Connor McKinney  
4-21-20

Penguin Pacer Instructions

In Penguin Pacer the goal of the game is to reach the finish area (the bottom right corner of the screen) in the least amount of time possible by navigating through the maze.

* Move the penguin from the top left corner of the screen to the bottom right corner using the arrow keys.
* Collecting fish gives you a 20% decrease to your time per fish collected. Lower times are better, so catching the fish can make quite a difference on larger mazes.
* After reaching the end you will no longer be able to move. Check the python console for your score, seed, and maze size if you would like to play the same maze again.
* The script must be re-run each time you wish to play.
* To play the same maze as someone else you must use the same seed and size(in each dimension).

Customization:

The Python console will ask you for three numbers each time you play:

* Seed
  + Seed determines the random generation of the map. Using the same seed over and over gives you the same map, allowing you to practice or challenge other people. Inputting the number “1” will cause the game to generate a random seed. The random seed will be printed to the console so that you can save it if you want to use it again.
* Number of columns
  + Number of columns is the longitudinal/horizontal number of maze nodes. More gives a larger maze.
* Number of rows
  + Number of rows in the latitudinal/vertical number of maze nodes. More gives a larger maze.

Using the same seed, say “9470” on differently sized maps (maps with different numbers of columns/rows) will cause different mazes to be generated. As a result, high scores are calculated based on the maze size as well as the maze seed.  
Notes:  
The resolution of the game window can be changed near the top of the “Penguin Pacer.py” file. Changing the resolution will likely cause things to look less than ideal and may or may not cause errors.

In my testing mazes between 10x10 and 30x30 seemed to be good sizes.

Files:  
You should have two files, “Penguin Pacer.py” and “handle\_scores.py” in the same folder.  
High scores are saved in a text file in the format (seed, number of columns, number of rows, score). The file can be found in “Scores\high scores.txt”. The “Scores” directory will be created in the same folder that the python scripts are run in.

Rubric:

* Reads from a file
  + The “handle\_scores” module reads from the “high scores” file to retrieve previous runs’ scores.
* Writes to a file
  + The “handle\_scores” module writes to the “high scores” file to save a run’s score.
* Uses at least one graphic element
  + The player character, fish collectibles, and finish area all use graphic elements. The walls of the maze also use pgzero rectangles.
* Use of randomness
  + The played seed can be random
  + The maze generation is random
  + The number of fish is random
  + The position of fish is random
* Multiple functions
  + There are many functions in the maze class, char class, and in the “handle\_scores” module.
* Must take input from keyboard or mouse
  + Takes keyboard input in the form of settings to generate the maze and to move the character.
* Use a dictionary somewhere
  + The “handle\_scores” module uses a dictionary in the “get\_high\_score” function to keep track of the current high score for the maze indicated.