**Digital Technologies & Hangarau Matihiko 3.8A**

**Level Three, Credits 6, Assessment Internal**

Introduction/Kupu Arataki

This assessment activity requires you to plan, develop and create a complex computer program.

You will be assessed on

* how effectively you use project management tools and techniques to plan and manage the development of a digital outcome
* how effectively you decompose the problem into smaller components, and test and refine your media outcome so that it is a high-quality response to the task
* how well you have addressed relevant implications
* how well you synthesise information from the planning, testing and trialling of components to develop a high-quality response to the task (e.g. well-structured, logical, flexible, robust and comprehensively tested program)
* discuss how this information assisted in the development of a high-quality outcome.

Problem Statement

I will create an interactive game of **battleship** played between two players. Each player will be able to place their ships at the start onto a 10x10 grid, each player has one grid. The ships being the 5 long carrier, 4 long battleship, 3 long destroyer and submarine and 2 long patrol boat. Once both players have placed their ships, they will then take turns to fire at a square in their opponents’ grid, not knowing where their ships are. The game will show whether it is a hit or a miss. Once all squares of a ship have been hit that ship will be sunk, once all ships of one player have been sunk the game is over and the player with the remaining ships is the winner.

SWOT Analysis

Conduct a SWOT analysis for the project management tools you are considering using for your project

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Strengths | Weaknesses | Opportunities | Threats |
| Version Control on GitHub | * Online, easy to access from multiple different computers, eg between school and home | * Have to upload zip files as opposed to uncompressed folder | * Allows for easy access to tools such as Kanban boards within the project setup | * GitHub servers could possibly go down / could lose internet connection |
| Version control with local files | * Easily manageable files all in one place | * Can’t transfer between computers as easily |  | * Hard drive / computer error could lead to losing files without backups anywhere else |
| AGILE | * Easy to change anything partway through the process if something doesn’t work or I get a new idea from something like feedback |  | * Chance to easily incorporate suggestions from feedback or completely redesign one little part if a user doesn’t like it |  |
| Waterfall | * Whole project comes together at once, instead of little things being made perfect at a time | * Harder to see where any problems might occur and could require a large redesign at the end |  |  |

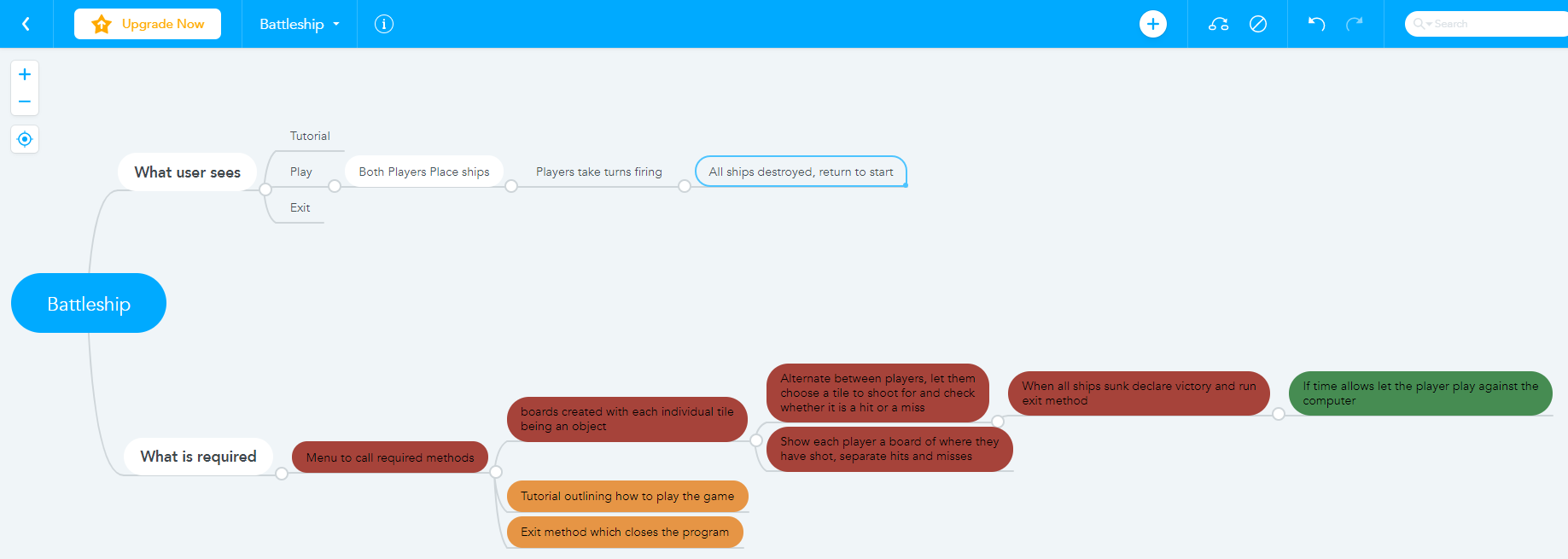
I will use github, this is because it will easily allow me to keep all work in one place easily. Code, planning, management etc. This will make it much easier to manage what I’ve done and still have left to do and track my progress. I will upload a zip folder to github for each time I work on the project

I will use AGILE techniques to help manage my project, this is because for a game like this I think it is better to decompose it into smaller pieces which I make work one at a time. AGILE methods will allow me to get the board working properly for example before I need to work on allowing users to shoot at each other. As users have to interact directly with my outcome it is important that it is simple to use, therefore I need to be able to incorporate user feedback and redesign anything that doesn’t work for them. Having my project split into smaller pieces means that I can easily use this feedback without losing too much time with wasted work having to delete a large section of what I have done.

Decomposing the outcome

* ***decomposing the digital technologies outcome into smaller components***

*Decompose your digital technologies outcome into smaller components. User stories is one method that is commonly used in an AGILE methodology*



Here is an initial decomposition of my project, red being the most important, orange middle and green least important. This will allow me to plan out my project properly and structure what I need to work on first. Which will help throughout my sprints.

My initial focus will be to get a barebones version of the project working as quickly as possible, and then tidy it up with the time I have left. I will plan my sprints with some time left at the end so I can have a final shorter sprint to thoroughly test the program and check if there’s anything that is not working properly.

Considering Relevant Implications

* ***addressing relevant implications.***

Usability

Usability implications are about how simple it is for the user to use the program. How the program communicates things to them and the ability for them to just use it without having to think too much. In this case it is important that they can learn how to play from a quick tutorial, and so that there is not too much text that it is not interesting. This is important so that they can enjoy the game and not become bored due to a difficult interface. As this would cause them to stop playing and that negates the entire purpose of designing it. To address this, I will make sure that there is a small amount of text, and any menus have clear options of what to input to do what the user wants to do. I will make sure that my GUI is well designed and clearly indicates what each button does and where to click on the graphics pane if I incorporate that. I will make sure that if I use multiple menus they are consistent throughout, as [CS Field Guide](https://csfieldguide.org.nz/en/chapters/human-computer-interaction/usability-heuristics/) says that “Consistency (something being the same every time) is extremely useful for people using interfaces”. This means that I will make it consistent within the program, as well as making navigation similar to other programs. This will mean that the user does not need to learn a whole new way of navigating through the program. It will also mean that they don’t have to overthink what they need to input and can just enjoy the game. I will also have simple error messages explaining the mistake so that they can quickly get back to playing the game if there is any point where error messages are required.

Functionality

Functionality implications are about how well the program performs the specified task. This includes things such as not having bugs and controlling errors. For this program it is important to make sure that it does not have any bugs and that it runs smoothly, and that any errors are picked up and controlled, not crashing the program. This is important so that the user can enjoy using it, as enjoying the game is the only reason people will play it. Which will mean they can play this version of battleship without the interruption of bugs, which is important as it is a common game and if my version is not enjoyable to play it is easy to find another place to play it. To address this, I will thoroughly test my program and pick up any errors as early as possible. I will test every type of input so that I know that it will handle any errors such as invalid input quickly and easily. I will also make sure to have good version control so that I can fall back on an old version if an issue develops, every time I begin to make changes to the code I will save it as a new file. I will also try as much as possible to contain input to buttons or clicking a certain spot so that there is no need for the user to input their own text which could have accidental and unexpected typos which could lead to errors.

