the actual gameplay, which is just another screen) derives from GameScreen. This provides Update, HandleInput, and Draw methods, plus some logic for managing the transition state. It coordinates the transitions from one screen to another, calls their Update and Draw methods at the appropriate time and takes care of routing user input to whichever screen is on top of the stack.

As we can see, inside of this module there is an element called InputState which is a helper for reading input from the input devices. This class tracks both the current and previous state of the input devices. In this class we implemented the initialization and the communication between the application and the Kinect sensor based on the Shape Game example from Kinect for Windows SDK 1.0 [52]. There is also an element called Utility Classes, which includes tools for general purposes (converting vectors and matrices between the XNA and the Jitter Physics Environment, debug drawing of general and collision models, frame rate counters, etc.). This whole module consisted of 10 classes with 2060 code lines.

4.3.2 Game Management Module

This module consisted of screens derived from the GameScreen class with the objective of managing the game (this includes screens for menus, loading screens, the video story shown to motivate the introduction of the unfamiliar challenge and the ending screen). The whole module was formed by 11 classes with 1678 code lines.

4.3.3 Game Play Screens

This is where the gameplay was implemented for each of the versions. As shown in Figure 4.9, it has 3 screens:

1. Tutorial: this is where the tutorial session is implemented.

2. GamePlay1S: this screen corresponds to the One Screen version of the game.

3. GamePlay2S: this screen corresponds to the Dual Screen version of the game.

The 3 screens mentioned share an instance of the class Player and it is exactly the same for all of them. However, the EnemiesManager class is different, because in the Tutorial we have fixed parameters for the enemies. For the One and Dual Screen Version the enemies are created by the parameters defined by the Adaptive Module. The main difference between the EnemiesManager of the One and Dual Screens is the initial placement of the enemies (for the One Screen version the enemies are placed in front of the player and for the Dual Screen version they are placed at the front and at the back of the player). The LevelManagement class is the same for the 3 screens, the main function of this class is to load and place the models for the world and manage the physics of it. The GraphicsManagement consist of several classes with the objective of rendering to the game screen. Finally, the adaptive module is the same for GamePlay1S and GamePlay2S (as we will see in 4.3.4).

This whole part without counting the adaptive module consisted of 19 classes with 5929 code lines. Now we will explain the Adaptive Module in full detail.

4.3.4 The New Adaptive Module

We kept using the same parameters of the prototype for the new architecture (except for the number of neurons in each layer), perceptron's based layers, an input, hidden and output layer, back-propagation algorithm for training, learning rate of 0.01 and momentum of 0.9. For the first version of the game and considering it's game play, we defined the following parameters that would determine the difficulty of the game:

- Speed of the enemies
- Size of the enemies
- Number of enemies in a wave

And the parameters that we wanted to predict were:

- Number of enemies that the player would destroy in a wave with a given set of parameters (speed, size and number of enemies)
- Damage that the player would receive during the wave

As we have mentioned before, we developed another architecture for the second and third version but after testing the game we redesigned the gameplay and the first architecture was used for the three versions of the game.

Assigning the parameters that determine the difficulty to the input neurons and the ones that we want to predict to the output neurons, almost settles the complete architecture of the network. We only need to determine the number of neurons in the hidden layer.

We will continue to use the heuristic formula for determining the number of neurons of the hidden layer (3.4). Considering again $S = 1000$ and replacing the known values, we can determine that H $\approx$ 66.

We tried to reuse most of the code from the prototype. The NeuralNetwork and NeuralNetworkTrainer classes had no changes. The only class that suffered modifications of the adaptive module was NNetworkManager. We also defined a new training phase. So the new methodology for training the adaptive module is defined by:

- Defining a large range for the parameters (for example, the speed of the enemies must go from very slow to very fast). See Figure 4.10 for a comparison of the minimum and the maximum size of the enemies.
- A first phase of training by increasing linearly all the parameters and presenting them to a human player so we can cover all the range and determine a range of playable settings. As we mentioned before (in Section 3.3.4) this person should have medium level of experience with movement interaction games.
- A second phase of training that will take the results obtained from the first phase, then calculates randomly several combinations of patterns and choose the best (in this case, the one with the greater weighted sum of the

enemies destroyed minus the damage received) and present it to the final player.

After 30 seconds, the game will check the performance of the player and retrain the network with this new obtained information and the cycle will start again. After the first phase, we are aiming to obtain a balanced neural network, that will be the starting point for all the users of the system. Then during the second phase, the personalization of the parameters will take place according to the new information gathered for each player. As mentioned before, we changed the approach of assigning the value of a known formula to the output neurons of the neural network, but in order to maximize the difficulty of the game and the player's performance, we needed to implement the evaluation phase somewhere else and consider both aspects to maximize its value. We used the following formula:

$$f(x) = 0.5(Difficulty + Performance) \tag{4.1}$$

Where:
$Difficulty$: represents the combined weighted sum of how difficult are the parameters (0 if the enemies have the biggest size, the lowest speed and the wave has the minimum number of enemies. 1 if the enemies have the smallest size, the fastest speed and the wave has the maximum number of enemies). Each parameter had the same weight (1/3).

$Performance$: represents the combined weighted sum of how good the performance of the player is (0 if the player didn't destroy any enemy of the wave and received damage equal to its amount of energy and 1 if the player destroyed all the enemies of the wave and didn't receive any damage). Each parameter had a weight of 1/2.

The methodology for this part was the same as the one for the adaptive module of the first prototype. However the neural network was no longer learning the value of a known formula but the value of the parameters that define the performance of the player.

The New Adaptive Module Consisted of 6 classes (7 if we consider the differentiation between the enemies that was discarded) with 1061 code lines (1304 with the differentiation of enemies).

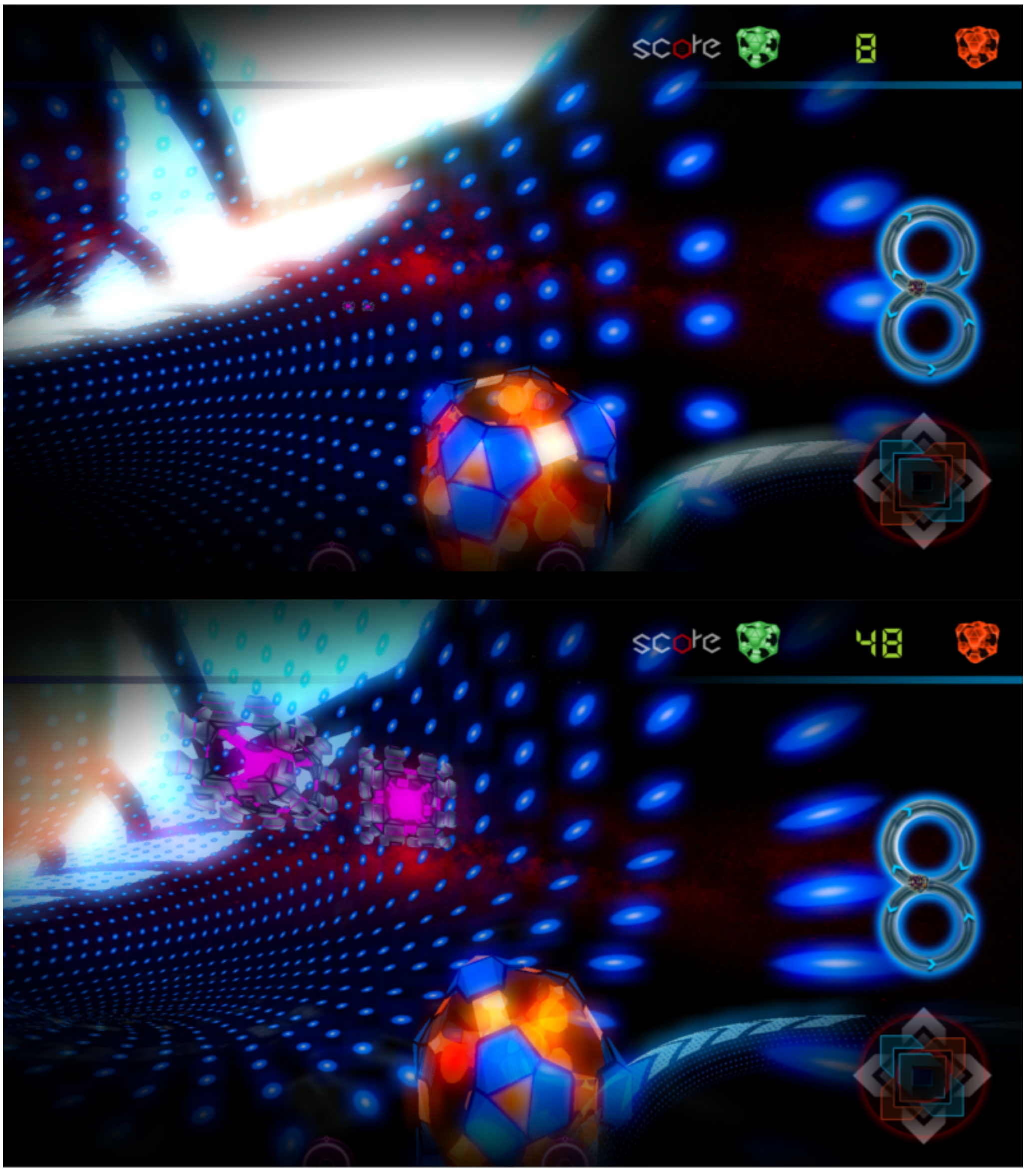

Figure 4.10: Comparison of the minimum size (top) and the maximum (bottom) size of the enemies.

