

Specification

Submit a zip file containing all the relevant Python code for this assignment.

Flappy Bird

In class we built a bare-bones version of Flappy Bird with gravity/jumping and a parallax background. It's your turn to add some functionality to the game and make it your own. You may choose to employ a state machine to handle all of the logic, or if you think it would be better you may manage state with a string. Your goal is to add the following features to your game.

Game States:

- Main Menu
- Countdown State
- Play State
- Pause State
- Game Over State
- High Score State

Game Features:

- Collisions between the bird/pipes/ground
- High Score Saving (top 3 scores of all time)
- 2 Player

Other:

- Sound Effects

Main Menu

A scrolling background with a text menu that has the options **1 Player**, **2 Player**, and **View High Scores**. The name of your game is displayed in a large font. The cursor (`assets/menu/cursor.png`) shows which option is currently selected, and the **ENTER** key is used to select that option. Indicate that the user must press **ENTER**. **1 Player** puts you into a one player version of flappy bird, **2 Player** puts you in a two player version of flappy bird and **View High Scores** takes you to the high scores screen.

Countdown State

A scrolling background with a number that counts from 3 to 2 to 1 then shows "GO" for an eighth of a second before dropping the player into the play state.

Play State

A bird constantly falls due to gravity and can hop with the **SPACE** key. The bird must avoid hitting pipes or the ground. Pipes are procedurally generated at random heights. The pipe gap may change as the gameplay goes on, or may stay the same. The pipes should move at the same speed as the foreground to make it seem like they are moving together. I suggest making a **Pipe** object that has a sprite of a pipe and a rect for a pipe. Also create a **PipePair** object that has two pipes (one inverted, one rightside-up) with a gap between them and a position that moves to the left with the same speed as the ground. The **PlayState** manages a list of **PipePair** objects, appending a new one after a certain amount of time, and deleting them after they've gone off the screen. A score should be displayed in the upper left hand corner, showing how many pipes the bird has currently passed through. The **P** key will enter the pause state. When the bird hits a pipe or the ground the game transitions to the game over state.

Pause State

A message appears "Press P" to upause the game. Pressing P will drop the player back into the play state with the same configuration as the game was in before pausing. Pipes and the bird do not move in the pause state.

Game Over State

When the bird hits the ground or a pipe the game is over and a new screen shows up with the score of the player. Display an image of 1 of 3 medals (bronze, silver, gold) depending on if the score is a top-3 score of all time. The top 3 scores should be stored in a text file called **high_scores.txt**. Show the overall high scores for the game. If a player has a higher score than that, update the high score text file to reflect that, and add a congratulatory message that flashes random colors every half second. You do **NOT** need to find a way for a user to specify WHO got the score, e.g. initials or anything, but it would be a nice touch if the person gets a high score. A good idea is to allow 3 letters to be selected arcade-style for the initials of who got the score. Left-Right arrow keys to select a letter in the initials and Up-Down cycles through each letter. I'd suggest using ASCII codes behind the scenes to hold the information about the letters. Press **ENTER** brings you back to the main menu.

High Score State

Show the top 3 scores (with initials if appropriate) with a gold, silver, and bronze medal next to the names. Press **ENTER** to return to the main menu.

2 Player Mode

If the player selects **2 Player** at the main menu, you enter a 2 player version of flappy bird with player1 controlling a bird and player2 controlling a differently colored bird. **SPACE** jumps for player1 while **TAB** jumps for player 2. Rather than keep score the normal way, the game plays until a single player hits the ground or a pipe. The game then transitions to a game over screen that displays who wins. If two players hit a pipe or the ground on the same frame, print that the game was a tie.