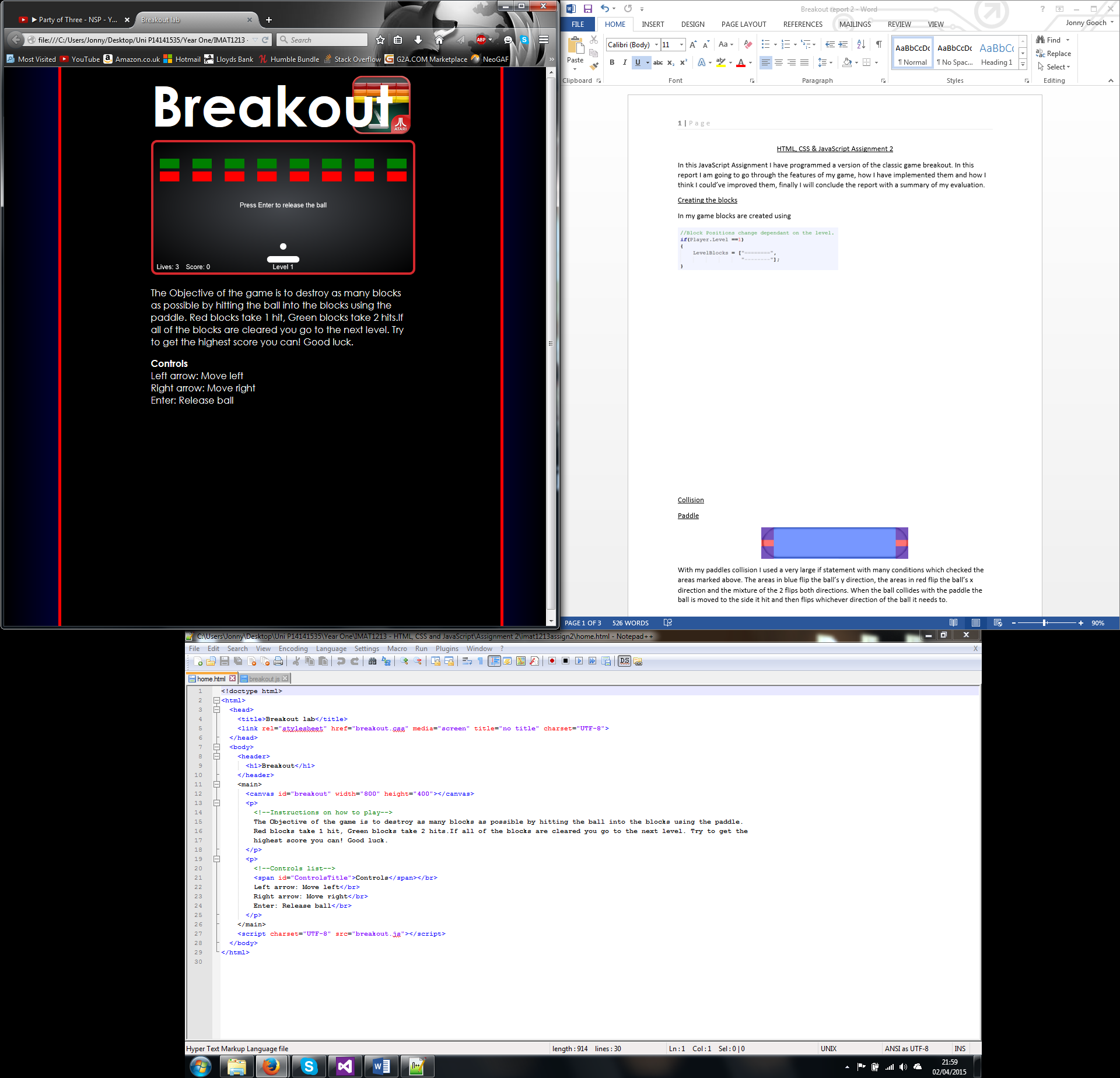