Chapter 5

EVALUATING THE ENERGOS GAME

In this chapter we will detail the two evaluations of The Energos Game. The first evaluation was qualitative one and the second quantitative. Their main objective was to: 1) ascertain if tightly integrated challenges could promote flow states; 2) determine the impact of motivating a non intuitive unfamiliar challenge regarding its potential to promote flow states; and 3) investigate the way in which the difficulty of each challenge present in The Energos Game (familiar and unfamiliar challenges) was perceived by the players.

For evaluating The Energos Game we coordinated efforts with Universum (Museo de las Ciencias), a popular science museum located in Mexico City so it could be presented inside the museum's facilities. This was planned in order to ensure an ecologically valid environment for the evaluation, but required much more work than evaluating it in a laboratory. The evaluation of the game comprised two parts. The first part consisted of a qualitative evaluation of the system using a within subjects design. The main objective of this first evaluation was to calibrate the procedure and materials of the experiment for the next evaluation. The second part consisted of a quantitative evaluation using a between subjects design. The objective of this evaluation, was to test hypotheses 3) and 4) mentioned in Section 3.2. Visitors of the museum who approached the module, were asked to play only one version of the game and answer a flow questionnaire to evaluate their experience. In this evaluation we compared the answers to the questionnaires for the three versions, looking for differences in the participants' experience.

The following sections will explain both parts of the evaluation in more detail.

5.1 The Qualitative Evaluation

The objectives of this preliminary evaluation were:

1. To determine if the tutorial was clear enough and to adjust the duration of each phase of the game (practice and gaming session).
2. To adjust the range of the movements to the module assigned in the museum (sections of the tapestry and distance of the player to the kinect sensor).
3. To test the overall performance of the game, to identify and solve any bug that could have passed unnoticed and to improve the playability of the game.

5.1.1 Experimental Design

For this experiment we used a within subjects design. The participants compared two versions of the game (unbalanced and balanced or vice versa) explicitly and gave their subjective opinion of the game and the full gaming session via an interview. We recorded on video their playing sessions for further analysis. For the balanced motivated version we used a verbal description as proxy of the video story because this was still in production.

5.1.2 Procedure

Participants played an introductory tutorial with the common elements to all the game versions, then they were divided into two groups: those who played first the unbalanced version of the game (One Screen Version, see Section 4.2) and then a balanced version (either the Dual Screen version or the motivated Dual Screen version) and those who played first a balanced version of the game and then the unbalanced version of the game. For the tutorial, players were explained the important concepts of each point listed below (based on a predetermined verbal description of each point) and then they had to make the movements required:

1. Moving trough the path that represents the world (see Section 4.2.1.2).
2. How to avoid to receive damage from the enemies (see Section 4.2.1.3).
3. Controlling the cursors that respond directly to the movement of the player's hands and rotating the camera (see Section 4.2.1.4).
4. Activating enemies and shooting them (see Section 4.2.1.5).
5. Receiving damage and its associated feedback (see Section 4.2.1.6).

To finish the tutorial the player had to score 500 points. After the tutorial was concluded (which took in most cases around 4 minutes), we introduced a warm up session with a duration of 5 minutes, using the same version they would play first. Then they proceeded with the gaming session for 10 minutes. After the first session, participants played the alternative version to the first one they played (balanced if the first one was unbalanced and vice versa) first for 5 minutes (so that players could experiment with the differences between the versions) and then for the final session for 10 minutes. Finally we conducted an interview where we inquired about:

1. Their general impressions of playing the game.
2. Their comparison between the two versions.
3. The utility of the tutorial.
4. The duration of the warm up session and the gaming session.
5. The comparison between their own perceived skill level and the difficulty of the game.
6. Improvements or things they did not like about the game.

5.1.3 Participants

For the selection of the participants, we tried to get experienced players either with shooters or with kinect games, so that their observations could help us to improve the game. For this initial evaluation, we selected a group of 9 participants, 4 (44%) females and 5 (56%) males. Their academic degree ranged from high school to university and their ages from 17 to 27 years. Of all the participants, 6 (66%) had from 0 to 10 hours of experience playing kinect games and 3 (34%) had from 30 to 60 hours. Also, 4 (44%) had more than 100 hours playing shooter games, 1 (11%) had from 10 to 30 hours and 4 (44%) had from 0 to 10 hours.

5.1.4 Results

We will divide the results obtained from the interviews and the videos of the gaming sessions according to their own relevance with the objectives of this evaluation.

Considering the first objective, all the players considered that the tutorial introduced them successfully to the controls and objectives of the game. About the adjustments to the duration of the warm up and the gaming sessions, all the players considered that the time for each phase was adequate. The players felt that the first 5 introductory minutes were enough to become familiar with the game. Regarding the gaming session, 10 minutes are enough to achieve a state of effortless attention (for the experiment described in Section 3.3.1 the gaming sessions lasted for 10 minutes and players experienced states of effortless attention) and the participants also considered it adequate (neither too long or too short).

For the second objective (adjusting the range of movements considering the dimensions of the module), there were 2 participants that complained about the right cursor because it had several little jumps out of position and therefore was very hard to aim with it. After analyzing the videos of the gaming sessions, we determined that this occurred because the player's position was too close to a window to his/her right (See Figure 5.1) and the light coming from it interfered with the correct detection of the player's right hand.

For the last objective (improve playability, identify and correct any bug) we found that for players experienced in the shooter sub-genre it was difficult to avoid enemies, because although this is a familiar challenge, it was implemented in an unusual way. In all shooters, the movement of the player's character takes as reference the position and orientation of the camera, instead, in our first implementation, we took as reference the path. We also found a critical bug, when the player took too much damage, the game crashed (we will detail this in the next subsection). Excluding those problems, participants felt comfortable with all the other aspects of the game (controls, graphics and music). About the comparison between the versions, all participants felt that the unbalanced version was easier (regardless the order in which they played) but they also felt that their level of concentration was higher in the balanced version.

Figure 5.1: The light of the window interfered with the correct detection of the player's right hand.

5.1.5 Discussion

Considering that the players deemed that the tutorial successfully introduced them to the mechanics of the game we decided to use it in the same way for the next evaluation. The same happened for the times of the different phases of the experiment, so we fixed the times for the quantitative evaluation exactly as those of the qualitative evaluation (Warm up session of 5 minutes and gaming session of 10 minutes).

Controlling the shooting reticles with the hands in order to aim is an essential mechanic to successfully play the game, so the interference with the light of the window was indeed a serious problem. Because of that, we moved the playing position further to the front, to avoid the light's interference. However, this caused another problem, because the player was quite close to the Kinect sensor and when he/she jumped to avoid an enemy, the upper part of his/her body was out of reach of the detection area of the sensor. For that reason, we disabled the jumping option from the next version. We considered that playing the game without the jumping option was better than playing it with a shooting reticle that was hard to control because we had other 3 similar options for avoiding enemies; however the coordination of both hands was an essential part of this game. In the sooting sub-genre (specially in FPS and 3PS) controlling the orientation of the camera is a very important challenge and the movements of the player are always relative to it. For that reason, we changed the code of the controls to take the reference from the orientation of the camera instead of the path.

Regarding the bug that made the game crash, after some time of checking the code and testing the game, we found the cause of this bug. Remembering the mechanic explained in Section 4.2, when the player received damage from any number of enemies, she/he could increase the energy of the character by damaging the enemies. This represented a problem for us because when we passed the training parameters to the adaptive module (enemies destroyed and damage received) the parameter of the damage received from the enemies could be greater that 1 because the player could receive more damage than her/his full energy (as the player can replenish it) and in the training phase, the adaptive module entered an infinite loop because there is no way of modify the synaptic weights to achieve an output value of a neuron greater than 1 (remembering that the sigmoid function is limited to the range of (-1,1)). This was the cause of the crash of the program. The solution implemented was that when the damage received in a wave was greater than 100 we passed a 1 (maximum value) to the output neurons of the neural network.

Finally, we devised a way to accelerate the process of finding a good combination of the difficulty parameters and also doing it in a smoother way. We took the parameter that had the biggest impact on the difficulty of the game (speed of the enemies) and controlled it directly by initializing it to a fixed value. After each wave, we trained the neural network with the new information and then we evaluated how much damage the player received, if the damage was below 10% of the total energy, we incremented the speed of the enemies by 2%, but if the damage was greater than 40% we decreased it by 4%. After that, the other parameters were determined as before, we created 444 random combinations (the maximum number of combinations without noticing lag between frames) for the value of the number of enemies and their size, predicted the player's response to that parameter and kept the one with the greater sum of the enemies destroyed minus the damage done by the enemies. All these changes were employed in the quantitative evaluation.