Aesthetics

Aesthetics implications are about how the outcome looks, this means that I must make it look as good as possible to engage the user and make them want to keep playing the game. Usability heuristics involve aesthetics and minimalist design. Minimalist design means having as few options as possible, while still maintaining full functionality. An example of this in my program will be how I design the GUI. I need to make sure that there is enough information so that they can play the game. But also, not so much that they become overwhelmed with it all. For battleship it is important to take feedback from users on some different possible looks and then decide which option looks the best from this feedback. I think it is important to set it up using colour rather than just black and white. I will also try to make the design as simple as possible so there is not too much going on, and so that it can be run on any machine.

Sprint Tracking

|  |  |  |
| --- | --- | --- |
| **Sprint Number** | **Start Date** | **End Date** |
| 1 | 6/9/21 | 19/9/21 |

* ***using recognised and appropriate project management tools and techniques to plan the development of a digital technologies outcome***

# Planning

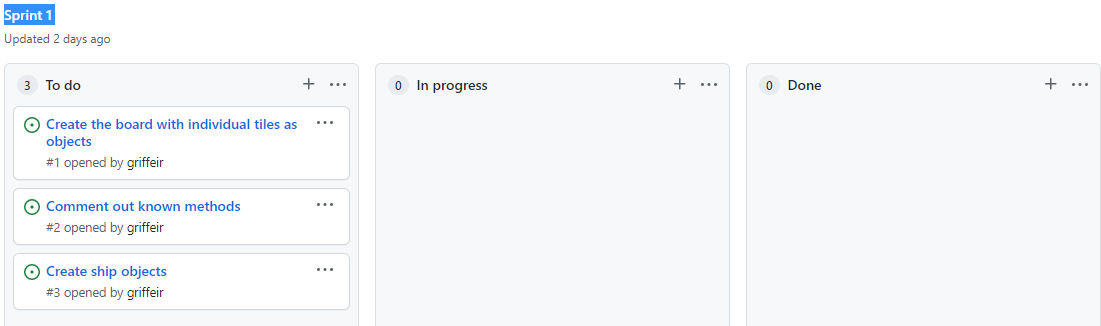
*What are you going to work on in this sprint?*

The main focus of my first sprint will be to set up the base of the project so that I have a strong foundation to build off of as I continue to work on my project. The main purpose will be to get the basic stuff functioning even if it doesn’t look perfect yet, I will then try to get feedback on how useable it is, rather than how it looks, as I will work on making it look nicer towards the end of the project.

To begin with I will create a plan of all necessary classes and methods and comment them out so I can easily work off of these later in the project. After this I will get the basic GUI elements set up so that I have an easy way to test the project as I go along. I will then figure out how to design the basic 10x10 board on which my project will be played on, this will involve creating tile objects and putting them in order to create the ‘board’ each of these tiles will be able to store whether a ship is on it, and whether it has been shot at. Alongside creating the board, I will also create the ship objects, these will have to be able to be stored across multiple tiles so I will have to figure out the best way to do this. Especially the placing of them when the user is setting up their ships will be difficult. I will not be placing them on the board this sprint.

I believe the board is the best thing to get working first as it is the core of my project, everything else will need to use it. So if I create it as early as possible I will then have time to make sure it works perfectly and design everything else so that it works with how my board is designed. This will also allow me to follow AGILE techniques well as it means if any requirements change, such as feedback from end users I can change the core of my project to meet the requirements earlier rather than later. For example, if I received feedback saying the board had an error I hadn’t tested that required the whole board to be changed then I could rework this early without having to get rid of all of the rest of the work I had done as well.

*Provide evidence (screenshot / photo) of your project management tool(s) being used to plan the development of your outcome at the beginning of your sprint here*



I have used a Kanban board to track progress. I will use one for each sprint, it will allow me to easily track how far I am through the sprint by moving notes between the columns. Each problem I need to work through will be made into an issue that is able to be moved between the columns

# Development

*What components are you going to trial?*

I will be trialling the main board. I am currently trialling what happens when I print the board and then click a square. Currently this is as if you were firing upon this square, I will add as if you were placing a ship later. I will be trialling different options for the size of my tiles. I think this is important because it relates to aesthetics implications, as well as usability. As if they are too small, they can be too hard to see what is meant to be clicked. As well as being hard to tell apart for people with impaired vision. However, if they are too big, they may not display in the smaller popup window and will need to be full screened. Which is an extra step some people will not want to take

Option 1: Option 2: Option 3:

Chart

Description automatically generatedChart, bar chart

Description automatically generated Chart

Description automatically generated

*Provide evidence (screenshot) of your version control*

## Feedback

|  |  |  |  |
| --- | --- | --- | --- |
| **Component** | Board display | | |
| **Name** | Will Turner | **Date** | 18/9/21 |
| **Feedback** | I think option 1 is the best option. It is a nice mix between the too sizes, the other ones being too small or too large.  Clicking a specific tile works well. However, when clicked and the square changes colour the black border should still show around the new colour. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Component** | Board display | | |
| **Name** | Alex Lubberink | **Date** | 19/9/21 |
| **Feedback** | I like the red on a hit, although when clicking tiles all together you can lose where the tiles are. Especially if option 2 is chosen and the tiles are that small. Therefore, I think option 1 is the best as in 3 the tiles are too big, and having to full screen to see a proper display is frustrating | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Component** | Board Display | | |
| **Name** | Tom Dominy | **Date** | 20/9/21 |
| **Feedback** | I think this looks alright, however there should be a different colour for when you miss a ship as opposed to when you hit it. I also agree that option 1 is the best here as it is the easiest to use, as you don’t have to enlarge it, and the best looking. | | |

*What is the outcome of this feedback?*

Using this feedback I know that I need to change how my tiles display. From now on I will make it display a black border around any tile which I change the colour of. This will make it a lot more obvious where tiles start and end and will look a lot better. I will also add in a different colour to display when the player misses a ship opposed to when they hit it.

# Testing

***Brief Description of what you are testing***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Type of test**  **(E, B, I)** | **Method to Test** | **Value(s) to enter** | **Expected result** | **Actual result (screen snip / time stamp)** | **Comments/ changes needed** |
| I | doMouse | Click outside the grid | Nothing happens |  | It is trying to check which square was clicked instead of first checking whether it was even in the grid. I will add a checker to see where they clicked and ignore it if outside the grid. |
| B |  | Click right on the edge of the grid |  |  | This works now for most places outside the grid, however, if you click perfectly on the edge of the grid it still causes an error |
|  |  |  |  |  | This works now, I added a try, catch statement to make sure this error doesn’t break the program. |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# Evaluation

*Sprint reflection and summary*

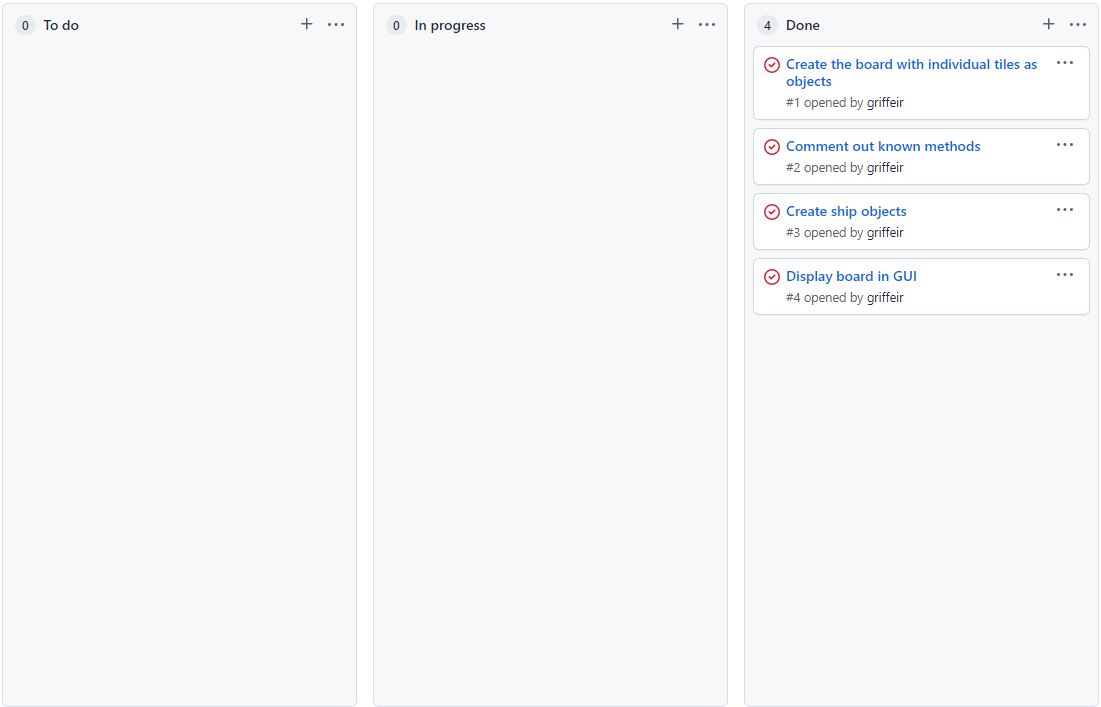
In this sprint I achieved the main part of what I wanted to, I got the board to work as a 2d array with tiles as objects. I also managed to print this board out in the GUI as a grid. If you click a tile it registers which tile was clicked. I now need to process whether they were placing a ship or trying to shoot one.

However, I did not get much further than this, I didn’t comment out all the methods I hoped to because I was focused on figuring out how I was going to get my board and ships working so I didn’t know exactly what to comment out. Overall, I think this was a successful sprint, however I think I will need to focus on getting more work done over the next sprint so that I have time in my final sprint to finish anything that I haven’t noticed until late in the development.

*What major changes and achievements did you complete in this sprint?*

The main major achievement this sprint was getting the board created with tiles in each position. I set each tiles x, y coordinates to its position in the board, and managed to get them to print out as a 10X10 board in the GUI. I also set it so that the user can click a specific tile and it will register which tile object has been referenced.

*Provide evidence (screenshot / photo) of your project management tool(s) being used to manage the development of your outcome at the end of your sprint here*



Sprint Tracking

|  |  |  |
| --- | --- | --- |
| **Sprint Number** | **Start Date** | **End Date** |
| 2 | 20/9/21 | 3/10/21 |

* ***using recognised and appropriate project management tools and techniques to plan the development of a digital technologies outcome***

# Planning

*What are you going to work on in this sprint?*

The main focus of this sprint is to allow ships to be added to the board. I will make the user click a tile and then press a key to choose which direction they want the ship to face. I will need to make sure that the ships can't go outside the grid and that they can't overlap another ship.

This is an important step as I need to have ships placed to be able to work on anything else further on in my code. After ships are able to be placed I can then work on being able to alternate turns and sink ships and everything else to finish the game.

*Provide evidence (screenshot / photo) of your project management tool(s) being used to plan the development of your outcome at the beginning of your sprint here*

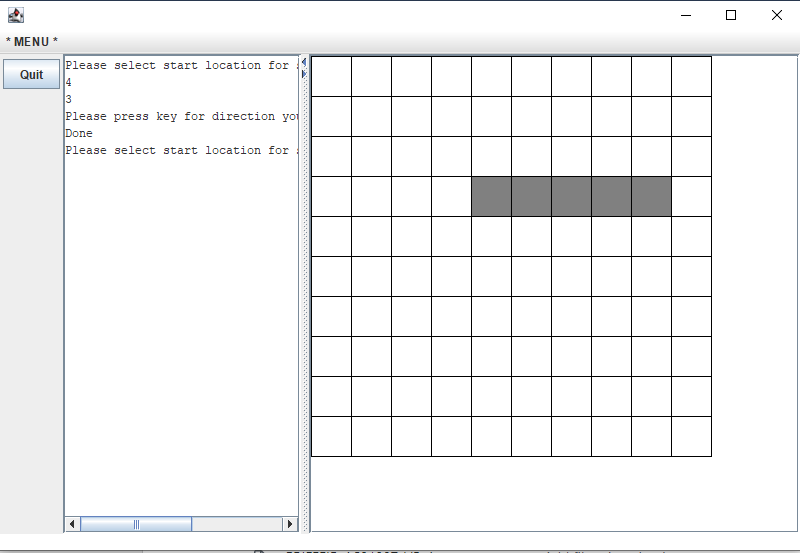
Graphical user interface, text, application

Description automatically generated

# Development

*What components are you going to trial?*

Placing ships – Press key for direction:



Placing ships – Click first and final tile:

