Super MeatBoy AI Performance

Research Written and Performed by
Doriana Othman

Problem Description

The rapid growth and development of the video game industry have led to an increasing demand for new challenges and immersive experiences among players. Gamers are constantly seeking ways to enhance their excitement and enjoyment while engaging with video games. One significant avenue for improving gaming experiences is the integration of Artificial Intelligence (AI) technology. AI can be utilized to enhance various aspects of games, such as Non-Playable Characters (NPCs), pathfinding, decision making, procedural modeling, and player experience modeling. By leveraging AI, developers can advance player experiences and create more dynamic and engaging gameplay.

In a previous project, my focus was on addressing the problem of developing a suitable AI for the popular video game Super Meat Boy. The primary objective of the AI was to navigate through the game's levels, reaching the target as quickly as possible while minimizing the number of deaths. Building upon this previous project, the current report aims to expand the research by timing the completion of the first three Light World Levels and the first three Dark World Levels by players. Additionally, the number of deaths incurred during these level completions will be tracked. The collected data from player performance will serve as a basis for comparison against an AI system that underwent the same amount of training time as the players' completion times.

By conducting this comparative analysis, the goal is to assess the performance of the AI system in relation to human players. This evaluation will shed light on the effectiveness of the AI model in replicating human-like gameplay and achieving optimal completion times with minimal deaths. The findings from this study will contribute to a deeper understanding of AI's potential in improving gaming experiences and provide insights into the viability of AI as a reliable and competitive component in video game design.

The subsequent sections of this report will detail the methodology employed to collect data from both human players and the AI system. The collected data will then be analyzed and compared to draw meaningful conclusions regarding the performance of the AI model. The report will conclude with a discussion of the implications of these findings and potential areas for further research and development in AI-driven gaming experiences.

Background Information:

Video Game Description- Super Meat Boy:

Super Meat Boy is a tough as nails platformer where you play as an animated cube of meat who's trying to save his girlfriend (who happens to be made of bandages) from an evil fetus in a jar wearing a tux. Our meaty hero will leap from walls, over seas of buzz saws, through crumbling caves and pools of old needles. Sacrificing his own well being to save his damsel in distress. Super Meat Boy brings the old school difficulty of classic NES titles like Mega Man 2, Ghost and Goblins and Super Mario Bros. 2 (The Japanese one) and streamlines them down to the essential no BS straight forward twitch reflex platforming.Ramping up in difficulty from hard to soul crushing SMB will drag Meat boy though haunted hospitals, salt factories and even hell itself. And if 300+ single player levels weren't enough SMB also throws in epic boss fights, a level editor and tons of unlockable secrets, warp zones and hidden characters.

Light World Levels:

In Super Meat Boy, “Light World” levels are the standard levels included in the base game. These levels are considered to be the normal difficulty in the game.

Dark World Levels:

In Super Meat Boy, “Dark World” levels are the harder levels included in the base game, which can normally only be accessed after getting an A+ rating in its “Light World” level version; this is achieved by reaching or beating that “Light World” level’s time limit. Each “Dark World” level will have a similar looking environment to its “Light World” counterpart. Increases in difficulty usually include more obstacles that will cause harm to Meat Boy or hinder pathways towards his goal.

Methods Used:

Player Data Collection:

20 players volunteered to play the official game *Super Meatboy* on my Laptop; each player used a controller when playing the game, as recommended by the developers themselves due to *Super Meatboy* being much more difficult to play using a keyboard and mouse. Each player was made aware of the controls before playing and also instructed on where to locate the controls if desired. Players were instructed to play the first three levels of the first Dark and first Light World. For each playthrough, I recorded the total number of deaths and total playtime in seconds for each level; *Super Meatboy* records the winning playtime i for each level, which was also recorded in the data. Once a player completed a level, they immediately moved to the next one; players did not replay any levels already completed.

Pygame:

To make the implementation of AI easier, Pygame was used to recreate the game Super Meat Boy. Key features of the character movement (Meat Boy) were kept; this includes walking, running, and wall jumping. Character animations were kept to a minimum and were only included to differentiate between movements during play. All graphics used are identical to Super Meat Boy; this was done to create familiarity between the pygame version and the actual version.

Level Selection:

For this report, The first three Light World and Dark World level designs were used from World 1 of Super Meat Boy. These levels were chosen for their simplicity of recreation, ease of introduction to game mechanics, and time scope of this project.

