Sources Cited:

- Tech With Tim. (2019, August 10). Python Flappy Bird AI Tutorial (with NEAT) - Creating the Bird [Video]. YouTube. <https://youtube.com/watch?v=MMxFDalOHsE>

\*A step by step walkthrough of the process of programming Flappy Bird and then training AI agents to play it.

- Coding With Russ. (2023, January 14). Get Started in Pygame in 10 minutes! [Video]. YouTube. <https://youtube.com/watch?v=PYVG3Pto8k8>

\*A basic runthrough of Pygame for beginners

- Tyagi, C. (2020, October 8). *Introduction to pygame*. GeeksforGeeks. https://www.geeksforgeeks.org/introduction-to-pygame/

\*Web resource for people new to using pygame

- Nuer, J. (2022, May 20). *A comprehensive guide to deep Q-learning*. Medium. https://medium.com/@jereminuerofficial/a-comprehensive-guide-to-deep-q-learning-8aeed632f52f

\*An accessible resource to help understand Deep Q-Learning.

- Nuer, J. (2022a, January 8). *Convolutional Neural Networks explained*. Medium. https://medium.com/@jereminuerofficial/convolutional-neural-networks-explained-6e8c04a8f29b

- Nuer, J. (2021, December 2). *Q-learning - an introduction*. Medium. https://medium.com/@jereminuerofficial/q-learning-an-introduction-f1392738bcf

|  |  |
| --- | --- |
| Description of Challenge | Attempted/Implemented Solutions |
| The first significant challenge we faced was figuring out how to clone the PLE repository and clone the Flappy Bird game code. | Because there was no specific repository mentioned in the instructions, some searching was required. Once the correct repositories were found, Claude was consultued to help make sure the code to clone the repositories and initialize the game environment was correct. |
| We encountered an issue when initializing that suggests some of the image asset files cannot be accessed for some reason. | With the assistance of AI we were able to approach the accessing of assets differently, by implementing a method in which the assets are directly downloaded from the source. Another option we considered was running the game on gymnasium instead of pygame to see if the assets downloaded without error. But we decided to stay with pygame for simplicity. |
| Because we are not fluent in python, we relied on AI to do a majority of the coding, and this led to a lot of minor c oding errors. | We patiently debugged by feeding the errors into Claude and implementing solutions repeatedly until we got the code running. |
| AI not learning effectively from basic reward system. | With the help of AI we created a more detailed reward system that gave out additional positive rewards for actions such as staying alive, good positioning near pipes, successfully passing pipes and surviving for longer periods of time. |
| Now that the code is running, we are monitoring the performance and noticing signs of poor performance. Noticeably, the training is taking a long time, the performance patterns are inconsistent, the epsilon is decaying too rapidly, and the agent is not learning well from mistakes. | We attempted many different small tweaks to the code, re-ran the code, observed the training results for some time (occasionally several hours of monitoring), had Claude analyze the results and implemented the suggested changes until we arrived at a version that showed the most improvement. |
| Left our most successful model to train overnight because the training time is so extensive and woke to find something had interrupted the runtime and the results were lost. | This was a pretty significant setback because of how long the model is taking to train. And due to whatever error happened, we lost the data that had been produced so we couldn’t even really assess how successful that iteration was. Given no other choices, we re-started the training. |
| Because the model is suggesting that it will take up to 18 hours to finish training and we would like to see results before we have to submit this project, we are attempting to get the training to run faster. | We tried tweaking a few major parts of the code including: switching to an even ligher weight model architecture than MobileNetV2, reducing the image size, optimizing memory management and usage, and experimenting with hyperparameters like batch size, learning rate, and epsilon decay. |

Reflective Entries:

Devin:

Shakira:

Christian:

Piero: