Linköping University | Department of Computer and Information Science
Bachelors' thesis, 16 ECTS | Computer and Information Science
Autumn term 2022 | LIU-IDA/LITH-EX-G--22/084--SE

2D Beats: Fast paced full body movement game using Godot Engine

John Karlsson

Tutor: Erik Berglund
Examiner: Anders Fröberg

LINKÖPINGS UNIVERSITET

Abstract

The world is quickly moving to a more and more digital world, everything is online and computerized. This is convenient but also increases the amount of time spent in front of various types of devices and displays, not getting the exercise we as humans need. Gaming or the act of playing video games is very much a part of this and is traditionally thought of as a sedentary activity. However, there is a type of video games that does not fall into this category, that is exergames. Exergames distinguish themselves from sedentary games in that they use the human body for control, leading to their use to combat the lack of exercise. This thesis describes the implementation and testing of one such exergame that is a part of a larger project at Linköping University that aims to create a library of games to be used in organizations where sitting for long periods is common. The game is implemented using the game engine Godot and an ML solution called MediaPipe for movement tracking and recognition. The goal was to create an easy to play game providing moderate to high physical activity. Testing showed that the game provided the desired physical exertion within a few consecutive rounds and exertion remained on roughly the same level even if the user was experienced.

Keywords
Exergames, Physical activity, MediaPipe, Godot, Game Development

Acknowledgement

A special thanks to Erik Berglund for giving me the opportunity to work on this project and all the help and advice received. A thanks also to all the creators on itch.io and Kenney at kenney.nl that provide free assets for game developers. Without help, advice, and free assets making this game and thesis would not have been possible.

Table of Contents

Images, Figures, Tables and Maps

A listing of all tables and figures included in the thesis.

Figures

Tables

1. Introduction

As the world moves toward an ever more digital future, more and more people spend an increasing amount of time in front of their tv, computer, or phone. According to Guthold et al. about a quarter of the world's population was not getting the exercise needed to stay healthy, with even higher numbers in high-income western nations [2]. So why spend so much time inactive? Mainly because of the ease of access to entertainment through tv, movies, and games. According to Lenhart et al. the vast majority of teens play video games [8]. That is why Exergames may be of importance if we want to stay healthy. Exergames are games that are controlled by a physical action, for example by moving left, right, or raising arms above head. Combining these movements effectively with an action in-game and an appropriate tempo, and a physically demanding exergame is born.

Exergames are not a new concept, there are examples of exergames as far back as the 80's with the Atari Joyboard, or perhaps more famously Dance Dance Revolution in 1998 and the Wii in 2006 or Xbox Kinect in 2010 [1]. The success and popularity of exergames first rose with the success of Dance Dance Revolution, spreading to arcades all over the world during the late 90's and early 2000's, and still creating new versions of the game as of 2022. Exergames' popularity then further grew with the release and subsequent success of the Wii, selling over 100 million consoles and 900 million software copies[1]. With this increase in popularity the notion that gaming was inherently sedentary slowly started to change.

This thesis focuses on creating a simple exergame that takes inspiration mainly from the VR game Beat Saber. Beat Saber is a VR rhythm game where the objective is to hit blocks in rhythm with a song, the song and timings of the blocks to the song are premade maps often called beatmaps. The decision to create a game like Beat Saber was made because of its success and physically demanding nature.

1.1 Motivation

At Linköping University there is a team at the Department of Computer and Information Sciences working on a gamification project with the goal of creating a library of games. This library of exergames is intended for companies and organizations where sitting for large portions of the workday is common. The library is provided through a gamified framework where the idea is for employees to play one of the short exergames roughly every thirty minutes and compete with their colleagues over high scores in the games.

1.1 Aim

This thesis aims to create and examine a well-designed and polished exercise game that can be easily picked up and played by office workers during short breaks and provides moderate to high physical activity.

1.1 Research Questions

1. How can an exergame be created to achieve a high degree of exertion?
2. Does the game require a high degree of exertion for an experienced user?

[1] Nintendo. IR Information : Sales Data – Dedicated Video Game Sales Units https://www.nintendo.co.jp/ir/en/finance/hard_soft/ (viewed 2022-10-21)

2. Background

In this chapter background knowledge needed for the coming chapters is presented.

2.2 Godot

Godot is a free and open-source game engine that uses GDScript as its scripting language, where GDScript is a high-level programming language that uses syntax similar to Python. Building games in Godot revolves around scenes and nodes. Nodes are premade game design elements, for example the AudioStreamPlayer and Sprite nodes, where one plays audio files and the other displays a texture or image. These nodes can then be combined in a scene to make a game or a component of a game, additionally scenes can be combined in scenes. Another important feature of Godot are Signals, for example the node Area2D has a signal called area_entered which will detect if another Area2D overlaps and emit, or send, the signal. This can then either manually or automatically be connected to a function, thus running the function when the area is entered.[2]

2.3 MediaPipe

MediaPipe is an open-source project created by Google that offers customizable machine learning solutions. MediaPipe contains multiple different premade solutions, for example face detection, face mesh, object detection, and pose.[3] The Pose solution infers 33 different 3D landmarks on the whole body from video utilizing their previous BlazePose research. Optionally MediaPipe Pose can predict a full-body segmentation mask, separating human and background. However, this project uses landmarks for movement tracking. MediaPipe is able to achieve real-time performance on most phones and computers, whereas other approaches mainly use powerful desktop computers.[4]

[2] Godot Engine. Godot Engine - Features. https://godotengine.org/features#design (Viewed 2022-10-20)
[3] Google. Home – mediapipe. https://google.github.io/mediapipe/ (Viewed 2022-10-20)
[4] Google. Pose – mediapipe. https://google.github.io/mediapipe/solutions/pose (Viewed 2022-10-20)

3. Theory

3.1 Exergames

What are exergames? One study by Sinclair et al. describes exergames as "… the use of video games in an exercise activity" [4]. Oh et al. defines exergaming as, "… experiential activity in which playing exergames or any videogames that requires physical exertion or movements that are more than sedentary activities and also include strength, balance, and flexibility activities" [5].