Level Creation:

To make implementation easier, each level was implemented as a list of strings with characters that represent different behaviors. Every level had 46 rows of 70 characters each; P represented Player, T represented Target, X represented Wall, D represented Death. Where these characters are located match where its represented behavior is on the screen; this style of implementation helps pygame determine how Meat Boy (the player) should react in the game. If the programmer or user wants to see this visually represented, the “render_blocks” option can be set to True in “settings.py”. All of the behavior related to how the player reacts in the game can be found in “level.py”.

AI Methods:

To create an AI, machine learning was used with a combination of Genetic Algorithms and Neural Networks during training. The general process of training functions as follows: a predetermined number of Meat Boys are created per generation. In each generation, each Meat

Boy has its own Neural Network that determines the pathway it takes and when Meatboy should jump. For each generation, the fitness of each Meat Boy is determined by the length of its pathway and if that Meat Boy is alive; the average and best fitness of each generation is then reported. The next generation is then created based on the predetermined percentage of "good" and "bad" Meat Boys to keep, with the remainder being newly generated Meat Boys. Training happens for as long as the user desires; the weights produced can be saved and/or loaded if wanted through "settings.py".

Pathway Algorithm:
To determine the best pathway distance for each Meat Boy, BFS was used; this was specifically used as a fitness measure to rank the Meat Boys in each generation, however, if the actual path taken from the Neural Network was better it will use that as its fitness value. Although A* would be the better algorithmic choice, due to the amount of time that would be needed to generate weights for each level and the focus of the project, it was decided that BFS would be the best approach for this project iteration.

Neural Networks:
The Neural Networks in this implementation represent the "brain" of a Meat Boy; this determines the path that a Meat Boy takes and also determines if it should jump. The weight values of each Meat Boy are only changed based on the predetermined percentage chance set by the programmer. The Neural Networks initial input layer, hidden layer, and outer layer values are also initialized by the programmer.

Genetic Algorithms:
Genetic Algorithms is the method used to improve each generation of Meat Boys. This is done by using an initial cut off percentage to determine the number of "good" and "bad" Meat Boys of each generation based on their ranking; the number of "good" and "bad" Meat Boys kept for the new generation is decided by initial predetermined percentages set by the programmer. The remaining Meatboys needed to fill the new generation are then determined by picking random "good" Meat Boy parents and creating new children based on a combination of their parent's weights from their Neural Networks; the percentage taken from each parent is based on a predetermined percentage value. Another predetermined percentage also decides if these newly created children will have their weights modified. For each new generation, all of the "bad" Meatboys kept will have their weights modified.

<u>**Results:**</u>

For this report, I used the player data to determine the training time in seconds for each AI created. Specifically for each level, the players selected were based on the best time taken to win and the worst time taken to win; the average of all players' times taken to win were computed and used for training as well. The duration of training itself was based on the selected players total play time, while the duration the Meat Boy(s) were permitted to stay alive per generation was determined by the same selected player's time taken to win. The overall goal is that Meat Boy can safely reach its target, bandage girl, without failure. Although many different variations of settings can be used, I have decided to use the settings below based on experimentation from the previous project iteration. For each level tested, both Dark World and Light World, the Meat Boy with the best weights was used for its Neural Network. Since all of the obstacles in these levels were static, it was unnecessary to conduct multiple tests as the same results would be achieved. Because testing will produce the same result each time due to the static gameplay and weight data, the number of deaths were calculated from training.

Observations:

