LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN
Department "Institut für Informatik"
Lehr- und Forschungseinheit Medieninformatik
Prof. Dr. Heinrich Hußmann

Bachelor Thesis

Design of Various Controls and Motivation Aspects for a Multiplayer Biofeedback Game Using Unity 3D

Manuel Frederik Graf

info@manuelgraf.com

Matrikelnummer: 10335080

Bearbeitungszeitraum: 01.02.2014 bis 21.06.2014

Betreuer: Dr. Alejandro Mendoza García (Technische Universität München, Institut für Informatik VI)

Hochschullehrer: Prof. Dr. Alois Knoll (Technische Universität München, Institut für Informatik VI)

Abstract

Nowadays more and more children and teenagers spend their time watching television or playing video games instead of engaging in sports. We can use this knowledge and use video games for rehabilitation purposes with patients which have heart diseases or recently had heart surgery, especially if a strict cardio training is needed. This thesis describes the design and integration of control mechanisms and motivational aspects into a video multiplayer game using an ergometer in combination with a joystick or a motion sensor. A preliminary study with young healthy students showed that the developed game motivated participants to replay and an aerobic exercise level can be reached, even during short sessions. Some aspects in the game aim to motivate players to exercise by competition and biofeedback as well as to increase or decrease the intensity of the player's physical activity. This thesis analyzes the game's motivational features, using current psychological approaches, and proposes further improvements. Still, future longitudinal studies should evaluate the success of the designed principles on long-term motivation.

Inhaltsangabe

Viele junge Menschen verbringen täglich mehrere Stunden vor Computerbildschirmen und Fernsehern und finden entweder keine Zeit für sportliche Betätigung oder werden eben durch diese Geräte in ihren Bann gezogen. Dies kann man nutzen, gerade wenn ein rigoroses Ausdauertraining benötigt wird, wie beispielsweise bei der Rehabilitation von Patienten mit Herzdefekt oder kürzlicher Herzoperation. Diese Arbeit beschreibt die Gestaltung und Implementation eines Kontrollmechanismus und Motivationsfaktoren für ein Mehrspieler-PC-Rennspiel unter Verwendung eines Ergometers in Verbindung mit einem Joystick oder einem Bewegungssensor. Eine vorläufige Studie mit jungen, gesunden Studenten zeigte bereits, dass das Spiel sie zum erneuten Spielen motivierte. Desweiteren konnte ach in kurzen Trainingseinheiten ein aerobes Trainingsziel erreicht werden. Bestimmte Aspekte des Spiels sollen den Spieler durch Wettbewerb mit Mitspielern und Biofeedback animieren Kardiotraining zu betreiben und dessen körperliche Anstrengung zu regulieren. In dieser Arbeit werden die Motivatoren des entwickelten Spiels anhand von aktuellen psychologischen Erkenntnissen analysiert und Vorschläge unterbreitet, wie diese noch verbessert werden können. Zukünftige Langzeitstudien sollten die hier entwickelten Konzepte in Hinsicht auf ihre motivierende Wirkung prüfen.

Scope

This project is done in collaboration with the German Heart Center and and the Informatics department from the Technical University of Munich. The goal of the project is to research and develop systems that help children with heart disorders to recover physical fitness after having an surgical treatment. The system uses electronic games to help motivate the children to perform exercise. Playing games will cause the children to perform moves that are supporting the rehabilitation process. The patients physical engagement is measured by biomedical sensors and used for controlling the intensity and frequency of moves that are stressed by playing the game.

The objective of this thesis is to develop a multiplayer game under the Unity 3d game engine to be integrated into the biofeedback system. With this game engine virtual environments and characters may be easily created. The game will receive user actions such as hand movements, peddling from an ergometer, posture, which should influence the virtual character on screen. Several levels of difficulty should be created to be used by the biofeedback system depending on the user physical condition.

Confirmation

I confirm that I independently prepared the thesis and that I used only the references and auxiliary means indicated in the thesis.

....

München, den 31. März 2014

Table of Contents

1 Introduction

1.1 Motivation

Recovering from heart surgery can be a very long process, in which patients often need special treatment such as physiotherapy and additional cardiovascular exercise. Since the effectiveness of these exercises is based on constant repetition, especially younger patients like children and adolescents might get bored in the process. Video games have proved to be an effective way of engagement for children, easily getting childrens' attention and will to play. According to a report from the NPD Group, a leading market research company in the USA, the number of children engaging in video game play rose consistently over the last few years [1][2]. Taking this into consideration, one possibility to increase children's motivation for exercise seems to be to connect the physical exercise to a video game that helps patients regaining their physical fitness while also entertaining them. Additionally playing against an opponent such as another person or a virtual character may also bring motivation to the player [3].

Therefore the main motivation for this thesis is to develop a racing game with the Unity 3D SDK that offers intuitive, body activating controls and biofeedback to increase the player's motivation to exercise as well as to adjust intensity of physical activity (PA) depending on the measured heart rate (HR). The game allows for multiple players to compete against each other or play individually. Also, a Software Framework “AutoMedic” with network capabilities serves as an Interface to different sensors for the tracking of player data under the supervision of a medical assistant, who can change game or exercise related parameters. The game itself aims to motivate children and adolescents that need or had surgical treatment of heart disorders, to accomplish their cardiovascular training and regain physical fitness.

The game and AutoMedic implementation should offer the possibility to play via LAN, Internet or with two Clients on the same PC. This thesis is a collaborative work, where one part focused on the development of the game's multiplayer requirements [4] while the interaction and motivation aspects are the objective of this thesis. Additionally, this thesis covers the design of player controls and usages for the biofeedback information. To discover how playing the game and different control mechanisms affect autonomic functions of the players body, preliminary test sessions in collaboration with the University of Applied Sciences Munich were carried out with a small number of technically and medically skilled students. These alpha tests for the game *MagiKart* and the subsequent questionnaire allowed the detection of potential difficulties and problems concerning gameplay and controls. Once these issues are solved the game may be introduced for rehabilitation.

1.2 Background

1.2.1 Serious Games and “Exergames”

In the last decade it has been widely assumed that playing video games would harm the mental and/or physical health of children. Especially discussions about the influence of some video games on children's aggression potential and about the addicting factor of online-based games arose [5]. Nevertheless game developers and behavioral scientists gained more interest in the field of Serious Games (SG). Nowadays, these games are used for various purposes such as military, education or healthcare.

Their main purpose is not just simple entertainment but in the case of healthcare "to entertain players while attempting to modify some aspect of their health behavior" [6]. Given a rising number of children are overweight [7]**,** show too little PA and the fact that video games are becoming more popular and ubiquitous one can easily conclude that connecting physical exercise to a computer game is a way to motivate children to engage in physical activity.

This conclusion is already but not exclusively concept-proven by *Nintendo's Wii* console. The *Wii* offers different types of fitness games in which the player can simulate actual sports like Tennis or Yoga with specifically designed game peripherals. Also, these games are not specifically designed solely for children but for entertaining the whole family. Some of them even show the player's body mass index (bmi) progress over time and thus motivating them to exercise and to compete with other family members. Other systems that use this approach are Motion Sensors for Microsoft Xbox ("Microsoft Kinect") and Sony Playstation ("PS Move"). Most of the games for these systems would not be classified as SGs but Exergames (EG, Exercise Games). Whether to fight childhood obesity or to help children recover after surgical heart treatment, EGs are an easy way to promote PA.

1.2.2 Biofeedback

Biofeedback can be defined as the process of providing a person with perceptible evidence of otherwise unperceived autonomic physiological functions like heart rate or blood pressure [8]. This makes the patient understand the reactions of his body to specific events up to being able to consciously manipulate these factors. Biofeedback is already used in different kinds of therapy ranging from simple Relaxation like *mindrest.me*[1], a biofeedback head mounted display, to treating diseases like chronic pain, high blood pressure or migraine headache. In conjunction with what already has been said about SGs and EGs, biofeedback becomes even more interesting in terms of player motivation and education about the right cardio training intensity.

Therefore adding biofeedback to an EG helps the player to understand his own body reactions and limitations in relation to certain events or exercises. Students of the TU Darmstadt researched the problems of existing EG like *Wii Fit, Wii Sports* or *EA Sports Active.* The research found that these EG's are missing the consideration of current vital parameters in gameplay [9]. To address this problem, the developed game for this thesis uses Biofeedback response and change of in-game content depending on vital parameters to address this problem.

Dr. Alejandro Mendoza Garcia from the Technische Universität München and the German Heart Center in Munich attempt to equip video games created by students with biofeedback to research the impact of different game designs, control concepts and biofeedback itself on the body response. The developed games' goal is to be used efficiently for rehabilitation.

One of the mentioned biofeedback games is the "Road Game" **[10]**. The goal of the game was to collect coins over a generated track with inclines and declines. It has been tested and evaluated by students of the University of Applied Sciences Munich. The results of the road game test sessions stated that the desired training effect can be accomplished, but the gameplay itself was not engaging enough to play it regularly. Due to this and some other minor problems, the idea arose to create motivation by competition and a more popular game concept. The racing game *MagiKart* (Magic and Kart), designed and programmed by Manuel Graf and Michael Prummer, sought to offer a game concept similar to the well-known Nintendo racing game "Mario Kart".

1 http://entspannungsbrille.de/

2 Tools

This chapter will briefly describe the different tools and important software programs that were used for the development of the game. 2D graphics and textures were created using Adobe Photoshop CS5. Creation of 3D Meshes and application of textures was done via Autodesk Maya 2013. Game Sounds and Music were created with Audacity 2.0.5 and Apple Garageband. The most important software frameworks used are the AutoMedic Framework and the Unity 3D SDK.

2.1 AutoMedic Framework and MDC

The *AutoMedic Framework* was developed in the *German Heart Center* by Dr. Alejandro Mendoza as a system that enables the quick interconnection of different components such as peripherals, sensors and actuators ([11],[12]). It provides a graphical interface for a type of visual object oriented programming language. The main focus of the framework is the automation of medical devices. It consists of several libraries programmed in C++ and Qt, which is a cross-platform user interface (UI) framework for C++. These libraries contain different components which can be loaded dynamically depending on the need of a specific application. The applications themselves are a set of files written in XML (Extensible Markup Language) containing the used components and their interconnections. The Medical Device Control (MDC) is a part of the AutoMedic Framework that automatically loads the configuration file and the needed libraries required for a specific component.

An essential component within the AutoMedic framework is the abstract class *BaseObject.* It provides dynamic arrays for input and output ports (pointers) which provide interconnection between components and a *calculate()* method to perform calculations whose results can then be sent to other components via output ports.

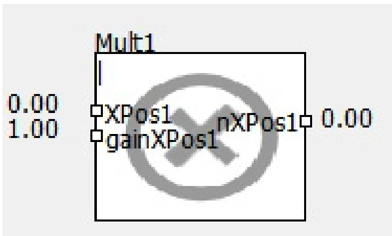


Figure 1: *Basic multiplier component that multiplies input values (left) and sends it to another component via the output port (right)*

Some essential components for the development of the game this thesis describes were the following:

- A *Motion* component that loads drivers for usage of the *Asus Xtion Motion Sensor*. Libraries used to connect to the Motion Sensor and to track user's movements by skeleton detection were OpenNi[2] and NiTE[3].
- *Object* components which are general purpose components, derived from the *BaseObject* class that are used for interconnection and control. These can define constants, be graphical user interface (GUI) elements, mathematical functions or even *Object* Groups that contain other *Objects.*

2 http://www.openni.org/
3 http://www.openni.org/files/nite/

- A *GameConnect* component (*Figure 2*), which automatically detects instances of running Unity games and connects to them. It continuously sends specified values to the game via the *Transmission Control Protocol* (*TCP*). This component can also receive values from the game, which the AutoMedic Framework can visualize or store in a database.

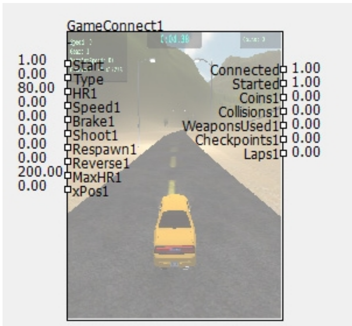


Figure 2: *GameConnect component with input and output ports*

- Additionally specific components containing drivers and libraries for different sensors and peripherals like gamepads, the *Kettler ergometer* or the *Zephyr BioHarness (Figure 3)*

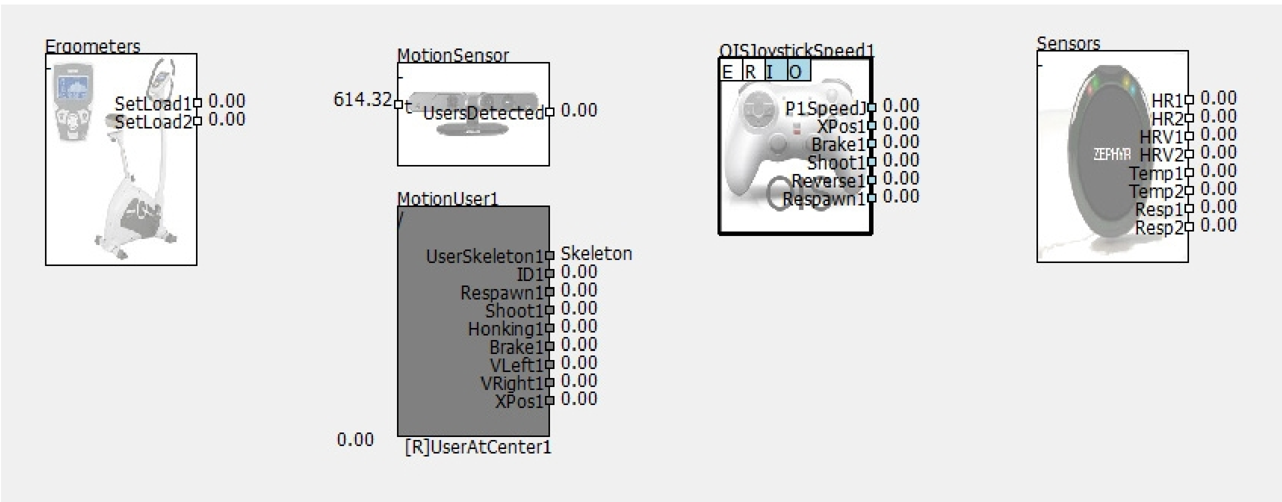


Figure 3: *AutoMedic MDC components for ergometer, motion sensor, gamepad and Bioharness connection*

Beside the pure technical side of loading libraries and drivers, the purpose of the AutoMedic Framework is to enable a medical supervisor to adjust the game or patient related values based on the visual feedback about the patient's vital parameters. He can, for instance, alter user inputs to boost or lower a player's virtual speed or override the target or current heart rate for a player of the current training session.

Using the AutoMedic Framework holds various advantages:

- There is just one application for controlling the training session and visualizing the patients vital parameters.

- arbitrary peripherals can be connected but AutoMedic sends the same normalized information to the game. (holds the risk of ambiguous signals)
- Remote control of the *GameConnect* possible due to communication over network. (Holds the risk very high network traffic when connecting to many components with one AutoMedic Instance.)

2.2 Unity 3D SDK

There are some complimentary popular game development authoring tools, softwares and frameworks accessible on the market. For the responsive biofeedback game *MagiKart* Unity SDK version 4.1 was selected as a developing environment. The Unity 3D SDK offers state-of-the-art DirectX11 graphics for Windows and makes extensive use of fallbacks to ensure that all Shaders run smoothly on any target device and operating system. This is desired so the game can be used online, regardless of which operating system is used in the particular clinic. For now, the AutoMedic Framework is just tested on *Windows* and *Linux* operating systems, but it holds the possibility to run on *Mac OS X* as well.

Figure 4: *Unity 3D SDK Logo*

Unity working stages are always different S*cenes* which offer a hierarchy for *GameObjects*. GameObjects are containers that can either be empty, contain child GameObjects hold *Components*, which implement actual functionality[4]. These Components control every behavior of the GameObject they are attached to. These behaviors can contain audio, rendering, physics or animation options .

One can also attach own scripts written in either C# or Javascript which all serve as behavioral scripts, not as instantiable classes per se. *Scripts* can be accessed from other GameObjects after placing a GameObject that holds the script into the *Scene*. This implies the programming part of an Unity game is different from other object oriented framework programming, because it is not possible to simply instantiate GameObjects and set values via constructors like in other programming environments. The upside is the visual WYSIWYG Editor of Unity that allows manipulation of public variables in the Component Inspector (*Figure 5*).