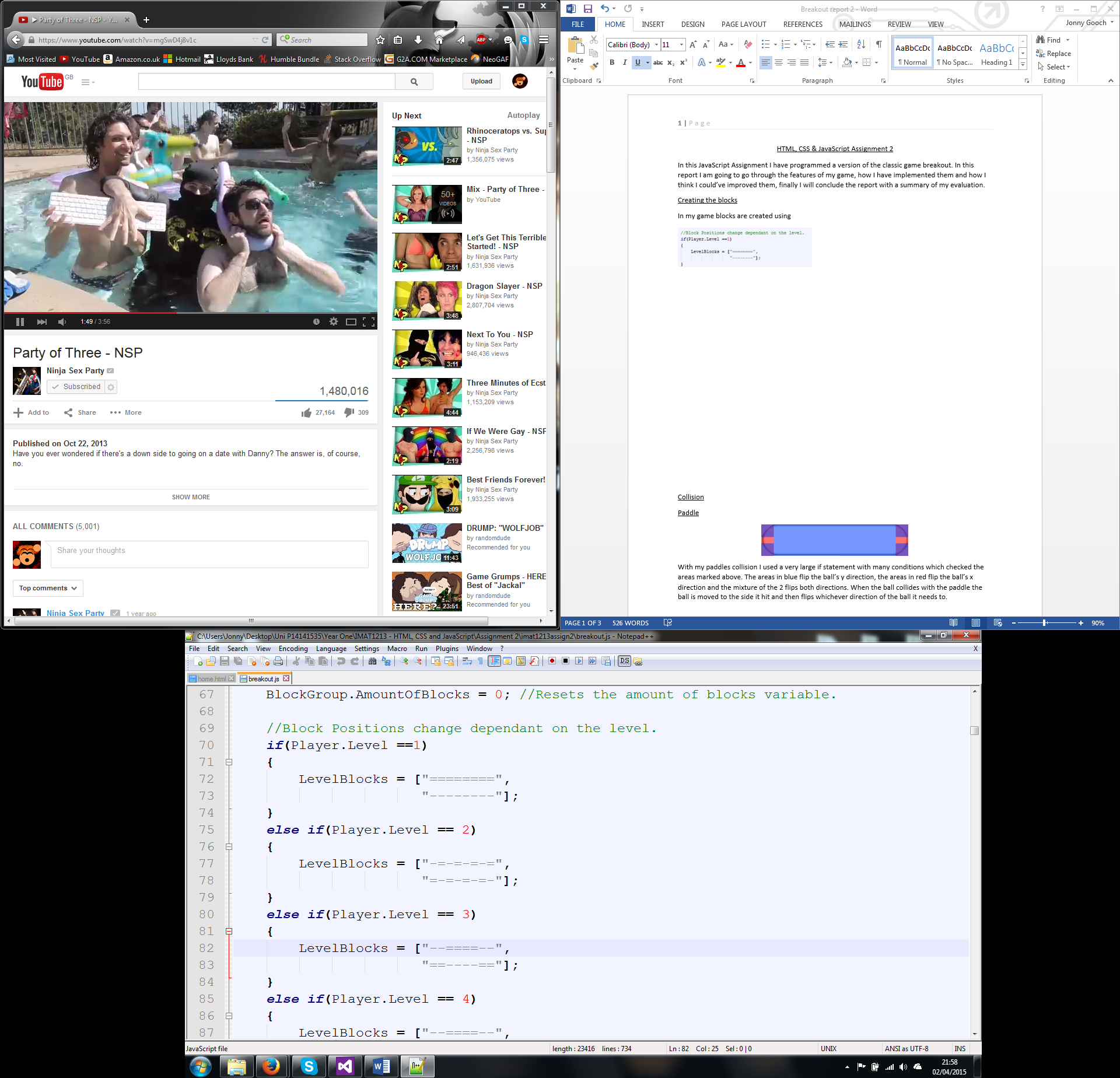
HTML, CSS & JavaScript Assignment 2