5.2 The Quantitative Evaluation

In this evaluation we were interested first in determining if any of the versions of The Energos Game could promote states of effortless attention. If this was the case, then we would determine which version (or versions) were responsible for that effect. That could ascertain whether the motivation of the unfamiliar challenge helped or not in the promotion of flow states. We also investigated how the participants perceived the different challenges involved in the gameplay. In order to have more reliable results we tried to get as many people as possible, so that the correlation analysis could have enough data. The Energos Game was presented in Universum, a popular science museum, so that visitors could participate by playing the game and filling up a questionnaire to evaluate their experience. We were interested in how players perceived the different challenges, so we asked about the perceived difficulty of each challenge (coordinating the movements of the hands to aim, moving the avatar through the path, make quick movements to evade the enemies and the generating a particular strategy while playing). The game was in exhibition for 2 months and we gathered 106 questionnaires of people who played the game.

5.2.1 Experimental Design

For this experiment we used a between subjects design. Here, participants played a version of the game (randomly assigned) and answered a flow questionnaire derived from well-known and widely used instruments in flow research[17][20] with a 9 Points Likert scale (See Appendix B for the complete questionnaire). The independent variable was the session played while the dependent variable was the comparison of gaming experience.

5.2.2 Procedure

For this experiment, participants were the visitors of the Universum museum who approached the module on their own will. The module had a generic description of the project that anyone could read before entering, inviting the visitors to participate in this investigation and compete for the greatest score, it also informed that the entire participation process would last about 25 minutes. Once inside of the module, the participant played the same tutorial used for the qualitative evaluation (described in Section 5.1.2). After the tutorial was concluded, participants were told about the board that contained the top 3 scores for each version and the version they were about to play. In that way they knew beforehand the scores to beat. Then, we introduced a warm up session using the same version they would play with a duration of 5 minutes. During this part, players were encouraged to clear any doubts remaining by asking the responsible of the module. After this warm up, they played a 10 minutes gaming session. Finally they were asked to fill in the questionnaire.

5.2.3 Participants

There were 106 participants, 30 female (28%) and 76 male (72%). Their academic level ranged from elementary school to university, and their ages from 8 to 55 years. Of all the participants, 76 (72%) had from 0 to 10 hours of experience playing kinect games, 15 (14%) were inside the range of more than 10 to 30 hours, 10 (9%) had played from 30 to 60 hours, 2 (2%) more than 60 to 100 hours and 3 (3%) had played kinect games for more than 100 hours. None of them had played The Energos Game before.

5.2.4 Analysis

Questionnaires were analyzed looking for correlations between concentration level, concentration effort, alteration of the sense of time, self consciousness, intrinsic motivation and control. The level of concentration should be inversely correlated with concentration effort and self-consciousness and directly correlated with an alteration of the sense of time, intrinsic motivation and control. Regarding the analysis of challenges, we analyzed the answers related to the perceived difficulty of each challenge present in the game searching for significant differences either between the versions or specific challenges. This analysis could give us information about whether participants of a certain condition (One Screen, Dual Screen or Motivated Dual Screen) perceived in a different way the difficulty of each challenge.

5.2.5 The Evaluation Space

As we have mentioned before, both evaluations took place inside the museum's facilities. The evaluation space consisted of an isolated module of around 5m x 3m. It had an acrylic window to the right (see Figure 5.1) so that the gaming session could be seen from the outside. At the opposite end of the module, there was a 42" LCD TV where the game was showed. The kinect sensor was placed in front of the TV (see Figure 4.6); for the qualitative evaluation the player was positioned at a distance of 3 meters from the kinect sensor and for the qualitative evaluation at a distance of 2 meters.

5.2.6 Results

The results of the study can be divided into those associated with the participant's playing experience and those related to the analysis of the challenges and their perceived difficulty. The data obtained for playing experience did not comply with normality assumption so tests had to be non-parametric. However, the answers associated with the challenges-skill balance and those related with the difficulty of the different challenges of the game had normal distributions (See questions 9.a to 9.e in Appendix B).

We searched for significant differences for the intrinsic motivation, time distortion, concentration, concentration effort, self consciousness and control related answers between the different conditions of the experiment (for example, to find if participants of a particular condition tended to have higher levels of concentration than the participants of the other conditions). However, there was no significant value for this analysis (see Table 5.1).

Considering the entire data from the three conditions, the answers for the level of concentration (*how much did you concentrate?*) correlated inversely with those of concentration effort (*how hard was it to concentrate?*), but not with those

	Intrinsic Motivation	Time Distortion	Concentration
Chi-Square	0.997	0.36	0.287
df	2	2	2
Asymp. Sig.	0.607	0.982	0.866
	Concentration Effort	Self Consciousness	Control
Chi-Square	1.986	0.723	0.265
df	2	2	2
Asymp. Sig.	0.370	0.697	0.876

Table 5.1: Correlations for the questions related to the playing experience and the different conditions of the experiment. Correlation is significant (2-tailed) at the 0.05 level(*).

related to self-consciousness (see table 5.2); and directly with those of intrinsic motivation and sense of control but not with those related to the distortion of the sense of time.

		Concentration
Effort	Correlation Sig.	$-3.47*$ 0.000
Time Distortion	Correlation Sig.	-0.152 0.120
Control	Correlation Sig.	0.216$*$ 0.026
Self consciousness	Correlation Sig.	-0.178 0.068
Intrinsic motivation	Correlation Sig.	0.273$*$ 0.005

Table 5.2: Spearman correlations for flow indicators. N=106, correlation is significant (2-tailed) at the 0.05 level(*)

Analyzing the histogram for the answers to the question related to self consciousness (see Figure 5.2), we can see that the great majority of participants answered that they did not feel self conscious. This could explain that the level of concentration did not correlated inversely with self consciousness, because not even the participants who did not experience states of effortless concentration felt self conscious. Regarding the answers to time distortion and the lack of correlation with the level of concentration we believe that was caused by the conditions of the experiment (we will discuss this in more detail in the next section).

Then we analyzed the correlations between the level of concentration and the concentration effort of each condition separately.

For the One Screen version there was no correlation between the level of concentration and the concentration effort. However, the balanced versions (Dual

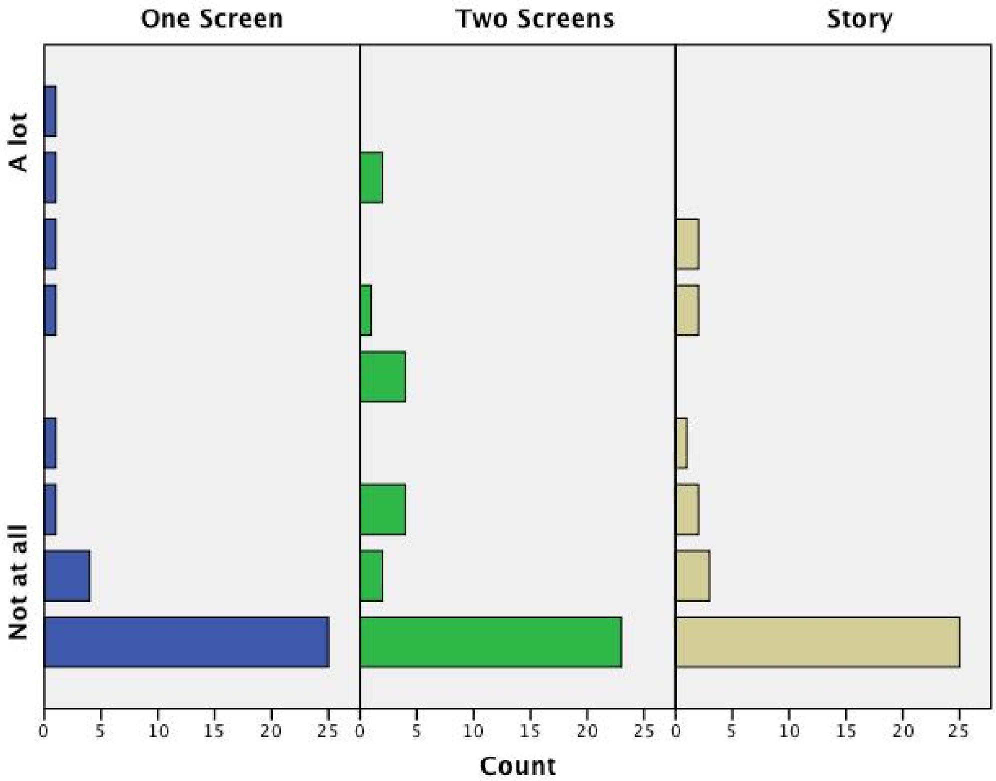

Figure 5.2: Histogram for the answers to the question related to the participants' level of self consciousness.

Condition	Correlation Sig.
One Screen	-3.09 0.071 N = 35
Two Screens	$-3.53*$ 0.035 N = 36
Two Screens with story	$-4.53**$ 0.006 N = 35

Table 5.3: Spearman correlations for each condition between level of concentration and concentration effort. Correlation is significant (2-tailed) at the 0.05 level(*), significant correlations (2-tailed) at the 0.01 level (**).

Screen and Motivated Dual Screen) presented significant values (See Table 5.3). This suggests that the balanced versions were responsible for the effect of effortless attention in the game. However the Motivated Dual Screen version had a higher significance level (of 0.01). This suggests that although overall the level of attention was similar for the 3 versions, for the unbalanced version it could have been effortfull attention.

For the analysis of the answers related to the difficulty of the challenges we used a Repeated Measures ANOVA with 5 factors within subjects (questions related to the challenges) and 3 between subjects (conditions of the experiment) to see whether there were differences between conditions in the difficulty of specific challenges. We found a significant difference when we analyzed all the questions ($F(99) = 16.04 p < 0.01$) , suggesting that participants perceived some of those challenges being more difficult than the others. According to the graph 5.3 showing the mean of the perceived difficulty for each challenge, it can be seen that the most difficult challenge was coordinating the movements of the hands to aim and destroy the enemies. This was corroborated by a post-hoc t-test ($t = 6.709, p < .01$).

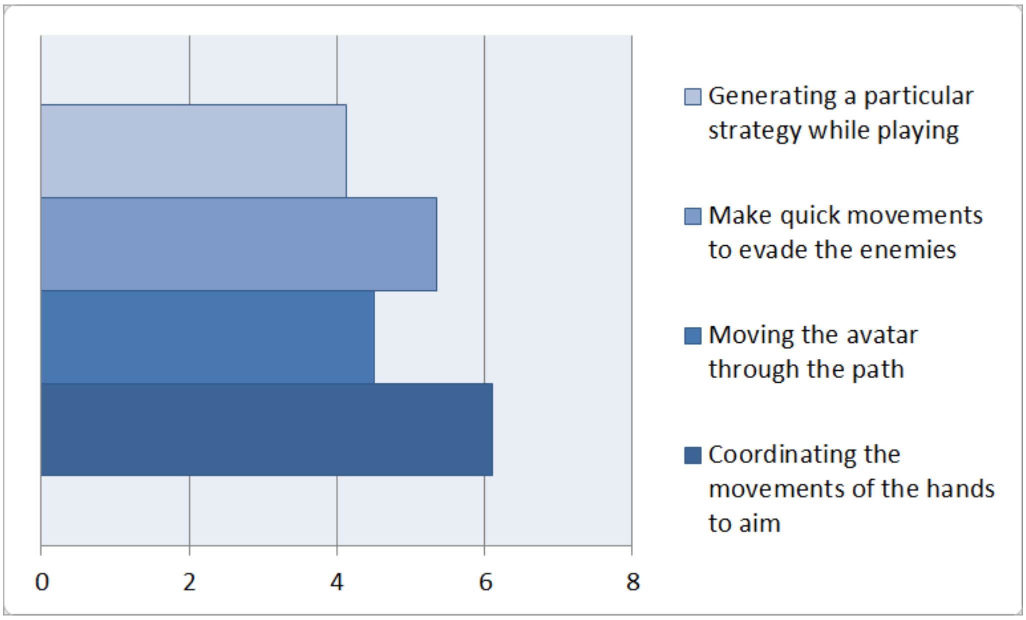

Figure 5.3: Mean value of the perceived difficulty for each challenge of The Energos Game.

5.2.7 Discussion

This evaluation compared the answers of the questionnaires for the three versions, looking for differences in the participants experience and their perception of the different challenges involved in the game. The results of the playing experience suggest that, although overall the data for the three conditions indicated that episodes of high concentration tended to be associated with states of flow, the balanced versions seemed to be mainly responsible for this effect. This confirms the importance of the balance between the cognitive and physical components of the challenges in order to promote flow states (found in Section 3.3.1). However, an interesting observation is that even though both of the balanced versions promoted flow states, the motivated Dual Screen version had a greater significance (at a level of 0.01). This suggest that the justification had a positive impact on the episodes of effortless attention.

As we can see in Table 5.2, the concentration level did not correlate inversely with self-consciousness nor directly with the distortion of the sense of time. We believe that this is due to the conditions of the experiment. Considering the self-consciousness part, the result shows that the participants who played the game did not feel ashamed, even without experiencing states of flow. We associate this to the personalities of the participants, because those who approached the module, are likely to have an extroverted personality and they would not feel self-conscious just by playing a game. With regards to the time distortion effect, we believe that this was the result of two main causes: 1)The description of the experiment out of the module informed about the duration of the whole process, which helped the participants to be aware of how much time it would take them to participate. 2)The majority of the participants were accompanied, some of those companions waited outside or entered the module, which could have caused the participants to be conscious of how much time passed.

Regarding the results related to the perceived challenges, we determined that the most difficult challenge of the game was to coordinate the hands to control the shooting reticles in order to attack the enemies for all 3 versions. This suggest that the dominant challenge of the entire game was aiming with the hands. The parameters that determined the difficulty had a greater impact on the challenge of coordinating hands to destroy enemies rather than on the generation of a strategy, movement through the road or dodge the enemies. When we defined the adaptive system, the game was thought and designed as a shooter. So it makes sense that this challenge was perceived as the hardest one because the main and hardest challenge of shooting games is to aim with the reticles to destroy the enemies. This shows that challenge is not a monolithic concept and its design needs a careful understanding in the way challenge impacts the different parts of the composite challenges, and of the overall gameplay.

Chapter 6

OVERALL DISCUSSION AND CONCLUSIONS

6.1 Overall Discussion

The study reported in this thesis evaluated the impact of several characteristics in movement interaction games that we hypothesized as important in order to promote flow states. The main findings suggest that:

1. Balanced activities (intellectually and physically balanced) are more likely to promote flow states.

2. Loosely integrated challenges could be perceived as a distraction interfering with the participant's level of concentration.

3. The proposed adaptive system based on neural networks had an adequate performance.

4. Motivating non intuitive unfamiliar challenges has a positive impact on the promotion of episodes of effortless attention.

5. Participants of the three conditions perceived the challenge of aiming with their hands as the hardest one.

This was a study that took place incrementally, the first two experiments (See Section 3.3) were the initial approach to the characterization of those challenges. The results of the experiment of balanced but loosely integrated challenges indicate that video games with movement based interaction are able to promote flow states (considering a definition of flow as effortless attention, See Section 2.1.2.1) but also suggests that the balance of cognitive and physical challenges is an essential part. Another important result is that an appropriate integration of the elements of the composite challenges is important because if it is done in a loosely way (when either the intellectual or physical parts are not relevant to the gameplay), players could feel that particular challenges are extraneous to the game and could consider them as a distraction.

Regarding the experiment that tested the adaptive system prototype, the results suggests that the proposed adaptive system had an adequate performance when trying to maintain a balance between the perceived difficulty of the game and the player's perceived skills. Although this is not the central point of this thesis, to the best of our knowledge, an adaptive system based on a neural network has not been used before to maintain a balance between the difficulty of the game and the player skills. This module could be enhanced in several ways, those improvements could vary from:

- Changing the neural network architecture (for example using different activation functions, adding more hidden layers, or even using architectures other than Feedforward, etc.).

- Improving the training phase (using more people to determine the best initial settings, creating personalized training stages for each player, etc.)
- Devising a more focused way to generate the patterns (not just randomly) to even most drastic changes like replacing the neural network with another optimization mechanism (like genetic algorithms or even simpler heuristics). But always considering that the key factor here is not the relation between the objective game difficulty and the objective player's skills; but the player's perception of their own competence.

Taking into account the results of these first experiments, we developed a movement interaction game. For its design, we selected a cognitive and a physical challenge from commercial kinect games and integrated them into a shooter (taking as inspiration a popular shooter that used a Kinect interface). This integration created a combination of familiar (recurrent challenges in a particular genre or sub-genre) and unfamiliar challenges. Unfamiliar challenges could be intuitive (when they resemble challenges of the real world, like trying to avoid something that is about to hit you by stepping to a side or ducking) or not (if it is completely new to the player). In order to investigate if understanding the motivation of an unfamiliar challenge inside the game is relevant for the promotion of states of effortless attention, we introduced a narrative that gave a context to the players on: 1) Who the main characters are 2) How did they get where they are 3) Why there are two screens at the same time. Then, we compared a version that implemented the narrative against another where no introduction was given to the players, they just played the game with no further explanation of the story or why there were two screens. To try to corroborate the results described in Section 3.3.3.5 we compared it with an unbalanced version that implemented familiar challenges with an intuitive unfamiliar challenge (by removing the cognitive challenge and leaving the physical challenge).

The comparison of those 3 versions indicated again that the balanced versions were responsible for the states of effortless attention, however the motivated version had a level of significance of 0.01 and the unmotivated version only at a level of 0.05. This suggests that the motivation had a positive impact on the episodes of effortless attention of the participants.