4 Unity Technologies, 2014, under: https://docs.unity3d.com/Documentation/Components/class-GameObject.html

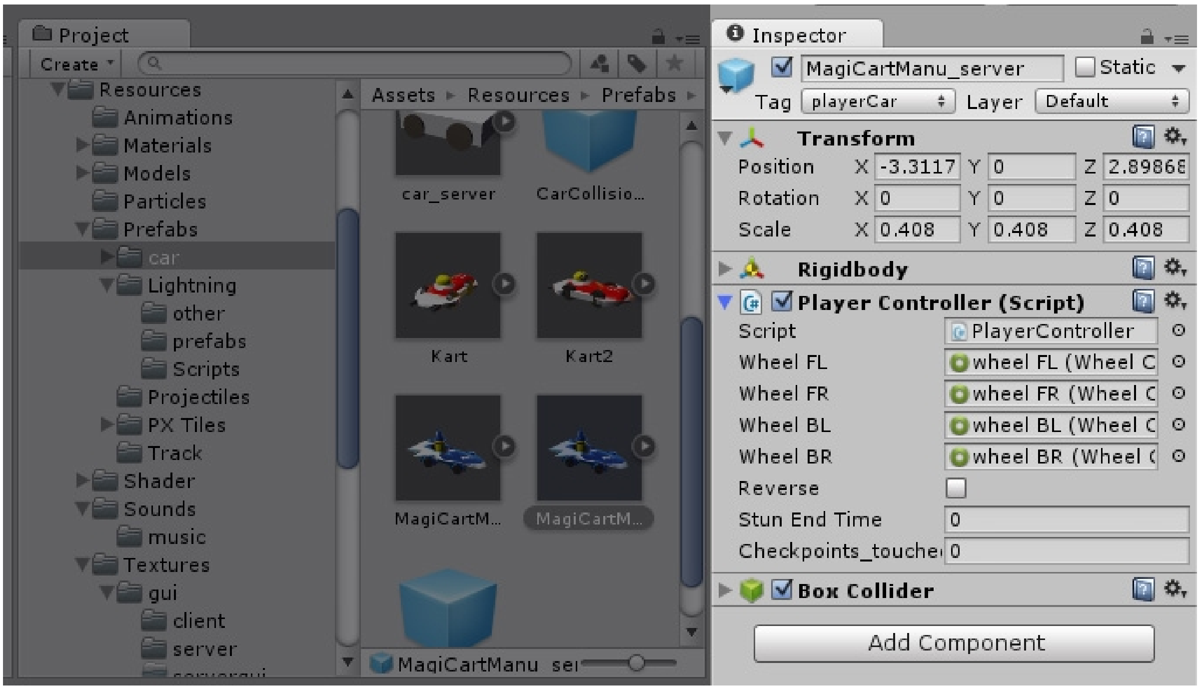


Figure 5: *Unity component Inspector that shows components attached to a vehicle, granting it physics (Rigidbody), collision (Box Collider) and controlling behavior (Player Controller).*

This also holds risks for the maintainability of the project code. Values set in the editor overwrite default values written in the code, which can produce nearly impossible to track (from MonoDevelop) inconsistencies for programmers.

The main reason for the usage of Unity was the fact that the scripts needed for the AutoMedic communication already existed for former biofeedback games created for the German Heart Center ([10], [13]). The First Scene loaded in the game client is the *Initialize* Scene that listens to the specified TCP port. If a *GameConnect* component uses the same port, Unity establishes a TCP connection to the AutoMedic Framework. Unity then receives a package of values and a 1 Byte identifier that signals that initial settings are sent. Afterwards, Unity continuously receives packages with another Byte identifier, marking packages as input values specified in the GameConnect component and stores the data within the *MDCServerData* script where they can be accessed by other client scripts.

However, the Unity SDK is a great choice for creating no-budget or prototype video games due to reasons like performance overview, easy implementation of custom shaders, support for version control, support of all major platforms and last but not least the active online community and sheer mass of tutorials.

3. Development of the Multiplayer Biofeedback Game

3.1 Game Ideas

In the beginning of the project it has been sought to create a biofeedback game that helps people exercise. Sketches of some games were created which combine the multiplayer component with the usage of vital parameters and hold the possibility to implement motion sensor controls. Our first idea was to create a series of competitive minigames. Like the "Tesla Coil Zap" game where the player has to shoot incoming objects with lightnings spawned from a screen centered Tesla Coil that is strengthened by paddling and the reduction of the target heart rate error.

Given the limited time we had to submit the thesis, we have instead decided to create a multiplayer racing kart game, which can by controlled by an ergometer combined with either a Joystick or a motion sensor. Others reasons for the chosen path were the increased immersion effect of a racing game while cycling and an ergometer allows better control of the stress put to the patient, increasing and decreasing his heart rate. Additionally, adding a multiplayer element to the game should increase players' motivation.

3.2. Target Audience and Setting

The target group for *MagiKart* are children who underwent surgery or suffer from a heart disease or disorder and therefore need to be encouraged to exercise regularly. Given the fact that multiplayer EGs can have a positive impact on the exercise routine, at least if there is parental or peer participation [14], the intention of the game was not solely to appeal to children but to adolescents or even adults as well.

One of the initial aspects proposed in the game was to add an educational element additional to the entertainment itself. We began to think about factors that motivated us playing video games as we were children. Regarding these, health education was not assumed to be a motivational factor for playing a video game.

A study [15] performed by the Massachusetts General Hospital also shows what is children's motivation for video game play (*Figure 6*). The study collected data from 1.254 public school students, almost all of them between the ages of 12 to 14, on their motivation to play electronic games.

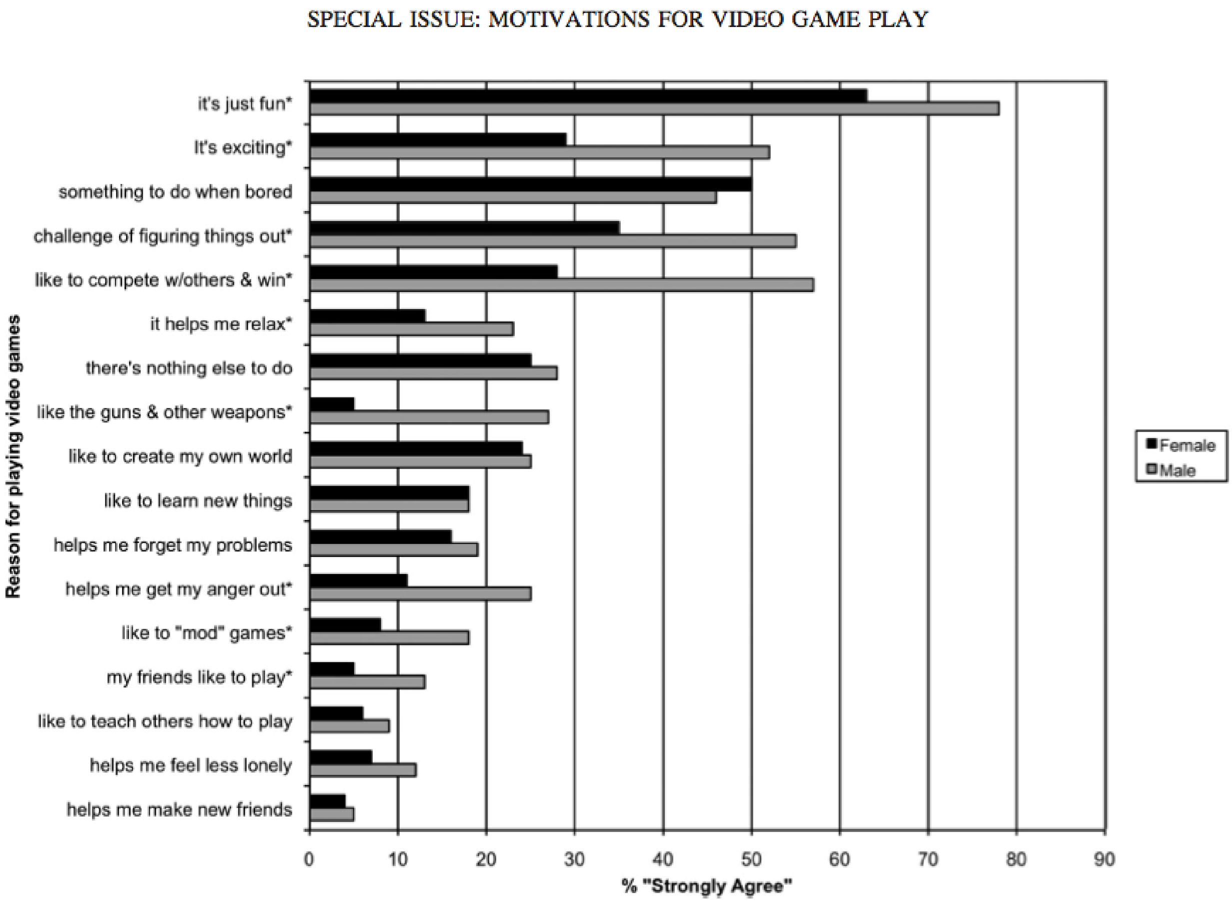


Figure 6: *"I play electronic games because. . . ." Reasons for playing video games for boys and girls. [15], p.181.*

The students had to rate 17 motivational aspects on a 4-point-scale. Only 18% strongly agreed that they are motivated by learning new things. Although this might seem like a high percentage, the "learning new things" option was just the 13th most option rated as "strongly agree" among the boys and the sixth most commonly picked one by the girls'. The most commonly rated options were "It's just fun", "it's exciting", "something to do when bored", "challenge of figuring things out" and "compete with others & win" [15].

Considering these results, making the game "serious" was not as necessary. Seeing that for most children the actual fun of playing a game and to compete with others seems to be the greatest motivation, it allowed us to create an entertaining cartoon setting without having to highlight unhealthy behavior, which might have been boring or even discouraging. Playing *MagiKart* with an ergometer should be sufficient enough to teach the player to which extent he should exercise by biofeedback and in-game rewards. Focusing on the aspect of entertainment classifies the game rather as an EG than a SG, although there is still an educational nature to the game since it can teach the player to some extent about his body and it's reaction to a specific amount of exercise. Additionally, in the long term, it can show more physically concerned players a certain improvement of conditions.

To emphasize the entertaining nature of the game characters, sound effects, music and game objects were designed comically. The playable characters ("Minis") and their voices are faintly reminiscent of the currently popular *Minions* from the computer-animated 3D comedy movie *Despicable Me*[5]. These Minions

5 Despicable Me, Dir. Coffin, P., Renaud, C.. Universal Pictures, 2010. Movie. In: Internet Movie Database: http://www.imdb.com/title/tt1323594/

as well as Minis are far from being associated with actual human beings, they are rather seen as punchbags. This association and slapstick-like voices and sound effects soften the remorse about using weapons against your enemy players, therefore encouraging their usage. The weapons/power-ups have a magical visual style and appearance, to further enhance the desired the comic fantasy look and feel.

3.3. Game Mechanics

Working with ergometers prompted developing a racing game with a commonly known game concept to easily achieve higher immersion for the player. In the race mode the player steers a vehicle on a circular racing track, trying to reach the goal faster than other players. Also, in *MagiKart* features from different popular racing games were adapted to enhance fun while playing the game with other players as well as with just one player. At predefined checkpoints the player can pick up useful power-ups, which are covered in Section “3.5.4. Power-ups“.

For players who do not enjoy racing and shooting an alternative game mode has been implemented. Here, the players will not be able to pickup power-ups. They can collect coins instead and by the end of the training session,, the player who collected the most coins will win the competition.

A main part of developing the game was the multi user support, designed and implemented as described in [4]. This was achieved by programming an authoritative master server application in Unity that allows players to compete either via local area network or the internet. One can easily connect to any started game in the whole network. Even if there are no real fellow players to play with, the user can choose to race against simple AI karts. A split screen functionality was not implemented but instead the same result is achieved by running 2 clients on the same computer and controlling both of them via one MDC instance that gathers two players' inputs and sends the data to two independent GameConnect components.

3.4. Game Elements

Apart from serving their main purpose, the game elements in MagiKart had to consider the player's heart rate, affect his or her's specific PA intensity and last but not least make the game more fun to play and thus increase the player's motivation. Calculating every game objects behavior within the server also made it necessary to create most elements twice, once for the client and for the server side. The server does not need to cope with the extensive use of particle systems and visual effects while the client is relieved of the calculation of physics and collisions can therefore focus on receiving the positions of GameObjects. This was achieved by either creating two versions of the same asset which differ in the usage of particle systems (none for the server asset) and colliders (none for the client version) or by setting a boolean flag in the assets script that holds the information whether it is used on client or server side.

3.4.1 Tracks

When it comes to racing games, a huge motivational aspect for players are the tracks they are driving. We considered some workflows for creating them. Firstly it had to be decided whether to create complete levels that consist of a single static 3D-Mesh or dynamic levels that uses tiled and reusable portions of tracks to create a higher variety of tracks.

Dynamically Generated Tracks

Dynamically generated tracks have some advantages over static ones. Since a training session always has to be supervised by a medical assistant, he or she could alter the course of the game, e.g. when the track goes up- or downhill. This holds the possibility to make the world adapt to the players specific condition up to forking the track into different selective routes which only are drivable for specific players.

This is especially interesting when the ergometer's load depends on the track's slopes. (Which it does in the *Road Game*, but not in *MagiKart*). Considering long-term motivation, dynamic tracks are more entertaining than static ones, because their appearance changes every time the user plays it, therefore offering a new environment in every game session. This can be a motivating factor for players who like to explore. Nevertheless creating such a subsystem that also allows to generate circular racing tracks would go beyond the constraints of this thesis.

Static Tracks

On the other hand, there is the possibility to create static tracks which consist of a big 3D Mesh that can be imported directly into Unity 3D. Static tracks hold the advantage to plan the course's appearance. Therefore it is easier to estimate the difficulty of a track, for being able to eliminate all difficult parts of the track where users are struggling. It can also be implemented in a shorter amount of time as it could with a dynamic generated track, especially if the tracks are circular and contain parts like crossing roads with ski jumps. Nevertheless, this comes at expense of maintainability of the tracks because we, as well as programmers that use the system afterwards, needed to manipulate the original file of the track and reimport it every time a small change on the track's appearance is needed.

Solution: Static Tracks from Tiles

We decided to implement a mix of static and dynamic tracks that combine both advantages and are extensible to further programming. *MagiKart* uses road parts, that can be easily put together in the Unity Editor, for each of them is a tile of 24 x 24 virtual meters. This enabled us to create a variety of tracks to drive on in a very short amount of time. The tiled parts were stored as *Prefabs,* so they can be edited independently, automatically updating all occurrences inside a scene. Hence, this system offers maintainability as well as extendability and scalability. It also paces the way for dynamically generating racing tracks. The first available track segments consisted of a straight road segment, a slope and a 90-degree turn.

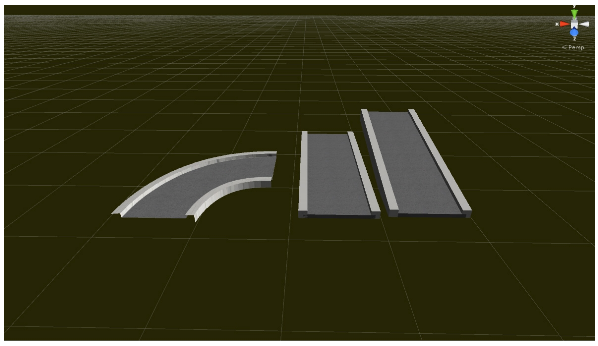


Figure 7: Available Road segments at the beginning of the test sessions

3.4.2 Vehicles

One of the main difficulties in implementing the game was the programming of the car. We decided to design the mechanics of the car to react almost physically correct, but we ran into problems concerning the vehicles behavior when boosting it's speed. For the center of mass has to be below the car to keep it on the ground, increasing the car's speed made the front wheels to loose connection to the ground. Therefore the *Windboost's* and Speedpad's bonus speed had to be reduced significantly. what lowered their impact immensely.

Using Unity's *Wheel Colliders* kept us from designing vehicles that do not use actual wheels, like spaceships or magical vehicles. Whilst wheels could just be emulated by invisible *Wheel Colliders*, the vehicle still would react like it had wheels. This could have been irritating for the player, so we decided to choose normal cars for vehicles (*Figure 8*).

