Game Design influence on gaming experience in Pacman

2022-1st Semester

Interdisciplinary Project

Student ID	20182002
Name	Kasymzhan Abdyldayev
School	ECE
1 track	Computer Science & Engineering
2 track	Biomedical Engineering
Advisor 1	Myeongjae Jeon
Advisor 2	Hajin Kim

I. Introduction

1) Topic and Purpose of Research

Game industry is one of the most developing industries nowadays. Some statistics claim that in 2020, the gaming industry generated $155 billion in revenue, by 2025, analysts predict the industry will generate more than $260 billion in revenue[1].

In this interdisciplinary project I tried to assess how game design and implementation affects the gaming experience of players. “Pacman”, one of the most popular retro games, is chosen as a game of our interest. To conduct research, I implemented “Pacman” from scratch using Java-17 programming language and let people play my implementation and the original. Then conducted a simple survey that shed a further light on how to improve my version and make it better, as well as highlighting the key gaming experience influencing parts in this particular game.

You can download the source code from github.com/Qasym/pacman.

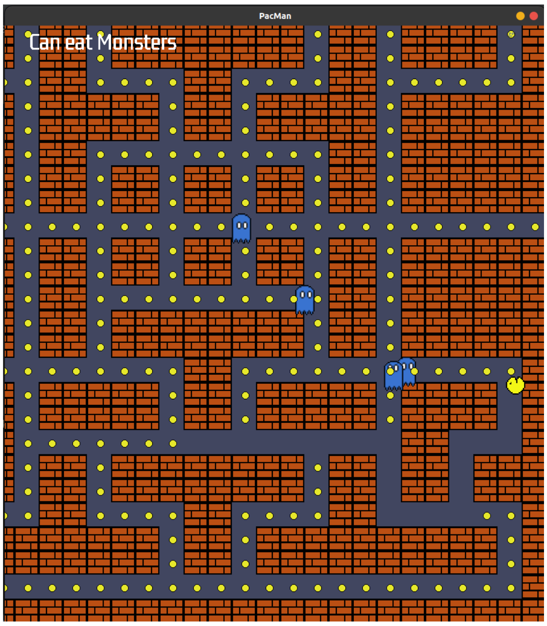
II. Main Subject

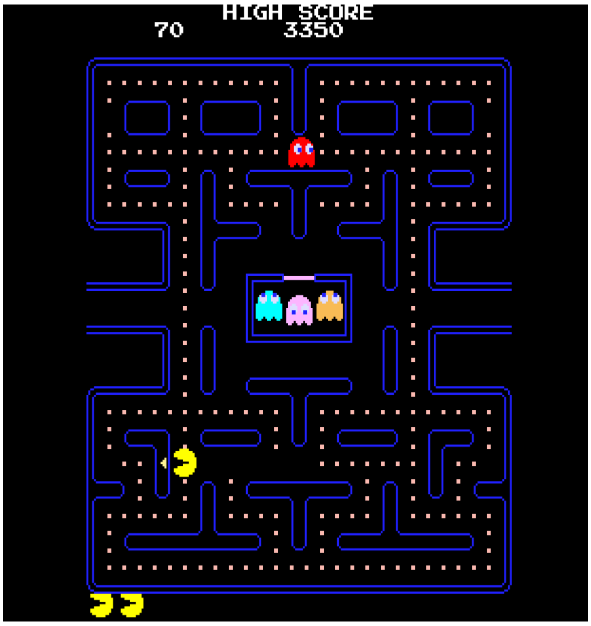
1) Research Planning

The first thing that needs to be done is to create two different versions of the same game and then compare how each element of game design affects the way the game is perceived. To do so, I am going to use this[2] version of “Pacman” and **refer to it as the original**, and I am going to implement my own game. From now on, I am going to go into the implementation details with a purpose of showing the differences that these 2 versions have, original and my respectively.

The first thing that appears as you open the game is the main menu, main menu music. It contains two buttons, “Play” and “Settings”, there is nothing unusual about this part, let’s press the “Play” button and start our game.

If you opened the original game I referred to, you can immediately notice that the size of the map that Pacman stays in in my version of the game is much larger than in the original, I created a camera that is centered on Pacman and follows him as he moves across the map (figure on the left), whilst in the original game we see the whole map right away (figure on the right).