With regard of the analysis of the perceived difficulty of the challenges, we were interested in determining whether participants of a certain condition (One Screen, Dual Screen or Motivated Dual Screen) perceived in a different way the difficulty of each challenge. There were 4 main challenges present in The Energos Game:

1. Coordinating the movements of the hands to aim and destroy the enemies.
2. Moving through the path.

3. Avoiding the enemies.

4. Generating a strategy to destroy enemies in a efficient way.

In the questionnaire we asked explicitly about the perceived difficulty of each challenge and also about the overall difficulty of the game. From the analysis of those questions, aiming to destroy the enemies was perceived as the hardest challenge. This was a surprise to us, because we hypothesized that this would be true only in the balanced versions (those with dual screen) and not in the unbalanced version (one screen) but it turned out to be true for all the versions. When we devised the adaptive module, we defined that it should modify the size, number and speed of the enemies. These parameters affected mostly how hard it was to destroy the enemies and not so much the creation of strategies nor avoiding enemies. Considering also that this is the main challenge in almost all the shooter games, it makes sense that this challenge was perceived as the hardest one. However, an interesting improvement could be to consider factors that impact directly on the physical and cognitive parts of the composite challenges (for example, varying the speed of the movements required or their magnitude for the physical part and creating different enemies, so that the player should prioritize his/her attacks on some enemies first). Still, players perceived the overall difficulty of the game in balance with their own abilities.

To the best of our knowledge, this study is the first to investigate how different challenges can affect the player's experience and also how to introduce such challenges to promote flow states in games that use movement based interaction. Integrating familiar challenges of a popular genre in video games with new challenges can make the resulting game balanced and able to promote flow states; also, the introduction of a motivation for such new challenges (when they are not intuitive) could help to this end too. However, a careful analysis on how those challenges affect the overall difficulty of the game and each of the familiar challenges present in a genre is essential. Also, it is important to ensure a balance between the perceived difficulty of the game and the own player's perceived level of skill.

The results also support the embodied view of flow proposed by Romero and Calvillo-Gámez[58], which emphasizes the importance of the body not as a separate element but on its relationship with the intellectual aspects of the person's behavior; they also confirm the results of Romero[59] about the importance of balance in the composite challenges.

The fact that the level of attention was inversely correlated with the perceived effort and also correlated with the rest of the main characteristics of the flow state confirms that flow can be operationalized and conceptualized as a state of effortless attention. Csikszentmihalyi and Nakamura (2010)[20] also used a similar method to operationalize the concept.

This study represent the initial steps on how to properly integrate the physical and cognitive parts of composite challenges and also on the refinement on how to measure each aspect.

Thinking about the limitations of the study we can think of three main improvements:

1. Integrating parameters to the adaptive module that impact directly on the how difficult the challenges are. For example, to increase the difficulty of the challenge of avoiding enemies, we could vary the range of movements required to activate the dodging move in the game to increase the physical activity and try to maximize it too. For the challenge of generating a strategy, a differentiation of enemies could be made so that particular enemies were more important than others. This could make the players to their attention on those particular enemies. The parameter here would be the number of "special" enemies.

2. Another improvement would be to limit the lower age of the participants to 15 years. The reason behind this limitation is that participants who were younger that 15 years, were sometimes confused with both the mechanics of the game and the phrasing of the questions.

3. The last 2 improvements have to do with how the motivation for the unfamiliar challenge was introduced. The script of the story elements could be improved or completely redesigned by a talented storyteller.

4. As we have mentioned before, the motivation was introduced by a video that showed the story of the game whit a duration of 3 minutes and 40 seconds. This part could also be improved by creating an interactive animation, so that the player could have a greater level of participation during the motivation of unfamiliar challenges.

6.2 Conclusions

We have presented a study with the main objective of characterizing challenges of games with movement based interaction designed to promote states of effortless attention or flow. This study took place incrementally with 3 main experiments. The results of the study that investigated the ability to promote flow of a balanced but loosely integrated game suggests that integration and balance are indeed important aspects of this promotion. Balanced challenges (even if they are loosely integrated), promoted states of effortless attention. However according to the opinion of the participants, the session in which they had to keep a mental count also tended to promote the highest levels of concentration effort.

Regarding the challenge-skill balance as the main requirement to promote flow, the results of the experiment that tested its performance suggest that an adaptive system based on a neural network is capable of maintaining such balance. Despite having different players (in terms of their playing experience) participants felt that the difficulty of the game was just a little above their skills, suggesting that it was challenging but not overwhelming.

The results of the unfamiliar challenges experiment, on the one hand confirms the importance of balance in composite challenges and on the other suggests that motivating a non intuitive unfamiliar challenge has a positive impact on their potential to promote flow states, although the unmotivated balanced version promoted flow states too. This suggest that although motivating unfamiliar challenges could not be essential, its a factor that should be considered.

The work presented has important implications in supporting the embodied view of flow proposed by Romero and Calvillo-Gámez[58], in refining the way challenges are measured in flow studies, as well as in guiding the design of movement interaction applications that aim to promote flow in users.

For this study, challenge elements have been limited to either physical or cognitive. However, according to Ellis et al.[29], they might be associated not only with the intellectual and physical parts but with an emotional part of the person as well. A question worth pursuing is how these 3 elements could be included in the design of a digital application.

A further issue relates to method and measurement. Traditionally, flow studies have employed questionnaires as an instrument for measurement mainly; however other methods, physiological correlates (Manzano et al.[47]) or phenomenological methods for the study of subjective experience (Petitmengin[55]; Hurlburt and Akhter[41]) could be very appropriate to investigate challenges in flow.

Bibliography

[1] Random class reference in msdn. http://msdn.microsoft.com/en-us/library/ts6se2ek(v=vs.100).aspx. Accessed 31 January 2013. [cited at p. 104, 107]

[2] Project home for jitter physics in google code, 2011. http://code.google.com/p/jitterphysics/. Accessed 31 January 2013. [cited at p. 60]

[3] Main page of the kinect for windows project, 2012. http://www.microsoft.com/en-us/kinectforwindows/. Accessed 31 January 2013. [cited at p. 43, 60]

[4] Main page of the openkinect project, 2012. http://openkinect.org/wiki/Main_Page. Accessed 31 January 2013. [cited at p. 43]

[5] Main page of the openni project, 2012. http://www.openni.org/. Accessed 31 January 2013. [cited at p. 43]

[6] E. Adams. Fundamentals of game design. *New Riders*, 2010. [cited at p. 16, 44, 45]

[7] Hopcroft J. Ullman J. Aho, A. The design and analysis of computer algorithms. *Addison-Wesley, Reading Mass*, 1974. [cited at p. 24]

[8] Ramalho G. Gomes A.S. Andrade, G. and V. Corruble. Dynamic game balancing: An evaluation of user satisfaction. *American Association for Artificial Intelligence*, 2006. [cited at p. 24]

[9] Entertainment Software Association. Game industry facts (sales and genre data), 2012. http://www.theesa.com/facts/salesandgenre.asp. Accessed 31 March 2013. [cited at p. 12]

[10] Whan W.K. Patel D. Bianchi-Berthouze, N. Does body movement engage you more in digital game play? and why? In Prada R. Picard R.W. Paiva, A.C.R., editor, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pages 102–113. Springer, Heidelberg, 2007. [cited at p. 14]

[11] B. Bruya. Introduction: Toward a theory of attention that includes effortless attention. In B. Bruya, editor, *Effortless Attention: A New Perspective in the Cognitive Science of Attention and Action*. MIT Press, 2010. [cited at p. 15, 113]

[12] McNeill M. Charles D. Burke, J. Optimizing engagement for stroke rehabilitation using serious games. *The Visual Computer 25*, 2009. [cited at p. 45]

[13] Jenova Chen. Flow in games: A jenova chen mfa thesis, 2006. http://jenovachen.com/flowingames/introduction.htm. Accessed 26 October 2012. [cited at p. 5, 38]

[14] Jenova Chen. Viewpoint flow in games (and everything else) a well-designed game transports its players to their personal flow zones, delivering genuine feelings of pleasure and happiness. *Communications of the ACM*, 50(4):31–34, 2007. [cited at p. 10, 11, 113]

[15] Ryan Clements. Kinect adventures review. does the kinect pack-in sell us on controller-free gaming? IGN, 2010. http://www.ign.com/articles/2010/11/04/kinect-adventures-review. Accessed 07 January 2013. [cited at p. 19]

[16] Cairnes P Shah P Carroll M. Cox, A. Not doing but thinking: The role of challenge in gaming experience. *CHI*, 2012. [cited at p. 10]

[17] Hektner J. Schmidt J. Csikszentmihalyi, M. Experience sampling method: Measuring the quality of everyday life. *SAGE*, 2006. [cited at p. 79]

[18] M. Csikszentmihalyi. *Beyond boredom and anxiety.* Jossey-Bass Publishers, 1975. [cited at p. 10]

[19] M. Csikszentmihalyi. *Flow: The Psychology of Optimal Experience.* Harper Perennial, 1990. [cited at p. 10, 11, 12, 47]