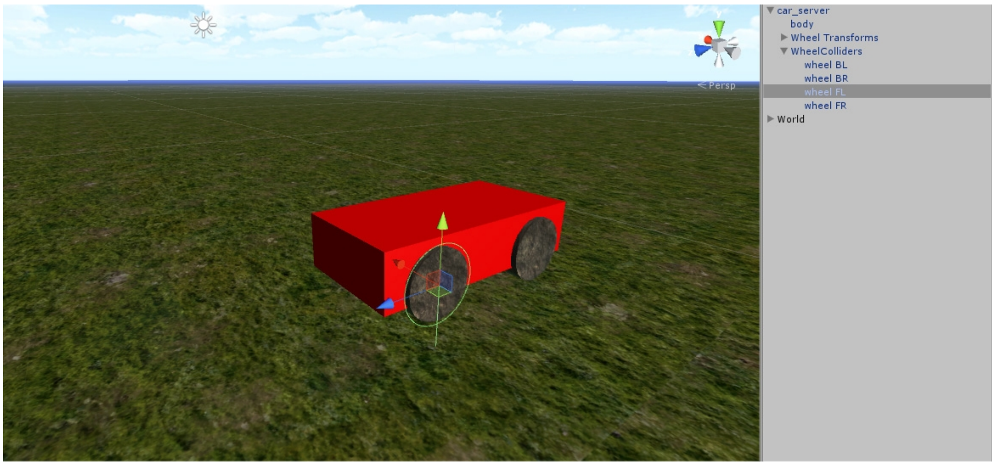


Figure 8: *Simple car model in Unity 3D Editor with one Wheel Collider selected*

3.4.3 Checkpoints

The checkpoints used in *MagiKart* are a crucial technical asset of the game as they have different functions. Yet, they have no motivational impact on the player, due to being invisible and the player doesn't get any notification but the elapsed time per lap. Basically a checkpoint is a trigger-type colliding node in a circular path that starts at the starting line of a track (*Figure 9*). The generated path serves as an ideal driving path that is used by the AI enemies as well as players in the “auto steer” and “auto drive” states. Besides that, they are needed for calculating each players race position in case they have absolved the same amounts of laps. They also serve as rescue point where the players car gets respawned. Depending on the game mode, either some selected checkpoints spawn power-ups or all of them spawn collectable coins.

Figure 9: *Top view of a track in Unity editor. The white dots represent checkpoints connected by a path*

3.5.4. Power-ups

Popular cartoon racing video games like "*Mario Kart*"[6], "*ModNation Racers*"[7] or "S*onic & SEGA All-Stars Racing*"[8] offer multiplayer support and the possibility to collect expendable power-ups along the race track by driving through a rotating box which is floating slightly above the track. This gives the player more or less critical boosts that help him or her win the race if used properly. These can either serve as a weapon to temporarily disturb other competitors or boost the abilities of the own character. The following section describes the different types of collectable weapons and boosts as they were implemented for the test sessions.

Four different types of power-ups can be collected within the game:

Name	Description	ID
Fireball	Straight linear projectile.	WeaponType.ROCKET
Thunderstorm	Very strong global disturber	WeaponType.THUNDER
Windboost	Vehicle speed boost	WeaponType.SPEEDBOOST
Ice field	Droppable area that slows pursuing enemies down.	WeaponType.MINE

Table 1: *Power-ups at the beginning of the test sessions*

There are many racing games that do not use weapons at all. Nevertheless one key motivation was "I play video games because I like the guns & other weapons" and almost 30% of the boys strongly agreed with this statement, making this aspect the 7th most strongly agreed option among boys [15]. Only 5% of the girls strongly agreed with this. After finding the visual setting for the game, all of the power-ups got a magical look and feel and the weapons appearance has been changed to be less gender-specific. The straight projectile became a fireball, the global weapon remained a thunderstorm, the "nitro"-boost became a wind-elemental boost and the droppable slow area got the visual style of frozen ground (*Figure 10*).

6 *Super Mario Kart.* SNES module. Japan: Nintendo EAD ,1992.
7 *ModNation Racers*. PlayStation Network Download. USA: Sony Computer Entertainment Worldwide Studios, 2010.
8 Sonic & Sega All-Stars Racing. Blu-Ray Disc, Wii-Disc, DVD-ROM. UK: Sumo Digital, 2010.

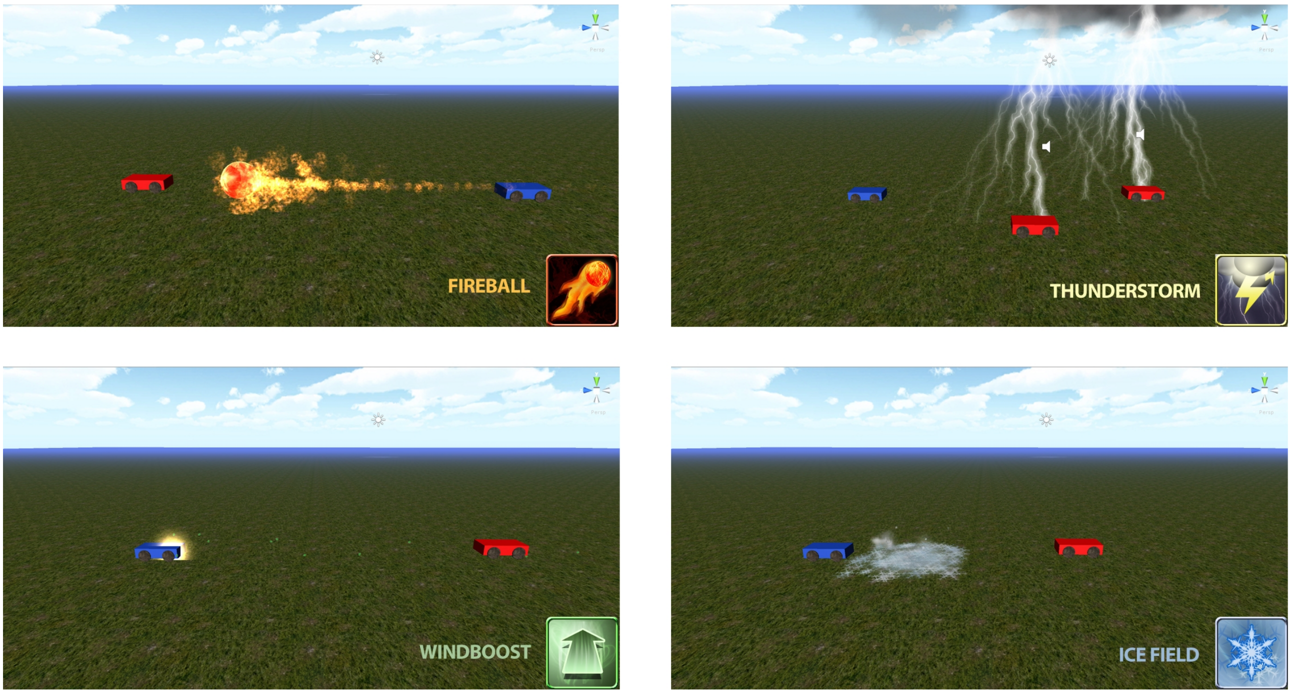


Figure 10: *Power-up appearances at the beginning of the test sessions*

3.5 Game Control Concept

Besides making a game that motivates children and adolescents to play regularly, body activating controls were needed to ensure the desired training effect of the game. The following section shall explain which control mechanisms were implemented into the game and why. Basically, we used an ergometer in conjunction with either a motion sensor or a gamepad to control steering.

3.5.1 Ergometer

The ergometers output is normally measured by the momentum of the flywheel that is inside of it. Since we implemented the vehicle to interpolate its velocity by script, we needed to get the information on the paddling speed not by the ergometers output itself but by mounting magnets on the pedals and a magnet sensor at a spot where the magnets pass by when the player cycles (*Figure 11*).

Figure 11: *Ergometer pedal with attached magnets and sensor*

The MDC counts the time between magnets passing by and calculates an average value of the paddles rounds per minute (rpm). To brake, the player has to stop paddling for two seconds. Cycling below 0.5 rpm sends a virtual speed of zero and stopping the car in the game. This holds advantages as well as disadvantages. This mechanic provides the game with the actual paddled rpm, which then can be used to calculate the virtual speed for the vehicle. Calculating this with the normal output of the ergometer would have undesirable results for the racing game.

Since the momentum of the ergometer would be used, the player had no option to rapidly brake or accelerate what can be useful in a fast-paced racing game. On the other hand this approach involves dangers that can be coped with. Cycling very slowly will result in a brake command, So this is a risk for slow cycling players. To cope with that, the minimum rpm before a break command is sent has to be adjusted by the medical assistant. Especially with increased ergometer's load this can lead to problems, because the load correlates with the momentum of the ergometer.

Normally, if the load is increased, cycling is harder, but also the ergometer's speed output is equal or higher even if the player is paddling slower. This does not hold for our approach. Increasing the load makes the player slower at an equal amount of exertion. This may seem counterproductive at first glance, but it is not for the reason that players need to be pushed to reach their target HR. If a player is currently in exert state, he has to put more effort into cycling to raise his HR. Therefore the ergometer's load is raised and thus the virtual speed is decreased.

This penalization is active until the player reaches his target HR or being in exert state for over 60 seconds. Reaching the training goal therefore grants not only the relief of putting less effort into cycling but also results in a higher virtual speed due to faster paddling. To adapt to the users fitness level the sent virtual speed can also be adjusted by the medical assistant via a steering gain value slider ranging from 0 to 1 (default 0.33) that allows to increase or decrease the sent virtual speed individually for each user.

3.5.2 Motion Sensor and Gamepad

The Motion Sensor is one of the possible input methods beside the ergometer. We used an Asus Xtion Motion Sensor to capture the player's movement. The AutoMedic Framework uses the OpenNi open source library to detect the user's skeleton and gestures. Using the motion sensor in conjunction with an ergometer, there are some limitations to gesture variety. Since the subject remains seated on the ergometer and it's handlebars occlude a big part of the user's lower body, skeleton detection is harder than usually. For this reason gestures can only make use of arms, head and torso properly. The motion is captured and evaluated by MDC and sent directly to the game via the *GameConnect* Node (*Figure 12*).

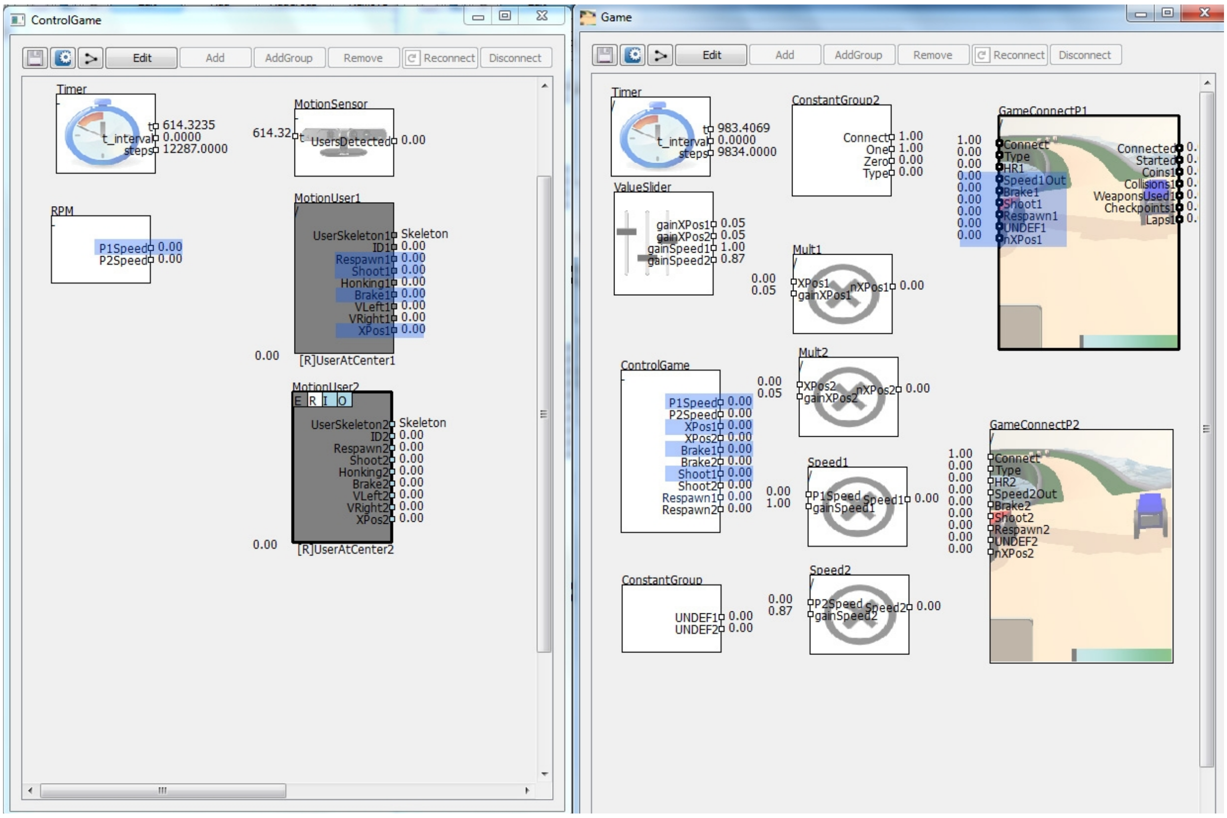


Figure 12: *AutoMedic setup for motion sensor controls. Values that are sent from the motion sensors and magnets are highlighted in blue.*

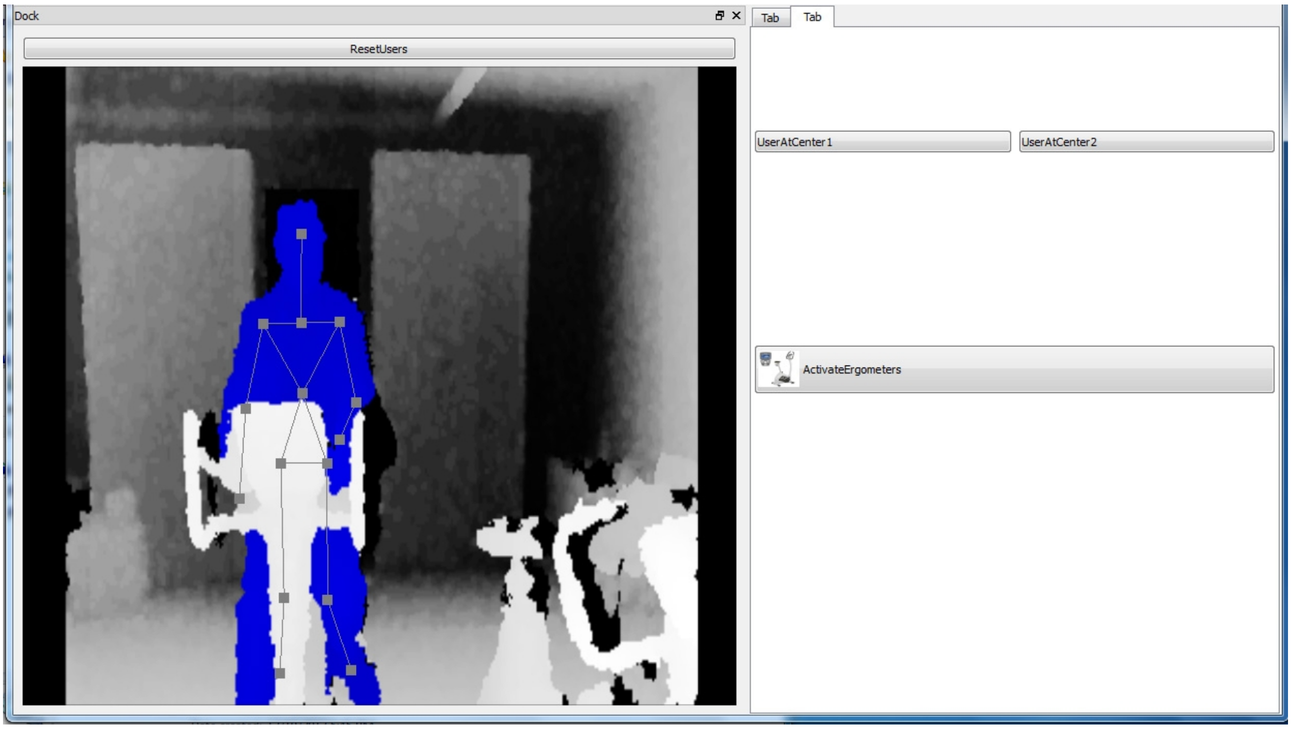


Figure 13: *Motion Sensor context window with one detected user. the GUI offers possibilities to re-scan for users (Button ResetUsers) and set the current inclination as centered (UserAtCenterX)*

For the steering part of the vehicle the inclination of the player's torso is calculated by averaging the x-axis position of head, shoulders and waist joints. The calculated value, normalized to range from [-10,10] within MDC, is used to determine the speed of the y-axis wheel rotation as well as the maximum angle. To adapt to the user's basic stance, the MDC needs to know exactly which position is "centered". Therefore a button was implemented in the Motion Sensor context window (*Figure 13*) that saves the current calculated inclination of all detected players as the centered position and uses this to calculate the inclination difference.

Since the gesture recognition requires recompiling the AutoMedic Framework, a slider was also implemented in the MDC interface to control a factor the calculated inclination is multiplied with, ranging from 0 to 1. This serves the purpose to adapt to the users flexibility. Since some patients that had surgical treatment have limited mobility, steering might have to be adjusted individually. Thus for every player that is connected via MDC there exist such a value slider to adjust his steering sensitivity.

Motion Gestures

To trigger events in the game, gestures were needed for actions like using a power-up, braking or repositioning the car to the last checkpoint (respawn) in case the player is stuck at some point. Intuitive gestures were sought, since [16] proposed that IC can affect and enhance both *presence* and *perceived competence* and therefore intrinsic motivation. Therefore, the following gestures were implemented (*Figure 14*):

- Using a power-up like shooting a fireball is triggered by an overhead throwing gesture, moving your left or right arm forward above the head (to prevent it from triggering while moving on the ergometer naturally).
- Resetting the vehicle's position to the last passed checkpoint (respawning) is triggered by clapping with both hands.