*Provide evidence (screenshot) of your version control*

Table

Description automatically generated

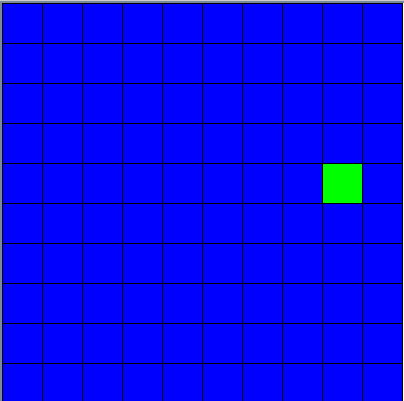
## Feedback

|  |  |  |  |
| --- | --- | --- | --- |
| **Component** | Ship placement | | |
| **Name** | Will Turner | **Date** | 29/9/21 |
| **Feedback** | Ship placement is easy, I think that pressing a key for the direction to place in is better than clicking final tile. I think the only change necessary is to make it display the first tile you click in a different way. Although it would be good to let the user know that the keypress to place a ship is w, a, s, d instead of arrow keys. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Component** | Ship placement | | |
| **Name** | Alex Lubberink | **Date** | 29/9/21 |
| **Feedback** | I think the grey looks fine, but I think the surrounding sea should be blue to make it look more interesting. However, I think there should be a quick tutorial or message explaining what keys need to be pressed. | | |

*What is the outcome of this feedback?*

I now know that my ship placement works well, but I will change the colour of the first tile you click so that it’s easier to see. I will make this tile pop up as green, and then be coloured grey when the ship has been placed. I will also make the surrounding sea blue, and when I have added ships I will make a different colour for hitting them or the sea.



This now displays as a green square when pressing the first tile, letting the user know that they have started placing a ship. The surrounding ocean also displays as blue now, which makes it more interesting to look at and more fun to play.

I have also added a help button, this button checks which phase of the game you are in, either placing ships or shooting at opponent’s ships, then prints out a message saying what they should do in that scenario.

# Testing

***Brief Description of what you are testing***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Type of test**  **(E, B, I)** | **Method to Test** | **Value(s) to enter** | **Expected result** | **Actual result (screen snip / time stamp)** | **Comments/ changes needed** |
|  |  | Pressing a key which would put ship out of board, as well as every direction |  |  | This is working for every ship other than against the left and right edge where constraint is >= 0. This is because of cords starting from 0 and ship length being 1 greater than the coord. e.g. coord (0,3) finalY would be -1 so just needed to add 1 to it |
|  |  | Pressing a key that would cause a ship to be placed inside another ship |  |  | This is working for placing a ship upwards. However, I need to get it working for other directions. |
|  |  |  |  |  | This now works, I loop through each tile ship will be on and make sure it doesn’t already have a ship. If one tile does ship doesn’t place |
|  |  | Clicking a different tile while waiting for key press | Don’t let them place there |  | This does not work yet.  I made a Boolean which sets to true after the first tile has been clicked. My mouse listener will check to make sure this is false before recording the click position |

# Evaluation

*Sprint reflection and summary*

This was a successful sprint, I think I managed my time well and got everything I needed to complete completed. This leaves a good amount of work still to do, but definitely not too much to handle, so my final sprint won’t be too busy.

My next steps are to use the board and the ships I created in the last two sprints and make the base battleship game work. This will involve switching between players turns and letting them place ships or shoot at the oppositions board trying to hit a ship.

*What major changes and achievements did you complete in this sprint?*

The major achievement I made this sprint was ship placement, I can now place ships on the board, which is the core for the rest of the project from here, so this was very important. I also added a help button which allows the user to click it and get a message detailing what to do from there. This is important because it helps to address usability implications. Because if the user didn’t know what to do and couldn’t figure it out the game wouldn’t be accessible for everyone.

*Provide evidence (screenshot / photo) of your project management tool(s) being used to manage the development of your outcome at the end of your sprint here*

*Graphical user interface, text, application

Description automatically generated*

Sprint Tracking

|  |  |  |
| --- | --- | --- |
| **Sprint Number** | **Start Date** | **End Date** |
| 3 | 4/10/21 | 17/10/21 |

* ***using recognised and appropriate project management tools and techniques to plan the development of a digital technologies outcome***

# Planning

*What are you going to work on in this sprint?*

My focus of this sprint will be to finish the game, I need to get the whole game working before the end of this project. So my main focus will be to get the core game working. Once I have done this, if I have time remaining, I will work on making the game as smooth and user friendly as possible.

