CSSE1001 – Assignment 3

Subject – CSSE1001

Student – Daniel Eddy (s4480588)

Due – 29/10/2018 (extension granted)

Part 1 – Required Features

For easier comparison, figures one and two illustrate the queue before part 2 was implemented. Figure 1 illustrates the queue when no students have joined and figure 2 demonstrates the queue when there are multiple students joined.

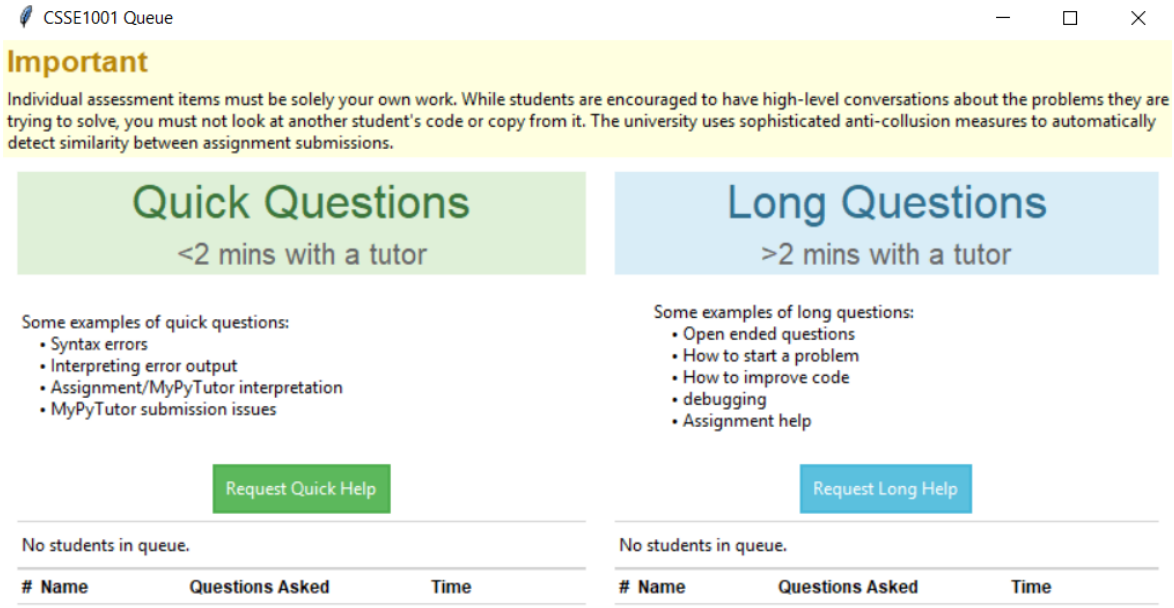

Figure 1. Queue with no students joined.