- Braking, or respectively driving backwards if the calculated virtual speed within the game is already 0, was firstly done by a gesture similar to applying an emergency brake (pulling down the emergency cord). Since that gesture was most commonly recognized erroneously and therefore interfered with the player's driving, an easier to recognize gesture was chosen, but at the expense of intuitive controls. The new gesture for braking was implemented to be triggered when both arms are lifted overhead simultaneously.

Since most player's in the test respawned themselves instead of driving backwards, the last gesture was not considered too important. The problem of wrongly performed player movement, mentioned in another study [17], did not affect our decisions since the set-up of the game always requires an inaugurated medical assistant supervising the game session, there was no need to explain the gestures within the game.

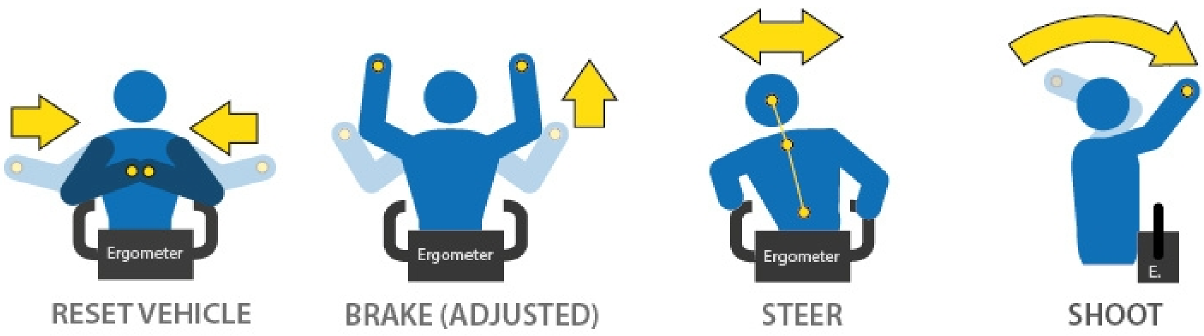


***Figure 14:** Implemented gesture controls. Yellow dots represent the most important skeleton joints for a specific gesture.*

Gamepad Controls

As another input method we used a gamepad, similar in design to a *Sony PlayStation 2* gamepad. Only 3 Buttons and the analog stick had assigned functionality. The stick was controlling the steering angle, normalized to the same interval as with the inclination recognition by the motion sensor, the shoulder buttons L2, R2 and the Circle-Button were used to trigger position reset, brake and shooting.

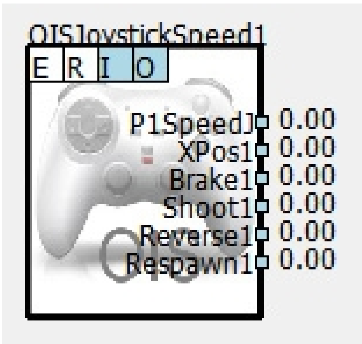


***Figure 15:** AutoMedic gamepad component with outputs for speed, steering and commands*

3.5.3 Graphical User Interfaces

For the medical assistant controls were sought to game relevant values according to the player's condition in addition to the existing automatic adaptations. There are two GUIs which allow such manipulations. Firstly, the GUI of the AutoMedic Framework, that allows to manipulate any value sent to the game for one player with previously discussed factor value sliders. Secondly, the GUI of the *MagiKart* Server Application, that offers possibilities to manage the hosted game and connected players. The separation of those makes sense since the AutoMedic MDC mostly serves mainly the manipulation of the clients. That way, the server application always stays independent, regardless of whether the MDC is used for a client or not.

The supervisor has full control over the training session with these two interfaces. Besides controlling multiplying factors for user input values, he can emulate user input, start the exercise, configure the the Motion Sensor or override the calculated maximum heart rate of a specific player via the AutoMedic Framework MDC application (*Figure 16*).

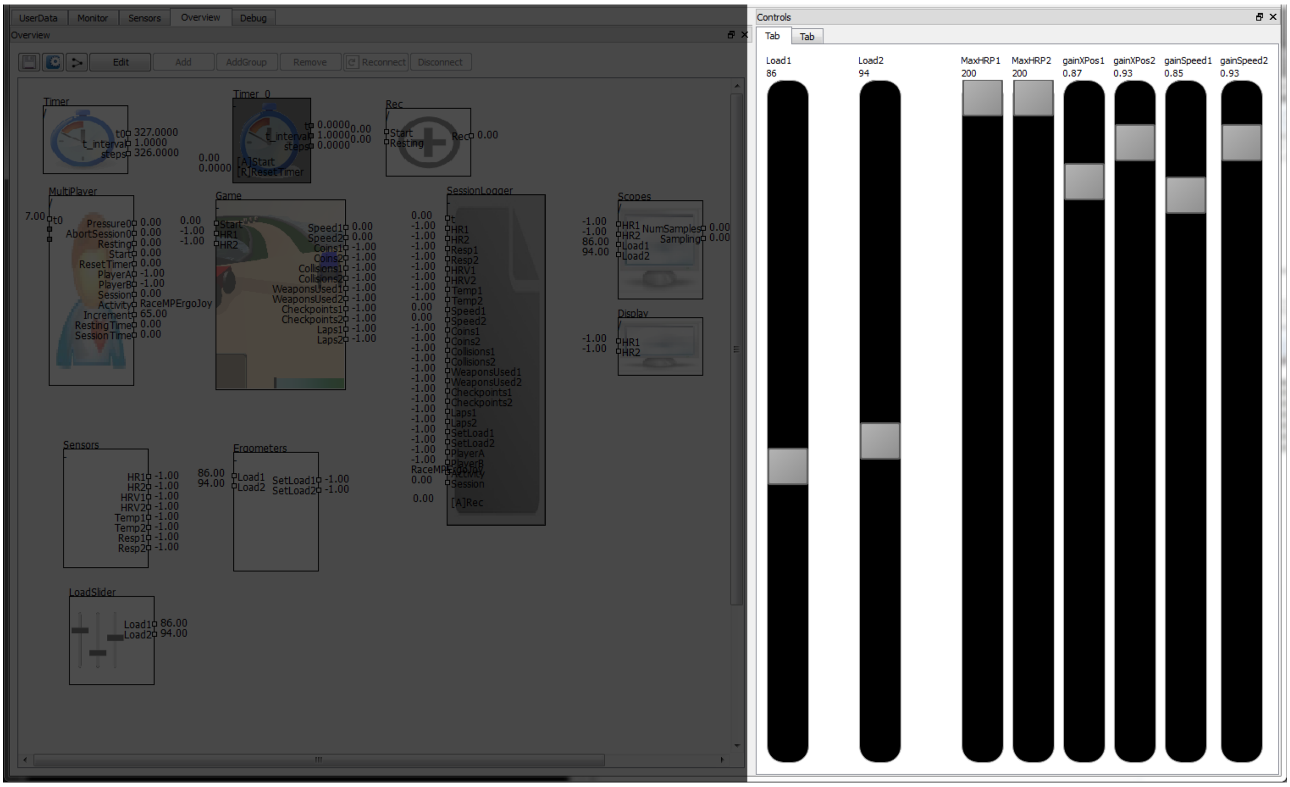


Figure 16: *AutoMedic MDC Interface for the medical supervisor. Sliders for values like ergometer load, maximum heart rate and speed and steer multiplier are highlighted on the right side*

The server application GUI offers controls for choosing the level that is going to be played, adding AI opponents, selecting game mode or granting players special benefits like a random power-up or auto-drive state (*Figure 17*). In the early stages, it also offered the possibility to emulate that a player got hit by a projectile in case he is stuck at some part of the track.

Figure 17: *MagiKart server application GUI for a running game*

3.6 Preliminary Study

Having a working prototype of the game with a basic control concept, multiplayer functionality and game design, preliminary studies were carried out in the context of a practicum of the University of Applied Sciences Munich. different EGs developed by the German Heart Center were tested in three sessions over a two weeks period, including the game *MagiKart*. The goal of the tests was to detect problems with the game and it's controls that prevent the game's usage for rehabilitation purposes.

3.6.1 Experiment Setup

Participants, all of them being technically and medically skilled and between 23 to 27 years old, were divided into different groups which were asked to perform different activities. 18 participants (13 male, 4 female, 1 not specified) were asked to play a *MagiKart* track while cycling and absolve 3 laps on the selected track. During the first week, the participants controlled the game with a gamepad, and in the second week with a motion sensor. After all the testing sessions were completed, the participants were given a questionnaire, which was designed to help determining the participants' perception of the game.

The setup consisted of two ergometers facing a wall the game was projected on and one PC that hosted two *MagiKart* clients, which were presented to the players, as well as the server application. A Motion Sensor was placed on a slightly elevated platform to prevent the players' occlusion by the ergometer handlebars as much as possible. While two players cycled and played the game, another participant had to control the training sessions as a supervisor.

Figure 18: *Experiment setup with 2 participants seated on ergometers and one supervisor*

3.6.2 Sensor Usage

Used sensors were two ergometers with magnets attached to the paddles and magnetic field sensors attached to the ergometer, to recognize the actually cycled rounds per minute (rpm). The *Zephyr Bioharness 3* chest strap was used to measures multiple vital parameters and sends it wirelessly to a receiver. In our case, we tracked the user's HR and respiration values over time to send it to the game. Electrodes are more likely to detach from the skin, especially when the users sweat or move a lot. Therefore we chose a wireless breast strap to capture vital parameters. For movement tracking, a *Asus Xtion Motion Sensor* was used.

KETTLER ERGOMETER

ZEPHYR BIOHARNESS CHEST STRAP

GAMEPAD

ASUS XTION MOTION SENSOR

Figure 19: *Used Sensors and peripherals*

3.6.3 MDC Implementation

The data accumulated on the participants was entered into the training application of the AutoMedic Framework and consisted of variables like age, weight, whether they smoke or not. The AutoMedic application was configured to send different values to the game namely cycling speed, steering angle, heart rate and boolean values of performed commands like shooting or respawning (*Figure 20*). The game returned values to the AutoMedic application for every player individually that were assumed to help understanding impacts of game events on the user's vital parameters. Those were the number of checkpoints passed, the number of collisions with walls or other players and the number of power-ups that were used (for the competitive racing game mode).

Figure 20: *AutoMedic GameConnect component with values sent to (left) and received from (right) the game*

As it was the first time multiple gamepads were used with the AutoMedic framework, there were initial problems to cope with, since the joysticks couldn't be distinguished. While it may not have been a problem playing with multiple computers and hence with multiple instances of the MDC, it certainly was when managing multiple clients with one computer. This obstacle was addresses by remapping the buttons of one gamepad.

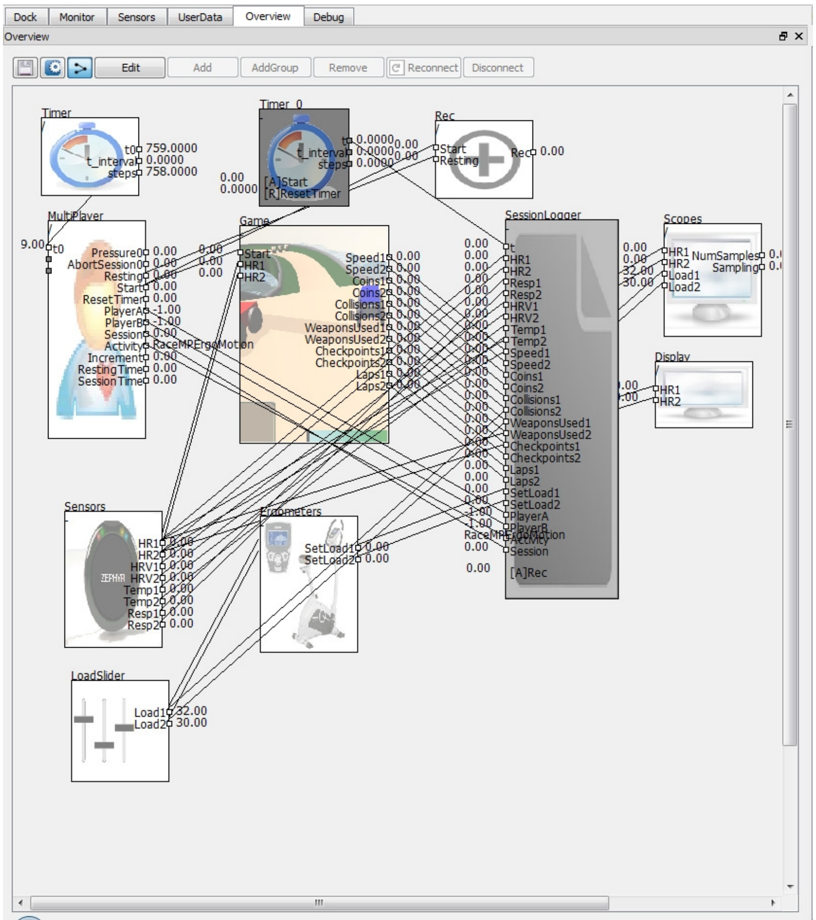


Figure 21: *AutoMedic setup (with lines indicating component connections) for the preliminary test sessions. The "Game" component in the middle is a group object that contains components like shown in* ***Figure 12***

3.6.4 Test Results

The sessions carried out by the students brought out a high number of program failures due to the connection of multiple sensors and peripherals required for multiplayer purposes. This was previously not as visible in single player configurations and needs further testing. For this reason data sets were reduced to a very small number of games played with the joystick (10) and the motion sensor (4). Additionally only 9 students sent in the questionnaire about their personal perception of the game.

Firstly, data that was collected in sessions which used the the motion sensor cannot be adequately compared to the ones that used a gamepad, since the steering by leaning to a side in conjunction with the cycling on the ergometer caused a lot of sensor misinterpretations, as participants stated in the questionnaire. This also was noticed in the players' average speed which was on average 40% lower compared to the gamepad, due to the high number of collisions and difficulty of the task.

Thus most sessions that used motion sensor input were canceled before one of the players reached the goal, since there was very limited time for each game. The average completion time for the track and the number of collisions also reflect this: The players average time for completing all three laps (measured time from reaching the first checkpoint to reaching the last) when playing with a gamepad was 4 minutes and 17 seconds with on average 19 collisions per run. Since most motion sensor game sessions were canceled before one player reached the goal, obtaining the allover completion time and collisions required estimation based on the passed checkpoints, respectively detected collisions, and the amount of time data was sampled. The estimated average completion time for sessions played with the motion sensor was 15 minutes and 54 seconds with approximately 147 collisions per session. From this we concluded that the controls we implemented via the motion sensor were much too difficult for players, especially in a fast-paced racing game like *MagiKart*, that requires not only to cycle fast but also to watch out for competitor, use power-ups and move your body correctly at the same time.

Looking at results of the survey, 44% of the participants found the motion sensor steering physically exhausting (*Figure 22*). Despite that they stated that the motion sensor controls were even more amusing than the gamepad controls, due to the heightened PA. Still, survey participants answered that they had fun playing *MagiKart* and would want to recreate the experience (*Figure 23*).

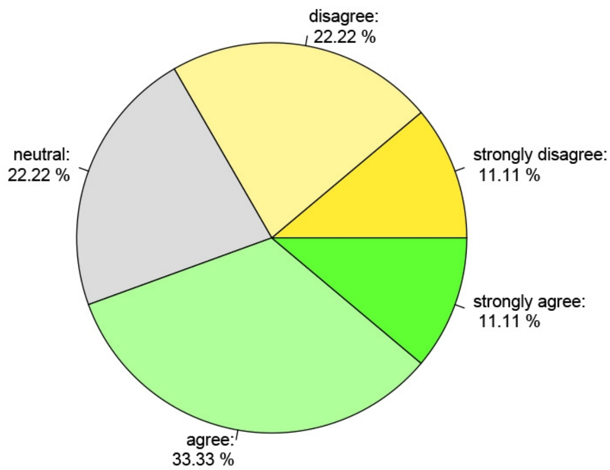


Figure 22: *Participants' answers to the statement "Controlling the game with a motion sensor and ergometer was physically exhausting"*

The data showed that the average heart rate of the participants in most cases reached the aerobic exercise goal of [0.6 * HR_{max}, 0.75 * HR_{max}] where HR_{max} is being calculated with the simple maximum heart rate formula *220-age* (*Figure 24*, *Figure 25*). Due to the limited data of the motion sensor sessions, it is not possible to adequately compare the effects of the control methods on the players heart rate. However, both input methods motivated players to put physical effort into the game. Since most participants' heart rate stayed on an aerobic level the duration of the training should be set to at least 30 minutes to achieve health benefits.

