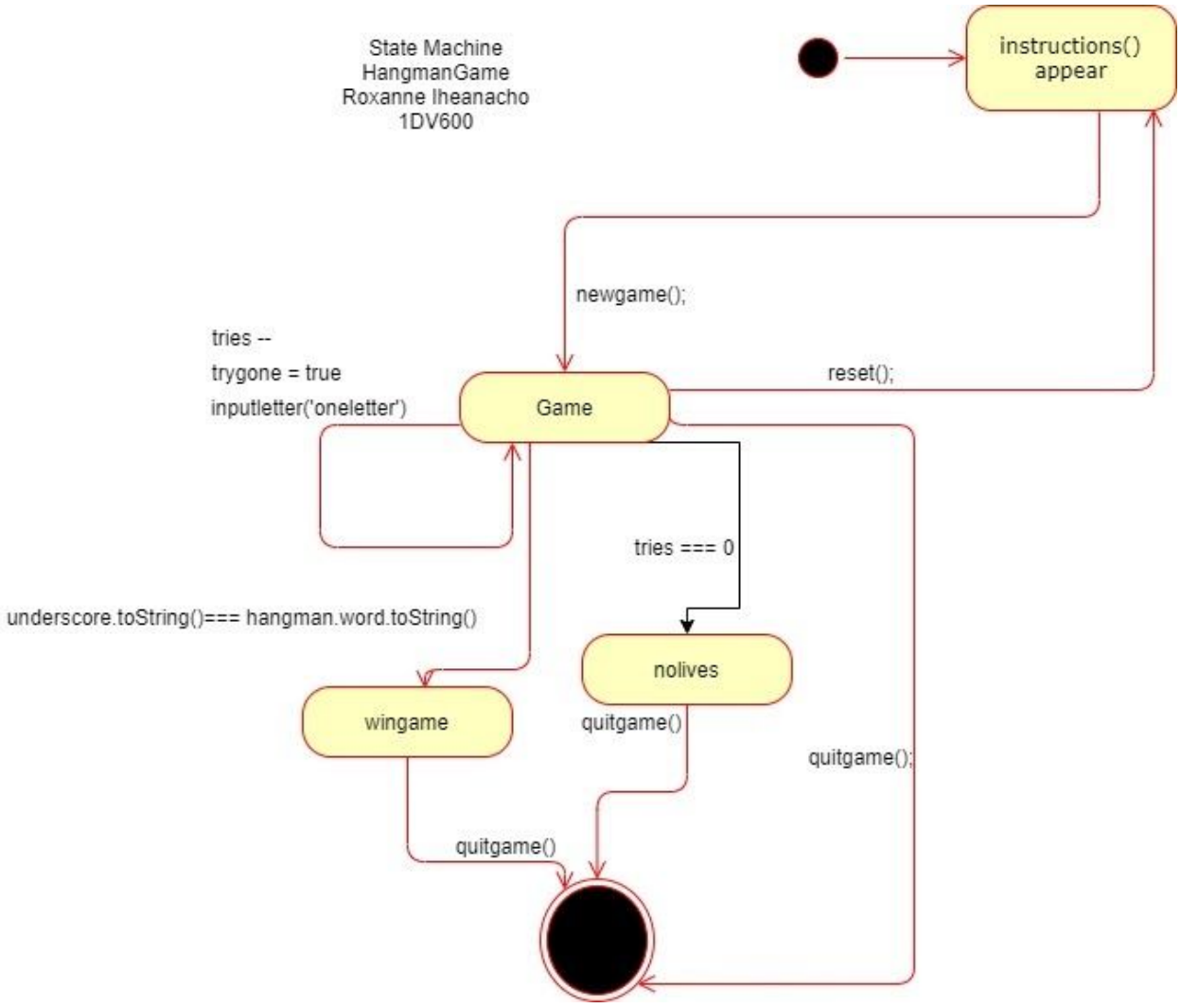
UML Diagrams
1DV600
Roxanne Iheanacho
20.03.2020

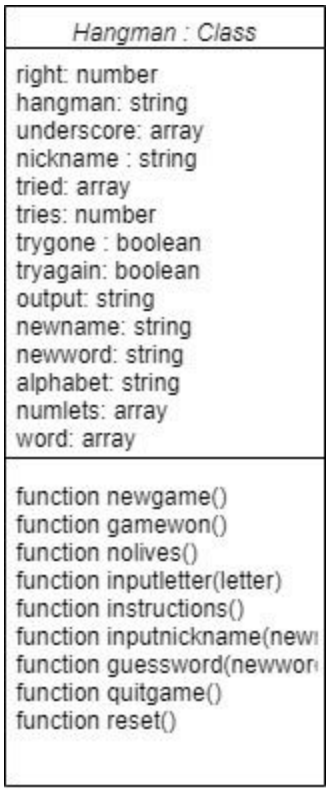
This diagram goes through the different states a player can be in. Instructions is the first state, which is a menu that allows the user to see what guards there are to get to the game state and the guards within the game state that loop back to the game and then quit, win or end with no lives that leads to the reset guard. The black dot with two circles around it indicate the end point of the state machine. This state machine diagram simplifies each state in order to clearly understand the specific states one can be in.

