The Report for Our Maze Game

I felt that the making of the game went very well, although after the master class it did not get the chance to do as much as I would like to have, I felt I achieved real progress at the master class.

After using the code that we were already supplied with, we had the basis of the game. The first that we began to develop was inserting the walls, after drawing out our plan of where we planned to put the walls we began finding the right co ordinates.

After creating a function and adding the walls, to our program we tested it out by using unit testing.

```
}
    walls[0].x=60;
    walls[0].y=0;
    walls[1].x=120;
    walls[1].y=0;
    walls[2].x=120;
    walls[2].y=60;
    walls[3].x=240;
    walls[3].y=60;
    walls[4].x=60;
    walls[4].y=120;
    walls[5].x=240;
    walls[5].y=180;
    walls[6].x=300;
    walls[6].y=180;
    walls[7].x=60;
    walls[7].y=240;
    walls[8].x=120;
    walls[8].y=240;
    walls[9].x=300;
    walls[9].y=240;
         */
```

We found that our walls were not at first properly working. After promptly correctly a slight problem with our co-ordinates we soon saw that this was the problem.

After the wall were inserted with began a new function called collision, which enabled the counter in the game to go back to the beginning of the maze if it hit a wall:

```
collision_box;

      collision_box *setUpWalls( void )

      int collide(collision_box box1,collision_box box2){
```

After we corrected a few errors found in the terminal, such as spelling mistakes and incorrect positioning of '(' and '{}' signs, we had it working.

It took a lot of adjusting and processing but eventually we had a fully functioning game. The next thing we did was the score, the scoreboard was later finalised by Samuel.

The few things we had left was the background, walls and counter:

```
    CCSS_load_resize_and_rotate("./resources/img/smiley.png",
0.05, 180);
    upLeft =
CCSS_load_resize_and_rotate("./resources/img/smiley.png",
0.05, 225);
    left =
CCSS_load_resize_and_rotate("./resources/img/smiley.png",
0.05, 270);
    downLeft =
CCSS_load_resize_and_rotate("./resources/img/smiley.png",
0.05, 315);
    down =
CCSS_load_resize_and_rotate("./resources/img/smiley.png",
0.05, 0);
    downRight =
CCSS_load_resize_and_rotate("./resources/img/smiley.png",
0.05, 45);
    right =
CCSS_load_resize_and_rotate("./resources/img/smiley.png",
0.05, 90);
          upRight =
CCSS_load_resize_and_rotate("./resources/img/smiley.png",
0.05, 135);

// Apply background to screen
     CCSS_apply_surface(0, 0, background, screen);
     // Apply our character
     CCSS_apply_surface(x, y, character, screen);
               CCSS_apply_surface(endBox.
```

After much testing and solving our problems we had successfully made our maze game by the end of the master class. We also added other functions in order to improve the game, which also needed fine-tuning, but eventually the game could be played. And once applying the program, we tested out the game as we went along it ensure the final result would be what we intended.

After the master class, only needing minor adjustments and extra

details added for example making sure that the counter faced in the way it was traveling.

I am very happy at how the game has ended up, and believe it is a real success. And the game is now compatible to play on most computers.