|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **CSY1018: Web Programming** | | | | | | | |
| Due for Issue  (week commencing): | **Sunday, 5th February, 2017** | | **Last Date for Submission:** | | | **Sunday 30th**  **April 2017**  **23:59:59** | |
| Agreed Date for late  submission: |  | | Module Tutor: | | | Thomas Butler | |
| Student ID: | 17418687 | | | | | | |
| Aspect (& weighting) | Excellent A | Good B | | Satisfactory C | Needs some more work  D | | Needs much more work  F |
| Functionality (40%) |  |  | |  |  | |  |
| Game Design (15%) |  |  | |  |  | |  |
| Program Design (15%) |  |  | |  |  | |  |
| Testing (10%) |  |  | |  |  | |  |
| Code Quality and Efficiency (15%) |  |  | |  |  | |  |
| Video Demonstration (5%) |  |  | |  |  | |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Specific aspects of the assignment that the marker likes: | | Specific aspects of the assignment that need more work: | | | |
| Tutor’s Signature: |  | Date: |  | Grade: |  |

Contents

[Introduction 2](#_Toc481263408)

[Game Design 3](#_Toc481263409)

[Program Design 3](#_Toc481263410)

[Testing 3](#_Toc481263411)

[Conclusion 4](#_Toc481263412)

[References 5](#_Toc481263413)

# Introduction

This was the second assignment received for the Web Development module from the University of Northampton. It required for game which was made using HTML and CSS to be given functions through the use of JavaScript. The basic layout was given, but without any functionality in the program. So the goal of this assignment was to make four horses and their respective riders go around the entire track in order to win a bet set at the beginning of the race. And upon that basic function, it was required that additional functions be implemented, such as the ability to specify the number of laps for the horses to run and display the chances each horse has of winning the next round.

# Game Design

Most of the solution to the game came from scrolling through helpful coding websites on the internet as well as seeking aid from my lecturer. It was extremely helpful that the CSS contained bits of code to change the position of the horses as well as the animation required to mimic movement, so all that there was to change the current position of each element in the game and as it came to an edge stop the current animation, change the standing position of the horse and start its running animation in the new direction. As each horse crossed the finish line for the last time, it simply stops the horses from moving and used the CSS element to make the horses appear to stand straight.

In order to make sure the same horse didn’t win each and every game, It simply gave each horse a random ‘speed value’ when they made a turn at each edge. As a result of that it became interesting to see which horse will win, as the horse that leads at the start of the game could be over taken by the other horses as their speed changes when they make a turn.

# Program Design

To reduce the amount of code in the game, and minimize the amount of repeated lines, functions were used that perform larger tasks and then that function is called where necessary. And also arrays were made of use to store values as well as assign values from. The screen size is taken and used as a variable, after that the animated elements, such as the horses, are placed on within the tracks and are given certain values which are actually the amount of space each horse can use for their respective tracks. And throughout the entire game each horse sticks to their allocated track sizes.

The problem was broken down into smaller tasks by using functions and methods throughout the game, so that those functions and methods could be called where needed and therefore reduce the large size of the game.

Different methods to approach the same problem were considered but ultimately none of them showed enough simplicity as this approach. Also given the time constraint imposed, they wasn’t much time to find and fix all the bugs and errors that would potentially arise and also have the whole assignment ready for submission.

# Testing

The basic method that was used to test this game was by running it over and over again while observing the minute changes that were made in the coding and editing it if the conditions were not met. But certain parts of the game such as changing the speeds at the turns were simply changed by setting its initial ‘speed value’ to suit the gameplay so that it can be observed much faster, instead of having to wait for the horse to finish the track on their own time. But for other smaller changes to be observed, the rest of the lines of code which were not useful at the time were made into comments and the lines of code that needed observation were simply run.

The game was run several times with different values for the bet amount as well giving several lap numbers. Each horse was used to run the track. Something that was very observable was the fact that out of all the horses, the ‘Brown’ and ‘Green’ horses always had a high chance of winning as compared with the ‘White’ and ‘Blue’ horses. It was inconclusive whether or not this is happened simply because the ‘Brown’ and ‘Green’ horses had a shorter track length to cross compared with the other two horses or if it was simply that the auto generated values for the speed variable were larger for the ‘Brown’ and ‘Green’ horses.

The first noticeable bug in the program occurred when the horses ran for the second consecutive time. The first time they all run smoothly and finish off in their respective tracks, but after the second run, they finish off all piling on top of one another. Which continues for the rest of the game until the page is refreshed and an entirely new game is taken. Another bug was found when the one of the horses makes a win and gets a lot of money, if that horse were to lose the next round all his previous winnings would be reduced as well as the current amount the player bet on that horse.

# Conclusion

This game contained a few errors such as the horses piling on after the first round, and the extra money reduced in the cases of a winning horse losing. Also the position board is slightly on the track and as the screen size is reduced further the position board is moves more onto the track and the horses are seen as to be running over the position board. Just as it has its errors there are several things that work well in the game. The horses keep to their lanes while running and the betting function used works correctly. The position board displaying the winnings works properly as well. And the horses are responsive to the web page when its size is changed. The horse will always run for the exact amounts of laps that the player specifies.

This game can be much better than what it is, certain improvements, such finding a way to fix the bugs in the code and adding more animations using the giving animation schemes in the assignment brief, could make this game much more user friendly and would colourful making people want to play it more. As well as animations making it a better game to play, the addition of sound and a start screen would make this game a lot more interactive for its players.

And finally in conclusion if I were to make a similar game in the future, I would incorporate a track that makes it fair to all the riders so that there is no cheating as well as add more animations and sounds. Simply so that the game will become more interesting to play and serve its basic purpose, and that is to be a fun means to pass the time.

# References

"Stack Overflow". *Stackoverflow.com*. N.p., 2017. Web. 29 Apr. 2017.

Crockford, Douglas. *Javascript*. 1st ed. Sebastopol: O'Reilly Media, Inc., 2008. Print.