Three Easy Game For CoVid-19

Jinqian Pan | panjin@kean.edu | CPS3320

Background:

Because of the novel coronavirus outbreak, many people can't go out, but they can only stay at home and go out to work. People's normal life is greatly affected. At this time, playing games becomes a good choice to pass the time. According to the latest data, steam (a video game digital distribution service by valve.) has more than 23 million people online at the same time, or more accurately, 23436186 players. As the epidemic continues to spread, more people may be forced to stay at home, so we expect that in the future, steam will also set a new record for online population. In this project, I used Python to write three puzzle games (snake, Tetris, 2048) to provide more game options for people.

Safe Goal:

Using PyGame to write Snake and Tetris.

Stretch Goal:

Writing 2048 with Tkinter.

Bulleted List Outlining:

For Snake:

1. Draw a window
2. Define the size of small squares
3. Draw a picture to start the game and wait for the event input
4. There are two modules in the main cycle of the program, one is to run the game module, the other is to display the end screen of the game module
5. Run the game module:
   (1) Random initialization sets a point as the starting point for snake eating
   (2) Starting from this point, create a greedy snake (array) with a length of 3 spaces
   (3) Random initialization sets a point as the starting point for snake eating
   (4) Initialize a direction of motion
   (5) Random position of an apple
   (6) Handling events in the game loop
   (7) Change the direction of the snake according to the key
   (8) Check if the game is over (hitting the border or hitting yourself)
   (9) Check if the snake can eat apple
   (10) Draw background, square, snake, apple, score and other game elements
6. Module to display the game ending screen
   (1) Draw game over
   (2) Wait for the user to press the key to restart the game

For Tetris:

1. Draw a window
2. Define the size of small squares
3. Draw a picture to start the game and wait for the event input
4. There are two modules in the main cycle of the program, one is to run the game module, the other is to display the end screen of the game module
5. Run the game module:

(1) The position of the random square at the top
(2) Handling events in the game loop
(3) Change the movement direction of the box according to the key
(4) Check if the game is over
(5) Check whether the blocks are connected in a row
(6) Draw game elements such as backgrounds, squares, scores, etc

6. Module to display the game ending screen

(1) Draw game over
(2) Wait for the user to press the key to restart the game

For 2048:

1. Draw a window
2. Define the size of small squares
3. Draw a picture to start the game and wait for the event input
4. There are two modules in the main cycle of the program, one is to run the game module, the other is to display the end screen of the game module
5. Run the game module:

(1) Handling events in the game loop
(2) Numbers appear at random
(3) Change the direction of all numbers according to the key
(4) Check if the game is over
(5) Check if two numbers are combined into a larger number
(6) Draw game elements such as backgrounds, squares, scores, etc

6. Module to display the game ending screen

(1) Draw game over
(2) Wait for the user to press the key to restart the game