There are multiple advantages exergames have over normal sedentary games, the most important of which being the exercise the player gets. Staiano et al. highlight the fact that obesity rates are skyrocketing and that especially among 12- to 17-year-olds exergames could be a means to combat the crisis [3]. This is because according to Lenhart et al. 99% of boys and 94% of girls ages 12 to 17 play video games [8]. Another benefit of exergames Staiano et al. highlight is that skills learned during play can transfer to real life activities, benefiting physical, social, and cognitive development [3].

There are several studies on whether exergames can be considered exercise. Marshall et al. performed a review of existing literature and found that exergames are comparable to exercise such as walking, jogging, and dancing under specific circumstances. Further, they found that only 9.8% of exergames met the standard for intense exercise and only 45% the standard for moderate exercise. They also found that resulting exercise strongly depends on game design, and that dancing games usually achieved moderate exercise [6]. Another review by Sween et al. found that a strong correlation between exergames and increased energy expenditure existed, and that most of the tested games met the standard for moderate exercise [7].

3.2 Game Design

Why are some games more successful than others? This is the question game design researchers try to answer by attempting to find the link between design elements in games and their success. Clanton presents two principles to follow in game design, "hook 'em fast and hard", and "keep 'em hooked". These principles are broken down into elements such as establishing a quest, providing a gentle on-ramp, pressure can be fun, and give hints not answers [11].

Sweetser et al. proposed a game-flow model based on Csikszentmihalyi's [15] concept of flow [12]. The game-flow model strives to be a general model of player enjoyment that applies to all game genres. The model consists of eight elements, concentration, challenge, player skills, control, clear goals, feedback, immersion, and social interaction. Each element has one or more criteria to be fulfilled if applicable. Sweetser et al. also note that not all elements and criteria can or should be fulfilled for every game, game-flow is a very general model [12].

Hunicke et al. created a framework for game design and game research that they call the MDA framework. MDA stands for Mechanics, Dynamics, and Aesthetics, which are the design counterparts of rules, system, and fun. Mechanics describe game components at the level of data representation and algorithms. Dynamics describe the run-time behavior of mechanics acting on players' input. Aesthetics describes desired emotional response by the player. They present eight aesthetic components that they mean make a game fun. These are sensation, fantasy, narrative, challenge, fellowship, discovery, expression, and submission [13]

3.3 Exergame Design

Designing an engaging and fun videogame is a difficult task, adding to that the additional challenges of designing a movement-based game and it becomes very difficult. Luckily there are studies that present a set of guidelines, for example the ones presented by Mueller et al. for developers and designers of movement-based games. The guidelines are based on previous research and reviewed by 14 movement-based game design experts with both commercial and academic backgrounds. A simplified version of their guidelines follows below,

- Embracing ambiguity of movements, forcing precise movements can be frustrating. Instead embrace the ambiguity of human movements and adjust for sensor error.
- Provide feedback on how well a movement was performed, well performed movements should give more points than a worse performed one.
- Maintain a balanced level of physical activity, for example by using short game cycles of intense physical activity or varying movements.
- Map movements in an imaginative way, the movement in game does not need to be the same as the one performed, unless it's needed for the game.
- Allow players to perform different kinds of movements to achieve the same outcome, supporting self-expression.

Mueller et al. also note that while overall the experts thought the guidelines would be useful and confirmed what they themself found, they had previously not followed all the guidelines and yet created engaging gameplay [9].

One of the bigger issues in game design is how to keep the player coming back to the game. Exergames are not exempted from this issue as Madsen et al. found that the willingness of the children in the study to return to the game decreased over time and after only four weeks roughly half found the game boring [10].

4. Method

After implementation of the game an evaluation will be carried out by having one person play the game 100 rounds to collect data. Data collected will be the total number of pixels moved, shoulder width, movement score, registered punches, points score, number of bonus points, and amount of multiplier breaks. In the last 30 rounds the number of selected paths is also measured.

Measurement	Explanation
movement_px	Total sideways movement by the players center mass position in pixels.
shoulder_width	Distance measured between the players shoulders
movement_score	The number of pixels moved divided by shoulder width.
registered_punches	Number of punches successfully registered.
points_score	Number of points scored during the run
bonus_points	Number of points that were a bonus because of the multiplier system.
multiplier_breaks	Number of times the player broke a multiplier they had built up
selected_paths	Number of paths that were active this round. (Only collected last 30 rounds)

Table 1. Measurements taken and a brief explanation for each.

The above measurements are taken during a gameplay round, a 2-minute-long period of playing the game continuously. The measurement values are calculated in various ways, some are straight forward, like points score or registered_punches, others like movement_px and shoulder_width are a bit more complex. In the movement_px case the difference in pixels between the sights current position and its position last frame is added to the movement_px variable.

Shoulder_width is an average width throughout the game round. Every frame the current shoulder width is calculated as the left shoulders distance from zero minus the right shoulders distance from zero in MediaPipes coordinate system. This calculated width is added to an array which is used to calculate the average shoulder width at the end of the round.

At the end of each round the measurements are added together and output and downloaded through the browser. The output file is a CSV file for ease of use. If the file does not already exist as a save file in Godot it is created and measurements added, otherwise the measurements are added to the end of the file keeping the previous rounds values.

With some of the more important measurements coming more or less directly from MediaPipe the accuracy of MediaPipe's output is important. Halder et al. performed a study on sign language recognition using MediaPipe and found that in all 3 different sign languages tested MediaPipe performed better than other methods. They also found that in their case MediaPipe had an average accuracy of 99% in most of their sign language datasets [14].

5. Results

5.1 Game implementation

How the game was implemented using Godot and MediaPipes features.