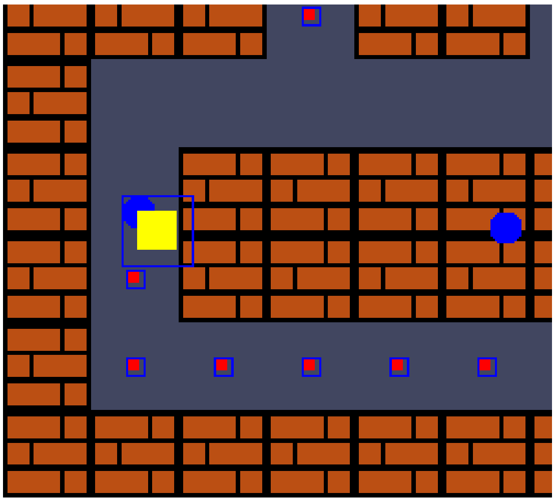
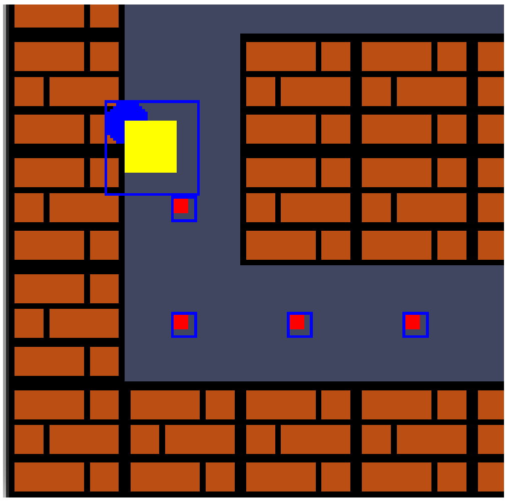


The next thing that differs my implementation and the original game is the absence of lifes of Pacman. In the original game the player is given 3 lives or attempts to finish the level, whilst my implementation does not give such an opportunity, you live only once.

Another point that is worth mentioning is the fact that the original game provides some form of music unlike my implementation which has music in the main menu but doesn't have any sounds in the game.

As for controls, the original game allows users to use both arrows and WASD keys, my implementation allows only WASD keys, though this feature can be adjusted very easily in the source code.

Now comes one of the most distinguishing features of my implementation, the movement. In the original game Pacman moves on "rails", meaning that there is no way for him to turn in any direction other than the directions that are available for him, those are limited by the corridor he is currently passing through. For instance, if Pacman moves on the lowest bottom line horizontally, he can only move left or right, unless he reaches the corners in which other directions are possible since they're intersections of two "rails". This design significantly simplifies turns for a player, since such design feature does not give player freedom to turn earlier, but forces him to turn correctly at each intersection. This is a strict movement design.
On the other hand, my implementation is not using "rails", instead, I defined collision bounds for each tile on the map and for each entity. Pacman can't move towards a tile if their collision bounds intersect, but as long as they do not, Pacman is free to move in any direction. Please, have a closer look at the figures below to see where the collision box for Pacman is located, blue rectangular area represents the area in which the Pacman sprite is drawn, yellow filled rectangle represents the collision box. Both figures show the same bottom left corner of the map, but with different Pacman locations, on the left figure Pacman is near the right wall, on the right figure Pacman is near the left wall. This design gives more freedom for player to move, but slightly increases the learnability of efficient turns (initially players get stuck to the wall due to early turn)

Another interesting aspect of Pacman is the intelligence defined for monsters. As all points before this, my implementation is slightly different than in the original game, though the whole inspiration was coming from the same source[3]. In the original game, there is a system of levels, at each level monsters behave slightly differently, for example they chase Pacman less and scatter more in the first level than in later levels, meanwhile in my implementation there is no notion of levels, the game is designed to be survival and how much can you score type of game, due to that monsters behave more aggressively immediately from the start.

If you played the original game, you might already have thought about power buffs that allow Pacman to eat monsters chasing him, eating a monster gives you a significant amount of points. My implementation also presents those buffs, additionally I have implemented a new type of buff which is called speed buff, it slows down monsters and doubles the points Pacman receives when he eats regular yellow balls.
We have to note that in the original game, eating a buff allows you to eat each monster only once, trying to eat it twice while Pacman has buff will result in "game over", my implementation gives much less points for eating a monster, but allows Pacman to eat them multiple times.

Minor differences include sprite differences, in the original game each monster has unique color and special effects when you eat them, my implementation does not present such features.

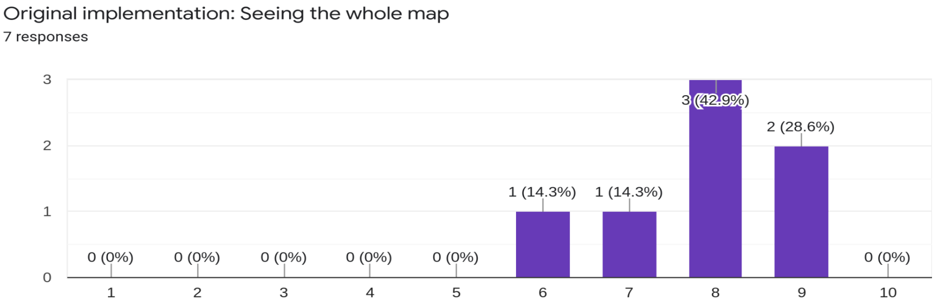
So far, I have specified the key differences in the game design between my implementation and the original game.
My research is about giving players an opportunity to play both games, and take a survey that assesses the differences I mentioned above.
You can take the survey here[4], also see the questions in the next section.

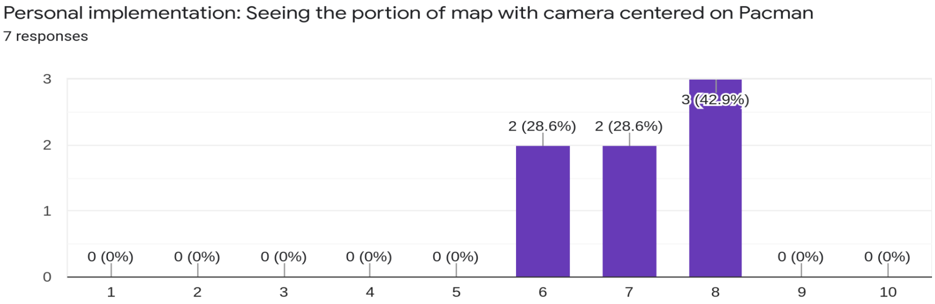
III. Conclusion and Discussion

1) Research Results

I asked 7 people to play both versions of the game and these are the results I obtained from the survey. The higher value indicates the higher enjoyment.

- Design choice 1: Seeing the whole map vs Seeing a portion of

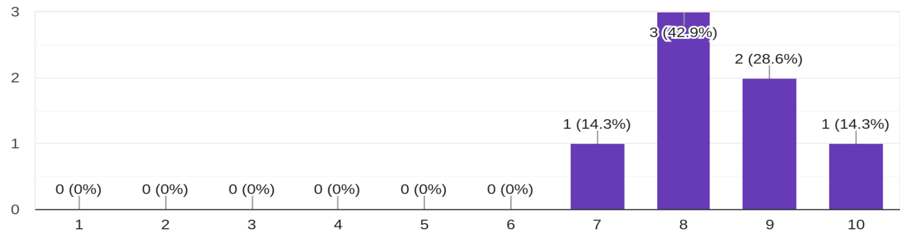




- Design choice 2: System of levels vs Survival mode

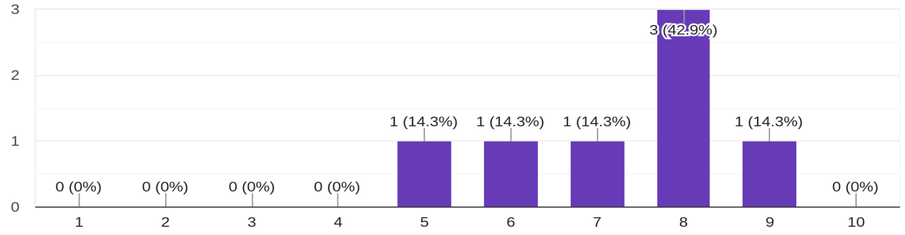
Original implementation: System of levels