In this JavaScript Assignment I have programmed a version of the classic game breakout. In this report I am going to go through the features of my game and how I have implemented them, finally I will conclude the report with a summary of my evaluation.

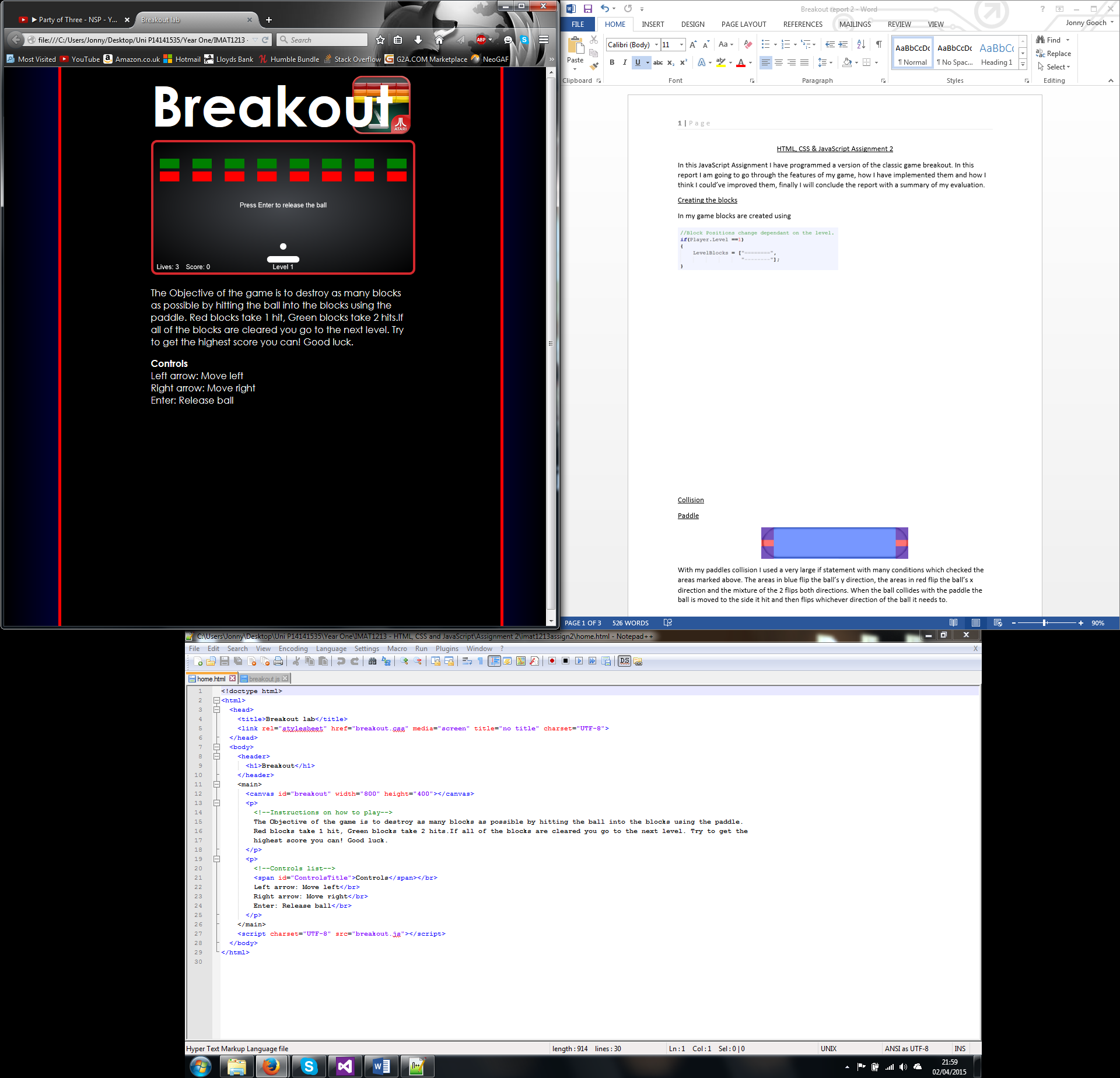
Creating the blocks

In my game blocks are created using a string of characters representing different types of blocks. If the character is “=” then the block is a double hit block, if the character is a “-“ then it is a single hit block and if there is a space then that means that no block is drawn. The amount of strings in the array of the blocks is the amount of rows containing blocks.