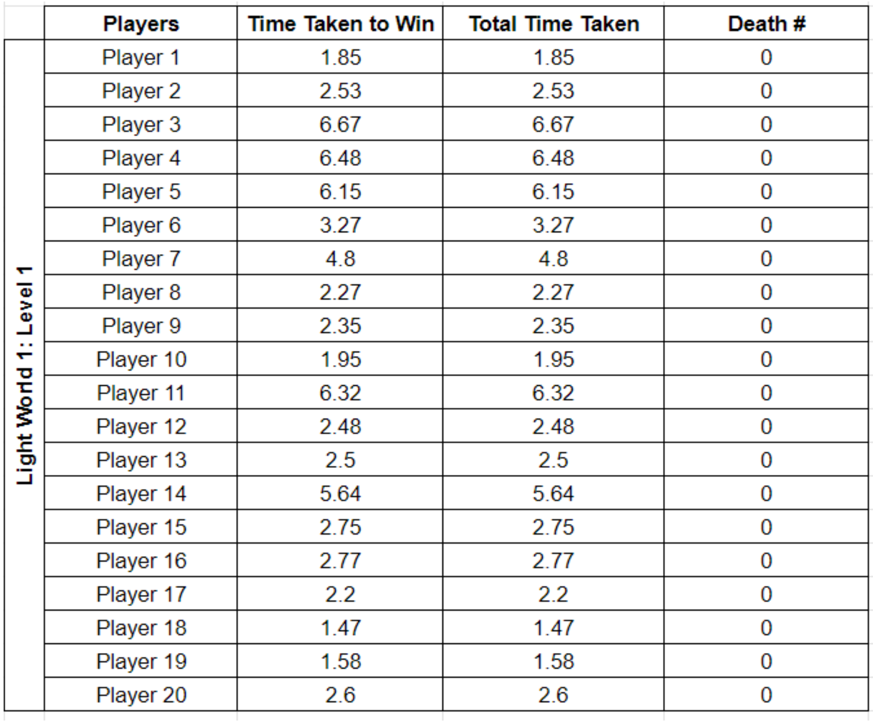
As expected, the AI’s across the board were mostly worse than the human players. In the previous report, it was mentioned that the AI had difficulty learning combinations of moves, such as wall jumping, so it was only natural that with the limited amount of time to train based on player times, it was going to perform poorly against human players.None of the AI’s trained with a Meatboy generation size of 1 were successful or ever close to being so; the machine learning methods used in this report normally require lots of feedback from training, so this was to be expected from a limited generation size. Also due to the way the AI is trained, it is willing to take a lot more risks than human players, causing the number of deaths to be drastically higher than humans. On the occasions the AI was successful, it usually would outperform the human player; in the levels the AI was successful, there were no risks of death and limited combinations of controls required.

Settings:

```
get_nn_inputs = True
max_alive_time = 7
render_blocks = True
save_img = True
tile_size_x = 10
tile_size_y = 10
screen_width = 700
screen_height = 468
scale_factor = 0.35 if lvl in [1, 2, 3, 6, 7, 8] else 0.2
jump_speed = -9 if lvl in [5, 7] else -10
GEN_SIZE = 100
generation_size = GEN_SIZE if train else 1
IN_INPUT, N_HIDDEN, N_OUTPUT = 3, 10, 2
MUTATION_WEIGHT_MODIFY_CHANCE = 0.1
MUTATION_ARRAY_MIX_PERC = 0.5
MUTATION_CUT_OFF = 0.5
MUTATION_BAD_TO_KEEP = 0.1
MUTATION_MODIFY_CHANCE_LIMIT = 0.1
```

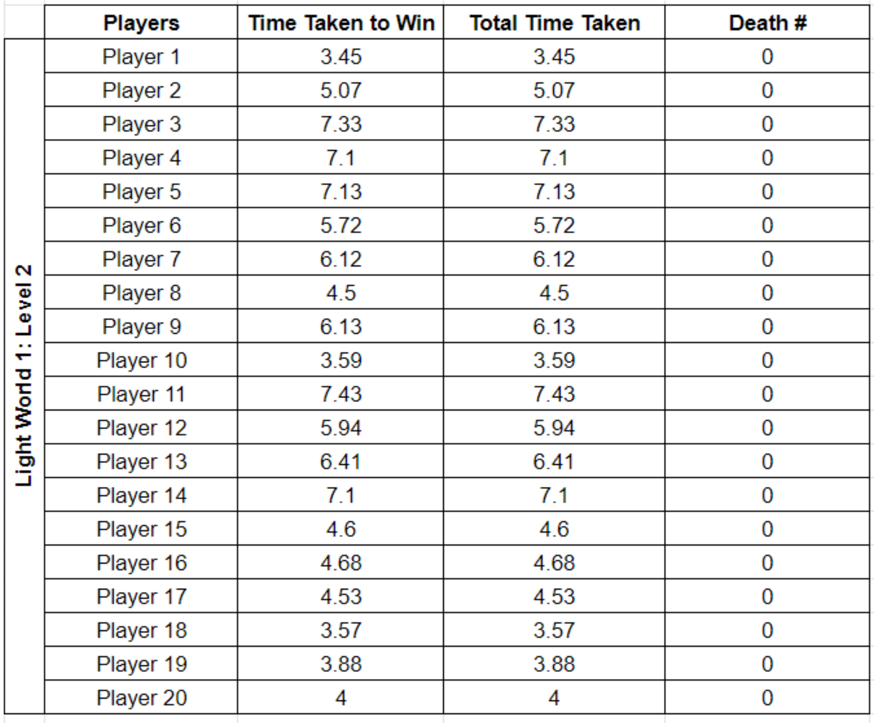
Player Data:

Light World 1: Level 1



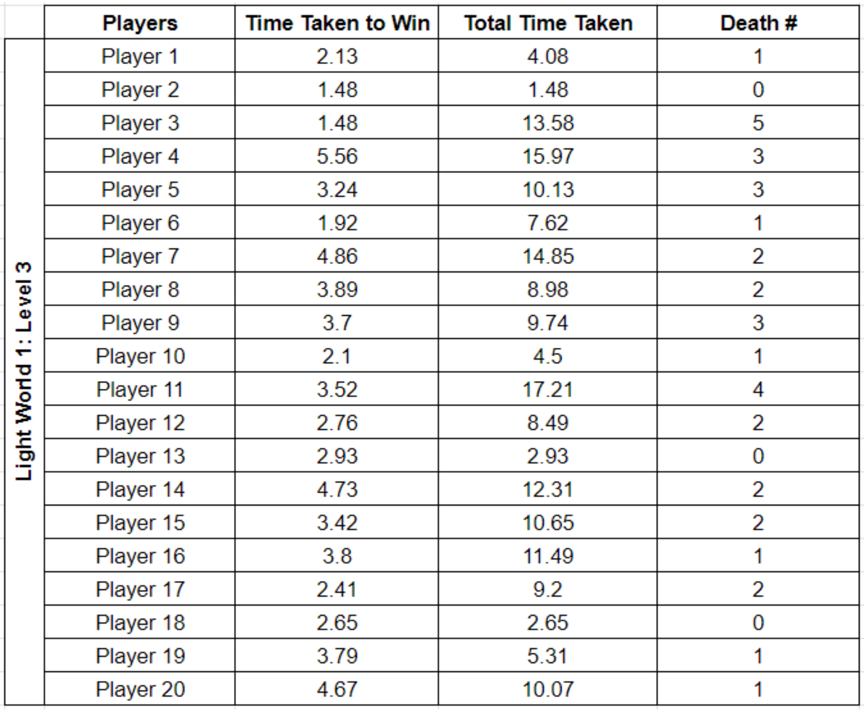
	Players	Time Taken to Win	Total Time Taken	Death #
Light World 1: Level 1	Player 1	1.85	1.85	0
	Player 2	2.53	2.53	0
	Player 3	6.67	6.67	0
	Player 4	6.48	6.48	0
	Player 5	6.15	6.15	0
	Player 6	3.27	3.27	0
	Player 7	4.8	4.8	0
	Player 8	2.27	2.27	0
	Player 9	2.35	2.35	0
	Player 10	1.95	1.95	0
	Player 11	6.32	6.32	0
	Player 12	2.48	2.48	0
	Player 13	2.5	2.5	0
	Player 14	5.64	5.64	0
	Player 15	2.75	2.75	0
	Player 16	2.77	2.77	0
	Player 17	2.2	2.2	0
	Player 18	1.47	1.47	0
	Player 19	1.58	1.58	0
	Player 20	2.6	2.6	0

Light World 1: Level 2



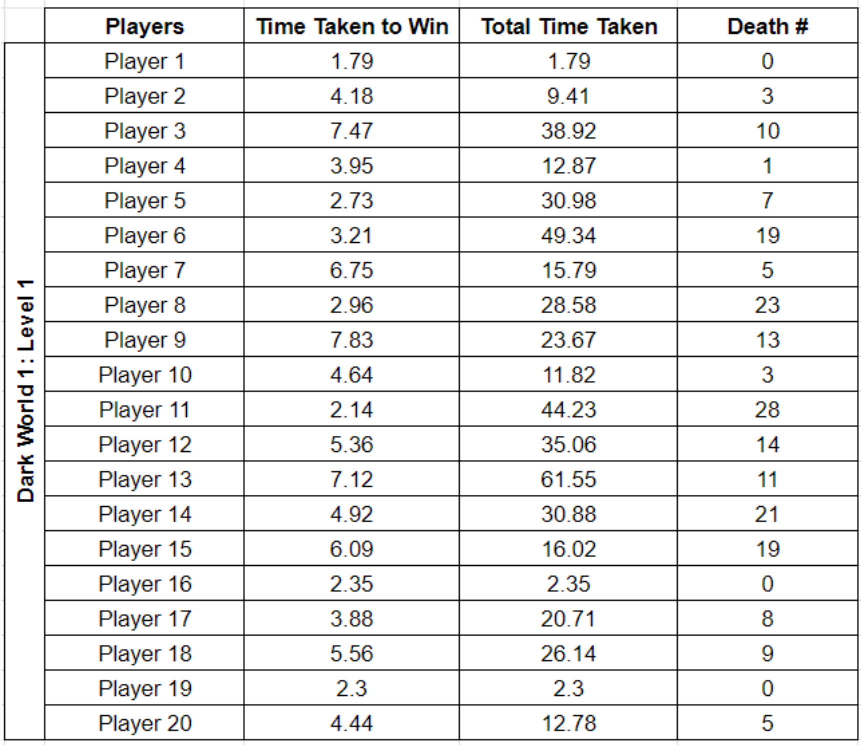
	Players	Time Taken to Win	Total Time Taken	Death #
Light World 1: Level 2	Player 1	3.45	3.45	0
	Player 2	5.07	5.07	0
	Player 3	7.33	7.33	0
	Player 4	7.1	7.1	0
	Player 5	7.13	7.13	0
	Player 6	5.72	5.72	0
	Player 7	6.12	6.12	0
	Player 8	4.5	4.5	0
	Player 9	6.13	6.13	0
	Player 10	3.59	3.59	0
	Player 11	7.43	7.43	0
	Player 12	5.94	5.94	0
	Player 13	6.41	6.41	0
	Player 14	7.1	7.1	0
	Player 15	4.6	4.6	0
	Player 16	4.68	4.68	0
	Player 17	4.53	4.53	0
	Player 18	3.57	3.57	0
	Player 19	3.88	3.88	0
	Player 20	4	4	0

