

Study on Enemy Adaptations

Elena Herzinger
Computer Science Masters Program
Full Sail University
Winter Park FL, USA
egherzinger@student.fullsail.edu

Abstract—This article will describe the work and results from the game and study conducted over the last two months. The data collection methods, data visualization tools, and application development will be shown and described.

Keywords—Dynamic Difficulty Adjustment, HCI, Data Visualization, Data Collection, Game Development

I. INTRODUCTION

This paper will look at game development, data collection methods, and data visualization tools used for a study looking at enemy dynamic difficulty adjustment. Dynamic difficulty adjustment, also known as DDA, is a way to adjust the game's challenges to the player's skill level [1]. This study looked at performance DDA, which are metrics derived from a weighted average of an in-game variable [1]. DDA can make a better game than a game with fixed difficulty [2]. Having DDA in a game can keep players engaged no matter what the players' skill level is [2]. In the case of this study, it was the player's health from the previous level that was adjusting the difficulty. Depending on what that number is, the number of enemies changes.

To run this study, I created a 2D platformer game in Unity. This game can change the number of enemies based on how the player did in the previous level. There is also a way to turn adaptability on and off for easy testing. The results of the tests were then used to create data visualizations for analysis.

II. DATA COLLECTION METHODS

To collect data from participants, I created a way to collect certain variables and write them to a comma-separated value (CSV) file. The variables I have collected are enemies killed, end health, and end coin amounts. These variables are counted and kept track of throughout the game and when the player ends the game, by either reaching the game over screen or you win screen. Once the CSV file was created, I was able to use the file to create some data visualizations.

I collected the data by playing the game myself and having others play my game a couple times. I mostly had experienced or occasional players participated, but they still helped me understand how my game was doing. The data I ended up using to create the visualizations is displayed in the table below.

Enemies Killed	End Health	End Coins
12	6	15
8	3	15
10	0	13
12	1	14
17	6	18
18	4	19

Table 1. Data Used for Visualizations

III. DATA VISUALIZATION TOOLS

I created my visualizations in Jupyter Notebook with the Matplotlib library. Jupyter Notebook is a program commonly used to create data visualizations and work with Python libraries [3]. It's easy to use and good for debugging when visualizations fail [3]. Matplotlib is a 2D visualization library [4]. It can create a wide range of visualizations and is completely customizable [4].

With those tools I made a couple different variables. The first one I made was three scatter plots. I had a scatter plot for each combination of the variables being assessed. The graphs didn't show too much, but it's still a good way to see the comparison between the variables.