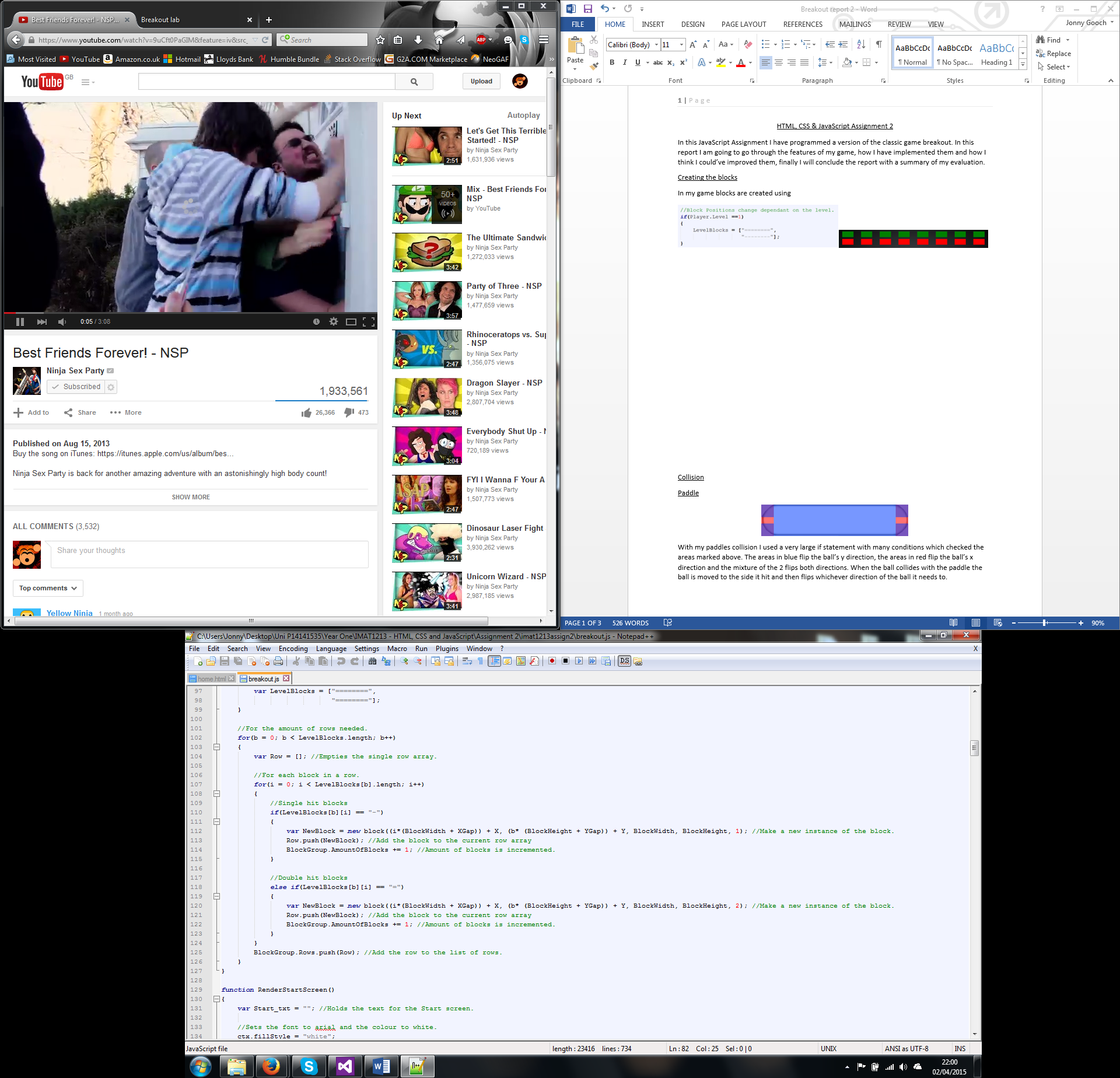


*Example of Level 1’s layout.*

When the strings are entered into the algorithm the code checks each character to determine the block type and generates a block appropriate to that type. To position the blocks I have used 4 variables to make sure the blocks are spaced evenly. These are the X coordinate of the starting block, the gap space between the block, the size of the block and the number of the block. This means that I can move the x / y coordinate so many block spaces so that it is aligned with all of the other blocks.