Light World 1: Level 3



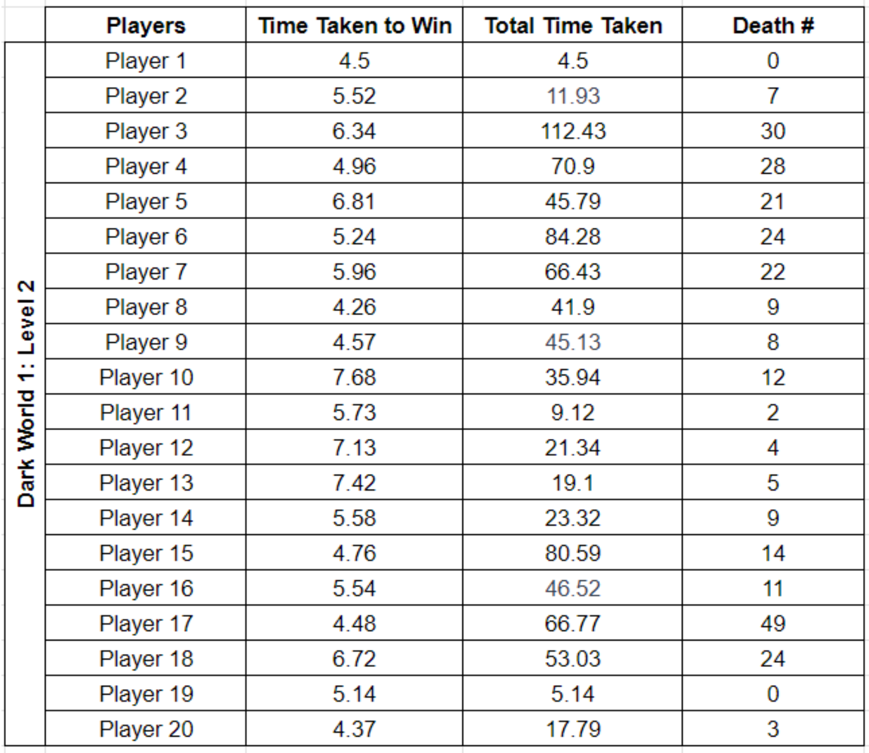
	Players	Time Taken to Win	Total Time Taken	Death #
Light World 1: Level 3	Player 1	2.13	4.08	1
	Player 2	1.48	1.48	0
	Player 3	1.48	13.58	5
	Player 4	5.56	15.97	3
	Player 5	3.24	10.13	3
	Player 6	1.92	7.62	1
	Player 7	4.86	14.85	2
	Player 8	3.89	8.98	2
	Player 9	3.7	9.74	3
	Player 10	2.1	4.5	1
	Player 11	3.52	17.21	4
	Player 12	2.76	8.49	2
	Player 13	2.93	2.93	0
	Player 14	4.73	12.31	2
	Player 15	3.42	10.65	2
	Player 16	3.8	11.49	1
	Player 17	2.41	9.2	2
	Player 18	2.65	2.65	0
	Player 19	3.79	5.31	1
	Player 20	4.67	10.07	1