7 responses



Personal implementation: Survival mode

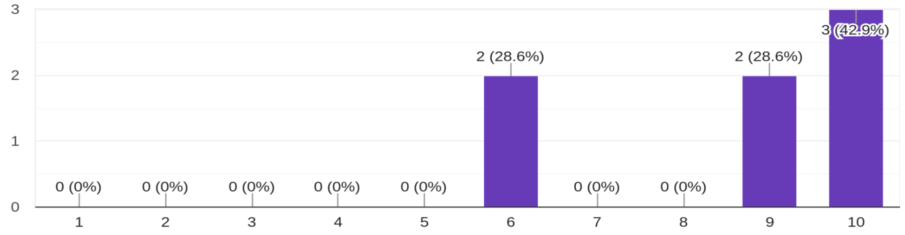
7 responses



- Design choice 3: Sounds during the game vs No sounds during the game

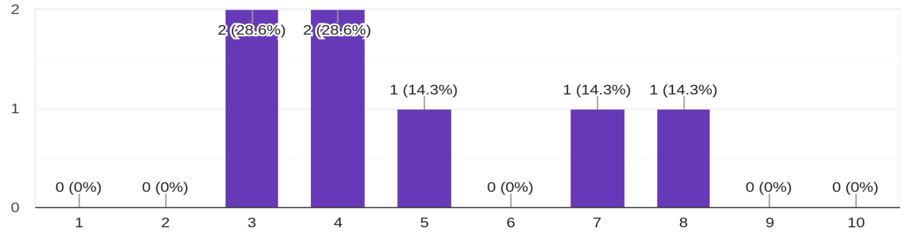
Original implementation: Sounds during the game

7 responses



Personal implementation: No sounds during the game

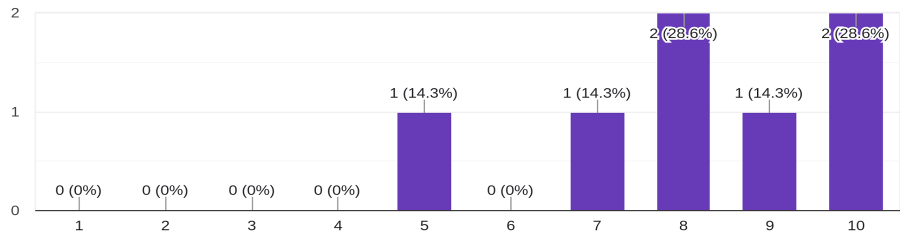
7 responses



- Design choice 4: WASD and arrow keys as controls vs Only WASD keys as controls

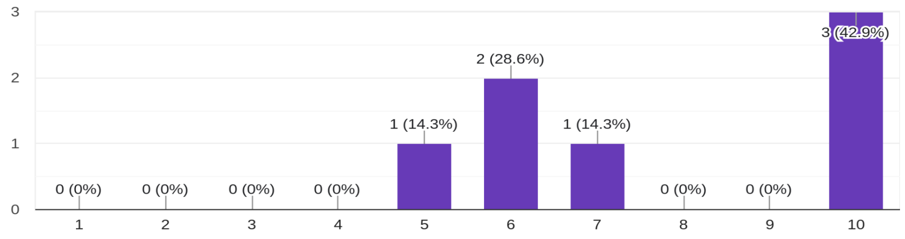
Original implementation: WASD and arrow keys as controls

7 responses



Personal implementation: Only WASD keys as controls

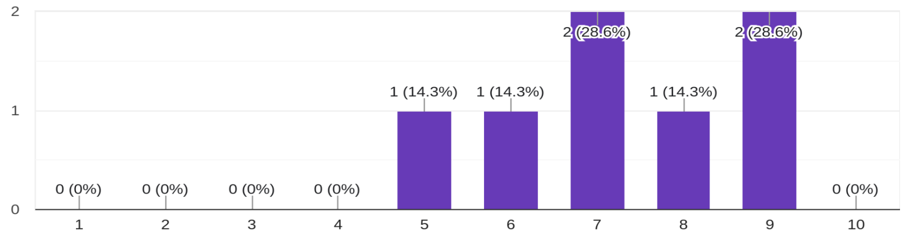
7 responses



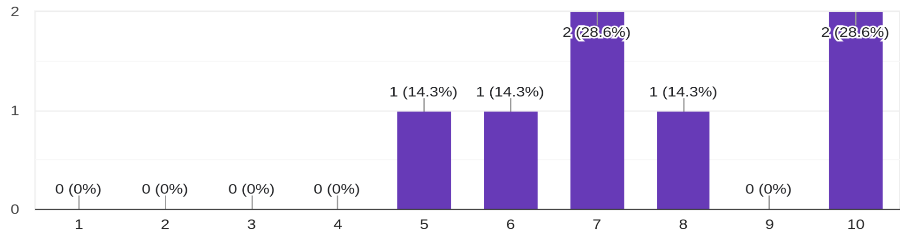
- Design choice 5: Strict movement vs Collision based movement