[20] Nakamura J. Csikszentmihalyi, M. Effortless attention in everyday life: A systematic phenomenology. In B. Bruya, editor, *Effortless Attention: A New Perspective in the Cognitive Science of Attention and Action*, pages 179–189. The MIT Press, Cambridge, MA, 2010. [cited at p. 38, 39, 79, 88]

[21] S. & Nakamura J. Csikszentmihalyi, M.; Abuhamdeh. Flow. In A. Elliot, editor, *Handbook of Competence and Motivation*, pages 598–698. New York: The Guilford Press, 2005. [cited at p. 10, 11]

[22] M. Curtiss. Why games work and the science of learning. *Alion Science and Technology*, 2011. [cited at p. 12]

[23] E. Debold. Flow with soul: An interview with dr. mihalyi csikszentmihalyi. *What Is Enlightenment Magazine*, 2002. [cited at p. 10]

[24] digitalerr0r. Kinect fundamentals: Basic programming, June 2011. http://digitalerr0r.wordpress.com/2011/06/20/kinect-fundamentals-2-basic-programming/. Accessed 01 September 2011. [cited at p. 43]

[25] digitalerr0r. Kinect fundamentals: Installation and setup, June 2011. http://digitalerr0r.wordpress.com/2011/06/20/kinect-fundamentals-1-installation-setup/. Accessed 01 September 2011. [cited at p. 43]

[26] Sarah Ditum. Child of eden - review. Theguardian, 2011. http://www.guardian.co.uk/technology/gamesblog/2011/jun/14/child-of-eden-kinect-review. Accessed 10 February 2013. [cited at p. 21]

[27] Sarah Ditum. Dr kawashima's body and brain exercises - review. Theguardian, 2011. http://www.guardian.co.uk/technology/gamesblog/2011/feb/07/dr-kawashima-body-brain-exercises. Accessed 20 January 2013. [cited at p. 21, 22]

[28] P. Dourish. Where the action is: The foundations of embodied interaction. *The MIT Press, Cambridge, MA.*, 2001. [cited at p. 14]

[29] Voelkl J. Morris C. Ellis, G. Measurement and analysis issues with explanation of variance in daily experiences using the flow model. *Journal of Leisure Research 26*, pages 337–356, 1994. [cited at p. 15, 90]

[30] Entertainment Software Association (ESA). Sales and genre data of video games in 2011, 2011. http://www.theesa.com/facts/salesandgenre.asp. Accessed 01 May 2013. [cited at p. 16]

[31] M. Zhang P. Finneran, C. Flow in computer-mediated environments: Promises and challenges. *Communications of the Association for Information Systems*, pages 82–101, 2005. [cited at p. 14, 16, 38]

[32] JC Fletcher. Body and brain connection's pac-man minigame on video. Joystiq.com, 2010. http://www.joystiq.com/2010/11/23/body-and-brain-connections-pac-man-minigame-on-video. Accessed 20 January 2013. [cited at p. 22]

[33] Matt Furtado. Kinect hands-on: Body and brain connection hands-on preview. Examiner.com, 2010. http://www.examiner.com/article/kinect-hands-on-body-and-brain-connection-hands-on-preview. Accessed 20 January 2013. [cited at p. 22]

[34] Vanden Abeele V. Husson J. et al. Geurts, L. Digital games for physical therapy: fulfilling the need for calibration and adaptation. *Proceedings of the fifth international conference on Tangible, embedded and embodied interaction*, 2011. [cited at p. 45]

[35] R. Grootjans. *XNA 3.0 Game Programming Recipes: A Problem-Solution Approach*. Apresspod Series. Apress, 2009. [cited at p. 43]

[36] S.S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 2 edition, 1999. [cited at p. 25, 28]

[37] A. Hejdenberg. The psychology behind games. Gamasutra.com, 2005. http://www.gamasutra.com/view/feature/2289/the_psychology_behind_games.php. Accessed 22 October 2012. [cited at p. 11]

[38] Pepyne D. Ho, Y. Simple explanation of the no free lunch theorem of optimization. *Proceedings of the 40th IEEE Conference of Decision and Control. Orlando Florida USA*, December 2001. [cited at p. 47, 48]

[39] Novak T. P. Hoffman, D. L. Flow online: Lessons learned and future prospects. *Journal of Interactive Marketing*, pages 23–34, 2009. [cited at p. 14]

[40] Robertson Holt. *Examining Video Game Immersion as a Flow State*. PhD thesis, Department of Psychology, Brock University, St. Catharines, Ontario, Canada, 2000. [cited at p. 12]

[41] Akhter S. Hurlburt, R. The descriptive experience sampling method. *Phenomenology and the Cognitive Sciences 5*, pages 271–301, 2006. [cited at p. 90]

[42] IIMAS-UNAM. Seminario de inteligencia artificial, March 2011. http://turing.iimas.unam.mx/ seminarioia/index.html. Accessed 31 March 2013. [cited at p. 52]

[43] A Jarvinen. *Games without Frontiers: Theories and Methods for Game Studies and Design*. PhD thesis, Department of Literature and the Arts, University of Tampere, Finland, 2008. [cited at p. 13]

[44] & Landhuer A Keller, J. The flow model revisited. In S. Engeser, editor, *Advances in Flow Research*, pages 51–64. Springer New York, 2012. [cited at p. 11]

[45] Johannes Keller and Frederik Blomann. Locus of control and the flow experience. *Eur. J. Pers*, 22(7):589607, 2008. [cited at p. 11, 38]

[46] A. Leigh. Microsoft: Kinect hits 10 million units, 10 million games, 2011. http://www.gamasutra.com/view/news/33430/Microsoft_Kinect_Hits_10_Million_Units_10_Million_Games.ph Accessed 15 September 2012. [cited at p. 6]

[47] Theorell T. Harmat L. Ullen F. Manzano, O. The psychophysiology of flow during piano playing. *Emotion 10 (3)*, pages 301–311, 2010. [cited at p. 90]

[48] Liam Martin. Kinect adventures (xbox 360). Digital spy, 2010. http://www.digitalspy.co.uk/gaming/review/a285914/kinect-adventures-xbox-360.html. Accessed 07 January 2013. [cited at p. 20]

[49] Griffin McElroy. Child of eden review: A momentary masterpiece. Joystiq.com, 2011. http://www.joystiq.com/2011/06/14/child-of-eden-review/. Accessed 10 February 2013. [cited at p. 21]

[50] Microsoft. Xna game studio 4.0 main page on msdn, 2005. http://msdn.microsoft.com/en-us/library/bb200104.aspx. Accessed 20 March 2013. [cited at p. 60]

[51] Microsoft. Game state management sample, 2011. http://xbox.create.msdn.com/en-US/education/catalog/sample/game_state_management. Accessed 31 January 2013. [cited at p. 68]

[52] Microsoft. Developer downloads (kinect for windows), 2012. http://www.microsoft.com/en-us/kinectforwindows/Develop/Developer-Downloads.aspx. Accessed 31 January 2013. [cited at p. 68]

[53] Nintendo. Wii website, 2012. http://goo.gl/MxRXF8. Accessed 31 February 2013. [cited at p. 16]

[54] Bianchi-Berthouze N. Dijk B. Nijholt A. Pasch, M. Immersion in movement-based interaction. In Anton Nijholt, Dennis Reidsma, and Hendri Hondorp, editors, *Intelligent Technologies for Interactive Entertainment*, pages 169–180. Springer Berlin Heidelberg, 2009. [cited at p. 14]

[55] C Petitmengin. Describing one's subjective experience in the second person: An interview method for the science of consciousness. *Phenomenology and the Cognitive Sciences 5*, pages 229–269, 2006. [cited at p. 90]

[56] New York Post. Microsoft gears up for big launch of kinect controller-free device for xbox, 2010. http://goo.gl/pqvHS. Retrieved October 20, 2010. Accessed 01 September 2011. [cited at p. 43]

[57] Eric Qualls. Body and brain connection review (x360). About.com, 2012. http://xbox.about.com/od/Kinect-Game-Reviews/fr/Body-And-Brain-Connection-Review-X360.htm. Accessed 18 January 2013. [cited at p. 22]

[58] Calvillo-Gamez E. H. Romero, P. Towards an embodied view of flow. In Semeraro-G. d. G. M. Lops P. Masthoff J. Grasso F. Ham J. Ricci, F., editor, *Proceedings of 2nd Workshop on User Models for Motivational Systems: The affective and the rational routes to persuasion (UMMS 2011)*, pages 100–105. CEUR Workshop Proceeding, 2011. [cited at p. 9, 14, 88, 90]

[59] Calvillo-Gamez E. H. Romero, P. Effortless attention and composite challenges in movement interaction. *British Informatics Society Ltd. Proceedings of HCI 2012 - People and Computers XXVI*, 2012. [cited at p. 14, 15, 35, 88]

[60] A. Rostamizadeh and Mohri-M. Talwalker, A. Foundations of machine learning. *The MIT Press*, 2012. [cited at p. 24]

[61] John Sherry. Flow and media enjoyment. *Communication Theory*, 14(4):328347, 2004. [cited at p. 12, 57]

