



# CSY1063

## Web Development

### Week 13

[Chris.Rafferty@northampton.ac.uk](mailto:Chris.Rafferty@northampton.ac.uk)



# Learning Objectives

- This week we will be covering
  - How to submit the assignment
    - AS2 - Source Code Zip
    - AS2 - Source Code Word Document (Turnitin)
    - AS2 - Report Word Document
  - Revision of Weeks 8 - 12
  - Report writing
    - Referencing
    - Testing
    - Evaluation



# Video Demonstration

- A video going over the assignment submission can be found here
- [https://mymedia.northampton.ac.uk/media/CSY1063+AS2+Submission+Guide.mp4/0\\_5i1vnpyg](https://mymedia.northampton.ac.uk/media/CSY1063+AS2+Submission+Guide.mp4/0_5i1vnpyg)



# Recap W8 - W12

- The rubric for assignment two is broken up into five different sections
  - Functionality (60%)
  - Evaluation (10%)
  - Testing (10%)
  - Code Quality and Efficiency (15%)
  - Video Demonstration (5%)

# Functionality (D)

Criteria	Week
When the start button is pressed, the game should begin, and the start button should no longer be visible.	Week 8 & 9
The player should not be able to move until the start button has been pressed.	Week 9
Stop the player from walking through the maze's walls (blue squares).	Week 10
When the player touches a point (small white circle element), the point is removed from the maze.	Week 10
Update the score for every point the player collects (<p> tag in .score).	Week 8 - 10
A game-over message will appear when the player collects all the points.	Week 8
If the player touches an enemy (green circle element), a game-over message will appear, and the player will display the dead animation (css class .dead).	Week 9 & 10

# Functionality (C)

Criteria	Week
Randomise the position of enemies at the start of the game.	Week 11
Prevent the enemies from being created outside the maze and only allow them to appear where there is a free space (0 in the maze array)	Week 11
The enemies randomly move around in the maze. If they hit a wall, they should move in a new direction (they should not get stuck)	Week 9
Enemies stop moving when the game-over state has been reached	Week 9
Instead of the game-over message display "Restart?" to allow the player to restart the game if they lose, the game should reset back to its original state upon clicking restart	Week 11
Implement the arrow buttons. The player will continue moving in that direction when an arrow button is clicked. Only one arrow direction can be used at a time. The player should only move in that direction	Week 9

# Functionality (B)

Criteria	Week
At the end of the game, ask the player to enter their name (use a prompt()) and save their name along with their score using local storage	Week 12
Display the five highest player scores in the leaderboard (.leaderboard div)	Week 12
The leaderboard should be organised in order from the highest score to the lowest score	Week 12
Add the player's lives using JavaScript (not HTML) at the start of the game	Week 11
When the player collides with an enemy, remove a life instead of ending the game. The player should use the hit animation (.hit css class) and be unable to move for 1.5 seconds while the animation is being played	Week 10
If all three lives are lost, display the game-over/restart message	Week 10

# Functionality (A)

Criteria	Week
Add levels of increasing difficulty. Once the player has collected all the points in the game, a new maze layout is shown. The difficulty can be increased by adding new enemies for each level. For example, level 1 has one enemy, while level 5 has five. As the levels increase, so too should the difficulty	Week 11
Bonus marks are available if there are an infinite number of levels. These are not premade mazes; the maze layout is automatically randomised each time.	Week 11
Add two unique features to the game—the more complex the features, the higher the mark. For example, a power-up or a character customizer could be implemented. State what the additional features are in your report.	Week 8 - 13



# Report - Testing

Criteria	Week
How did you test that your code worked?	Week 13
Could you test certain aspects of the code without running the entire game and waiting for the correct condition to be met?	Week 13
What tests did you carry out and what were the outcomes?	Week 13
What bugs did you discover during testing?	Week 13

# Report - Evaluation

Criteria	Week
A list of known bugs/weaknesses in the game	Na
What works well?	Na
What improvements could be made?	Na
What else would you have done if you had more time?	Na
How easy would it be to extend the game to add more functionality?	Na
If you had to build a similar game in the future, what would you do differently and why?	Na



# Video demonstration

- You need to record a video demonstration for this assignment that includes the following:
  - Demonstrates multiple outcomes e.g. winning/losing
  - Demonstrates known bugs
- You can use OBS to record the video demonstration (it's free)
  - <https://obsproject.com/>
- To host the video use either Mediaspace (affiliated with the university) or YouTube
  - <https://mymedia.northampton.ac.uk/>
  - <https://www.youtube.com/>



# Report writing

- The report for the assignment is worth 20% of your overall grade
- 20% can make all the difference
- For report requires:
  - A checklist of completed functionality
  - Testing
  - Evaluation
- It's good practice to get in the habit of adding contents tables, page numbers and cover pages to any academic reports you produce during your time at the university

# Referencing

- An optional section to the report is for references
  - For this assignment you will need to reference any code that is not your own (or from the lectures)
- To do this you need to comment in the code and add a reference to your references section

```
//This code was taken from "https://developer.mozilla.org/en-US/docs/Web/JavaScript"  
function randomEnemy() {
```



# Harvard referencing

- The official referencing standard at the University of Northampton is Harvard
- While this is not something that is being accessed in the first year it is good practice to adopt this for the second and third years
- Harvard references typically look like
- MDN Contributors (2023). JavaScript. [online] MDN Web Docs. Available at: <https://developer.mozilla.org/en-US/docs/Web/javascript>