Original implementation: Strict movement scheme

7 responses

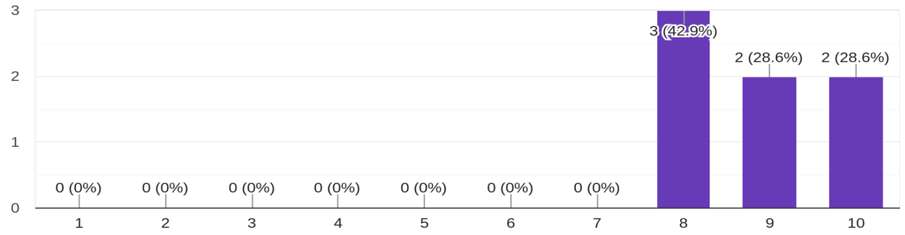


Personal implementation: Collision based movement scheme
7 responses

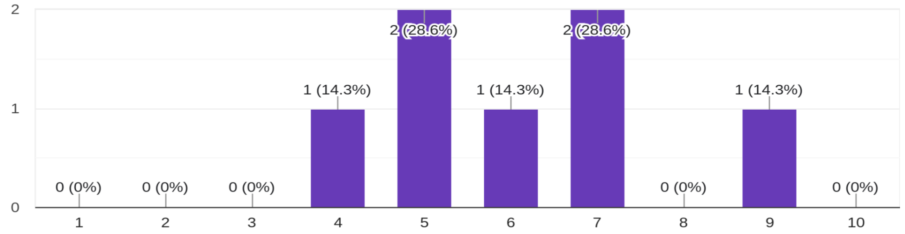


- Design choice 6: Raising difficulty vs Constant difficulty

Original implementation: Difficulty raising with each level
7 responses



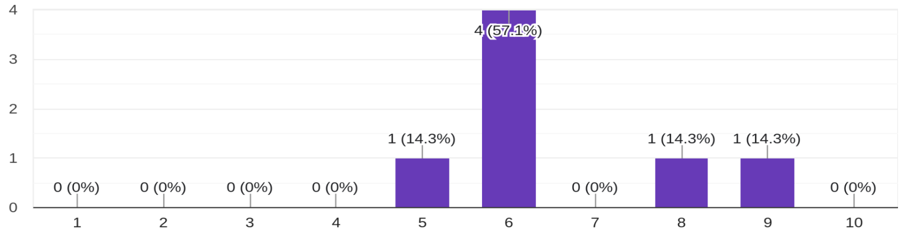
Personal implementation: Constant difficulty throughout the game
7 responses



- Design choice 7: Single usage buff vs Multiple usage buff

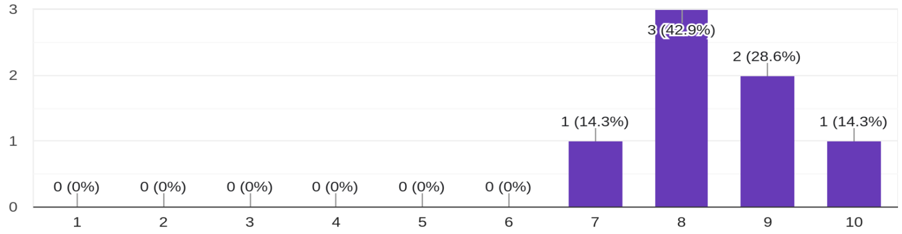
Original implementation: Power buffs allow Pacman to eat only once