5.1.1 Scene Structure

2D Beats is implemented in four main scenes, HitBlock, BlockPathSpawner, PlayerAndSight, and RythmGameMain. There is also a fifth scene that collects all the above scenes into one and contains the premade UI elements and connections to MediaPipe and the larger Gamification project. This scene is provided by the Gamification project team and is called top_node, the scene tree for top node can be seen in figure 1.

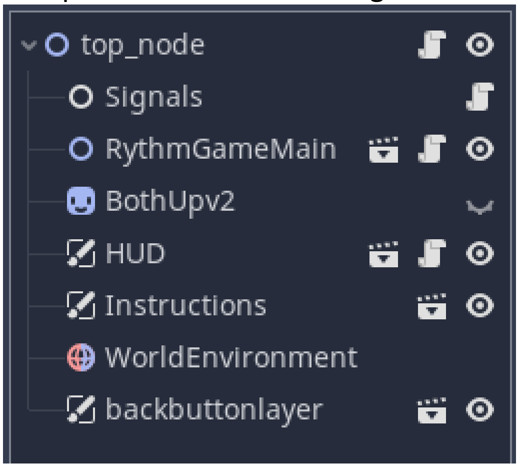

Figure 1. top_node scene tree, with nodes

The HitBlock scene is responsible for adding points when hit, and randomly selecting its own color. The HitBlock is implemented using an Area2D node because this makes hit detection easy, simply using Godots built in area entered signal. BlockPathSpawner is responsible for randomly selecting between four and six premade paths, drawing these paths, and then continuously spawning HitBlocks on these paths with a 1.5s interval. PlayerAndSight is responsible for moving the sight using data received from either the keyboard or the camera, and then snapping the player to the closest path to the sight.

The reason a sight is used over just moving the player directly is because of the snap system, if the player was moved directly it would be less clear what happens when a snap occurs. With a sight there is always something connected directly to the physical players' movement, showing clearly where you are in the game world uninterrupted. Meanwhile without a sight there would be at least a second or so where the player would have to stop at a snap position before being able to move again, breaking that movement and possibly confusing the physical player as to where he/she is in the game world.

The PlayerAndSight scene also has a subscene purely for the player, this scene handles the animation of the player sprite. And RythmGameMain collects all the above scenes into one main one, as well as adds some HUD elements and audio nodes.

5.1.2 Using MediaPipe landmarks

The Pose Landmark Model from MediaPipe is used by the gamification project, it contains 33 landmarks. Of which 2D Beats uses 5, the nose, both shoulders, and both elbows. The nose and elbows are used for punch detection. Since the chosen punch movement was straight up a punch can be detected by checking if the right or left elbow is above the nose. The reason why the elbows position is used and not for example the wrists is because using the elbows provides a more challenging movement.

The shoulders are used to get the sideways movement in-game by adding the shoulders coordinates together and dividing by two to get a center mass position, providing a more reliable tracking than for example the nose. MediaPipe outputs coordinates in a in the form of a value between 0 and 1, which usually needs to be interpreted into something more usable in Godot.

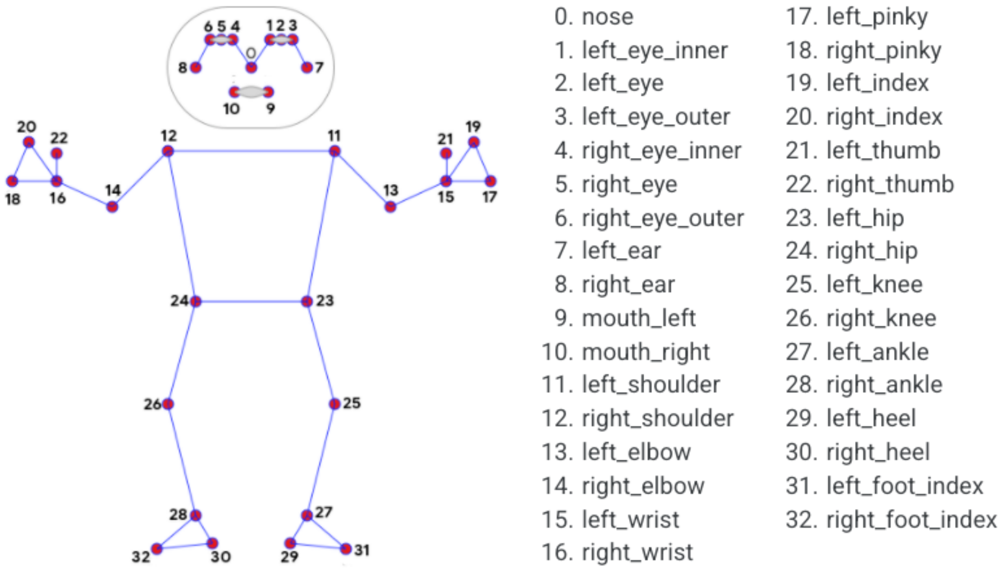

Figure 2. MediaPipes 33 different landmarks and what they represent. **Source:** Google. Pose – mediapipe. https://google.github.io/mediapipe/solutions/pose (Viewed 2022-10-20)

5.1.3 Controlling the player

Player control is handled in two different scenes, top_node and PlayerAndSight. In top_node translation from MediaPipe output values to usable values is handled. The sights position is calculated as follows, first get the left and right shoulders x coordinate from MediaPipe and calculate the middle point of these coordinates by adding them and divide by two. Then clamp this middle point between 0.1 and 0.9 to ensure movement outside screen is impossible, and finally multiply the clamped value with the display resolutions width, in this case 1920, to get a pixel coordinate. The reason why this value is needed over for example using the noses x coordinate is because it’s a more stable point to measure actual side movement. The nose can much easier be moved sideways by for example, rotating the head sideways, which could be a way to then cheat. Finally this center position between shoulders in pixels is passed to PlayerAndSight where the sight is moved using a lerp function.