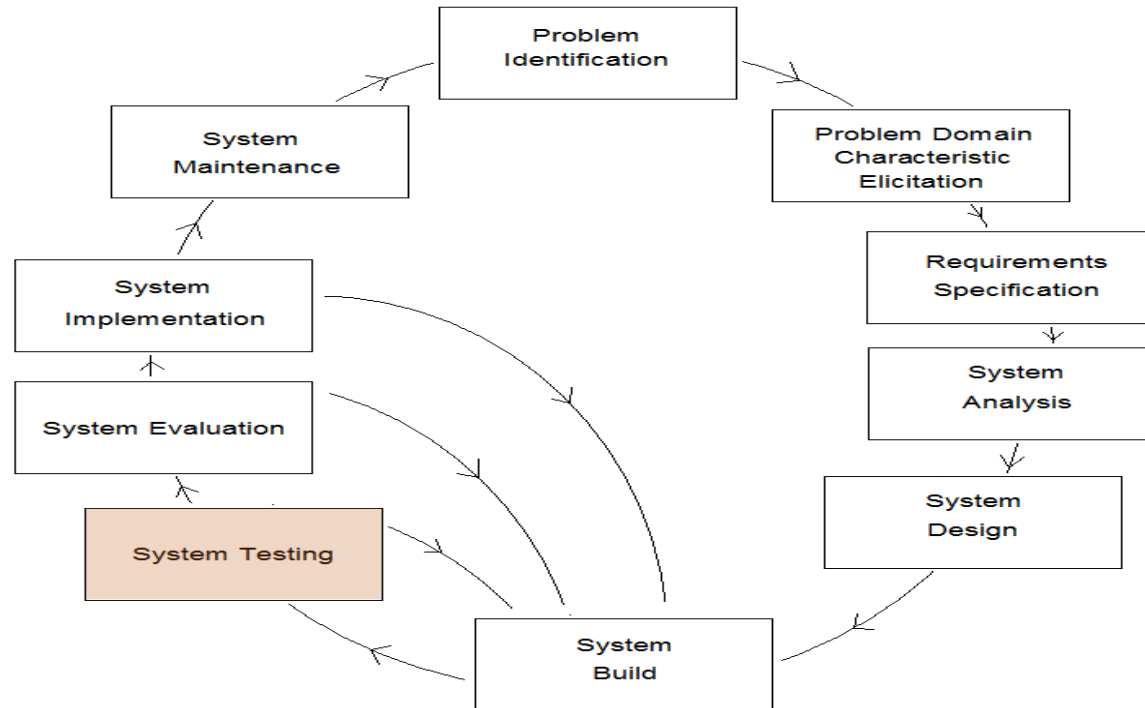


# Referencing tools

- There are a few tools that make referencing a lot easier
  - Word also provides some functionality for referencing
- MyBib is an online (free) option is <https://www.mybib.com/tools/harvard-referencing-generator>
- Zotero is more advanced (more for MSc/PhD) <https://www.zotero.org/>

# Testing

- Testing is an essential step in the software life cycle (CSY1064)







# Testing pt2

- Testing evaluates a system to ensure it meets the specified requirements and functions correctly
- It is an essential quality assurance measure aimed at identifying defects or bugs in the software, ensuring it is reliable, functional and usable
- The two categories of testing we will look at for this module are
  - Black-box testing
  - White-box testing

# Black-box testing

- Black-box testing is used to check a systems functionality ensuring it meets the requirements it aimed to resolve.
- In the assignment a requirement is the start button disappears when clicked
- Black-box testing for the start button could include

Requirement	Expected output	Actual output	Result	Notes
When the start button is pressed, the start button should no longer be visible	The start button should disappear when clicked	The user clicks start and the button disappears from view	Y	Test successful
When the start button is pressed, the game should begin	The player should not move until the start button has been pressed	The player could move before pressing the start button	N	Error in code which allows the player to move before game has started

# White-box testing

- White-box testing is used to test the actual code itself identifying any logical errors.
- The code for the start button involves a click event and finding the element by a class/id
- White-box testing for the start button code could include

Feature	Test	Result	Notes
Start button	Left clicking the start button should set the div to display: none;	Start div was set to display: none	Test successful
Start button	Right clicking the start button should not trigger the click event	Div did not disappear	Test successful
Start button	Using an external mouse to click the start button should set the div to display: none;	Start div was set to display: none	Test successful
Start button	Clicking the start button on a mobile device using a touch input should set the div to display: none;	Start div was set to display: none	Test successful



# Testing pt3

- To get the maximum number of marks for the testing section it should be clear you have attempted at least a few different tests for the assignment.
- For black-box testing you could test for each requirement you have implemented from the brief/checklist
  - Can a new name and score be saved in the leaderboard
- For white-box test the code itself from those requirements
  - What happens if you don't enter a name and click save for the leaderboard?



# Evaluation

- The evaluation needs to cover the following questions
  - A list of known bugs/weaknesses in the game
  - What works well?
  - What improvements could be made?
  - What else would you have done if you had more time?
  - How easy would it be to extend the game to add more functionality?
  - If you had to build a similar game in the future, what would you do differently and why?



# Report

- The recommended structure for your report
  - Cover page
  - Checklist of completed functionality
  - Testing
  - Evaluation
  - Video demonstration (URL)
  - References (optional)
- Including the video demonstration the report is worth 25% of your grade, don't leave it to the last minute!