7 responses



Personal implementation: Power buffs allow Pacman to eat multiple times

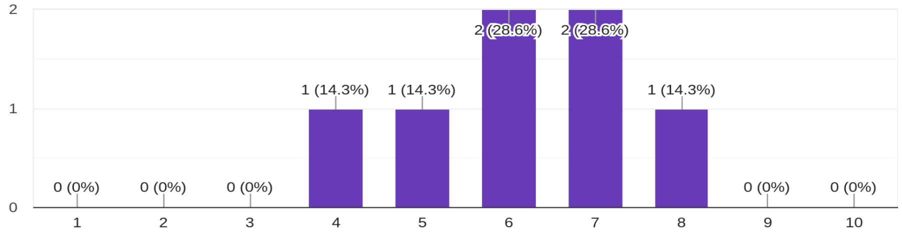
7 responses



- Design choice 8: Single type of buff vs Multiple types of buffs

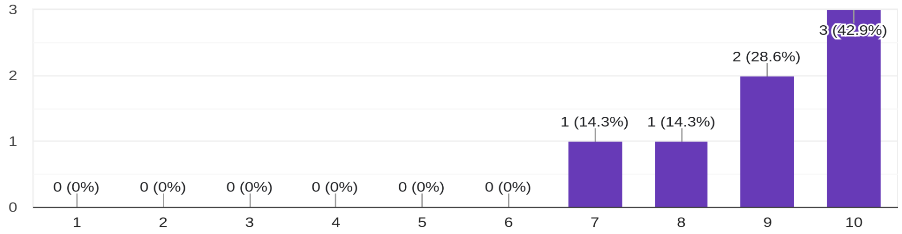
Original implementation: Only one type of buffs

7 responses

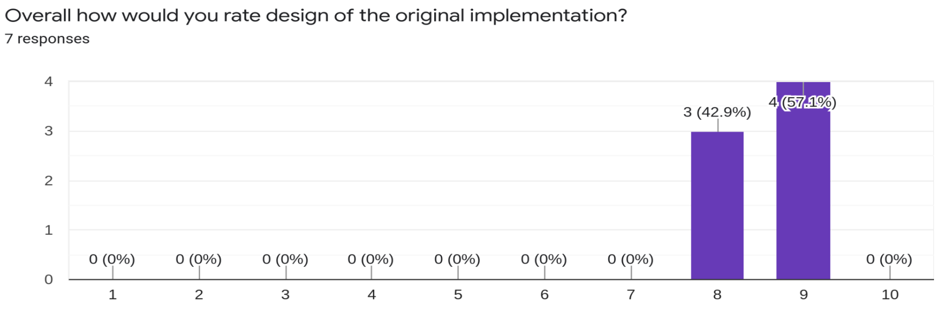


Personal implementation: Multiple types of buffs

7 responses

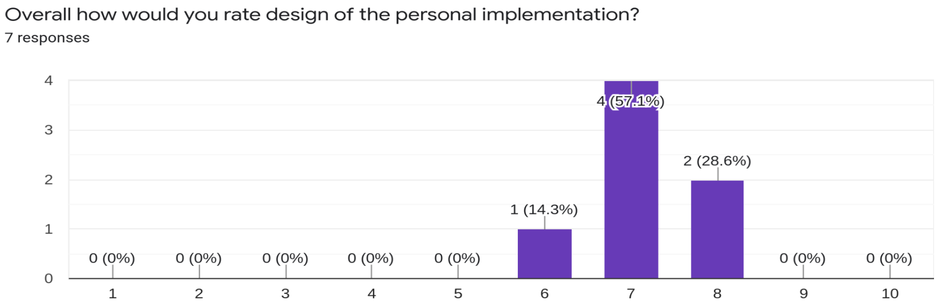


- Design choice 9: Overall design & appeal



Overall how would you rate design of the original implementation?

7 responses



Overall how would you rate design of the personal implementation?

7 responses

2) Discussion

Going through all the key differences and conducting a survey, we can come to the following conclusions.

Design choice 1. Seeing the whole map is slightly better according to the survey than having a camera focused on the player, it may be due to the fact that people can see the location of each monster, whilst with the fixed camera players cannot do so.

Design choice 2. Looking at the chart we can see that the system of levels is more preferable by players, though the graph doesn't show any sharp difference between those two. Probably this slight difference is due to the fact that the leveling system allows players to adapt gradually, whilst survival mode with constant difficulty puts players in a big stress right after the start of the game.