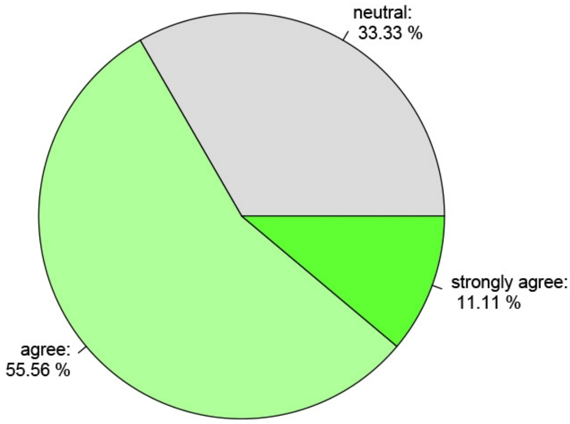


Figure 23: *Answers to "The game was fun and I want to play it again."*

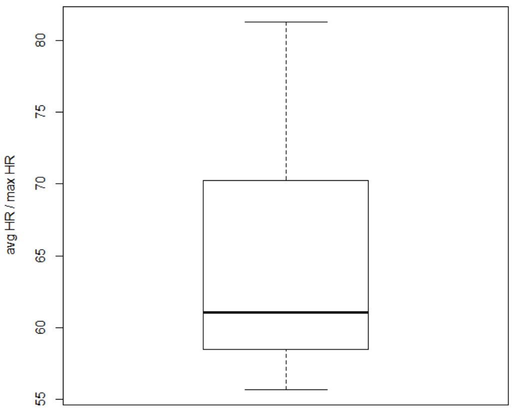


Figure 24: Percentage of maximum heart rate reached at average when controlling the game with an ergometer and motion sensor

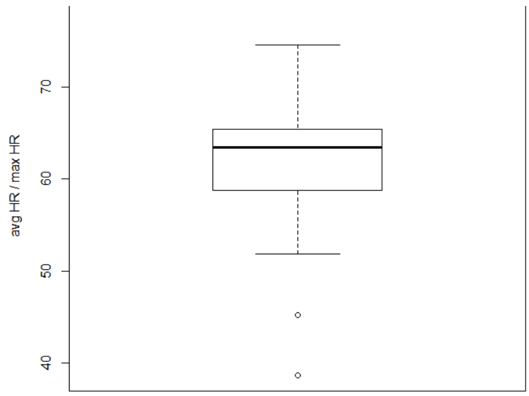


Figure 25: Percentage of maximum heartrate reached at average when controlling the game with an ergometer and gamepad

Comparing genders, female participants were even more likely to collide with the tracks walls as the male players (60% more collisions in one session with the gamepad, 36% more with motion sensor). This can be explained with the game being a virtual 3D-environment and that certain visual skills like spatial rotation or aiming are typically less developed in girls [15]. Other values did not have significant differences between the genders, although our data suggests that female participants were more likely to put higher physical effort into the game session with motion sensor controls than males. But it is supposed that these findings are inconclusive due to the small number of

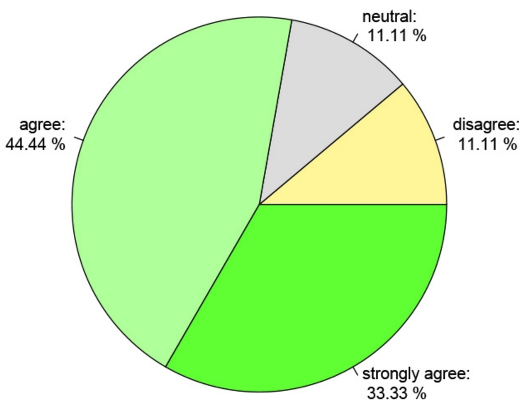


Figure 26: Survey participants' response to "The Game is suitable for children rehabilitation."

participants.

In additional comments, survey members were asked if they think that the game is suitable for children rehabilitation (*Figure 26*). Because of the game's capability to distract the player's attention away from the exercise itself, many of them agreed upon this as long as some major issues with the game were resolved (namely a properly working ergometer load adaption and easier-to-perform motion sensor controls). Additionally, they wanted more tracks and vehicles too choose from, what was the deciding factor to create multiple vehicles and stay with the approach of static tracks from tiles.

Concerning the power-ups, there were only some extreme examples that showed how the weapons affected the game's outcome. When supervisors “cheated” and distributed weapons requested by a player, this player had a significant higher chance to win the race. One survey participant even stated the power-ups had no effect at all. Nevertheless all of them agreed that the power-up system made the game more interesting and some of them were motivated to play the game because of it (*Figure 27*).

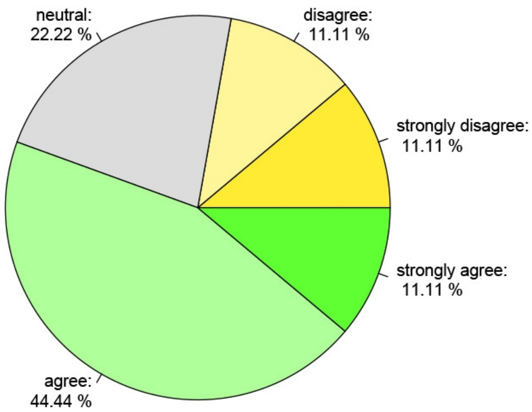


Figure 27: *"The collectable power-ups motivated me to play again"*

For the tests the power-ups were implemented with just one form. The *Fireball* Power-up shot a flaming sphere in the forward direction of the player. On impact with other players, they got thrown up in the air and were unable to move for 2 seconds. Although aiming was difficult especially for the female participants, players were always amused about hitting other players with the projectile. The *Thunderstorm* was by far the mightiest of all power-ups at this time as it globally hit all other players after a short delay, leaving them unable to move for a limited period of time. The fact that most of the participants sought this weapon indicated that it was overpowered. Since a huge problem with the motion sensor was the high steering sensitivity, and the track was very narrow at this point, using the *Windboost* mostly led to crashes with a wall. Instead of increasing player motivation it increased frustration. The same applied to the droppable *Ice field*, as it could easily be bypassed due to its small size.

3.6.5 Error Sources

Apart from the already mentioned sensor issues, some other error sources became obvious during the test sessions. Since all of the the participants stayed in the same room for the whole duration of the test sessions and the due to the multi player nature of the game, there was a lot of talking and laughing. While this is desirable to happen for a game, especially the sensor for vital parameters can be tricked by such actions, as it measures chest contraction to determine heart rate and respiration. To address this problem, one had to repeat the study with spatially divided player groups.

Another problem was the occlusion of the players by the ergometer's handlebars. This sometimes prevented the correct recognition of the player's virtual skeleton and the performed gestures, since whole parts of the player's lower body were not visible for the sensor. Given the steer control by leaning to one side or the other already interfered with the cycling on the ergometer, this also hindered properly controlling the car. Additionally, the ergometers' load change did not work properly. Problems with the used hardware resulted in load adjustments not being applied at all or being applied with an immense time delay.

Granting the participant that served as a medical supervisor abilities to influence the game did hold some problems for some of the offered options. In the first session, the supervisor's server interface contained an option to emulate that a player was hit by a Fireball or Thunderstorm to manually punish single players. This option was quite frequently misused as well as manual distribution of a random power-up for fun among peers, invalidating conclusions based on the used weapons.

Based on the experience of this test a proposed protocol is suggested for further analysis:

- A strict protocol should be established where resting and cool down periods are carried out and the start of the physical activity is synchronized with the start of the game to avoid falsifying results on average heart rates.

- Specific scenarios should be addresses such as what should define the end of a multiplayer session. If it it ends when one player finished a specific number of laps, the session can be terminated too soon once the leading opponent has finished the session. Another possibility is to set a fixed time and see how many laps can be finished during this time.
- The goal of the session should be defined to either reach an anaerobic target, with a reduced duration of the session, or to stay in an aerobic exercise, allowing the game to continue for a longer period of time.

3.7 Adaptation of Game Elements and Controls

Noticing which game elements or controls caused the most problems for users, some changes to the power-ups and controls were needed. The button that emulated a hit was removed right after the first session to prevent it's usage. Since the possibility to grant a specific player a power-up can be useful in therapy sessions, this option stayed in the game. Also, it had to be differentiated what player should have access to which power-up depending on his current effort and position in the race for balancing and training purposes. This led to the conclusion that they need different forms, what we tried to achieve by creating different power levels for the power-ups.

3.7.1 Controls

Many users struggled with the motion sensor steering, since there was a noticeable delay of approximately 0.3 seconds between leaning and the according reaction of the game to rotate the wheels. Due to this delay, many users often oversteered and tried to compensate for this by countersteering. When leaning to one side and after the inaccurate attempt to return in center position the users also countersteered unwittingly. These problems led to many crashes with the track's walls. To address them, a steer value snapping was implemented. Leaning to a side and returning to an inclination value of [-2,2] sets the sent steer value to 0 as long as the player leans to one side again, letting the car drive straight forward.

Additionally, to support players controls when using the motion sensor, *Speedpads* were introduced with the ability to grant the player “auto-steer” state if he uses the motion sensor control method. The *Windboost* also adopts this on higher levels, to further support players controlling the game with a motion sensor. The implemented gestures were mostly perceived intuitive, but were not always recognized properly. Still finding another gesture for steering instead of leaning may be reasonable since leaning and cycling interfere when performed at the same time.

Due to the large amount of collisions, the track's walls were changed in a way that colliding players are bounced away from the wall and additionally rotated towards the next checkpoint to ensure they do not get stuck after colliding.

3.7.2 Tracks and Vehicles

The outcome of the preliminary tests prompted to make changes to the tracks as well. The played track was scaled up afterwards by 150% to offer more space to steer the vehicle. Additionally more track segments and tracks were implemented into the game, not just offering 90-degrees curves but also curves that are less steep (*Figure 28*). New tracks were created with those additional track segments before the last test session. Having less steep curves, players collided less with the tracks walls when playing with a motion sensor.

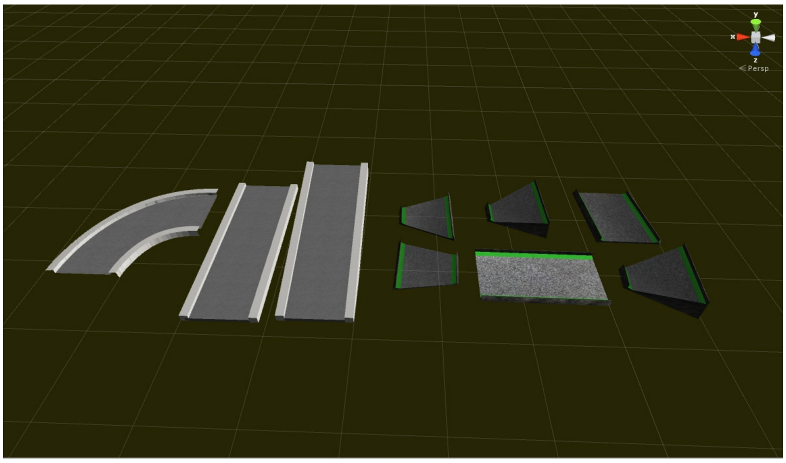


Figure 28: *Introduced new track segments*

To satisfy the wish for different player models, two vehicle models were implemented (*Figure 29*) which differ in appearance and game related values like acceleration and maximum speed. These values pave the way for the long-term motivational personalization aspect of upgrading the car's appearance and performance within the game.

Figure 29: *Main menu of the MagiKart client with vehicle selection*

3.7.3 Power-up Levels and Distribution

These power levels for weapons were introduced for different reasons. One is to consider the players heart rate and reward the right amount of PA with increased power within the game, the other one is to balance the game, giving weaker players a slightly higher chance to beat stronger competitors.

Each power-up has three different power levels which can be distinguished by intensity and/or appearance. The higher the current weapon's level, the higher the impact on the game (level 1 equals a weak impact, level 2 a medium impact and level 3 a strong impact). Every collected weapon has a weapon level that is calculated from the player's position in the race and his or her's current heart rate. The minimum weapon level is always 1.

Figure 30: *GUI graphics of Thunderstorm power-ups with level indicators*

According to [17], an individual's optimum heart rate for aerobic exercise is the interval [0.6 HR_{max}, 0.75 HR_{max}] with HR_{max} being it's maximum heart rate. If the player's current heart rate is far below (< 0.4 HR_{max}) this optimum value, e.g. at the beginning of the training, the user requires increased PA and therefore motivation to put more effort into the game. At this point, the weapon strength remains the same, i.e. weak. Raising the heart rate to the interval]0.4 HR_{max}, 0.6 HR_{max}[grants one bonus level for the weapon, reaching the mentioned optimum heart rate gives a bonus of 2 points on top of the base level. Exceeding the optimum heart rate interval for a specific amount of time reduces the base weapon level by one.

Not only the vital parameters but also the current race position affects the level of the power-ups. Being on the first place reduces your power-up level by one. Being on the last place adds one level. All in all, this system leads to several cases:

1. the winning player never gets a strong power-up. Even at optimum heart rate, he just gets a medium one at best.
2. The losing player gets a strong power-up, even if he is not at optimum heart rate and thus gets at least a medium one as long as he is not exercising at a much too high intensity.
3. For all other players placed in between the first and the last place, the power-up level solely adjusts to vital parameters and therefore to the effort the player is putting into the game. These can be either weak (no effort), medium (medium effort) or strong (optimum training intensity).

Power-ups the user picks up at this point has exactly the calculated power level. The player can check the level of any pickup indicated by the star count directly above the power-up icon in the lower left portion of the screen. The different characteristics of the power-ups and their impact on the game were designed as follows:

The *Fireball* power shoots a straight moving projectile in the viewing direction of the user's car that stuns all enemies hit. The weakest version is as large as the player's model. At Medium power level the shot projectile's size is increased by factor 2, at strong power level by factor 5, covering nearly half of the track's width. (*Figure 31*)

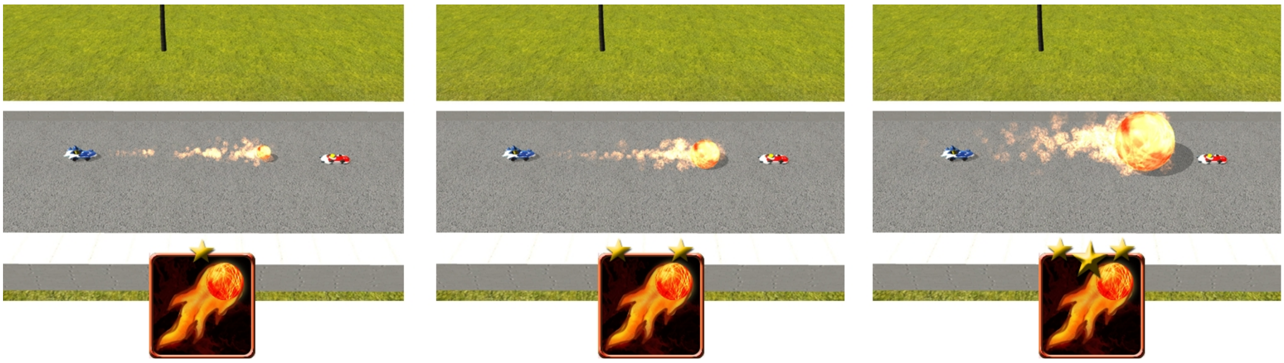


Figure 31: *Fireball power-up levels*

The weakest *Thunderstorm* power sends out a jerky lightning from the front side of the user's vehicle. Every other player hit by a *RayCast* of the length of the asset is hit multiple times and stunned for the whole duration of the spark (2 seconds) plus another 2 seconds before he can continue driving. At medium level, a thunder cloud appears at the nearest other player's location and stuns him after a short delay. The strongest version of the Thunderstorm spawns such a cloud at every other enemies location, thus stunning all other players. While the weakest version is especially useful in the beginning of the game,when many players tend to be located near each other, the strong power provides players who are performing worst the chance to catch up more easily. (*Figure 32*)

