Technical Report

Functions Checklist

The assignment that was presented was that of making a character in a game move to avoid falling bombs, with each attempt having 3 lives. Of the functions that were required to fulfil this game the following ones are the ones which were achieved in the making of this game.

1. When the start button is pressed the game should begin and the start button should no longer be visible.
2. Bombs get created at a random position at the top of the screen
3. Bombs fall down the screen towards the player
4. When the bomb hits the ground (green grass area) it explodes
5. If the player is in the radius of the bomb, print “game over” and set the player’s animation to dead (css class “character dead”)
6. Bombs get created at a random position at the top of the screen
7. Bombs fall down the screen towards the player
8. When the bomb hits the ground (green grass area) it explodes
9. If the player is in the radius of the bomb, print “game over” and set the player’s animation to dead (css class “character dead”)
10. Make the game more difficult by randomizing the bomb’s speed and frequency at which they are dropped.
11. Make the bombs fall at different angles rather than straight down.

Testing Procedure

During the construction of the game black box testing was done upon completing a function requirement, working towards the expected result. During the production of the game whilst coding in each aspect, testing was performed upon completion of the block of code for that function requirement. To test the new feature that had been coded to the game the developers tool in the browser was used, accessing the browsers console to view any problems that would be present in the entered code.

With the involvement of integer variables tools such as ‘console.log()’ were used to track the movement of the element on the page and the effect it had on the game. Testing certain aspects of the code without waiting for the correct conditions to be met was possible, for example in the earlier stages of the coding of the game to determine that bombs would spawn randomly and fall and explode on the grass, one bomb was being worked on only monitoring the outcome of just one bomb, and once all the required conditions were met the block of code would just be placed inside a loop creating multiple bombs already executing the expected outcomes, therefore increasing efficiency in the block of code.

The following table shows the black-box tests that were done once the game assignment was complete to evaluate the outcome of the game and debug any bugs within the final code.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test** | **Procedure** | **Expected Result** | **Result** | **Comment** |
| 1 | Start button disappears once clicked | Click the start button to make it disappear | Start button disappears from screen and cannot be clicked again. | Expected Result |  |
| 2 | Bombs randomly formed on top of screen and falling towards the player | Click the start button and have the bombs randomly falling from the top of the screen. | Once the start button is clicked bombs form at the top and start falling. | Expected Result | Bombs formed randomly with a delay of 2 seconds per bomb |
| 3 | Bombs drop at random angles, speed, and frequency | Click start button and have the bombs falling at different angles and speed | Bombs spawn at random positions falling at random angles and speeds | Expected Result | Bombs fall as expected, but with a bias to right direction due to the use of Math.floor() |
| 4 | Bombs explode at random heights on grass | Start the game and as the bombs drop observe if they explode at random heights | Explosion of the bombs occurs randomly on the grass | Expected Result |  |
| 5 | Player life deduction upon hit by a bomb | Have the character hit by a single bomb | should result in a life being deducted on the top right corner and a hit animation on the player | Once hit a life is deducted from the three lives, with the player displaying a hit animation in the down direction only | When hit by a bomb the hit animation for down direction is shown only. |
| 6 | Player dies when hit 3 times and all lives are gone | Player gets hit by a bomb 3 times, and on the 3rd hit displays the character as dead | On the 3rd hit the player dies displaying the dead animation with all the life circles gone. Unable to move the player after dead animation displayed. | Expected result |  |
| 7 | Display Game over and play again button after player dies | Get hit by bombs three times and make the character die | ‘Game over’ text is displayed on the screen and play again button shows up on the screen. | Expected Result |  |
| 8 | Play Again button restarts game | Click on the play again button when the character dies | The game restarts with all the elements of the game reset including the lives brought back. | The page refreshes and everything is reset but will start the game when start game is pressed. | This button refreshes the page allowing user to then click start to commence the game. |

**Bugs Discovered**

When the above tests were completed there were some bugs discovered with the code for the game, resulting in the game having some features not functioning at a 100%. The bugs that were discovered are as follows:

1. The movement of the player came already pre-coded as a base to commence the assignment. There was a bug found that the player could walk on to the sky beyond the grass if the up-direction button was held at the same time with either a left or right button.
2. With the player dying after losing all 3 lives all timers are set to stop, which led to one problem that would occasionally show up. If the player dies the same time a bomb explodes on the grass the interval that clears the bombs explosion will be stopped resulting in the explosion going on even though the game would’ve stopped.
3. One other problem that would show up at random intervals is the life deduction of the player when hit by a bomb. When the character was hit a life is deducted but during some of the test trials when hit by a single bomb two lives would be subtracted instead of one, resulting in a quicker death for the player.

Evaluation

With the game started the bombs fall randomly at random angles, also falling at random speeds and frequencies increasing the difficulty of the game from the basic requirement. The current outcome of the game has resulted in a functional version of the game, but if more time had been allocated some features that would be added are as follows. Having the bombs falling at random with random speeds and frequencies within the walls of the web browser and either having them exploding when they hit the walls on their way down or bouncing off the walls heading towards the grass section of the game.

Extending the game to add more functionality to the game would not be too much of a problem as most additions would be done into the interval that contains the main function that runs the game. Some features that could be added to the game if more time had ben permitted would be the scoring system to hold data about the high score and the score each player would have achieved. Another feature that would have been implemented is that of having levels to the bombs dropping with a set interval of about 10 bombs incremented with each level.

If a similar game had to be built in the future some features that would be different from the current version of the game are that of having a selection panel where the player can customise the appearance of their character before commencing the game. Another feature that could be added is that of having bombs of different sizes falling with the explosion radius being different depending on the size of the bomb.