Coordinate( X / Y ) + (Block Dimension( Width / Height ) + Gap ( X / Y ) \* Amount of blocks)

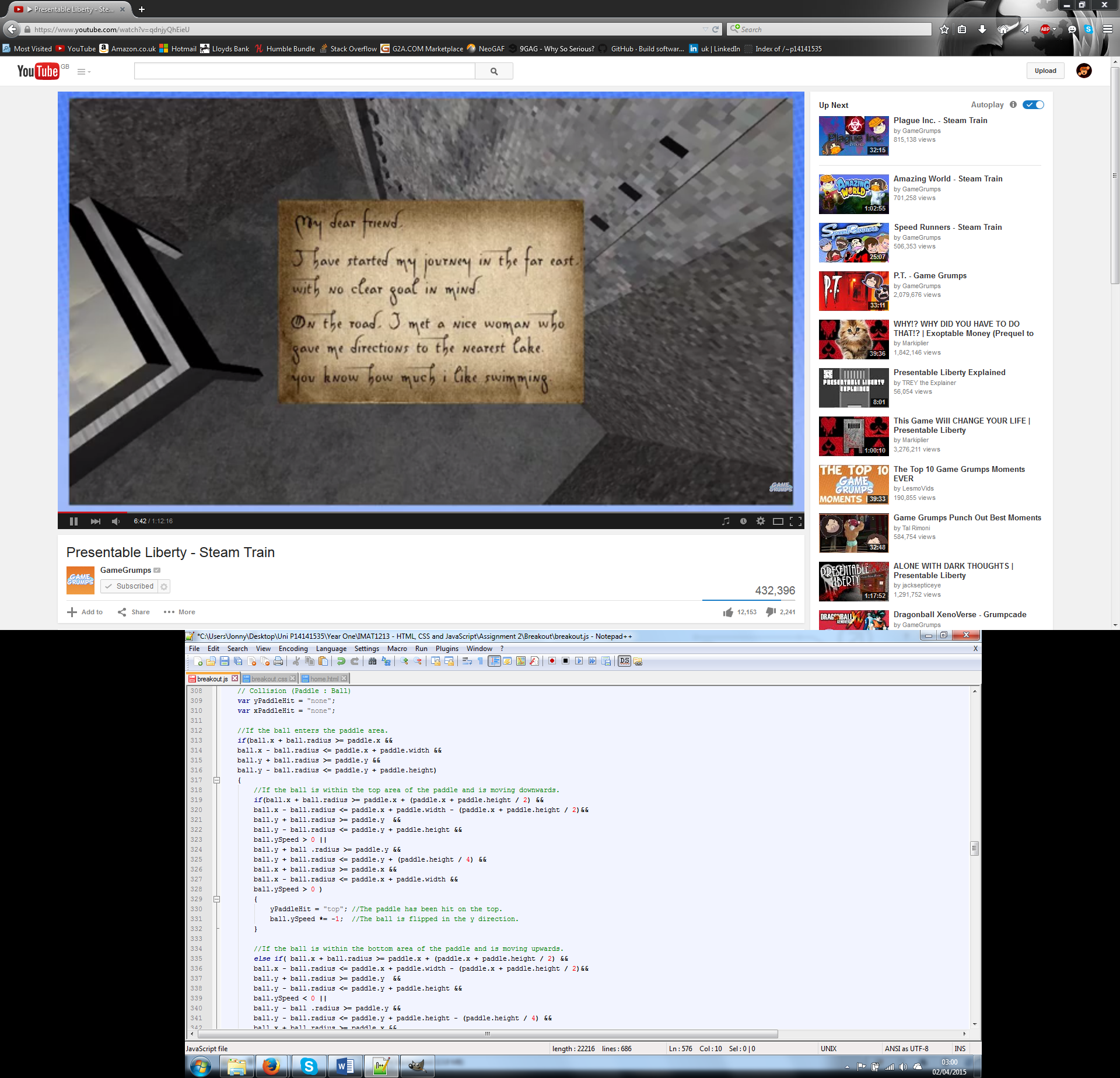


*The algorithm which takes the strings and makes them to blocks.*

Collision

C:\Users\Jonny\Desktop\Untitled.png 

With my collision I used a large if statement with many conditions which check the areas marked above. The areas in blue flip the ball’s y direction, the areas in red flip the ball’s x direction and the mixture of the 2 flips both directions. When the ball collides with an object the ball is moved to the side it hit and then flips whichever direction of the ball it needs to.