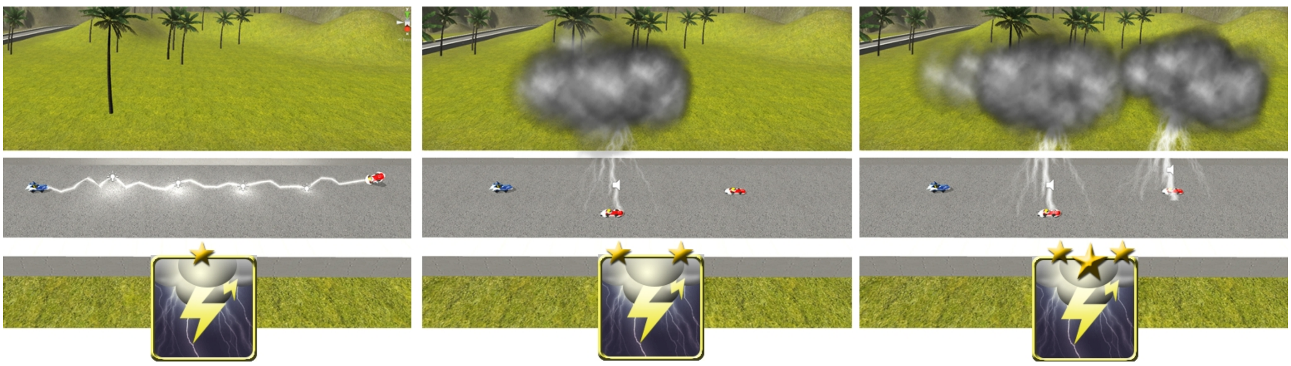


Figure 32: *Thunderstorm power-up levels*

The *Windboost* power-up gives the user a slight speed boost for a short amount of time. Used at medium level, the speed boost's duration is increased and additionally the user is granted a state called “auto steer”, which disables the need to steer his vehicle as it then steers automatically on a path the AI uses as well. This functionality became important, seeing that many users struggled with the motion sensor steering. Using a strong wind power-up increases the temporary speed boost and it's duration even more as well as making the user invulnerable to any powers from other players. Additionally, medium and strong versions of this power-up grant the player the state “auto drive”, when his heart rate is exceeding the recommended training intensity. His vehicle then not only steers but handles acceleration automatically for the duration of the power-up as well to decrease his heart rate. The appearance of this power-up does not change at higher levels. (*Figure 33*)

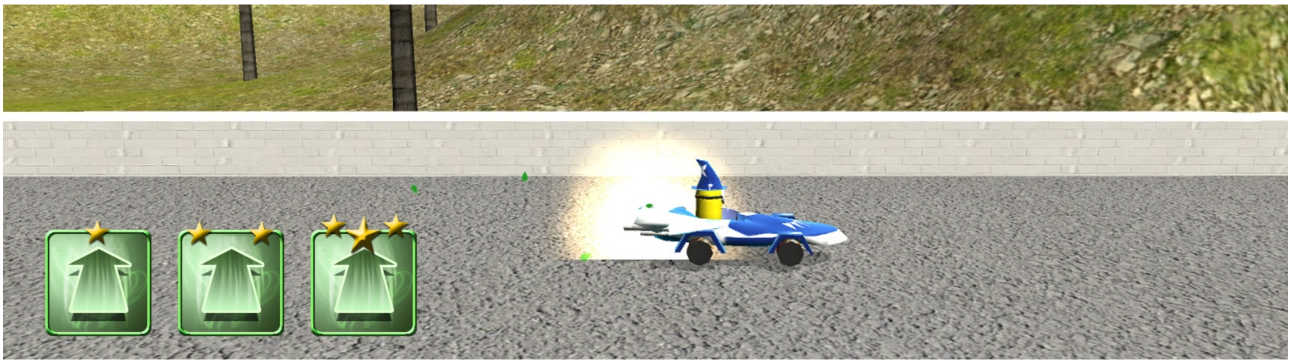


Figure 33: *Windboost power-up appearance at all levels*

The last obtainable power-up is the *Ice Field*. This power-up has been adjusted the most, due to the occasional changes of the car physics. In the beginning it just lowered the motor torque of any enemies driving on it. Noticing that this was hardly having any impact on the game, we changed it's effect to make crossing enemies slip, what made driving over it much more punishing. Additionally with every level the frictions' stiffness as well as the size of the affected area increases significantly, covering the whole width of the track at it's strongest form. This makes it positively impossible to bypass and can have a huge impact on the game if placed right before a curve or a ski jump. Also, the time the field stays before disappearing becomes higher with higher levels, to ensure that even in later stages of the game, when the players may be more scattered as in the beginning, enemies still encounter it before it vanishes. (*Figure 34*)

Figure 34: *Ice field power-up levels*

In Addition to the level penalty for being placed first, we noticed that a distribution system is needed which considers the racing position, because some power-ups were too powerful for the leading player, while some are more useful for struggling players. For that reason, we implemented the weapon distribution as in *Table 2*.

	Fireball	Thunderstorm	Windboost	Ice Field
First place	25.00%	0.00%	25.00%	50.00%
Last place	16,6%	33.33%	33.33%	16,6%
Other place	25.00%	25.00%	25.00%	25.00%

Table 2: *Distribution of power-ups depending on race position*

This Distribution ensures, that first and last placed players are more likely to get power-ups that are suitable for their particular racing position. The leading player does not get the most powerful weapon *Thunderstorm* but is more likely to get the *Ice Field* which is most powerful when you are in front of other players. For being the last player, he or she probably is more in need of a speed boost or a global stopper than anyone else.

3.7.4 Biofeedback

For the preliminary tests a simple visualization of the players heart rate, which had solely a cosmetic purpose, was implemented into the game. After the evaluation of the game's effects on the player's vital parameters, the game was altered to make better use of them. The game must not visualize the player's heart rate alone but also adapt to his rate of exertion by changing game values. This section will describe which mechanisms were implemented to make use of the player's vital parameters while playing *MagiKart*.

Visualizing the Patients Body Functions

Some parts of the graphical user interface (GUI) were designed to reflect the player's current vital parameters besides changing invisible game related parameters. It was decided to prioritize the heart rate as the most important parameter during the exercise. Thus in every game mode there is a visible heart rate indicator at the lower left corner of the heads-up display (HUD). This heart rate indicator shows an empty heart at the beginning of the exercise, which fills up gradually, depending on how close the patient's heart rate gets to an optimal interval of $[0.6 * HR_{max}, 0.75 * HR_{max}]$. This allows the patient to see how his or her heart is performing. If the patient is resting, the heart rate shows an empty heart. Being at 75% of HR_{max} the heart is filled up completely. If the heart rate exceeds the optimum interval, a danger sign is drawn on top of the heart icon to indicate that the player is exerting too much. Additionally, any overexerting player's vehicle will start to burn.

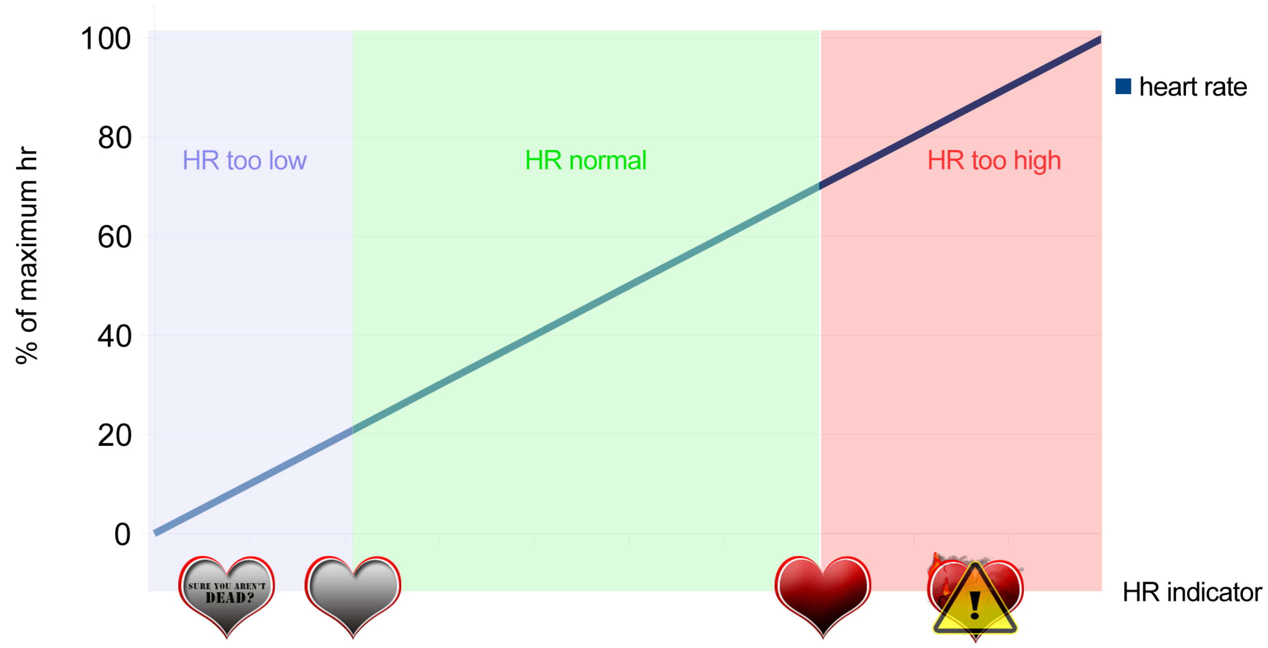


Figure 35: *Heart rate indicator states at specific heart rates*

Like some training professionals in [17] proposed, there are basically 3 states that the player can be in while exercising. Either keep training at the current extent and reduce or increase the current physical activity. Therefore, the HUD also reflects these training states according to the following finite state automata (Figure 36).

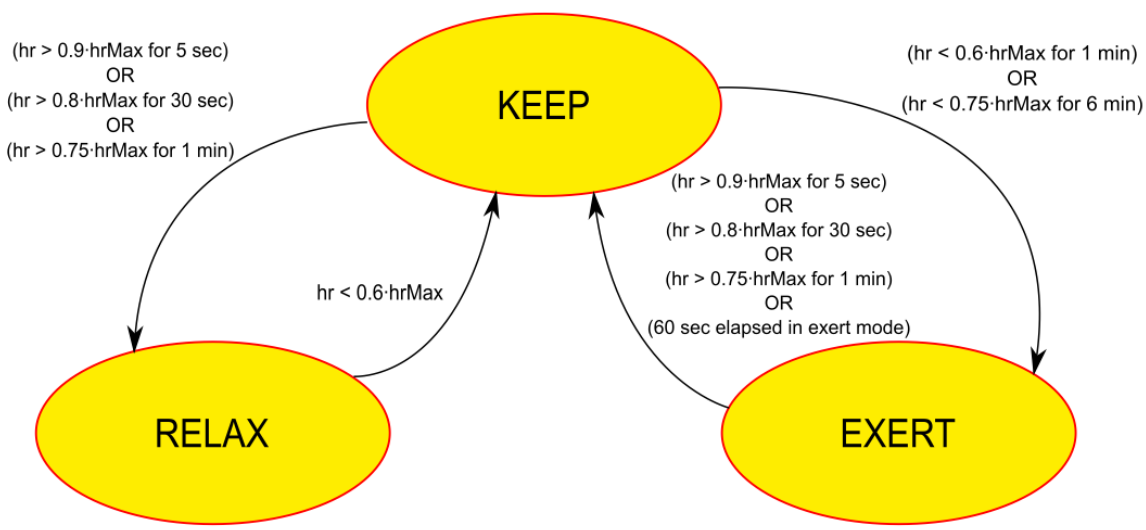


Figure 36: *Finite state automata to switch among exercise modes as proposed in [17], p. 7*

On changing the training state, positive motivational instructions are shown at the center of the screen to either decrease or increase PA or to keep the exercise at the current level (). These serve not only as a clarification of the current heart rate but also as a positive feedback on how the player is performing and to convey competence to the users that perform right. (Figure 37)

Figure 37: *Game instructions to increase, keep or decrease physical activity*

Although, especially for children, one actually cannot just rely on the vital parameters but you have to adjust the training intensity individually on how the children's exertion is perceived by a medical assistant. John H. Stock at Cardon Children's Medical Center proposed that a better indicator for a children's exertion is whether it can answer to your questions while exercising ([19]). Being a game with a high level of immersion and action which needs to be concentrated on this may not be a valid way of measurement. Since this game is played under the supervision of a medical assistant, he has to observe the respiration of the patient as well and adjust the maximum heart rate to fit the exertion level of a juvenile player.

Changing Ergometer Load

For the preliminary tests the adjustment of the ergometer's load did not work properly. Nevertheless, the adaptation engine of the *AutoMedic* framework is capable of taking cycling speed and current heart rate into consideration and raise the ergometer's load if the heart rate remains too low for a certain amount of time. Introducing the training states as mentioned in [17], the ergometer load is also increased in exert mode to raise physical stress and lowered in relax mode to decrease the player's PA . The load can also be adjusted directly by the medical supervisor using a slider handle.

Body Functions Influence on the Game

Besides the visible influences of the player's vital parameters on the game GUI there are also some invisible game parameters the biofeedback affects. Since *MagiKart* is a multiplayer racing game and the main goal is to absolve the track faster than others or finish most laps in a given timeframe, the only parameters that could be manipulated individually for each vehicle without manipulating game flow for other players are it's speed and steering. As already mentioned earlier, the closer the patients HR is to the calculated target HR the stronger the power-ups. Especially the strong or medium *Windboost* helps users with steering. Furthermore, the vehicles maximum speed and acceleration are adjusted via the heart rate, reaching maximum possible speed just while having a HR between 0.6* HRmax and 0.75 * HRmax. This speed bonus rewards players for being at their target heart rate, allowing them to drive faster on average as worse performing players.

Nevertheless this system still holds a wide variety of possibilities to change game parameters depending on the players vital parameters. A dynamically generated track that can be altered on the fly could adjust to the players performance, integrating slopes when needed. For multi-player usage this must happen individually. So the track would have to fork, to offer each player a sidetrack that fits his needs. Also, the AI could be lowered in difficulty when the player is struggling. Since the AI is already quite simple to beat, it would be necessary to make it more challenging for usage with the vital parameters.

3.8 Player Motivation

Taking psychological aspects into consideration, players are first off motivated by their physical and psychological needs. Jesse Schell connected the motivation of video game players with Abraham Maslow's *Theory of human motives* ([20], pp. 126-128), proposing that most enjoyment that comes from playing games serve the need for self-esteem.

Having that in mind and looking at this fourth level of Maslow's Hierarchy of needs (*Figure 38*), people can enjoy mastery and achievement as well as recognition and respect.

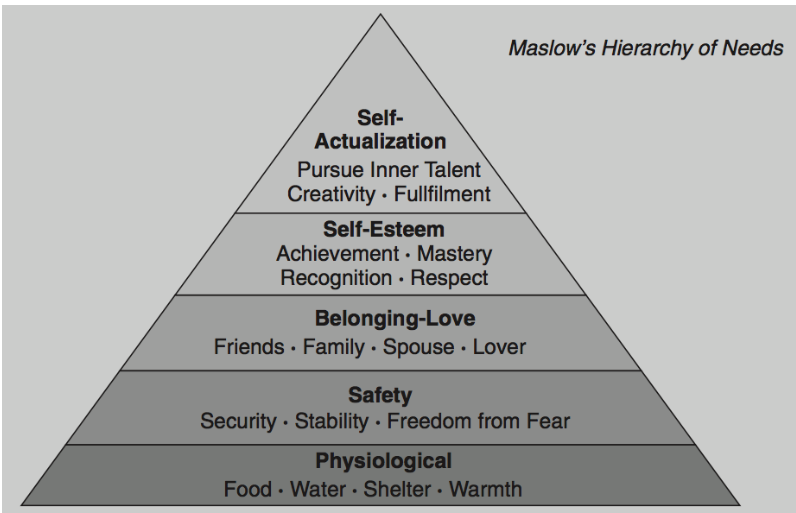


Figure 38: *"Maslow's Hierarchy of Needs" as cited in [20], p. 126*

The Game that was developed for this thesis already satisfies some of those, since being successful in most games requires mastery of the mechanics and controls as well as achieving the goal set by the game. Being a multiplayer game, *MagiKart* also satisfies the need for recognition and respect for the performed activities. By playing a multiplayer game with peers (in the case of a rehabilitation environment: other patients), winners will get recognition and respect from the other players as well as from the game itself via performance feedback. Schell even puts multiplayer games at a lower level of the hierarchy, *Belonging-Love*, for its potential to connect people and encourages developers to question their game at which stage of the hierarchy of needs a newly created game is operating. Additionally, he brings up the question whether games, which can access even lower stages, even exist. This raises the question if *MagiKart* can be seen as a game trying to operate at all stages of this hierarchy. The goal of the game is motivating people to pursue physical fitness (physiological need) under the supervision of a medical professional (safety) by playing and exercising with others (belonging, recognition) regularly (mastery, achievement) with the possibility of progress. Although this may be far-fetched, EGs in general aim to pursue a goal that is beyond pure entertainment, being able to satisfy even more basic needs like physical well-being. Since our game is basically meant for rehabilitation, it's primary goal is to serve the player's health, which can be seen as operating on the most basic level of human needs that Maslow mentioned. It is questionable whether the fact that the game serves rehabilitation motivates patients to play it or the motivation to play the game just makes them pursue their training goal, reversing cause and consequence. Nonetheless, gaming motivation apparently and obviously is connected to the satisfaction of psychological needs.

There are more current and more frequently used approaches for determining player's motivations in video games than this one, e.g. [21], which categorizes massively multiuser online (MMO) players as either non-exclusively seeking Achievement, Socialization or Immersion. Although, [16] states that these are based on the individual diversity of player types and “these categories or typologies largely reflect the structure and content of current games, rather than the fundamental or underlying motives and satisfactions that can […] sustain participation […].” ([16], p.345).