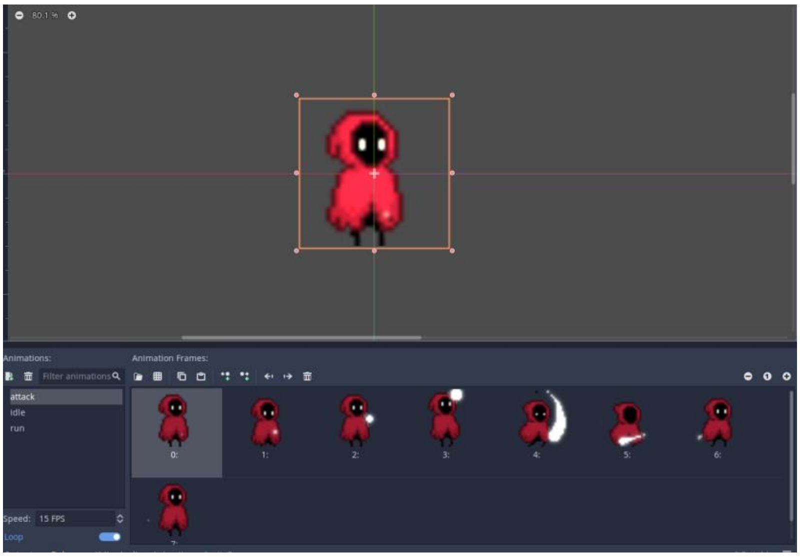

Figure 3. The player sprite and it's animations

Punches or attacks are registered by raising either the left or right arm above the nose, practically this is done by checking that the left or right arms elbow y coordinate is above the noses y coordinate and then signaling PlayerAndSight that an attack has been registered. Then the playing of animations and moving the attack hitbox is handled in PlayerAndSight. The player has three animation states, idle, running, attacking. Which state is active is handled by the Player node and depends on which velocity the sight has and in the case of the attack animation if an attack has been signaled.

5.1.5 Spawning Blocks

The spawning of blocks and selection of paths is handled by BlockPathSpawner. Firstly, a set of between 4 and 6 paths needs to be selected from a total of 11 possible. This is done by randomly selecting one of the first four paths as a sort of 'seed' path, and then every other path after the seed-path. The reason why every other is selected is that the paths would be far too close to each other otherwise, it would be very difficult, if not impossible to snap to and hit a HitBlock on the correct path. Once the selection is done, the paths are drawn on screen to be visible since a path in Godot is not a visible node by default.

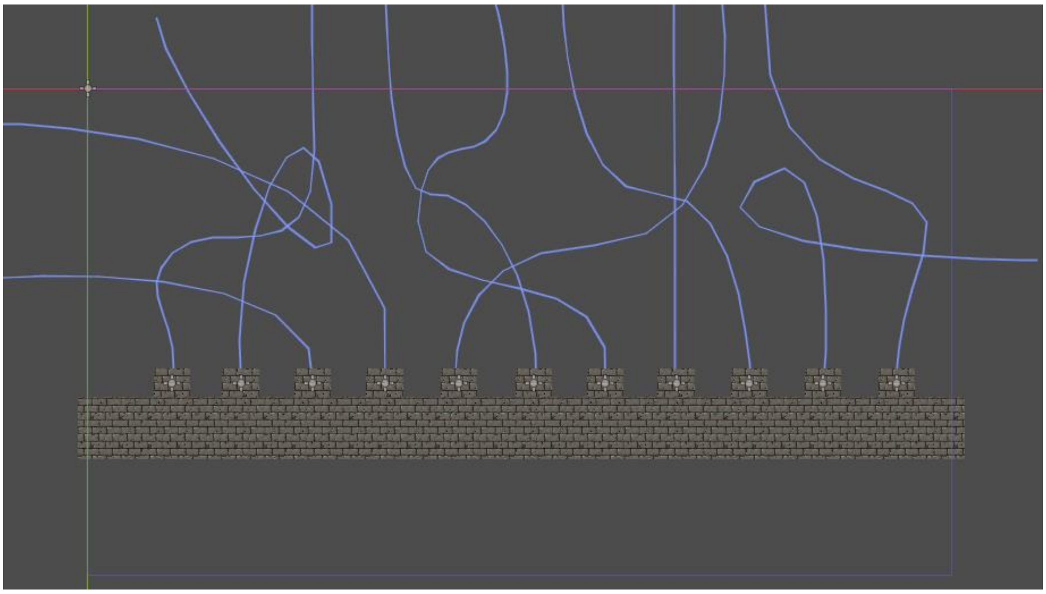

Figure 4. Spawner with all possible paths visible.

Once paths are selected and drawn the spawners’ main task can be started. A new HitBlock is spawned on a random path every 1.5 seconds. Once spawned it starts moving from one end of the path to the other with a speed calculated to ensure the HitBlock arrives at the end in 3 seconds. When it arrives at the end of the path it is added to a deletion queue and then deleted.

5.1.4 Scoring system

The scoring system is handled mainly by two scenes, HitBlock and BlockPathSpawner. As mentioned earlier each HitBlock adds points to a total when they get hit. The number of points added depends on two variables, the color of the HitBlock and what if any multiplier is active. The color of the HitBlock determines what base number of points it has, red is one, yellow is three, blue is five, and green is ten. When hit the base number of points multiplied by the multiplier is added to a total score. The reason why there are blocks with different points and colors is because the spawner was originally intended to spawn more blocks so that the player would have to choose the most valuable blocks to hit and maybe miss some others. This idea was moved away from during development but the original version of the HitBlock stayed because no reason to change it was found during development.