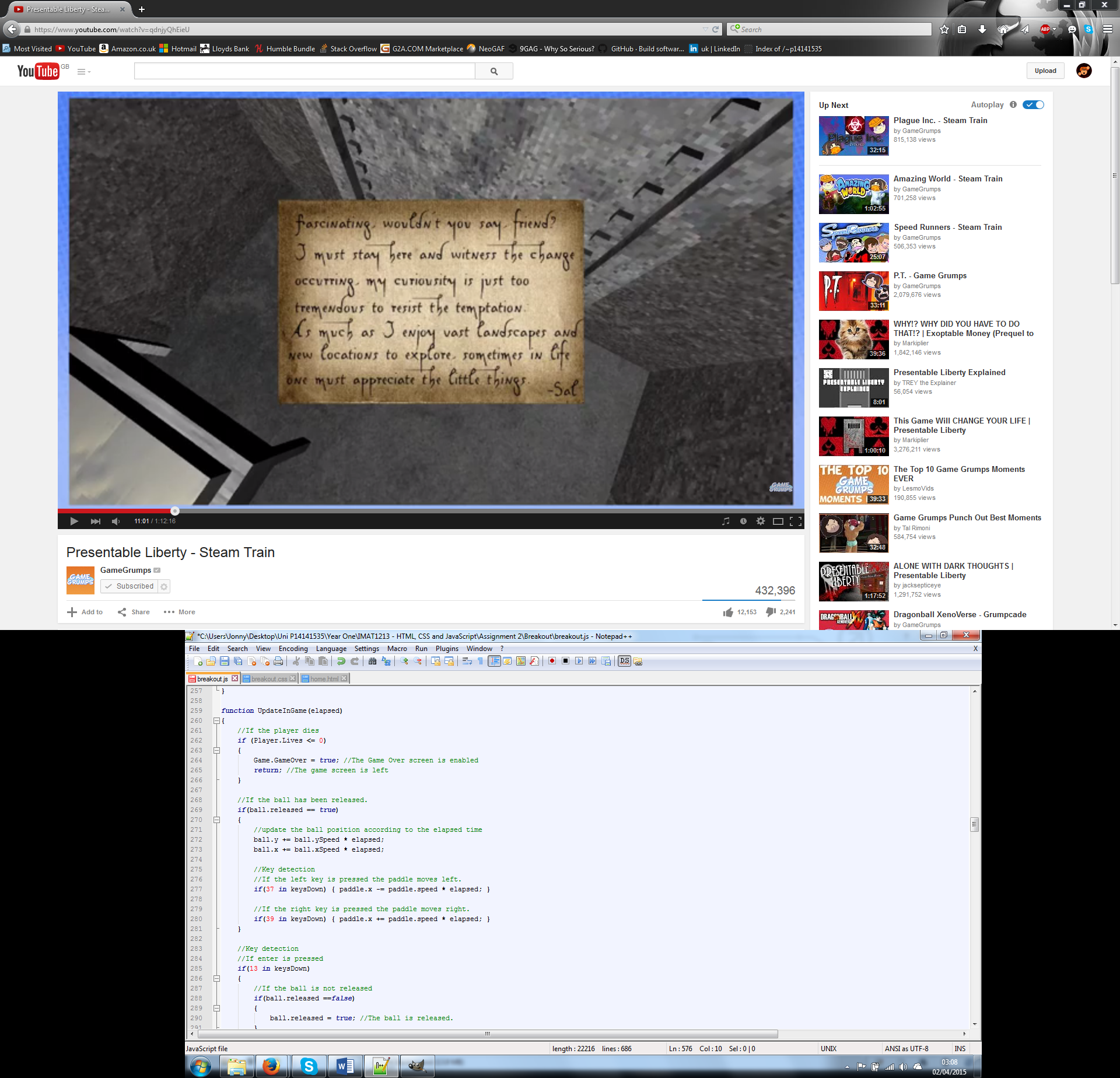


*Example of collision (top of paddle)*

For the computer to determine which side was hit; the ball’s direction is checked. For instance if the ball is moving down in cannot hit the bottom and therefore it must’ve hit the top of the object.

This collision works particularly well as it provides semi realistic collision although it doesn’t provide many different angles as I found this to be a rather large challenge just getting the ball to collide correctly.

There is also a bug in my collision with the paddle where occasionally the ball will surpass through it, I think this bug occurs due to the nature of the ball’s movement. As the ball’s movement is dependent on time elapsed I feel the ball will sometimes jump further than the space the paddle accommodates.



*Ball’s movement affected by time elapsed.*

Conclusion

For this assignment I think I have done a good job creating this game although I feel it can be further improved by working further on the collision, adding power ups and by adding a make your own level option. My game does have the potential to make these possible but I had not made them for my game.