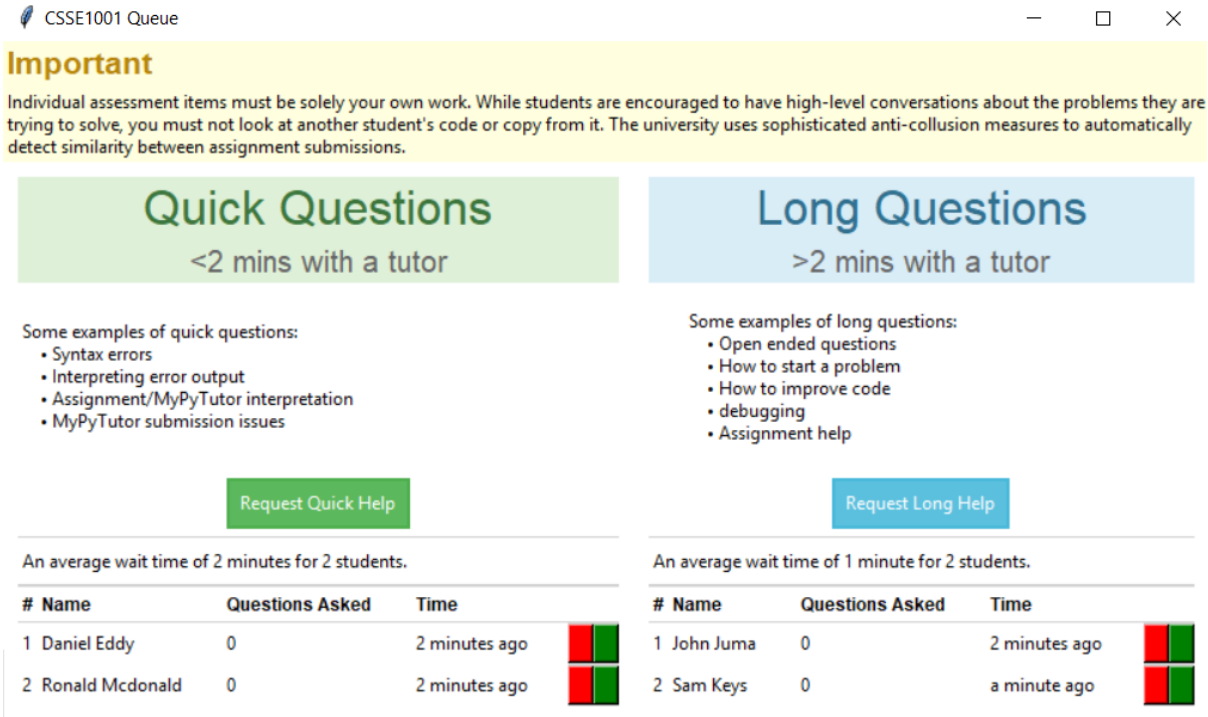

Figure 2. Queue with several students joined.

Part 2 – Additional Features

For the purposes of the Additional Features of this assessment, I decided to implement a game which the students can play while waiting to have their questions answered. As illustrated in figure 3, the game which I coded and implemented, is a basic game of snake, where the user maneuvers a snake using the arrow keys, in order to eat the fruit ©, while avoiding hitting the snakes head against the walls or its own tail.

Brief User Manuel

To access the additional feature (snake game), the user must navigate to the menu button at the top left corner of the queue. The associated drop-down menu contains a button labelled Snake which activates the snake game at the bottom of the queue. Once the game is activated, the user controls the snake using the arrow keys, in a bid to eat the fruit © and avoid its own tail and the walls. After each fruit is eaten, the snakes tail length increases by one. The user wins the game if all available space on the canvas is taken up by the snake, and loses if the snakes head hits a wall or its own tail.

Approach taken to writing the code.

Initially, I was not going to attempt the additional features, but after I saw that it is required if one wants to obtain a seven for the course, I decided to implement the game. In writing the code, it was difficult to determine the best method to do so. The first step I took was to create a canvas and determine how to place an object (or in this case a piece of text) onto the canvas. Secondly, I went about determining how to position an object on the canvas randomly and to subsequently identify the coordinates of the randomly position object. At this point, I was able to create a canvas and randomly place the snakes head on the canvas, with initial tail behind it, in addition to randomly positioning the initial 'fruit' on the canvas.

At this point, I attempted to use a recursion method to refresh the canvas after every 175 milliseconds, and subsequently implement the method for moving an object on the canvas. This was probably the hardest, as I needed to learn how to implement several new commands, including .move, and .coords. After this, I spent a substantial amount of time binding the arrow keys to functions which would change the snake's direction of travel.

From learning the commands associated with the previous steps, determining if the snake had eaten the fruit, hit a wall, or hit its own tail was fairly simply. The final step was to import and use the game.py module in the a3.py module. A substantial amount of time was spent on this step, as I was attempting to have the snake game open in its own pop-up window. I was unable to make the game open in a new window, and thus settled with opening the game at the bottom of the queue.