State Machine Diagram:

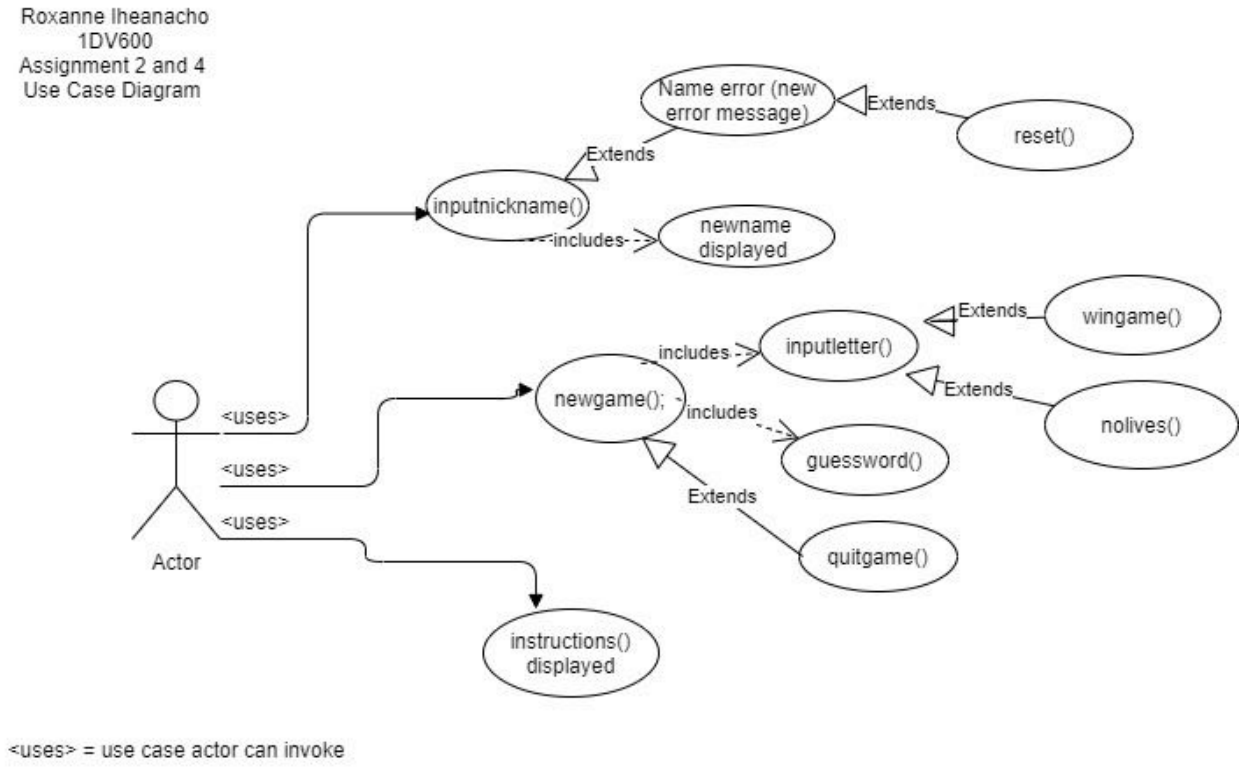


UML Diagrams
1DV600
Roxanne Iheanacho
20.03.2020

Here is the class diagram. It was specified that it is unlikely to only have one class, however given that the final product online had one file (one src code), there was therefore only one class. This class simplifies the understanding of functions, their calls, and what each attribute represents. You can see here all of the arrays are attributes of the hangman word. Then the number types are loops that loop through the code until the user has no tries left. Then there are the boolean attributes that indicate whether the game is playing or not, and string attributes are linked to user inputs.



UML Diagrams
1DV600
Roxanne Iheanacho
20.03.2020
Use Cases




**USE CASES**

UC 1 Input Name
Main scenario:

1. User wants name to display
2. Instructions shows inputnickname(‘’) as function to input in console to input name to present to user each time the game loops
3. User inputs nick name with inputnickname function in console
4. Game displays nickname playing in console everytime the game is played, and letter is entered
   Extensions:
   3.1 User fails to input nickname, or it is over 15 letters
   1. Error message is displayed
   2. Game is reset() when error message is presented

UC 2 NewGame
Main scenario:

UML Diagrams
1DV600
Roxanne Iheanacho
20.03.2020

1. User wants to play a new game
2. Console shows newgame(); should be entered in order to play a new game
3. User inputs newgame(); into function and new game is started
4. New word is presented and user is asked to guess word
5. User inputs guessed word into console which is then displayed
6. User is prompted to input letter in order to guess word
7. User inputs letter
8. Letter appears in the word next to the guessed word
   Extensions:
   3.1 User wants to quit the game
      1. User writes quitgame(); in console and game is quit
   7.1 User loses all lives
      1. Nolives function is activated, Nolives message is logged and quitgame(); state activates
   8.1 User inputs enough letters and winsgame();
      1. Wingame message displayed and quitgame(); function is extended

UC 3 Instructions Displayed
Main Scenario:

1. User wants to play a game and opens game in console.
2. User views instructions when opening the console which presents user the options of newgame(); inputnickname(); and guessword();
3. User is able to pick any option after viewing instructions for game.