Design choice 3. Sounds definitely affect how players enjoy the game, the chart looks very skewed for the original implementation. Personally, I also feel more satisfaction when sounds are on, and it immerses players more into the game. Considering how significant this design choice is, it is always a good decision for game developers to hire some sound designers for their games. Unfortunately, I couldn't design proper sounds for my implementation, thus didn't include any sounds except for the main menu.

Design choice 4. Controls. Controls are surely important, and the original game provides both options, however we have very few data points to conclude anything about this particular design choice. Possibly, players like it more with a usual WASD style movement, and they're confused when two options are provided for them.

Design choice 5. Pacman movement. The original game presents the strict "rail" movement in which pacman moves like on rails without the ability to turn whenever he wants. Although there are very few data points collected, we see that both versions show similar performance, meaning that this design choice is not significantly affecting the game experience compared to the original, probably, both implementations have pros and cons in their implementation of movement.

Design choice 6. Next aspect is about the difficulty of the game. The chart suggests that people like constant difficulty less than rising difficulty. That is most probably due to the reason I mentioned above, in design choice 2. Monster in my implementation definitely feels more aggressive than in the original game, at least at the beginning of it. This gives a lot of insight on how to pace the game, and how to properly introduce players to the game. Aggressiveness immediately at the start is not always a good option.

Design choice 7. This design choice is related to buffs. The data suggests that allowing users to eat monsters multiple times is a better design choice. However, there are multiple factors that affect the choice. As I stated earlier, monsters in my game are more aggressive thus allowing players to eat monsters multiple times becomes a remedy from the monsters attacks, and gives those precious moments of peace while running through the map. In the original game though, as the difficulty rises, players also want to have a more peaceful time, but as monsters recover after the player eats them, they immediately chase pacman putting the player into stress. In my opinion there is a trade-off between how stressful and easy the game is, difficulty makes the game interesting but if you make it too difficult or too easy, no one is going to like your game. Rewarding players with some power buff is a good design choice overall.

Design choice 8. It is related to the type of buffs that the game offers. We can clearly see that players prefer creative approaches with multiple different types of buffs. In my game, slowing and power buffs are implemented whilst in the original game you can only observe the latter buff. There is a good insight here, when designing a game, it is more likely that people will enjoy your game if the game is contentful and offers creative approaches to accomplish a goal.

We analyzed how users experienced both versions and conducted a survey. Overall, players enjoyed the original implementation more than my implementation. This research shed a lot of light on the pitfalls of my implementation, and which design choices are better than the others.

We also have to admit that the sample size was extremely low to derive some convincing conclusions, the limitation was the fact that my implementation at the moment of writing this report was only playable on my personal laptop.
It would be very inspiring to deploy my game and collect more answers from the players who played both games in the future.

3) Conclusion

Implementing a game from scratch was a wonderful experience, sometimes horrible at times of debugging. In this zero-to-one project I learnt a lot, how the game development process proceeds and how each design element influences the gaming experience. Looking at the results, I can clearly see that there is a lot of room for improvement and maybe I am going to change some of my design choices in the future for a better enjoyment of players.

Certainly, some of the aspects of my game would be much better if I didn't do it alone and collaborated with other people, for example artists to draw Pacman sprites, beat/music makers to create an appropriate sound support, or some experienced game designers to foresee possible issues that may arise during the play.

※ References

1. Beattie, A. (2022, February 8). How the video game industry is changing. Investopedia. Retrieved June 9, 2022, from https://www.investopedia.com/articles/investing/053115/how-video-game-industry-changing.asp#:~:text=In%202020%2C%20the%20gaming%20industry,than%20%24260%20billion%20in%20revenue.
2. Project, T. F. V. G. (n.d.). Free Pacman. FreePacman.Org. Retrieved June 9, 2022, from https://freepacman.org/
3. Birch, C. (n.d.). GameInternals - Understanding Pac-Man Ghost Behavior. Game Internals. Retrieved June 9, 2022, from https://gameinternals.com/understanding-pac-man-ghost-behavior
4. Abdyldayev, K. (2022, June 10). Game Design influence on gaming experience. Google Docs. https://docs.google.com/forms/d/e/1FAIpQLSf3rclEkYQU_nN8431Bclut0jOV8Y-L3aNcUHEzeF20KSqYnA/viewform?usp=send_form