Figure 5. The game right after hitting a HitBlock with the multiplier active

The multiplier system works as follows, each HitBlock tracks whether it was hit or not and later just before when the HitBlock is deleted 0.25 is added to a multiplier variable if the block was hit, otherwise the multiplier variable is reset to 1. For example, figure 4 shows the game after hitting a HitBlock with base number of points of three and with the multiplier 2.75, adding 8.25 to the total score and increasing the multiplier to 3.

5.2 Evaluation

Once implementation of the game finished the process of collecting data for evaluation started. This process consisted of playing the game for 100 rounds to collect data, this was done by the author. The results from this data collection process will be presented here. It is split into two sections, original data, and additional data. This is because during the data collection stage, it was discovered that two of the game systems impacted the amount of sideways movement and final total score respectively. And it was decided that the impact of these systems needed to be considered. Unfortunately, the first was discovered after roughly 40 rounds had been measured, so these 40 rounds were replayed to collect the additional data. The second was discovered around round 70, instead of replaying all 70 previous rounds only the remaining 30 rounds collected the new data.

5.2.1 Additional data

Starting with the additional data, the offending systems were the multiplier system and the path selection system. The multiplier system greatly impacted the total score, it was observed that more than half of the total score usually consisted of bonus points. This was later confirmed by measuring the number of bonus points which showed that on average 76% of the total points were bonus points. The results are present in figures 6 and 7, in figure 6 we see that a majority of the time 75-79% of total points is made up of bonus points and in figure 7 we can see the same thing but instead as a line graph over the 70 rounds.

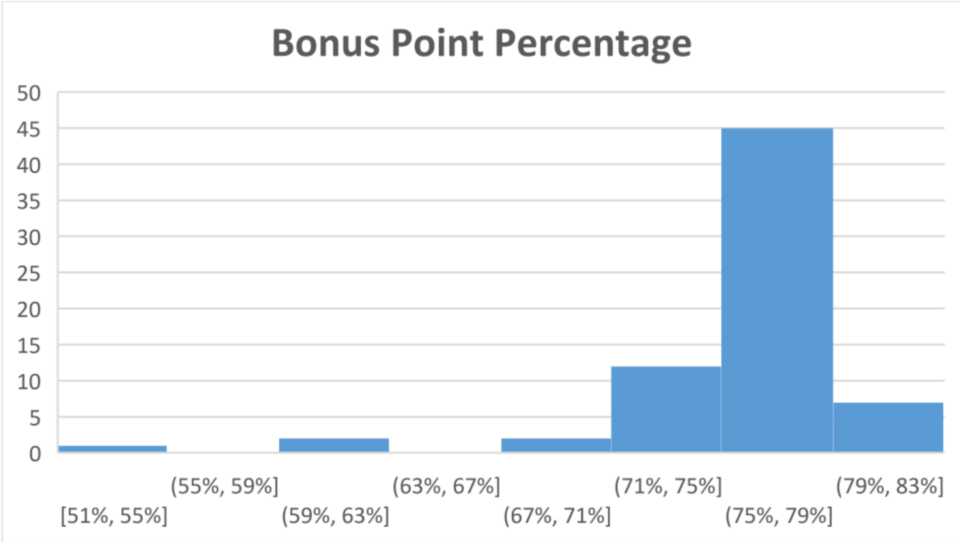

Figure 6. Histogram showing bonus points as percentage of total points.

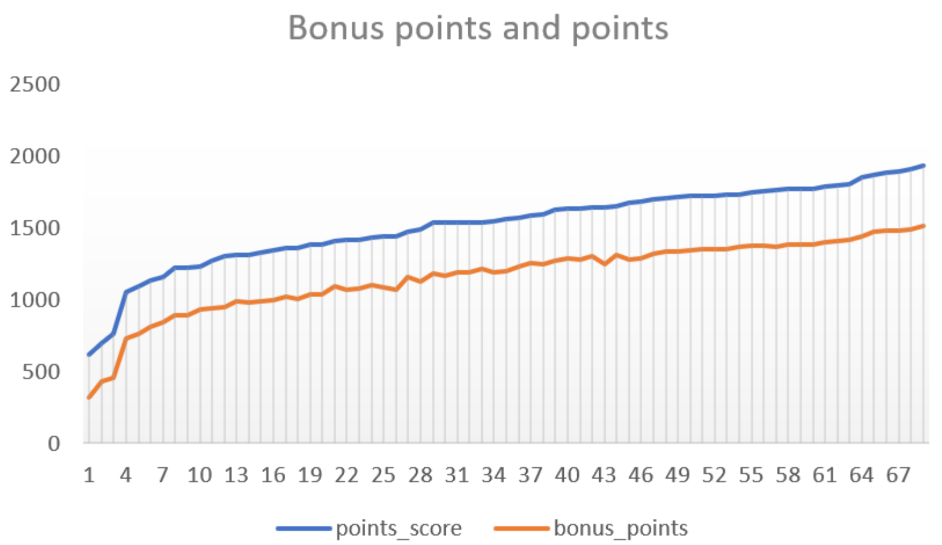

Figure 7. Line graph showing total points and bonus points over 70 rounds.

The path selection system impacted the amount of sideways movement and the movement score since they are connected. It was observed during the original 70 rounds that whenever the game gave the player 4 paths the movement score would be lower, and with 6 paths higher. This was then confirmed by measuring the number of paths each round and comparing with movement score, showing that the lowest scores were with 4 paths, median scores with 5 paths, and highest scores with 6 paths.