Dark World 1: Level 1



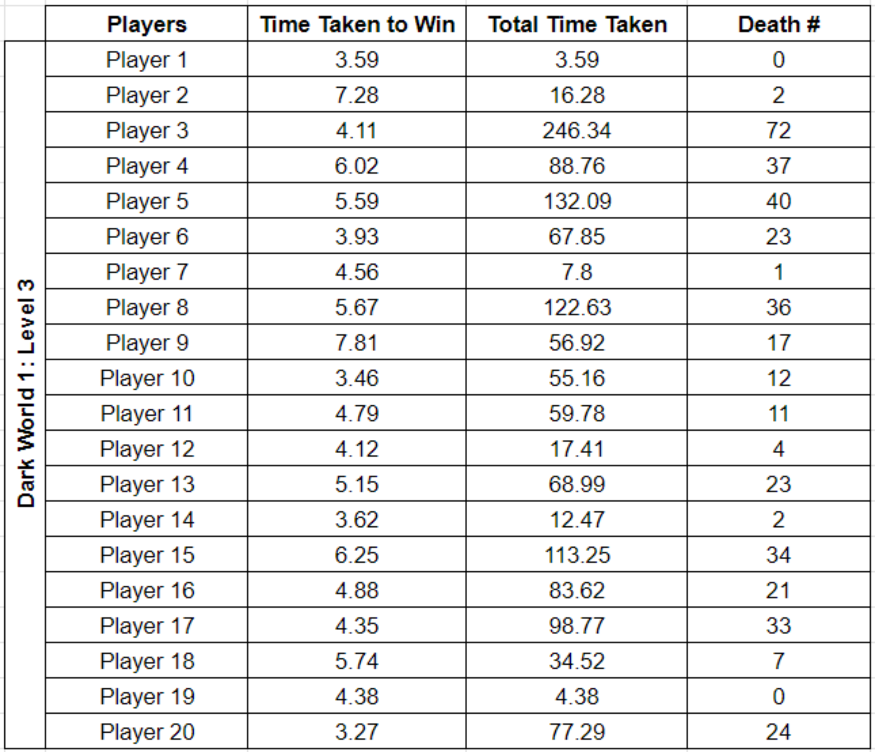
	Players	Time Taken to Win	Total Time Taken	Death #
Dark World 1: Level 1	Player 1	1.79	1.79	0
	Player 2	4.18	9.41	3
	Player 3	7.47	38.92	10
	Player 4	3.95	12.87	1
	Player 5	2.73	30.98	7
	Player 6	3.21	49.34	19
	Player 7	6.75	15.79	5
	Player 8	2.96	28.58	23
	Player 9	7.83	23.67	13
	Player 10	4.64	11.82	3
	Player 11	2.14	44.23	28
	Player 12	5.36	35.06	14
	Player 13	7.12	61.55	11
	Player 14	4.92	30.88	21
	Player 15	6.09	16.02	19
	Player 16	2.35	2.35	0
	Player 17	3.88	20.71	8
	Player 18	5.56	26.14	9
	Player 19	2.3	2.3	0
	Player 20	4.44	12.78	5

Dark World 1: Level 2



	Players	Time Taken to Win	Total Time Taken	Death #
Dark World 1: Level 2	Player 1	4.5	4.5	0
	Player 2	5.52	11.93	7
	Player 3	6.34	112.43	30
	Player 4	4.96	70.9	28
	Player 5	6.81	45.79	21
	Player 6	5.24	84.28	24
	Player 7	5.96	66.43	22
	Player 8	4.26	41.9	9
	Player 9	4.57	45.13	8
	Player 10	7.68	35.94	12
	Player 11	5.73	9.12	2
	Player 12	7.13	21.34	4
	Player 13	7.42	19.1	5
	Player 14	5.58	23.32	9
	Player 15	4.76	80.59	14
	Player 16	5.54	46.52	11
	Player 17	4.48	66.77	49
	Player 18	6.72	53.03	24
	Player 19	5.14	5.14	0
	Player 20	4.37	17.79	3

Dark World 1: Level 3



	Players	Time Taken to Win	Total Time Taken	Death #
Dark World 1: Level 3	Player 1	3.59	3.59	0
	Player 2	7.28	16.28	2
	Player 3	4.11	246.34	72
	Player 4	6.02	88.76	37
	Player 5	5.59	132.09	40
	Player 6	3.93	67.85	23
	Player 7	4.56	7.8	1
	Player 8	5.67	122.63	36
	Player 9	7.81	56.92	17
	Player 10	3.46	55.16	12
	Player 11	4.79	59.78	11
	Player 12	4.12	17.41	4
	Player 13	5.15	68.99	23
	Player 14	3.62	12.47	2
	Player 15	6.25	113.25	34
	Player 16	4.88	83.62	21
	Player 17	4.35	98.77	33
	Player 18	5.74	34.52	7
	Player 19	4.38	4.38	0
	Player 20	3.27	77.29	24

AI Data:

Table Background Info:

For the following tables, the training time in seconds used was based on the best, worst, and average players total time taken to complete a level. The max time a Meatboy was permitted to be alive per generation was determined by the best, worst, and average player's time taken to win in seconds. The number of deaths recorded were from the training sessions. Failure or success was determined by if the Meatboy was able to reach the target in its testing session. Each AI's successful test time was also compared to the player's time taken to win to determine which was better.

Light World 1: Level 1:

	Training Time	Time Taken to Win	Total Time Taken	Death #	Failure or Success	Better Than Player Time?
AI (Best Player Time): 1 Meatboy	1.58	N/A	N/A	0	Failure	No
AI (Best Player Time): 100 Meatboys	1.58	1.318837881	1.318837881	0	Success	Yes
AI (Worst Player Time): 1 Meatboy	6.67	N/A	N/A	0	Failure	No
AI (Worst Player Time): 100 Meatboys	6.67	1.41415	1.41415	0	Success	Yes
AI (Average Player Time): 1 Meatboy	3.4315	N/A	N/A	0	Failure	No
AI (Average Player Time): 100 Meatboys	3.4315	1.259601593	1.259601593	0	Success	Yes

Light World 1: Level 2:

	Training Time	Time Taken to Win	Total Time Taken	Death #	Failure or Success	Better Than Player Time?
AI (Best Player Time): 1 Meatboy	3.45	N/A	N/A	0	Failure	No
AI (Best Player Time): 100 Meatboys	3.45	N/A	N/A	0	Failure	No
AI (Worst Player Time): 1 Meatboy	7.43	N/A	N/A	0	Failure	No
AI (Worst Player Time): 100 Meatboys	7.43	N/A	N/A	0	Failure	No
AI (Average Player Time): 1 Meatboy	5.414	N/A	N/A	0	Failure	No
AI (Average Player Time): 100 Meatboys	5.414	N/A	N/A	0	Failure	No

Light World 1: Level 3

	Training Time	Time Taken to Win	Total Time Taken	Death #	Failure or Success	Better Than Player Time?
AI (Best Player Time): 1 Meatboy	1.48	N/A	N/A	2	Failure	No
AI (Best Player Time): 100 Meatboys	1.48	1.523944616	1.523944616	11	Success	No
AI (Worst Player Time): 1 Meatboy	15.97	N/A	N/A	0	Failure	No
AI (Worst Player Time): 100 Meatboys	15.97	1.664098501	1.664098501	0	Success	Yes
AI (Average Player Time): 1 Meatboy	3.252	N/A	N/A	6	Failure	No
AI (Average Player Time): 100 Meatboys	3.252	1.456321716	1.456321716	28	Success	Yes

Dark World 1: Level 1

	Training Time	Time Taken to Win	Total Time Taken	Death #	Failure or Success	Better Than Player Time?
AI (Best Player Time): 1 Meatboy	1.79	N/A	N/A	1	Failure	No
AI (Best Player Time): 100 Meatboys	1.79	N/A	N/A	69	Failure	No
AI (Worst Player Time): 1 Meatboy	23.67	N/A	N/A	304	Failure	No
AI (Worst Player Time): 100 Meatboys	23.67	N/A	N/A	42	Failure	No
AI (Average Player Time): 1 Meatboy	23.7595	N/A	N/A	34	Failure	No
AI (Average Player Time): 100 Meatboys	23.7595	N/A	N/A	372	Failure	No

Dark World 1: Level 2

	Training Time	Time Taken to Win	Total Time Taken	Death #	Failure or Success	Better Than Player Time?
AI (Best Player Time): 1 Meatboy	3.45	N/A	N/A	5	Failure	No
AI (Best Player Time): 100 Meatboys	3.45	N/A	N/A	201	Failure	No
AI (Worst Player Time): 1 Meatboy	35.94	N/A	N/A	65	Failure	No
AI (Worst Player Time): 100 Meatboys	35.94	N/A	N/A	637	Failure	No
AI (Average Player Time): 1 Meatboy	43.0975	N/A	N/A	78	Failure	No
AI (Average Player Time): 100 Meatboys	43.0975	N/A	N/A	855	Failure	No

Dark World 1: Level 3

	Training Time	Time Taken to Win	Total Time Taken	Death #	Failure or Success	Better Than Player Time?
AI (Best Player Time): 1 Meatboy	77.29	N/A	N/A	92	Failure	No
AI (Best Player Time): 100 Meatboys	77.29	N/A	N/A	577	Failure	No
AI (Worst Player Time): 1 Meatboy	56.92	N/A	N/A	23	Failure	No
AI (Worst Player Time): 100 Meatboys	56.92	N/A	N/A	441	Failure	No
AI (Average Player Time): 1 Meatboy	68.395	N/A	N/A	107	Failure	No
AI (Average Player Time): 100 Meatboys	68.395	N/A	N/A	662	Failure	No