[62] S. Siwek. Video games in the 21st century (the 2010 report). Technical report, Entertainment Software Association, 2010. [cited at p. 12]

[63] Craig Snow. Kinect adventures - review. gamrReview, 2010. http://www.gamrreview.com/review/82805/kinect-adventures/. Accessed 07 January 2013. [cited at p. 20]

[64] Kristine Steimer. Child of eden review on ign, 2011. http://www.ign.com/articles/2011/06/14/child-of-eden-review-2?page=2. Accessed 31 January 2013. [cited at p. 42]

[65] Penelope Sweetser and Peta Wyeth. Gameflow : a model for evaluating player enjoyment in games. *Computers in Entertainment*, 3(3):1–24, 2005. [cited at p. 12, 13, 58, 115]

[66] Ubisoft. Child of eden, 2011. http://child-of-eden.us.ubi.com/. Accessed 15 September 2012. [cited at p. 21]

[67] A. E Voiskounsky. Flow experience in cyberspace: Current studies and perspectives. In A. Barak, editor, *Psychological Aspects of Cyberspace: Theory, Research, Applications*, pages 70–101. Cambridge University Press, New York, 2008. [cited at p. 14]

[68] Wikipedia. Dead or alive series. http://en.wikipedia.org/wiki/Dead_or_Alive_(series). Accessed 01 September 2011. [cited at p. 18]

[69] Wikipedia. Donkey kong video game. http://en.wikipedia.org/wiki/Donkey_kong. Accessed 01 September 2011. [cited at p. 18]

[70] Wikipedia. Gears of war series. http://en.wikipedia.org/wiki/Gears_of_War_(series). Accessed 01 September 2011. [cited at p. 18]

[71] Wikipedia. Halo series. http://en.wikipedia.org/wiki/Halo_(series). Accessed 01 September 2011. [cited at p. 18]

[72] Wikipedia. King of fighters series. http://en.wikipedia.org/wiki/King_of_Fighters. Accessed 01 September 2011. [cited at p. 18]

[73] Wikipedia. Quake series. http://en.wikipedia.org/wiki/Quake_(series). Accessed 01 September 2011. [cited at p. 18]

[74] Wikipedia. Space panic video game. http://en.wikipedia.org/wiki/Space_Panic. Accessed 01 September 2011. [cited at p. 18]

[75] Wikipedia. Street fighter series. http://en.wikipedia.org/wiki/Street_fighter. Accessed 01 September 2011. [cited at p. 18]

[76] Wikipedia. Super mario bros video game. http://en.wikipedia.org/wiki/Super_Mario_Bros.. Accessed 01 September 2011. [cited at p. 18]

Appendices

Appendix A

Class Diagrams of the Adaptive Module

Class diagrams for the relevant classes of the adaptive module of the first prototype.

A.1 Class Diagram for NeuralNetworkTrainer

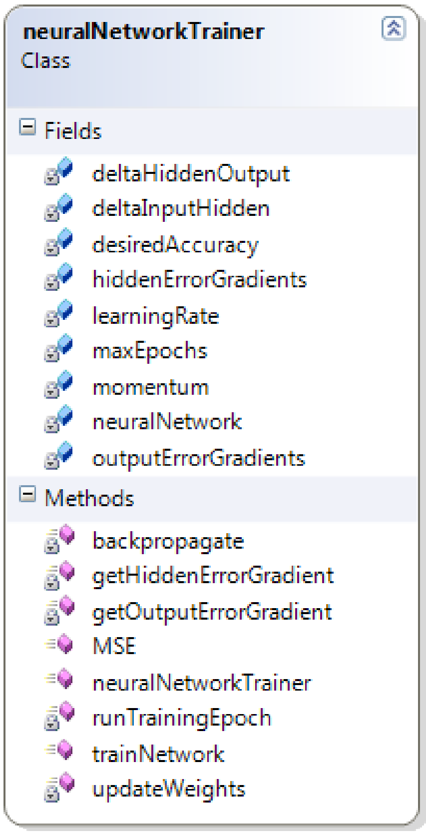

Figure A.1: Class Diagram for NeuralNetworkTrainer.

Fields

deltaHiddenOutput : Variations in the weights of the hidden layer to the output layer. These changes are calculated in the backpropagate. They indicate the adjustments to be made to neuralNetwork.wHiddenOutput.

deltaInputHidden : Variations in the weights of the hidden layer to the output layer. These changes are calculated in the backpropagate method. They indicate the adjustments to be made to neuralNetwork.wHiddenOutput.

desiredAccuracy : Desired accuracy for the training process. The value is normalized. When the MSE is smaller than 1 - desiredAccuracy the training stops.

hiddenErrorGradients : Array of doubles to storage the error gradients of the hidden layer.

learningRate : Learning rate.

maxEpochs : Max epochs for the training phase.

momentum : Momentum.

neuralNetwork : Instance of NeuralNetwork. It represents the neural network that will be trained.

outputErrorGradients : Array of doubles to storage the error gradients of the output layer.

Methods

backpropagate : Calculates deltaHiddenOutput and deltaInputHidden according to the desired outputs from the training example.

getHiddenErrorGradient : Calculates the error gradient for a neuron of the hidden layer. Receives as parameters the index of the neuron.

getOutputErrorGradient : Calculates the error gradient for a neuron of the output layer. Receives as parameters the desired value and the actual output value of the neuron.

MSE : Calculates the MSE of the neural network. Receives a training set to calculate it.

NeuralNetworkTrainer : Constructor of the class. Receives the instance initialized of NeuralNetwork, the learning rate, momentum, max epochs and the desired accuracy.

runTrainingEpoch : Consists of an epoch for the given training set. For each training example in the training set it calls feedForward to calculate the value of the output neurons for the input pattern, then calls backpropagate to calculate the changes to the weights and finally calls updateWeights.

trainNetwork : Calls runTrainingEpoch while the MSE is greater than 1 - desiredAccuracy.

updateWeights : Updates the values of the weights between the input-hidden layer and the hidden-output layer. The changes are saved in deltaInputHidden and deltaHiddenOutput.

A.2 Class Diagram for NNetworkManager

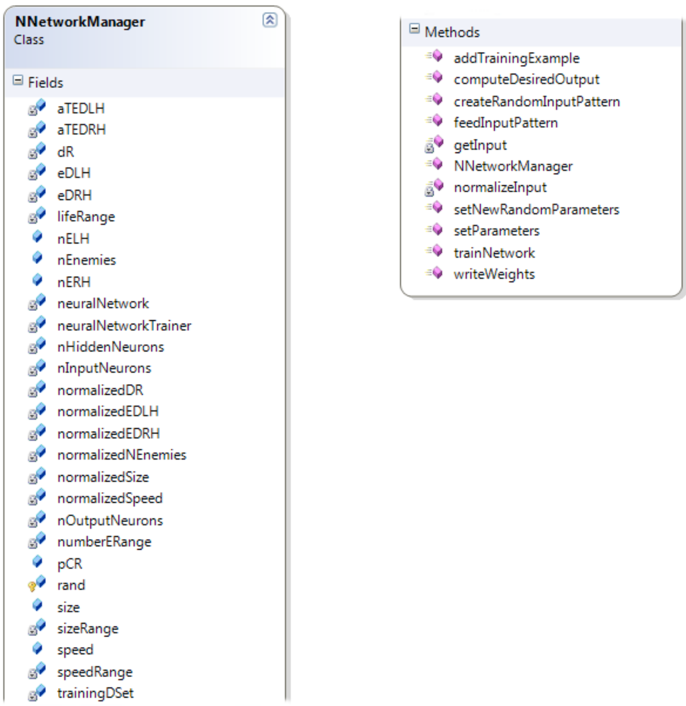

Figure A.2: Class Diagram for NNetworkManager.

Fields

aTEDLH : Average time to destroy a blue enemy (left hand).

aTEDRH : Average time to destroy a red enemy (right hand).

dR : Damage received during the wave.

eDLH : Number of blue enemies destroyed during the wave.

eDRH : Number of red enemies destroyed during the wave.

lifeRange : Range for the life of the player.

nELH : Number of total blue enemies of the wave.

nEnemies : Number of total blue and red enemies of the wave.

nERH : Number of total red enemies of the wave.

neuralNetwork : Instance of neuralNetwork.

neuralNetworkTrainer : Instance of neuralNetworkTrainer.

nHiddenNeurons : Number of neurons in the hidden neurons.

nInputNeurons : Number of neurons in the input neurons.

normalizedDR : Normalized value of the damage received.

normalizedEDLH : Normalized value of the number of blue enemies destroyed.

normalizedEDRH : Normalized value of the number of red enemies destroyed.

normalizedNEnemies : Normalized value of the number of blue and red enemies.

normalizedSize : Normalized value of the size of the enemies.

normalizedSpeed : Normalized value of the speed of the enemies.

nOutputNeurons : Number of neurons in the output neurons.

numberERange : Range to determine the minimum and maximum number of enemies in a wave.

pCR : Probability for an enemy to be red.

rand : Instance of the Random Class [1]. Used to generate random numbers.

size : Size of the enemies.

sizeRange : Range to determine the minimum and maximum size of the enemies in a wave.

speed : Speed of the enemies.

speedRange : Range to determine the minimum and maximum value for the speed of the enemies in a wave.

trainingDSet : Training set.