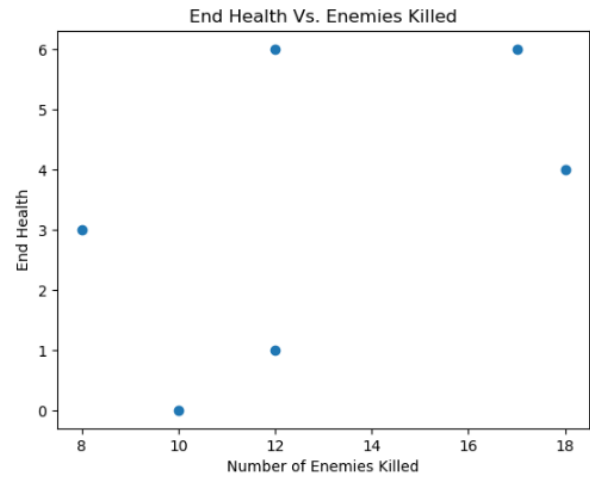


Fig 1. End Health vs. Enemies Killed

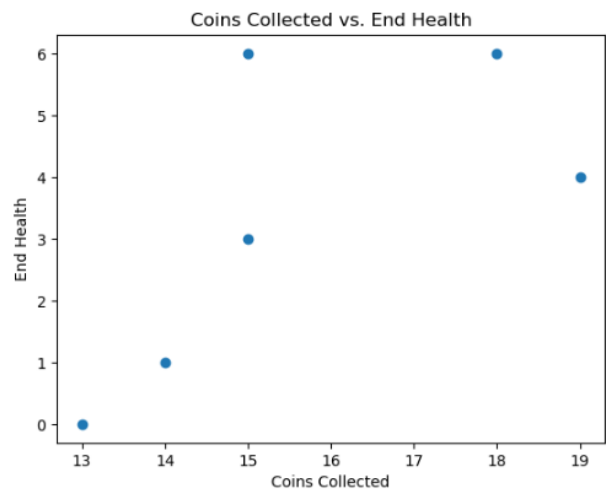


Fig 2. Coins Collected vs. End Health

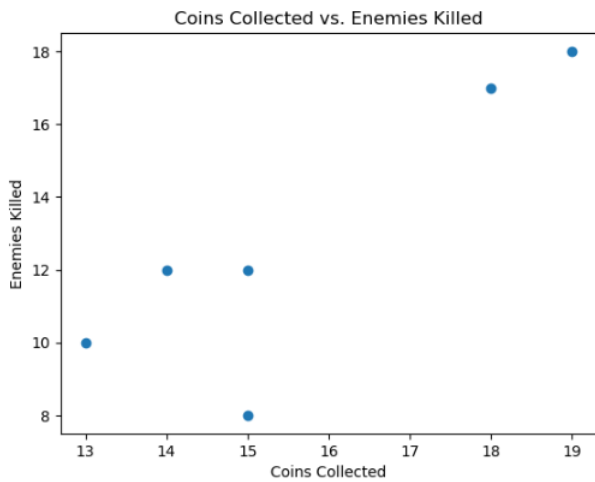


Fig 3. Coins Collected vs. Enemies Killed

The other visualizations I created were bar charts showing the frequency of each variable. These bar charts are great for understanding how players perform in each area. It's great to see if there is a trend in the variable.

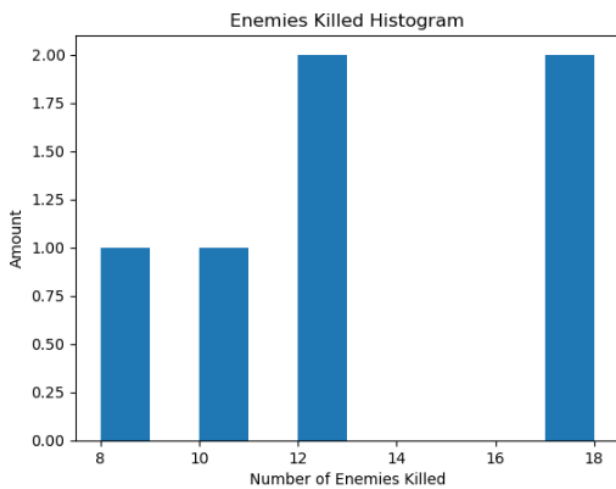


Fig 4. Enemies Killed Bar Chart

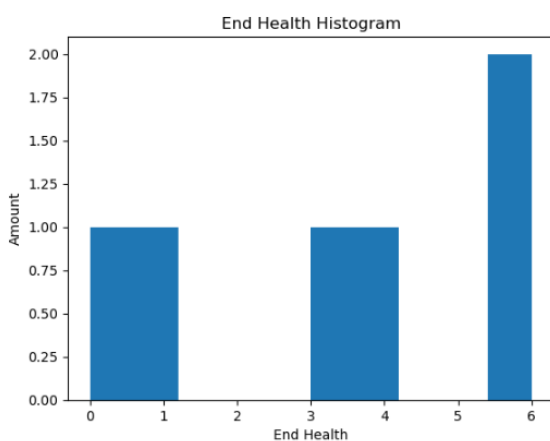


Fig 5. End Health Bar Chart

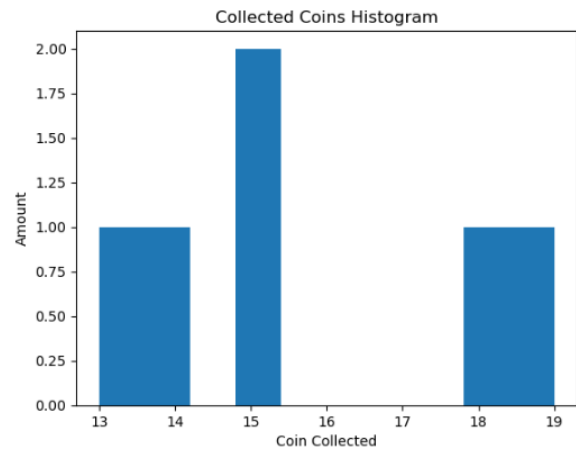


Fig 6. Collected Coins Bar Chart

The final visualization I did with this data was get the averages of all the variables looked at. It's another way of seeing how the players are doing. Average can help see if something is too hard or too easy. The averages for these results were 12.83 for the enemies killed, 3.33 for end health, and 15.66 for coins collected. These seem like reasonable results to me, and I don't think my game is too hard to too easy.