Conclusion:

The human players outperformed the AIs by a significant amount across the board. Given that each AI normally relies on the trial and error process of genetic algorithms to determine the weights of each Neural Network, it makes sense that the human players were able to defeat the AIs when completing each level in a short amount of time. It was also likely that because these were the first 3 levels of both difficulties of the first world in *Super Meatboy*, it is likely that human players were able to easily navigate these levels due to previous experience in platformer games, such as *Super Mario Bros*; in contrast, these AI do not have any previous knowledge other than the provided controls. For a future iteration of this project, I would likely design one of the most challenging levels of *Super Meatboy* into pygame and have the AI and human players play that for the first time using the same methodologies; doing this, the AI and human player will start with similar experiences to the game, which should hopefully create some more interesting results.

Literature Review:

One of the earliest examples of AI in a computerized game was from a game called Nim in 1952. Nim, despite being advanced for its time, featured a small box-like device that possessed the ability to consistently defeat fairly skilled and experienced players. This early instance showcased the potential of AI in gaming and laid the foundation for future developments in the field (Eugene, 1952). Shortly thereafter, the Ferranti Mark 1 machine from the University of Manchester ran a checkers program that also demonstrated remarkable intelligence, outsmarting several players (Abby, 2022). These early ventures into AI programming for gaming were groundbreaking, representing some of the earliest instances of computer programs designed to challenge human opponents. Another significant milestone in AI gaming was IBM's Deep Blue computer, which would later go on to defeat chess grandmaster Garry Kasparov, marking a pivotal moment in the progress of AI (IBM)**.**

Following these remarkable AI achievements, the popularity of single-player game modes began to rise, largely influenced by the technological advancements in AI. Initially, AI in games was limited to enemy movement patterns stored within the game's code. However, as computational power increased, game developers were able to introduce more complex and unpredictable elements into enemy movements, enhancing the gameplay experience. One notable example of early single-player video game AI is observed in the game Space Invaders, where the difficulty level increased between levels and enemies exhibited distinct movement patterns influenced by player input (BigData, 2021). This innovation brought a new level of challenge and excitement to video game AI, captivating players and paving the way for further advancements in the field.

The integration of AI into various game genres gained momentum in subsequent years. In 1980, Pacman revolutionized the use of AI patterns in maze games, becoming the first game to showcase such sophisticated AI mechanics. Four years later, Karate Champ introduced AI patterns in the context of fighting games, enhancing the realism and strategic depth of player-vs-computer combat (BigData, 2021). The sports game genre also witnessed the incorporation of AI, with titles like Madden Football, Earl Weaver Baseball, and Tony La Russa Baseball introducing AI opponents that provided challenging gameplay experiences. These developments eventually led to the creation of formal AI tools like finite state machines, which enabled game developers to implement more complex and adaptive AI systems in role-playing and real-time strategy games. However, some early games in these genres faced challenges due to incomplete information necessary for these algorithms. Fortunately, subsequent iterations and advancements in AI techniques allowed developers to overcome these issues and create more sophisticated and dynamic AI systems (Cui, 2010).

In modern games, AI has adopted a bottom-up approach to create new AI behaviors. This approach involves gathering information from the game environment and using it to generate new insights and actions. By modeling their behavior based on player behaviors and data, newer AI developments in games can provide more personalized and responsive experiences (Randoff).

References:

Abby. (2022, December 6). *Ferranti mark 1 computer - everything you need to know* ￼￼￼￼￼￼￼￼. History. https://history-computer.com/ferranti-mark-1-computer-guide/

Bigdata. (2021, March 24). *History of AI use in video game design*. Big Data Analytics News. https://bigdataanalyticsnews.com/history-of-artificial-intelligence-in-video-games/

Cui, Xiao & Shi, Hao. (2010). A*-based Pathfinding in Modern Computer Games. 11.

Deep Blue. IBM100 - Deep Blue. (n.d.). https://www.ibm.com/ibm/history/ibm100/us/en/icons/deepblue/

Eugene F. Grant, R. L. (1952, July 26). *It*. The New Yorker. https://www.newyorker.com/magazine/1952/08/02/it

Radoff.com. Jon Radoffs Internet Wonderland Emergent Intelligence in Games Comments. (n.d.). https://web.archive.org/web/20110219183426/http://radoff.com/blog/2011/02/17/emergent-intelligence-games/