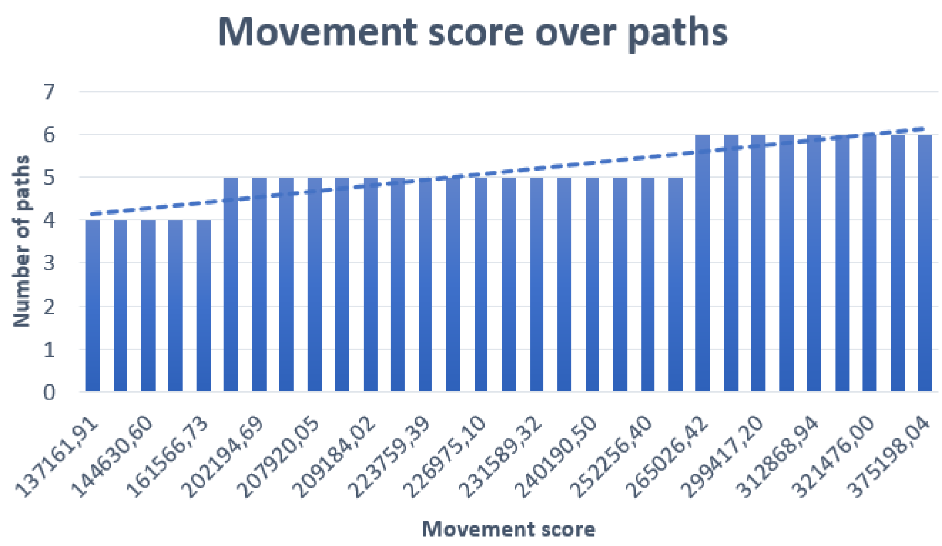

Figure 8. Bar graph showing change in movement score as number of paths increase.

5.2.2 Original Data

The original data was collected to show the evolution of movement scores over time, the relationship between movement scores and point scores, the relationship between multiplier breaks and scores, and finally registered punches. The evolution of movement scores over the rounds played can be observed in figure 9 below, we see a very jagged line that trends slightly downwards. The jaggedness being because of the multiple game systems that use randomness, meaning that for some rounds more paths were active and required more movement and for some not.

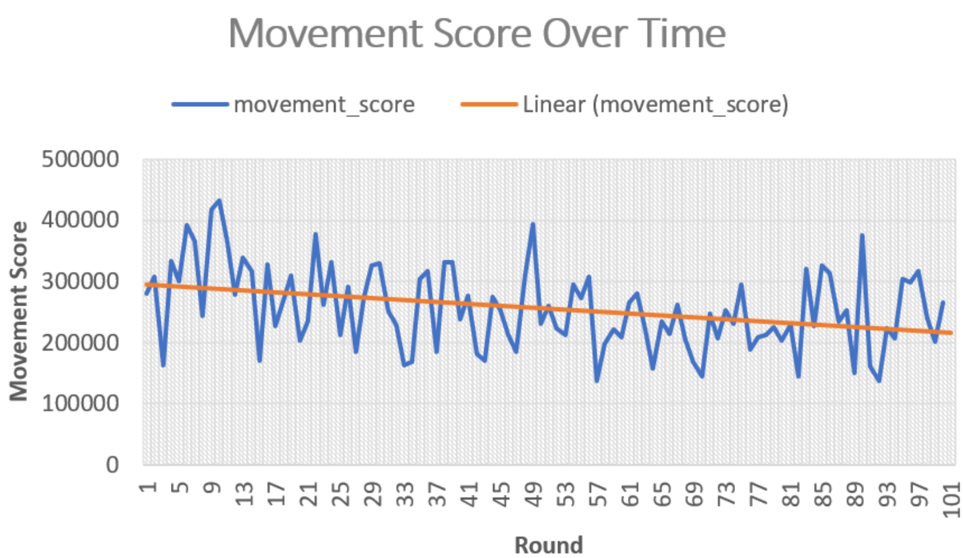

Figure 9. Movement scores over rounds, trendline in orange.

The relationship between movement score and point scores is presented in figure 10, like in figure 9 we see a jagged line because of the randomness used in the game but this time it does not trend either upward or downward, it is more or less straight. The relationship between multiplier breaks and point scores is presented in figure 11, we see a sawtooth like line which clearly trends downwards with increased multiplier breaks.

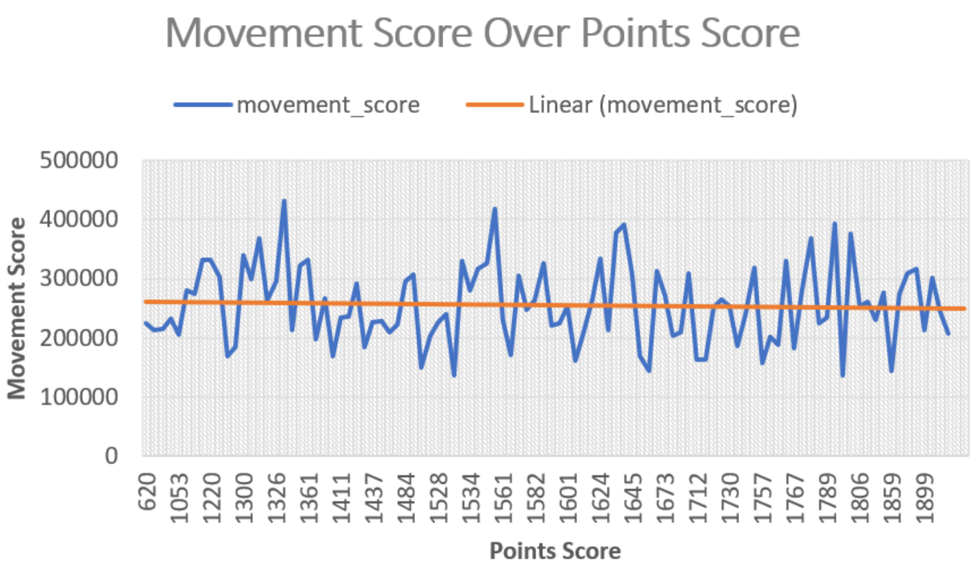

Figure 10. Movement scores relationship with points, trendline in orange.