Since another study's findings were that some commercial EGs (also: *Active Video Games*, AVG) do not increase PA due to the stagnating participation under naturalistic circumstances ([22]), the usage of a more psychological approach to gamer's motivation may help to address this problem. According to is the self-determination theory (SDT), which focuses on factors that support or weaken both intrinsic (the satisfaction derived from performing an action) and extrinsic (satisfaction from imposed objectives) motives of human behavior ([23], [24]). Intrinsic motivation is the main factor for engaging in play and sport ([25], [26]), which relates to the fact that participation in games and sports is generally voluntary, i.e. just for intrinsic satisfaction. Considering [15], this holds for children as well, since the most common answer in the study surveying childrens' reasons for playing games was “It's just fun.” A mini-theory of SDT, the cognitive evaluation theory (CET, [23], [27]) proposes that the perceived sense of autonomy, competence and relatedness supports intrinsic motivation. Ryan also assumed that intuitive controls (IC) and presence affect autonomy and competence, therefore intrinsic motivation as well.

While the sense of autonomy is typically high for voluntary actions like playing a video game, in our case it may be diminished by the fact that the players need to play *MagiKart* not just because they want to, but rather to recover from surgical treatment. Even within the game there is a restrictive set of rules and small boundaries of the virtual world, like in most racing games, further limiting autonomy. Although this apparent lack of exploring autonomy in racing games, they obviously encourage people to play, otherwise sales of racing games would plummet. Especially *Mario Kart*, the main source of inspiration for *MagiKart*, is a top-seller in console games.

Assumably, core motivators for playing *Mario Kart* instead of more realistic racing games are:

- an intuitive control mechanism that enhances *presence* or immersion, the fading perception of the medium as a medium (immersion through *Wii* steering wheel),
- a widely known and beloved set of Nintendo franchise characters and settings, that most people of a peer-group know, thus motivating even non-gamer to play along (*relatedness*)

- and game elements that can convey more *perceived competence* than a regular car racing game (Weapons, power-ups).

These assumptions correspond to indicators of the PDT. The need for perceived competence, that can be understood as the need for mastering challenges and feeling effective [23]. According to Richard M. Ryan, perceived competence is enhanced by factors such as “[...] to acquire new skills or abilities, to be optimally challenged, or to receive positive feedback […] [therefore enhancing] intrinsic motivation.” ([16], p.346) He proposed that a high level of autonomy and perceived competence leads to a higher motivation for recreating the experience of playing a game.

3.8.1 Competitive Gameplay

Stuart Gray researched the connection between physical extent in EGs and the motivational pull of head-to-head and leader board competitive games [28]. Firstly, he analyzed different conducted studies which vary in their results regarding the effect of multiplayer competitive games on intrinsic motivation. The analyzed studies led him to the conclusion that competitive gameplay has a positive effect on highly competitive individuals whereas it tends to have negative or no effect at all on the intrinsic motivation of lowly competitive individuals. His main study was conducted on 33 participants, aged from 15 to 65, with an average age of 25, over a period of 4 weeks, and was analyzing the impact of competitive EG play on energy expenditure. Participants who played any competitive game mode spent significantly more time in anaerobic training zones ($0.82 * HR_{max}$ - $1 * HR_{max}$), suggesting that competitive conditions are more effective for energy expenditure than non-competitive conditions.

3.8.1.1 Highscore

There are different models of competitive gameplay. The models researched in Start Gray's study ([28]) was firstly a leader board condition, in which the players try to achieve the highest game score which is then shown at the end of the game. In the case of a racing game this may be either a computed score for each unit of play or scoring the fastest lap time overall. Secondly, head-to-head competition, which makes two or more players play the game at the same time, trying to accomplish the goal faster than the others while playing simultaneously. Surprisingly, the extent of power expenditure was even significantly higher with the leader board competitive gameplay. It can be assumed that this arose from the motivation to not only to beat fellow players but also one's own personal record.

Designing motivational aspects for our racing game, we decided to implement a head-to-head competitive game with an integrated leader board for the fastest lap time or, respectively in coin collect mode, the highest coin score – therefore combining both competitive aspects to further increase energy expenditure and motivation. Also, if played alone, the leader board can also motivate to replay even with missing head-to-head competitors or against the relatively easy to beat AI. Even for unsuccessful players the leader board can maintain motivation, by setting and beating own records. The competitive part of the game not only enhances intrinsic but also extrinsic motivation, since the winning player is more likely to receive approval by his peers. Either through communication in a head-to-head-competition (space and time dependent) or via the scores shown on the leader board (space and time independent).

3.8.1.2 Motivation through Power-ups

As it is one of the core concepts of games like *Mario Kart* or *ModNation Racers,* a weapon system was implemented into *MagiKart*. As mentioned before, weapons in a game are not providing the biggest motivational pull, especially for female players. On the other hand, these power-ups convey a feeling of strength and competence when used. They are boosting the players morale, as they can have a strong impact on the games outcome by disturbing the competitors flow. This sense of competence can enhance intrinsic motivation. Being able to choose the point of time the power-up is used additionally conveys a sense of autonomy, theoretically increasing intrinsic motivation further. Getting stronger power-ups requires physical effort since the weapon levels increase with the user's heart rate.

3.8.2 Motivation Increase by Progress

Besides the motivation increase by mastering the game and it's controls and since session data is stored in a database, we can consider older training sessions in the current one, calculating changes in the user's vital parameters. We track the heart rate variability change, comparing minimum and maximum heart rate values which are good indicators for the patient's physical fitness. If the weight is entered correctly at the beginning of every session, there even can be made calculations on the player's body mass index changes. But not only the training progress is a long-term motivational factor, also progress within the game serves this purpose.

What yet has to be implemented in *MagiKart* is an upgrade system that allows the player to customize his vehicle with purely cosmetic or also game relevant upgrades that can be bought in a shop within the game with virtual currency which is earned for completing game sessions. Winning a game grants more virtual currency than losing, motivating player's to perform better. The bought upgrades for the car that boost vehicle values like acceleration, maximum speed or handling are stored within the user database, so player's can load their vehicle each session which offers long-term motivation and also theoretically satisfies the need for self-actualization and personalization, enhancing intrinsic as well as extrinsic (approval for the upgraded vehicle) motivation.

3.8.3 Motivation through Feedback

Schell states that “One deep need common to everyone is the need to be judged.” ([20], p.127) What players need in his opinion is feedback for their performed actions. This matches with SDT concepts that perceived competence enhances intrinsic motivation. For this reason we implemented feedback for the training effort the player is putting into the game. Like already described on pp. 30, The GUI reflects the current training state with the heart rate meter and instructional graphics that show the user whether to increase or decrease physical activity. These instructions can limit the feeling of autonomy, since they are proposing a course of action but also give positive motivational feedback if the player performs as intended by the game.

4. Conclusion and Following Work

After preliminary test sessions, perceptions of the medically skilled participants were, that the game was not yet suitable for children rehabilitation due to some software and hardware issues. However, most participants enjoyed the developed game and it's features. They also stated that the game has the capability to encourage children to pursue their rehabilitation, since it distracts players' attention from the exhausting exercise itself. Most of the difficulties occurred while using a motion sensor control mechanism, since some important gestures interfered with the movement that is accompanied by cycling. Also, many sensor failures and hardware issues with the used ergometers made some findings inconclusive. An alternative protocol is proposed that considers resting and cool down periods and fixed timeframes for each session. Adjustments to game content and controls were made after the test sessions to cope with most of the software problems.

Playing the competitive multiplayer racing game with a collectable power-up system that was created for this thesis motivated players to recreate the experience. Game development is an iterative process that involves many tests, especially in terms of long-term player motivation. Due to the limitation of time this could not be further pursued in this thesis. However, the already presented work showed to be motivating when used by different players and may serve as a ground work for further improvements.

For different kinds of players, different motivational aspects may be implemented. While some video game players are highly motivated by competition, others can be motivated by supporting fellow players, overcome obstacles or create and share game worlds by their own. The developed system offers the expandability for further improvement, such as additions to the track building set or new game modes.

Although game sessions were too short to achieve health benefits, players who played *MagiKart* reached an aerobic exercise target heart rate. Assumably, prolonging each session time and using ergometers with a working load adaption can result in even higher levels of exercise, reaching anaerobic exercise levels. As our test subjects were perfectly healthy, young adults, additional studies have yet to be carried out to determine the exercise effect on children or adolescents with heart problems. Those studies should not only test effects of the game on sick children but also the long-term motivation that helps them to stay the course. To achieve this, personalization of game content (e.g. upgrade vehicles appearance or game values like maximum speed) and the visualization of exercise and game progress can be helpful.

5. Appendix

5.1 Contents of the CD-ROM

AutoMedic/ compiled version of the Automedic Framework

Game/ holds the compiled Server and Client executables

Control/ holds a sample implementation of the AutoMedic Framework for testing.

Source/ programming source files (without resources)

Presentation/ disputation slides in .odp and .pdf

Thesis/ this thesis in pdf and odt format, as well as the digital literature

Starting the Game

1. Run Game/**Server.exe**
2. Run Game/Control/**run.bat**
3. the run.bat file opens 2 Client instances and one AutoMedic MDC Instance. When the Clients connected to the MDC, the "MDC connected" GUI Element will turn green and the main Menu is started in the Clients. After choosing a Vehicle and a Player name, click on **"Connect"**
4. In the Server Application, select the level (1, 2, 3 or dynamic), game mode ("racing" or "coin") or add AI opponents. After all players are shown in the list, click on "Load Level". The game now begins.
5. The Game can be controlled by setting values via the value sliders in the MDC window. SpeedX sets emulates the cycling speed, XposX the steering for a specific player.

5.2. Questionnaire

Umfrage zum Testing des Multiplayer-Rennspiels 'MagiKart'

* Erforderlich

ID *
Deine ID vom Testing

Geburtsdatum (tt.mm.yyyy) *

Geschlecht *
- ○ Männlich
- ○ Weiblich

Gaming Gewohnheiten

Ich spiele Videospiele... *
- ○ < einmal im Monat
- ○ einmal im Monat
- ○ einmal pro Woche
- ○ mehrmals die Woche
- ○ jeden Tag

...davon auf folgenden Geräten:
- ☐ Konsole (xBox, Playstation)
- ☐ Wii
- ☐ PC
- ☐ Handheld
- ☐ Smartphone
- ☐ Browsergame

... und folgende Genres:
- ☐ Adventure
- ☐ Aufbauspiel
- ☐ Beat-'Em-Up
- ☐ Casual Games (Doodle Jump, Bejeweled etc.)
- ☐ Puzzler
- ☐ Quiz
- ☐ Rennspiel
- ☐ Rollenspiel
- ☐ Shooter
- ☐ Simulation
- ☐ Sport
- ☐ Strategie

Ich spiele lieber... *
- ○ Singleplayer Spiele
- ○ Multiplayer Spiele
- ○ Egal

Ich halte mich für einen Hardcore-Gamer *
d.h. ich spiele auch gerne stundenlang intensiv und viele unterschiedliche Spiele.
- ○ 1 (trifft gar nicht zu)
- ○ 2 (trifft eher nicht zu)
- ○ 3 (trifft teils zu)
- ○ 4 (trifft eher nicht zu)
- ○ 5 (trifft voll und ganz zu)

Sportliche Betätigung

Ich treibe Sport.... *
- ○ < einmal im Monat
- ○ einmal im Monat
- ○ einmal pro Woche
- ○ mehrmals die Woche
- ○ jeden Tag

Wenn ja, welche Sportart?
- ☐ Mannschaftssport
- ☐ Kraftsport
- ☐ Ausdauersport/ Radsport
- ☐ Sonstiges:

Ich halte mich für sportlich *
- ○ 1 (trifft gar nicht zu)
- ○ 2 (trifft eher nicht zu)
- ○ 3 (trifft teils zu)
- ○ 4 (trifft eher zu)
- ○ 5 (trifft voll und ganz zu)

Steuerung des Spiels als "Patient"

bitte beantwortet die Fragen nach folgendem Bewertungsschema:
1: trifft gar nicht zu
2: trifft eher nicht zu
3: trifft teils zu,
4: trifft eher zu
5: trifft voll und ganz zu

Das Auto war mit dem Joystick gut steuerbar. *
- ○ 1 (trifft gar nicht zu)
- ○ 2
- ○ 3
- ○ 4
- ○ 5 (trifft voll und ganz zu)

Das Spiel mit dem Joystick zu steuern war geistig anstrengend. *
- ○ 1 (trifft gar nicht zu)
- ○ 2
- ○ 3
- ○ 4
- ○ 5 (trifft voll und ganz zu)

Das Spiel mit dem Joystick zu steuern war körperlich anstrengend. *
- ○ 1 (trifft gar nicht zu)
- ○ 2
- ○ 3
- ○ 4
- ○ 5 (trifft voll und ganz zu)

ich war an irgendeinem Punkt genervt von der Steuerung mit Ergometer und Joystick. *
- ○ 1 (trifft gar nicht zu)
- ○ 2
- ○ 3
- ○ 4
- ○ 5 (trifft voll und ganz zu)

Wenn ja, warum?

Das Auto war mit dem Bewegungssensor gut steuerbar. *
- ○ 1 (trifft gar nicht zu)
- ○ 2
- ○ 3
- ○ 4
- ○ 5 (trifft voll und ganz zu)

Die Gesten zum Steuern des Spiels waren intuitiv. *
(Waffe abfeuern: Ausholen und werfen, Lenkung: zur Seite neigen, Rückwärtsgang: Hände heben, Respawn: Hände klatschen)
- ○ 1 (trifft gar nicht zu)
- ○ 2
- ○ 3
- ○ 4
- ○ 5 (trifft voll und ganz zu)

Das Spiel mit dem Bewegungssensor zu steuern war geistig anstrengend. *
- ○ 1 (trifft gar nicht zu)
- ○ 2
- ○ 3
- ○ 4
- ○ 5 (trifft voll und ganz zu)

Das Spiel mit dem Bewegungssensor zu steuern war körperlich anstrengend. *
- ○ 1 (trifft gar nicht zu)
- ○ 2
- ○ 3
- ○ 4
- ○ 5 (trifft voll und ganz zu)

ich war an irgendeinem Punkt genervt von der Steuerung mit Ergometer und Bewegungssensor. *
- ○ 1 (trifft gar nicht zu)
- ○ 2
- ○ 3
- ○ 4
- ○ 5 (trifft voll und ganz zu)

Wenn ja, warum?

Steuerung des MDC als Betreuer

Ich war Betreuer *
d.h. ich startete das Spiel, steuerte den Server und beobachtete die Werte der Patienten
- Ja
- Nein (Weiter auf Seite 5)

Ich hätte mir im MDC mehr Möglichkeiten der Einflussnahme auf den Spielverlauf gewünscht.
z.B. Strecke während des Spiels für die Patienten anpassen
- 1 (trifft gar nicht zu)
- 2
- 3
- 4
- 5 (trifft voll und ganz zu)

Wenn ja, welche?

Die MDC-Steuerung ist meiner Meinung nach benutzerfreundlich und verständlich genug für einen eingelernten Betreuer
Verständliche Oberfläche, Daten gut einsehbar, etc.
- 1 (trifft gar nicht zu)
- 2
- 3
- 4
- 5 (trifft voll und ganz zu)

Der Schwierigkeitsgrad des Spiels konnte gut gesteuert werden.
z.B. Widerstand der Ergometer anpassen, Geschwindigkeit anpassen, etc.
- 1 (trifft gar nicht zu)
- 2
- 3
- 4
- 5 (trifft voll und ganz zu)

Game Design

In diesem Abschnitt geht es um das Spiel an sich und dessen Inhalte.

Das Spiel machte Spaß und ich würde es gerne erneut Spielen. *
- 1 (trifft gar nicht zu)
- 2
- 3
- 4
- 5 (trifft voll und ganz zu)

Die aufsammelbaren Powerups haben mir das Erreichen des Ziels erleichtert. *
- 1 (trifft gar nicht zu)
- 2
- 3
- 4
- 5 (trifft voll und ganz zu)

Wenn nicht, Haben manche Powerups das Vorankommen erschwert?
(z.B. Speedboost sorgte dafür, dass man öfter gegen Wand fährt etc.). Erläuterung bitte bei Sonstiges eintragen.
- Feuerball
- Gewitter
- Geschwindigkeitsboost
- Eisfeld
- Sonstiges:

Die aufsammelbaren Powerups haben mich motiviert zu spielen. *
- 1 (trifft gar nicht zu)
- 2
- 3
- 4
- 5 (trifft voll und ganz zu)

Die aufsammelbaren Powerups machten das Spiel interessanter. *
- 1 (trifft gar nicht zu)
- 2
- 3
- 4
- 5 (trifft voll und ganz zu)

Die gespielten Strecken waren zu schwer. *
- 1 (trifft gar nicht zu)
- 2
- 3
- 4
- 5 (trifft voll und ganz zu)

Die gespielten Strecken waren zu leicht. *
- 1 (trifft gar nicht zu)
- 2
- 3
- 4
- 5 (trifft voll und ganz zu)

Ich hätte mir mehr Hindernisse auf der Strecke gewünscht. *
- 1 (trifft gar nicht zu)
- 2
- 3
- 4
- 5 (trifft voll und ganz zu)

Eine geradere Endlosstrecke wäre einfacher zu bewältigen gewesen. *
- 1 (trifft gar nicht zu)
- 2
- 3
- 4
- 5 (trifft voll und ganz zu)

Gegen eine andere Person anzutreten spornte mich an. *
- 1 (trifft gar nicht zu)
- 2
- 3
- 4
- 5 (trifft voll und ganz zu)

Gegen eine andere Person anzutreten motivierte mich mehr als alleine zu spielen. *
- 1 (trifft gar nicht zu)
- 2
- 3
- 4
- 5 (trifft voll und ganz zu)

Einschätzungen der Motivationssteigerung von Videospielen

bitte beantwortet die Fragen nach folgendem Bewertungsschema:
1: trifft gar nicht zu
2: trifft eher nicht zu
3: trifft teils zu,
4: trifft eher zu
5: trifft voll und ganz zu

Das getestete Spiel eignet sich meiner Meinung nach zur Rehabilitation von Herzkranken (Kindern) *
- 1 (trifft gar nicht zu)
- 2
- 3
- 4
- 5 (trifft voll und ganz zu)

warum?

Ich halte allgemein Spiele zur Rehabilitation von Herzkranken Kindern für sinnvoll *
- 1 (trifft gar nicht zu)
- 2
- 3
- 4
- 5 (trifft voll und ganz zu)

Die Spiele-Session hatte einen merklichen Trainingseffekt. *
- 1 (trifft gar nicht zu)
- 2
- 3
- 4
- 5 (trifft voll und ganz zu)

Videospiele wie 'Magikart' würden mich Motivieren mehr Sport zu treiben. *
- 1 (trifft gar nicht zu)
- 2
- 3
- 4
- 5 (trifft voll und ganz zu)

Das Spiel würde durch erkennbaren Spielfortschritt durch häufigeres Spielen mehr motivieren. *
Beispielsweise durch Aufwertung des Autos für Münzen, die man nach jedem Rennen erhält und die im persönlichen Patientenprofil gespeichert werden.

- 1 (trifft gar nicht zu)
- 2
- 3
- 4
- 5 (trifft voll und ganz zu)

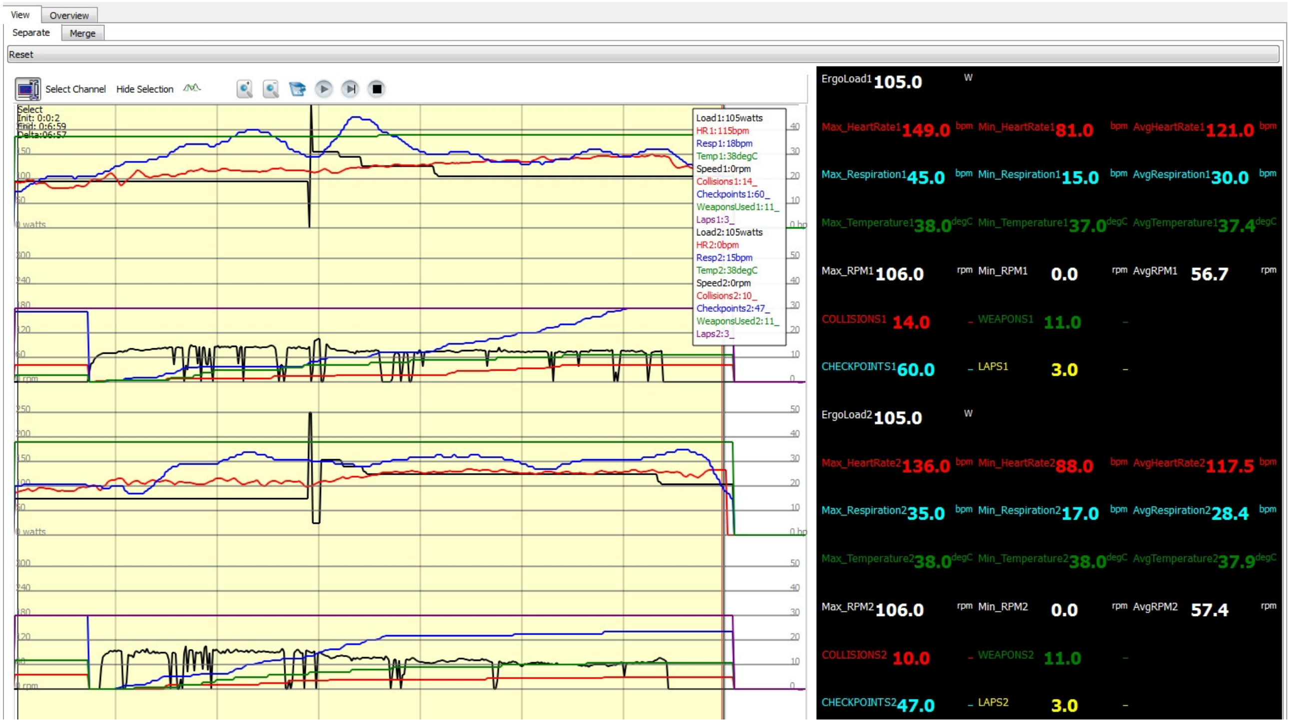
Zusätzliches

Was hat dir besonders gut gefallen?

Was würdest du verbessern?

Wenn ja, warum?

5.3 Test Session Result Sample



5.4 Important Scripts

5.4.1 Bouncywalls.cs

```
using UnityEngine;
using System.Collections;

public class BouncyWalls : MonoBehaviour {
 private float knockBack = 3f;

 void Awake() {
     if(GameObject.Find("Client"))
          Destroy(this);
 }

 void OnCollisionEnter(Collision col){
     GameObject player = col.collider.gameObject;
     if(player.tag == "playerCar" || player.tag == "aiCar"){
          IController c = (IController)player.GetComponent("IController");
          Server.Instance.SendCollisionCount(c.PlayerId);
     }
 }

 void OnCollisionStay(Collision col){
     GameObject player = col.collider.gameObject;
     if(player.tag == "playerCar" || player.tag == "aiCar"){

            IController controller = player.GetComponent(typeof(IController)) as
 IController;

          if(col.contacts.Length > 0){
               Vector3 towards = controller.Path[(controller.CurrentCheckPoint+1)
                              %controller.Path.Count].transform.position;

               Vector3 playerpos = player.transform.position;
               towards.y =playerpos.y;
               Vector3 dif = towards - playerpos;
               if(dif.magnitude < 20 ){
                    // if closest checkpoint too close, look at the next.
                    int i = 2;
                    towards = controller.Path[(controller.CurrentCheckPoint+i)
                         %controller.Path.Count].transform.position;
                    towards.y =playerpos.y;
                    dif = towards - playerpos;


               }
               player.transform.position += dif.normalized;
               player.transform.LookAt(towards);
               Vector3 pushForce = dif.normalized * knockBack;
               player.rigidbody.AddForce(pushForce,ForceMode.VelocityChange);
          }
     }

 }
}
```

5.4.1 Gesture Controls

```
void MotionUserObject::GetPlayerOutputs(){

[...]
            {
                /*
                vec.push_back("[0]Head");
                vec.push_back("[1]Neck");
                vec.push_back("[2]LeftShoulder");
                vec.push_back("[3]RightShoulder");
                vec.push_back("[4]LeftElbow");
                vec.push_back("[5]RightElbow");
                vec.push_back("[6]LeftHand");
                vec.push_back("[7]RightHand");
                vec.push_back("[8]Torso");
                vec.push_back("[9]LeftHip");
                vec.push_back("[10]RightHip");
                vec.push_back("[11]LeftKnee");
                vec.push_back("[12]RightKnee");
                vec.push_back("[13]LeftFoot");
                vec.push_back("[14]RightFoot");*/

                NitePoint3f leftHand=joints_new[6];
                NitePoint3f leftHandOld = joints_old[6];
                NitePoint3f leftElbow=joints_new[4];
                NitePoint3f leftElbowOld=joints_old[4];

                NitePoint3f rightHand=joints_new[7];
                NitePoint3f rightHandOld = joints_old[7];
                NitePoint3f rightElbow=joints_new[5];
                NitePoint3f rightElbowOld=joints_old[5];

                NitePoint3f head = joints_new[0];

                // z distance left and right
                float zDistanceLeft = leftHandOld.z -leftHand.z;
                float zDistanceRight = rightHandOld.z - rightHand.z;
                //cout<<zDistanceRight<<endl;
                //cout<<zDistanceLeft<<" "<<zDistanceRight<<endl;

                 float distanceHands=sqrt(pow(rightHand.x-
    leftHand.x,2)+pow(rightHand.y-                 leftHand.y,2));

                // Casting
                 bool leftArmMovedForwardOverhead = (zDistanceLeft > 130  &&
         leftHandOld.y > head.y && leftHand.y > head.y);
                  bool rightArmMovedForwardOverhead = (zDistanceRight > 130  &&
         rightHandOld.y > head.y && rightHand.y > head.y);
                 bool casting = (distanceHands > 60 && (leftArmMovedForwardOverhead ||
         rightArmMovedForwardOverhead));
                if(casting){
                    setOutputValue("Casting",1);
                }else{
                    setOutputValue("Casting",0);
                }
                float yDistanceRightHand = rightHand.y - rightHandOld.y ;
                float yDistanceLeftHand = leftHand.y - leftHandOld.y;
                float yDistanceRightElbow = rightElbow.y - rightElbowOld.y;
                float yDistanceLeftElbow = leftElbow.y - leftElbowOld.y;

                //cout<<"hithere"<<zDistanceRight<<endl ;
                setOutputValue("VLeft",yDistanceLeftHand);
                setOutputValue("VRight",yDistanceLeftElbow);
```

```


            // Pull left or right arm down
            bool leftHandCrossesHead = (leftHandOld.y > head.y && leftHand.y < head.y && leftHand.y > leftElbow.y && leftHandOld.y > leftElbowOld.y && yDistanceLeftHand < -100 && yDistanceLeftElbow < -20);
            bool rightHandCrossesHead = (rightHandOld.y > head.y && rightHand.y < head.y && rightHand.y > rightElbow.y && rightHandOld.y > rightElbowOld.y && yDistanceRightHand < -100 && yDistanceRightElbow < -20);
            bool honking = (leftHandCrossesHead || rightHandCrossesHead);
            if(honking){
                setOutputValue("Honking",1);
            }else{
                setOutputValue("Honking",0);
            }

            // Lift arms over head
            if(rightHand.y > head.y && leftHand.y > head.y){
                setOutputValue("Liftarms",1);
            }else{
                setOutputValue("Liftarms",0);
            }

            if(distanceHands > -60 && distanceHands < 60){
                setOutputValue("Clapping",1);
            }else{
                setOutputValue("Clapping",0);
            }


        }

        for (size_t i=0; i<= nite::JOINT_RIGHT_FOOT; i++){
            nite::JointType type = (nite::JointType)i;
            joints_old[i] = joints_new[i];
        }
    }else{
        _skeletonPort->setTracked(false);
    }
    return;
}
```

Bibliography

[1]: NPD Group. *he Video Game Industry Is Adding 2-17 Year-Old Gamers At A Rate Higher Than That Age Group's Population Growth*. 2011. Online in the internet: https://www.npd.com/wps/portal/npd/us/news/press-releases/pr_111011/ (02 March 2014).

[2]: NPD Group. *Kids are Gaming on Mobile Devices Almost as Much as They are on Consoles and Computers*. 2013. Online in the internet: https://www.npd.com/wps/portal/npd/us/news/press-releases/kids-are-gaming-on-mobile-devices-almost-as-much-as-they-are-on-consoles-and-computers/ (03 March, 2014).

[3]: Frederick-Recascino, C. M. & Schuster-Smith, H.. *Competition and Intrinsic Motivation in Physical Activity: A Comparison of Two Groups*. In: Journal of Sport Behavior, 26:3, 2003, 240-255.

[4]: Prummer, M.. *Design of a Server-Oriented Multiplayer Game for a Biofeedback System using Unity 3D*. Bachelor's Thesis, Ludwig-Maximillians-Universität München, 2014.

[5]: Ng, B. D. & Wiemer-Hastings, P.. *Addiction to the internet and online gaming*. In: CyberPsychology & Behavior, 8:2, 2005, 110-113.

[6]: Thompson, D., Baranowski, T., Buday, R., Baranowski, J., Thompson, V., Jago, R., Griffith, M. J.. *Serious Video Games for Health: How Behavioral Science Guided the Development of a Serious Video Game*. In: Simulation & Gaming, 41:4, 2010, 587-606.

[7]: World Health Organization. *Obesity and overweight*. 2013. Online in the internet: http://www.who.int/mediacentre/factsheets/fs311/en/ (02 March 2014).

[8]: Houghton Mifflin Company. *The American Heritage® Dictionary of the English Language*. Houghton Mifflin Company, , 2000.

[9]: Göbel, S. et al.. *Serious games for health: personalized exergames*. In: Proceedings of the international conference on Multimedia, Firenze, Italy, October 2010, ACM, 1663-1666.

[10]: Cho, J.-H.. *3D game development with Unity 3D for Biofeedback Systems*. Bachelor's Thesis, Technische Universität München, 2012.

[11]: Mendoza, A., Rodriguez, M., Baumgartner, B., Schreiber, U. & Knoll A. . *Embedded platform for automation of medical devices*. In: Computing in Cardiology, 38:1, 2011, 829-832.

[12]: Mendoza, A.,Baumgartner, B., Schreiber, U., Krane, M., Knoll, A. & Bauernschmitt, R.. *Automedic: Fuzzy control development platform for a mobile heart-lung machine*. In: IFMBE Proceedings, Munich, Germany, September 7-12 2009, World Congress on Medical Physics and Biomedical Engineering, 685-688.

[13]: Thölke, A.. *Integration of an Ergometer Based Biofeedback-System into the Process of Rehabilitation*. Bachelor's Thesis, Munich University of Applied Sciences, 2013.

[14]: Paez S., Maloney A., Kelsey K. et al. *Parental and environmental factors associated with physical activity among children participating in an active video game*. In: Pediatric Physical Therapy:, 21:3, 2009, 245-253.

[15]: Olson, C. . *Children's motivations for video game play in the context of normal development*. In: Review of General Psychology, 14:2, 2010, 180-187.

[16]: Ryan R. M., Rigby C. S. & Przybylski A.. *The Motivational Pull of Video Games: A Self-Determination Theory Approach*. In: Motivation and Emotion, 30:4, 2006, 344-361.

[17]: Buttussi, F., Chittaro, L., Ranon, R. & Verona, A.. *Adaptation of graphics and gameplay in fitness games by exploiting motion and physiological sensors.*. In: Smart Graphics, 8th International Symposium, SG 2007, Kyoto, Japan, 2007, Springer, 85-96.

[18]: Sherry, J. L.. *Flow and media enjoyment*. In: Communication Theory, 14:4, 2004, 328-347.

[19]: Stock MD, J. H.. *Target heart rates for kids*. 2012. Online in the internet: https://www.bannerhealth.com/Services/Health+And+Wellness/Ask+the+Expert/Pediatrics/_Target+heart+rates+for+kids.htm (01 Feb. 2014).

[20]: Schell, J.. *The Art of Game Design - A Book of Lenses*. , Burlington, MA, USA, 2008.

[21]: Yee, N.. *A Model of Player Motivations*. 2005. Online in the internet: http://www.nickyee.com/daedalus/archives/001298.php?page=4 (February 28, 2014).

[22]: Baranowski, T. ,Baranowski, J.,O'Connor, T.,Lu A. S. & Thompson, D.. *Is Enhanced Physical Activity Possible Using Active Videogames?*. In: Games for Health Journal: Research, Development, and Clinical Applications, 1:3, 2012, 228-232.

[23]: Deci, E. L. & Ryan, R. M.. *Intrinsic motivation and self-determination in human behavior*. , New York, 1985.

[24]: Deci, E. L. & Ryan, R. M.. *The" what" and" why" of goal pursuits: Human needs and the self-determination of behavior*. In: Psychological Inquiry, 11:4, 2000, 227-268.

[25]: Frederick, C., & Ryan R. M.. *Differences in motivation for sport and exercise and their relations with participation and mental health.*. In: Journal of Sport Behavior, 16:1, 1993, 124-146.

[26]: Frederick, C., & Ryan R. M.. *Self-determination in sport: A review using cognitive evaluation theory.*. In: International Journal of Sport Psychology, 26:, 1995, 5-23.

[27]: Ryan, R. M., & Deci, E. L.. *Intrinsic and extrinsic motivations: Classic definitions and new directions*. In: Contemporary educational psychology, 25:1, 2000, 54-67.

[28]: Gray, S.. *The Extent of Physical Exercise and Motivations in Exergames*. Honours Dissertation, Heriot-Watt University, 2013.