IV. APPLICATION DEVELOPMENT

A. Previously Done Work

Before this month I created most of my game and had already run the initial study. The game I made was a 2D platformer game like Super Mario Bros. It is aimed at beginners, with enemy adaptations supposed to help new players adjust to the process.

The enemy levels are controlled with an algorithm. The algorithm takes the health left after the previous level and then decides how many enemies to add. If there are 4 or more health points left, then 2 more enemies are added. If there are 2 or 3 health, then one randomly selected enemy will spawn. Finally, if there is only 1 health left, then no extra enemies are spawned and the base two remain.

From the study I did with this game, I determined that people enjoy enemy adaptations. I had them play the game with and without the adaptations, and all the participants enjoyed the version with it more.

B. Week 1

This first week I added things to my game based off some suggestions I got from my participants in my study. The first thing I did was add coins to the game that allow you to regain some health. Some of my participants wanted more things to do in the game, and coins were an easy thing to add.

I also added some music in the background. This was another suggestion from a participant. It was another easy addition that makes the game more enjoyable.

C. Week 2

In week two I started working on data collection and visualization systems. To my game I added a CSV writer that takes data collected by different counters in the game. This version received data on the number of enemies killed and health at the end of the game. I attached the writer to the game

over screen and the you win screen so the data writes when the player is done.

Once the CSV was written, I moved it into a Jupyter notebook to create the visualizations with Matplotlib. In this I made my first versions of visualizations, which look very similar to the ones shown previously.

D. Week 3

At the end of week two I conducted a survey on enemy adaptations, and from that survey I found a couple things to change. The first thing I did was add another level at the end of the game. This allowed me to adjust the enemies in the second level to three to help new players understand that enemy levels change.



Fig 7. Level 6 Layout

The next thing I did was add the coin data to the CSV writer. This was simple to do and got me some more data to work with.

The final thing was to set up a way to turn off the dynamic difficulty. I did this by adding a checkbox to the main menu that turns it on or off. I did this because my survey showed me that some people had mixed feelings on dynamic difficulty, so I wanted a way to let them turn it off if they didn't want it.

E. Week 4

In this final week of production, I mostly worked on finishing up buttons and other minor details. The first thing I

did was work on finishing the buttons on my main menu. To do that I removed the continue button and added an exit button. I also added a credits screen and attached it to the credits button. Now I have three working buttons on the main menu compared to the one I had previously.

I also finished up the buttons on the win and game-over screen. Now you can get to the main menu or exit from these screens, making the game easier to navigate.

V. CONCLUSION

As of this week I think that my game and visualization maker is completely done. This game seems to be a good way to test enemy adaptations for platformers. It should also help new players understand how to play platformers. This study and process were interesting to me, and I learned a lot about game development and data visualization processes.

REFERENCES

- [1] N. Fisher and A. K. Kulshreshth, "Exploring Dynamic Difficulty Adjustment Methods for Video Games," *Virtual worlds*, vol. 3, no. 2, pp. 230–255, Jun. 2024, doi: <https://doi.org/10.3390/virtualworlds3020012>.
- [2] N. Fisher and A. K. Kulshreshth, "Dynamic Difficulty Adjustment in Games: Concepts, Techniques, and Applications," *Intechopen.com*, Aug. 2025, doi: <https://doi.org/10.5772/intechopen.1011703>.
- [3] Y. Wang, W. Meijer, J. A. H. López, U. Nilsson, and D. Varró, "Why Do Machine Learning Notebooks Crash? An Empirical Study on Public Python Jupyter Notebooks," *IEEE Transactions on Software Engineering*, vol. 51, no. 7, pp. 2181–2196, Jul. 2025, doi: <https://doi.org/10.1109/tse.2025.3574500>.
- [4] A. Lavanya, S. Sindhuja, L. Gaurav, and W. Ali, "A Comprehensive Review of Data Visualization Tools: Features, Strengths, and Weaknesses," *International Journal of Computer Engineering in Research Trends*, vol. 10, no. 1, pp. 10–20, 2023, doi: <https://doi.org/10.22362/ijcert/2023/v10/i01/v10i0102>.