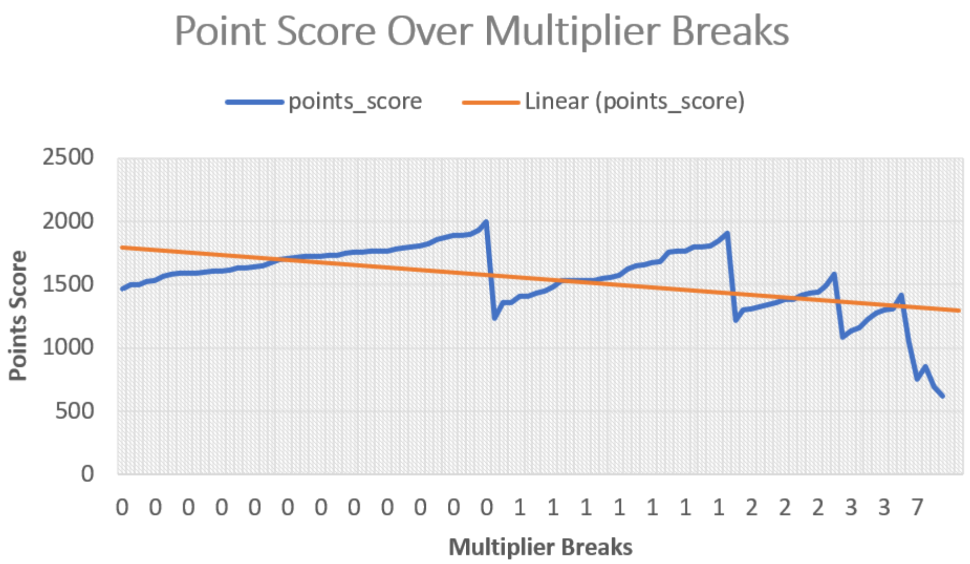

Figure 11. Point scores relationship with multiplier breaks, trendline in orange.

6. Discussion

6.1 Method

The biggest issue with the chosen method would be that it may not be representative of real-world usage of the game. This is because testing was only conducted using one subject, the author of this paper. Naturally the experience and physical ability of one person does not accurately represent the general public, not to mention the possible bias of the tester being the creator of the game. However, in this case, since we were looking to investigate whether an experienced user had a similar degree of exertion as a novice, the used method can still deliver reliable enough results. But with these apparent issues why not choose a different method? The reason this testing method was chosen is the short duration of the project as a whole and how little of the project time remained for testing once implementation was complete. There wasn't enough time to set up a study with more participants.

Because of the use of only one test subject data on whether the game was physically demanding was also limited to the general experience or feel over the 100 rounds of the testing. It is the hope and opinion of the author that the number of test rounds being relatively high will make up for some of the deficiencies in the method.

However, this method also offers some benefits, because the tester also made the game and thus already played it during development and know how the systems work it becomes very easy to see how an experienced player plays. For example, I observed during testing that there was an optimal distance from the camera where the sideways movement was easiest, roughly at 2 meters. The probable reason for this is because the sideways movement in-game converts the center mass coordinates directly to pixels not accounting for distance from camera, meaning that when standing closer a small physical movement converts to a bigger movement in-game and vice versa. The experienced nature of the player may also make it easier to trust the subjective opinion on how exerting the game turned out to be, which I found to be quite high but only under certain conditions. The exertion was high when a round with 6 active paths was played since the side movement then becomes higher because you need to move to more paths and usually further and faster than with less paths. And the exertion also usually became high withing 2-5 rounds played in succession, this then being because you usually got a round with more paths (5-6) in that time.

6.2 Results

6.2.1 Game Implementation

The implementation of 2D Beats could have been done in many different ways, for example the choice of game engine or game design choices would have changed the implementation. Choosing a different game engine would have led to differences mainly in the coding of the game but also what assets would be available since different engines have different marketplaces for assets. Although there are independent marketplaces like itch.io which were mainly used in this project since Godot's asset library is still quite limited, the point being that with a different engine it's possible that more assets would have been available. Whether this would have led to a better result is doubtful, the graphical fidelity of the game is not as important in this case since user enjoyment was not considered in this study. Due to the many decisions made during implementation and the fact that many of the games systems are implemented across different nodes describing the implementation to a degree where another study could replicate the game without any big differences would be very difficult. An effort was instead made to sufficiently describe the most important systems and how they interoperate in the hope that the core elements could be replicated.

Something that would most likely impact the result are the game design choices, for example the choice of movements to control the player and the design of the multiplier and path selection systems. The choice of different player control movements would naturally impact the result. For example, the chosen movement for attacking is to raise either wrist above the shoulder and then lower it to below shoulder height again to both make it a more physically challenging movement and preventing continuous attacking. Choosing a different movement here would change the resulting exertion either up or down, for example if the movement instead was to move either wrist above the head and then back the exertion would likely be higher. Regarding the design of the multiplier and path selection systems an alternative design could have made the game more fun to play. As it is now doing better in the game i.e., getting higher points score, is not directly tied to how much the player has moved. Ideally you would want how well the movements and how much you move to be better reflected in the points as Mueller et al note [9], otherwise there is no incentive to move more.

6.2.2 Evaluation

Figure 6 shows the percentage of total points that are bonus points and figure 11 the relationship between total points and multiplier breaks. These figures show that the multiplier system greatly influences the number of points received. However, figures 6 and 11 also suggest that either it’s too easy to get a high multiplier or more testing is needed to get data from rounds with lower multipliers. Which option is hard to determine as there only was one tester making a comparison of skill impossible, but it’s most likely possible to make the multiplier system harder without making the game too hard.

Arguably the most interesting of the results can be seen in figure 9 where the change in movement score over time can be observed, but only if you add a trendline since the line is very jagged. This is because of the randomness used quite heavily in the game, for one the number of paths is randomized, and secondly what value the HitBlocks have is randomized. This leads to some rounds giving more points than others because more high value HitBlocks were spawned, which is not great game design wise because the player is in effect not in control of how many points they earn. Over a large number of rounds the pattern still becomes clear but a player may not stick around for that long. This figure then also shows why and how the HitBlock maybe should have been redesigned when the behavior of the spawner was changed during development. The HitBlock should have or at least could have been redesigned to not randomize how many points it spawns with, in order to minimize the issue that some rounds give more points than others even if played similarly.

6.3 Source Criticism

Sources for information used to write this thesis were carefully chosen to be as reliable as possible. A reliable source was any scientific paper published by a reputable journal or previously cited in other papers a good number of times. All references listed under Reference List and used throughout the thesis follow these rules. There are also 4 non-scientific sources used for information on open-source projects and game console sales numbers, these are presented as footnotes so as to not confuse them with scientific sources.

7. Conclusion

To conclude the research questions will be answered and an answer to whether the thesis fulfilled its purpose will be given. Additionally, suggestions for possible future work will be discussed. The aim was to create an easy to pick up, polished, exergame that provided moderate to high physical exercise. This has been at least partially achieved, the game is easy to pick up and play with only two different controls and achieves moderate to high exercise within a few rounds. However, the game polish is questionable as there are game elements that could still be improved.

7.1 Research Questions

How can an exergame be created to achieve a high degree of exertion?
The game design took inspiration from Beat Saber because of its physically demanding nature, and movements for player control were chosen to be physically demanding. That the game achieved a high degree of exertion was later confirmed as the tester found that after playing around 2-5 rounds the exertion was high.

Does the game require a high degree of exertion for an experienced user?
Looking at the evaluation results we can quite clearly see that the movement score that measures how much the user has moved during a gameplay round trends slightly downwards as the rounds progress. This suggests that the degree of exertion decreases somewhat as the user gets more experienced. This is further backed up by the experienced exertion of the tester which was that after playing a few rounds, around 2-5, the exertion was high. This also did not change over the course of the 100 rounds.

7.2 Future Work

Due to the limited time spent on this project there are multiple areas that could have been further developed. First and foremost, the testing is lacking, in the event of future work more time should be allowed for testing and evaluation. This time should then be used to conduct testing with more than one subject which should result in more reliable and replicable results.

Another area that could have used more development is the implementation of the game, again due to time constraints the polish of the game is lacking. Balancing of certain features like the multiplier system could use work for example as previously discussed. And certain player animations don’t properly work using the camera as input which is something that clearly hurts the polish of the game.

Finally, the game implementation could have focused more on matching the amount of movement to the points score received at the end of each round. As it stands the amount of movement isn’t reflected in the points score to the degree that it’s noticeable between single rounds, this could give the user the impression that moving more is not beneficial.

8. Reference List

[1] M. D. Finco and R. W. Maass, "The history of exergames: promotion of exercise and active living through body interaction," 2014 IEEE 3rd International Conference on Serious Games and Applications for Health (SeGAH), 2014, pp. 1-6, doi: 10.1109/SeGAH.2014.7067100.

[2] R. Guthold, G. A. Stevens, L. M. Riley, and F. C. Bull, "Worldwide trends in insufficient physical activity from 2001 to 2016: a pooled analysis of 358 population-based surveys with 1·9 million participants," The Lancet Global Health, vol. 6, no. 10, pp. e1077–e1086, Oct. 2018, doi: 10.1016/S2214-109X(18)30357-7.

[3] Staiano, A. E., & Calvert, S. L. (2011). Exergames for physical education courses: Physical, social, and cognitive benefits. Child development perspectives, 5(2), 93-98.

[4] J. Sinclair, P. Hingston, and M. Masek, "Considerations for the design of exergames." In Proceedings of the 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia (pp. 289-295).

[5] Y. Oh, and S. Yang. (2010). "Defining exergames & exergaming." Proceedings of meaningful play, 2010, 21-23.

[6] Marshall, J., & Linehan, C. (2020). Are Exergames Exercise? A Scoping Review of the Short-Term Effects of Exertion Games. IEEE Transactions on Games, 13(2), 160-169.

[7] Sween, J., Wallington, S. F., Sheppard, V., Taylor, T., Llanos, A. A., & Adams-Campbell, L. L. (2014). The role of exergaming in improving physical activity: a review. Journal of Physical Activity and Health, 11(4), 864-870.

[8] Lenhart, A., Kahne, J., Middaugh, E., Macgill, A. R., Evans, C., & Vitak, J. (2008). Teens, Video Games, and Civics: Teens' Gaming Experiences Are Diverse and Include Significant Social Interaction and Civic Engagement. Pew internet & American life project.

[9] Mueller, F., & Isbister, K. (2014, April). Movement-based game guidelines. In Proceedings of the sigchi conference on human factors in computing systems (pp. 2191-2200).

[10] Madsen, K. A., Yen, S., Wlasiuk, L., Newman, T. B., & Lustig, R. (2007). Feasibility of a dance videogame to promote weight loss among overweight children and adolescents. Archives of pediatrics & adolescent medicine, 161(1), 105-107.

[11] Clanton, C. (1998, April). An interpreted demonstration of computer game design. In CHI 98 conference summary on Human factors in computing systems (pp. 1-2).

[12] Sweetser, P., Johnson, D., Wyeth, P., Anwar, A., Meng, Y., & Ozdowska, A. (2017). GameFlow in different game genres and platforms. Computers in Entertainment (CIE), 15(3), 1-24.

[13] Hunicke, R., LeBlanc, M., & Zubek, R. (2004, July). MDA: A formal approach to game design and game research. In Proceedings of the AAAI Workshop on Challenges in Game AI (Vol. 4, No. 1, p. 1722).

[14] Halder, A., & Tayade, A. (2021). Real-time vernacular sign language recognition using mediapipe and machine learning. Journal homepage: www. ijrpr. com ISSN, 2582, 7421.

[15] Csikszentmihalyi, M., & Csikzentmihaly, M. (1990). Flow: The psychology of optimal experience (Vol. 1990). New York: Harper & Row.