1. Which file stores the "knowledge" of playing flappy bird learned from Bellman Equation?

A. The Bellman Equation which uses the inputs - state and action, resides in the bot.py file. This file is responsible for the main reinforcement learning logic, which trains the flappy bird to take an action based on its respective state.

1. What is the game state used for the reinforcement learning in the given code?

A. The developer has created a flappy bird environment where the two states for the game are the bird’s vertical distance from the ground and horizontal distance to the nearest pipe. These two states will help the bot to learn on what actions to take for each frame.

1. What is the action space in a given game state?

A. The bot can take two actions based on its state. The variable is PlayerFlapped which defaults to False.

Action 1: PlayerFlapped (True) – In this action, the bird flaps upwards so that it can avoid the pipes in its path.

Action 2: PlayerFlapped (False) - In this action, the bird doesn’t flap, therefore as a result it falls down. This action is to avoid collision with the upper pipes.

1. What are the rewards on game states?

A. In the file bot.py we see the line “self.r = {0: 1, 1: -1000} # Reward function” in the \_\_init\_\_ function.

This means that if the bot chooses the right action it is rewarded by 1 point and if the bot crashes, it is penalized by -1000 points.

1. What does Bot.act(self, xdif, ydif, vel):  do?

A. This function adds the previous state, previous action, and the current state to a list so that it can learn from its experience, which allows the bot to choose the right action to take based on the current height of the bird and its horizontal distance to the next pipe. If the action taken is correct, then the bot is rewarded.

1. Which file and function updates the Q-table values using the Bellman Equation?

A. In the file bot.py, the update\_scores function uses the Bellman Equation to update the Q-table values which calls the dump\_qvalues function, which dumps the updated q-values to a JSON file.

1. What is the best score that you can get on running the program? Please provide a screen snapshot to show the best score.

A. A screenshot of a video game

Description automatically generated

The bot keeps improving for every iteration. It usually takes a very long time to get the best possible score. The highest score I obtained while running the code is greater than 1200.