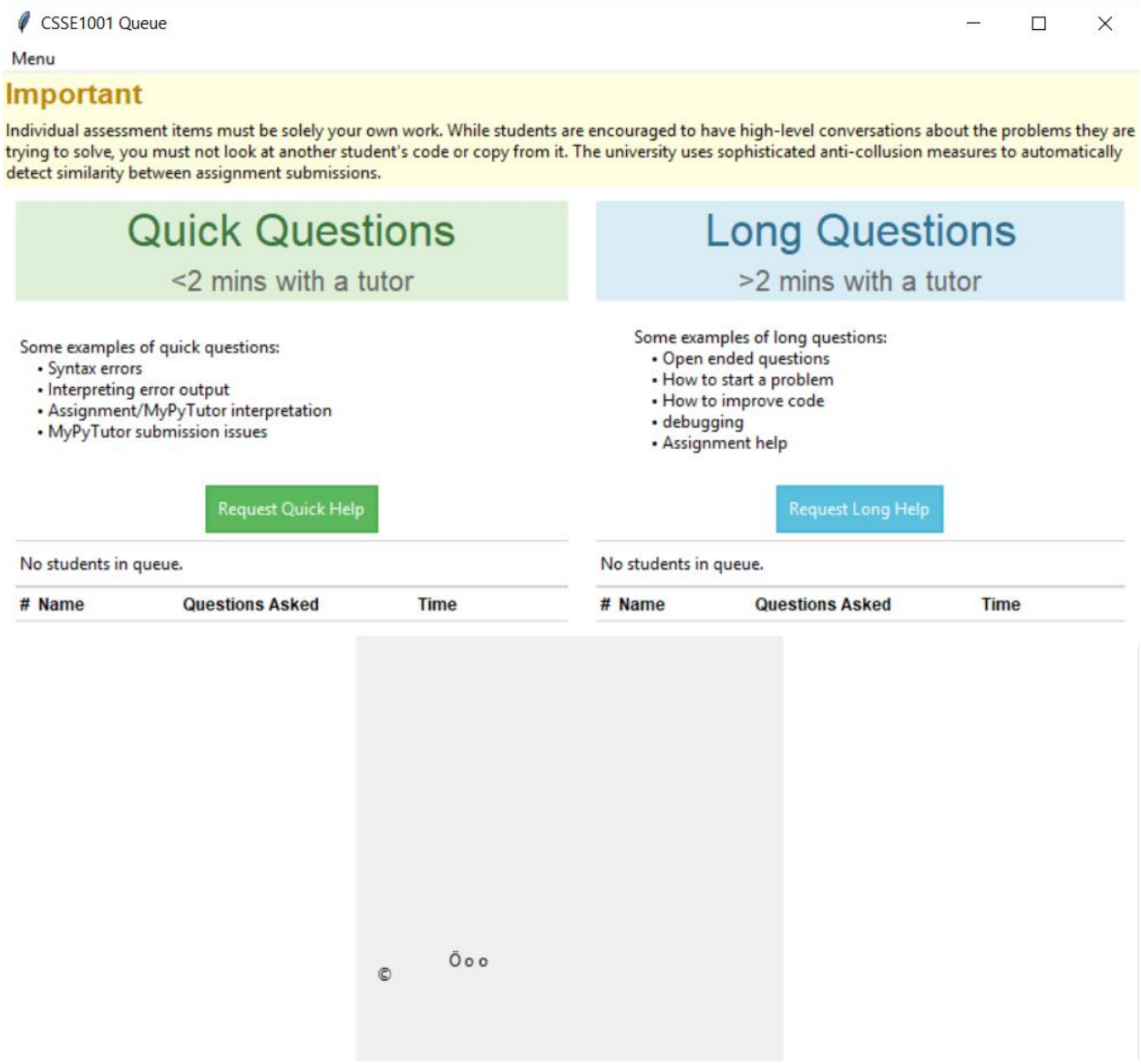

Figure 3. The snake game inside the Queue window