Methods

addTrainingExample : Adds a pattern to the training set. Receives the non normalized values of the next parameters: size, speed, pCR and number of enemies in the wave and the player's performance parameters: damage received, blue and red enemies destroyed and the average time for destroying a red and a blue enemy. It normalizes the input by calling normalizeInput and then adds the training example to trainingDSet.

computeDesiredOutput : Compute the value of the equations 3.2 and 3.1. Return their values in an array of doubles.

createRandomInputPattern : Creates and returns an array of doubles for the input neurons.

feedInputPattern : Receives an array of doubles of the size nInputNeurons. Calls neuralNetwork.feedInput for that array. Then calculates the value of equation 3.3 and returns it.

getInput : Constructs and returns an array of doubles with the normalized values of the parameters that corresponds with the input neurons.

NNetworkManager : Constructor of the class. Receives a bool, if true it loads the weights for the neural network from a CSV file. If false, the weights will be generated randomly. It also calls the constructors for neuralNetwork and neuralNetworkTrainer to construct them according to the settings specified.

normalizeInput : Normalizes the values of the difficulty parameters and saves them in their normalized variables.

setNewRandomParameters : Calculates a new set of values for the size, speed, pCR and number of total enemies.

setParameters : Receives an array of doubles of the size of nInputNeurons that is normalized. Calculates the non normalized values of the parameters and saves them in their corresponding variables.

trainNetwork : Calls neuralNetworkTrainer.trainNetwork with trainingDSet as parameter.

writeWeights : Write the current weights of the neural network in a CSV file.

A.3 Class Diagram for NeuralNetwork

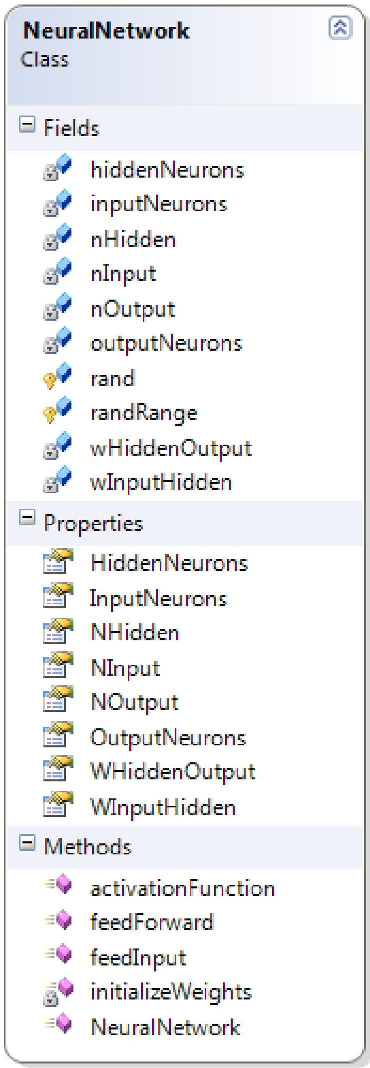

Figure A.3: Class Diagram for NeuralNetwork.

A.3.1 Fields

hiddenNeurons : Array of doubles that represent the hidden layer of the neural network. Its size is nHidden + 1, considering the bias neuron.

inputNeurons : Array of doubles that represent the input layer of the neural network. Its size is nInput + 1, considering the bias neuron.

nHidden : Number of neurons in the hidden layer.

nInput : Number of neurons in the input layer.

nOutput : Number of neurons in the output layer.

outputNeurons : Array of doubles that represent the output layer of the neural network. Its size is nOutput + 1, considering the bias neuron.

rand : Instance of the Random Class [1]. Used to generate random numbers.

randRange : Custom class to represent a range, used to simplify the calculation of random numbers between the specified range.

wHiddenOutput : An array of an array of doubles (double[ ][ ] wHiddenOutput;) that represents the weights from the hidden layer to the output layer.

wInputHidden : An array of an array of doubles (double[ ][ ] wInputHidden;) that represents the weights from the input layer to the hidden layer.

Properties

HiddenNeurons : Public property to obtain hiddenNeurons.

InputNeurons : Public property to obtain inputNeurons.

NHidden : Public property to obtain nHidden.

NInput : Public property to obtain nInput.

NOutput : Public property to obtain nOutput.

OutputNeurons : Public property to obtain outputNeurons.

WHiddenOutput : Public property to obtain or set wHiddenOutput.

WInputHidden : Public property to obtain or set wInputHidden.

Methods

activationFunction : Obtains the value of the activation function defined in equation 3.5.

feedForward : Calculates the values for the output neurons for a given input pattern.

feedInput : Given an input pattern, calls feedForward and returns an array with the resulting value of the output neurons.

initializeWeights : Initialize all the weights of the neural network in random values between 0 and 1.

NeuralNetwork : Constructor of the class. It takes as parameters the number of neurons in the input, hidden and output layers. Saves these values in nInput,nHidden and nOutput. Then creates the arrays inputNeurons, hiddenNeurons and outputNeurons. Finally creates the arrays for the weights (wInputHidden and wHiddenOutput) and calls initializeWeights.

Appendix B

Questionnaire

Following is the questionnaire used for the quantitative evaluation described in Section 5.2.

Evaluación del Nivel de Concentración

Por favor recuerda el videojuego que acabas de jugar y contesta las siguientes preguntas teniendo en mente la actividad que realizaste.
Indica en las siguientes frases, la opción que mejor se adecúe a tu opinión de la experiencia al jugar.
Estamos interesados primordialmente en tu opinión subjetiva respecto al momento en que estabas jugando.

Sobre tí...

1. **Tu nombre**: ________________________________
2. **Tu edad**: ______
3. **Sexo:** ○ M ○ F
4. **Grado actual de estudios**: ________________________________
5. **Horas estimadas de experiencia jugando juegos de kinect:**
   - ○ 0 a 10 horas.
   - ○ 10 a 30 horas.
   - ○ 30 a 60 horas.
   - ○ 60 a 100 horas.
   - ○ Más de 100 horas.
6. **Horas estimadas de experiencia jugando shooters:**
   - ○ 0 a 10 horas.
   - ○ 10 a 30 horas.
   - ○ 30 a 60 horas.
   - ○ 60 a 100 horas.
   - ○ Más de 100 horas.

Sobre el juego...

7a.	**Me encantaría jugar una vez más**	No, yo paso.	○—○—○—○—○—○—○—○—○	¡Claro que sí!
7b.	**No me dí cuenta del paso del tiempo mientras jugaba.**	Cierto, perdí la noción del tiempo.	○—○—○—○—○—○—○—○—○	Falso, siempre estuve conciente del tiempo.

Sobre tu concentración...

Comparando el juego con un episodio de máxima concentración que hayas vivido recientemente...

8a.	**¿Qué tan bien te pudiste concentrar?**	Nada, me la pasé pensando en otras cosas.	○—○—○—○—○—○—○—○—○	Totalmente, igual que en el episodio de máxima concentración.
8b.	**¿Qué tan difícil fué concentrarte?**	Fué muy fácil, no tuve que esforzarme para mantener mi atención.	○—○—○—○—○—○—○—○—○	Muy difícil, tuve que esforzarme mucho.

Sobre los retos que enfrentaste...

9a.	**En general, ¿qué tan difícil te pareció el juego?**	¡Muy fácil!.	○—○—○—○—○—○—○—○—○	¡Muy difícil!
9b.	**¿Qué tan difícil fué coordinar los movimientos de tus manos para destruir a los enemigos?**	¡Muy fácil!.	○—○—○—○—○—○—○—○—○	¡Muy difícil!
9c.	**¿Qué tan difícil fué mover al personaje?**	¡Muy fácil!.	○—○—○—○—○—○—○—○—○	¡Muy difícil!
9d.	**¿Qué tan difícil te fué esquivar a los enemigos?**	¡Muy fácil!.	○—○—○—○—○—○—○—○—○	¡Muy difícil!
9e.	**¿Qué tan difícil te fué pensar en una manera efectiva de jugar (matar más enemigos y recibir menos daño) ?**	¡Muy fácil!.	○—○—○—○—○—○—○—○—○	¡Muy difícil!

Sobre cómo te sentiste...

10a.	**¿Te sentiste cohibido o avergonzado?**	No, para nada.	○—○—○—○—○—○—○—○—○	Sí, bastante.
10b.	**¿Te sentiste en control de la situación?**	No.	○—○—○—○—○—○—○—○—○	Sí.
10c.	**El nivel de dificultad de la actividad comparado con el nivel de mis habilidades fué:**	Muy bajo, el juego es muy sencillo.	○—○—○—○—○—○—○—○—○	Demasiado alto, el juego es demasiado complicado.

11a. ¿Qué te pareció la experiencia de haber jugado el juego?

____

____

____

____

____

____

____

List of Figures

List of Tables