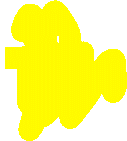
Matlab Rubric Comparison Snake

Graphical user interface, application

Description automatically generated



The game of snake produced from the MATLAB code covered all concepts that have been taught throughout all the practical in the course so far and have been implemented correctly and efficiently throughout the code.

User input and output: In the snake game the most important part of the game is the user controlling the game. The code prompts the user to control the game using the arrow keys or the WASD keys on the keyboard. This allows the user to move the snake to collect the fruit and when the game is over, there is a message displaying the game is over and the score obtained.

Visual of the application data: When the code is run a figure window appears which displays the snake game with the snake moving with an initial direction of ‘right’ on a 20 by 20 grid and a red circle randomly generated on the grid representing the fruit.

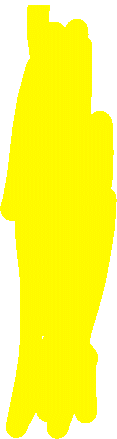
Vectors – Vectors in this code were mainly used to move the snake on the grid. As the snake moved along the grid when a direction change was detected the first element would of data.x would either be added or subtracted and then concatenated onto the end of the data.x apart from the last element.

Functions – Functions were used to produce a more concise code which dictated the movement of the snake, the overall data of the snake, the game loop and it also detected if the game was over or not.

Loops and Conditional execution – In this code a while loop was used for when the snake needed to grow or not as well as lots of if and else statements used for different purposes in different sections of the code.

Table

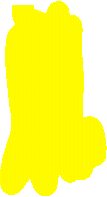
Description automatically generated



The game of snake has unique features from a normal game of snake as it each fruit consumed by the snake is 10 points which is displayed clearly at the top of the game and when the game is over a message is displayed in the command window saying that the game is over, and the score obtained. The game is very responsive, and the interface is easy to follow with clear colour indications of the colour of the snake and the fruit.

Text

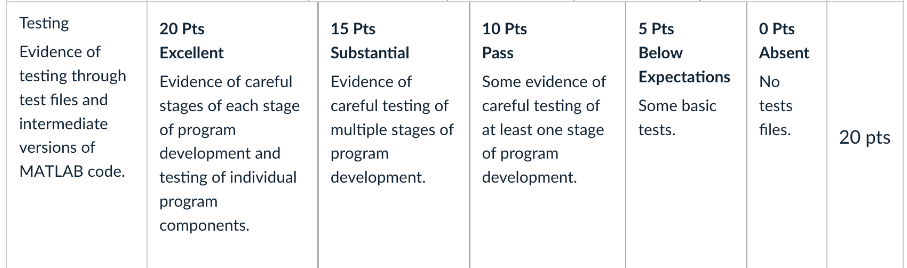
Description automatically generated

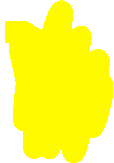


To develop the final game of snake there are many versions and files that were tested before getting the final code. The first file that was tested was the gameLoop which tested what would happen when the game was over which is named gameLoopVersion1. Following the attempt of the gameLoop function, a keypress function code was also attempted which moves the snake in all different directions named keyPressVersion1. This code was also tested with a test file to see if any errors would show up when it was run.

To store all the data and to update the new data when the game was being played a global variable was used and the first version that was coded was theGlobalDataVersion1 which was used to store the different data values and any new data values that the game would receive. It was the first use of the persistent function which was through the inspiration of a MATLAB discussion page online which has been referenced.

Once the initial versions of the codes were attempted errors were discovered so improvements and changes were made to the following codes to develop the final code. Once the final code was developed each function was tested again using test files to ensure that the code was running properly and as efficiently as possible.

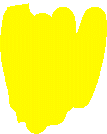




There is evidence of multiple test cases for the final code and versions before the final code during the development stages to ensure that the final code is the most complete and efficient code. Testing the code was used to discover any errors in the code and was able to help find the problem earlier in the process which is more effective than discovering the issue when a complete code is written.

A picture containing table

Description automatically generated



Each code in the test files and the final code have comments on each line of code explaining the purpose and why the code is used. The final code is seen to be quite compact due to thorough testing to find the most efficient and compact way to display the code