*Provide evidence (screenshot / photo) of your project management tool(s) being used to plan the development of your outcome at the beginning of your sprint here*

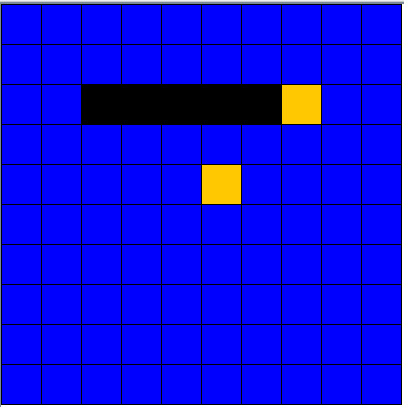
Graphical user interface, application

Description automatically generated

# Development

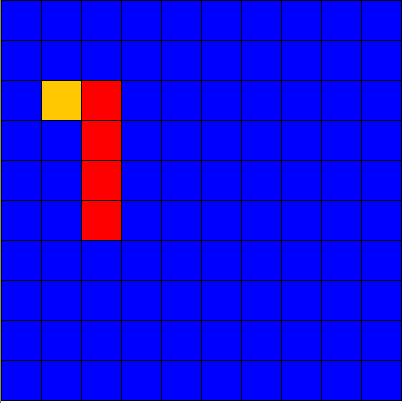
*What components are you going to trial?*

I will trial what happens when you hit a ship/sea/sink a ship



Sinking a ship causes it to turn black, ship hit = red, miss = orange

Or:



Sinking a ship just keeps it all as red, prints out a message in the text pane, rest is the same

*Provide evidence (screenshot) of your version control*

*Table

Description automatically generated with medium confidence*

## Feedback

|  |  |  |  |
| --- | --- | --- | --- |
| **Component** | Hit display | | |
| **Name** | Will Turner | **Date** | 9/10/21 |
| **Feedback** | I think both of these options look good, as the colours being added makes it more engaging.  However, option 1 is definitely the better option. As the ship changing colour when sunk is more engaging, and also helps the user to see whether or not they have to keep trying to sink this. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Component** | Hit display | | |
| **Name** | Min-Sung Jung | **Date** | 10/10/21 |
| **Feedback** | In my opinion, both examples deliver the message clearly, but I would go with the first option because making the ship go black when it sinks makes it suddenly clear that you have destroyed a ship. But I also think that a text message should also printed out, possibly giving the name of the ship that has been sunk. | | |

*What is the outcome of this feedback?*

Using this feedback I know that changing a ship to appear as black when sunk is the best option. A reason for this is that when playing a game, a sudden visual change when you’ve completed some task always gives a rush. Therefore, when the ship suddenly changes to black it keeps the user much more engaged than if it were to just stay the same colour and print out some text. However, I will also include a text message in the text pane, as this means I can include the ships name.

Chart, histogram

Description automatically generated

# Testing

***Brief Description of what you are testing***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Type of test**  **(E, B, I)** | **Method to Test** | **Value(s) to enter** | **Expected result** | **Actual result (screen snip / time stamp)** | **Comments/ changes needed** |
| E | hitShip() | Clicking the leftmost tile last | Paint ship black and set each tiles ship to null |  | This does not work, this is because I’m checking the other tiles ships against the one in the leftmost tile. Therefore, when this ship is null the others do not equal it so do not get deleted. I need to store the ship name in a variable before starting to delete things. |
|  |  |  |  |  | This works now. |
| E | hitShip | Hitting every ship | Every ship is black and print Congratulations, you have won |  | This works properly |
| E | tileClicked | Clicking a ship which has already been hit | Tell them that it has already been hit |  | This does not work properly yet. Because I have the switch turn call outside of the if statement. This means it switches turns even if the player hits a tile which they have already hit |
| B | placeShips | Clicking to place a ship on a tile that already |  |  | This does not work properly, I need to make sure that they cannot select a tile that already has a ship on it.  I fixed this by adding a check after the user clicks so that it won’t be processed as starting to place a ship if there is already a ship there |
|  | placeShips | Click in a place where the ship cannot be placed in any direction due to edge of board/other ships |  |  | This does not work, I need to make it so the user is able to click a new starting tile if they want to.  This is working now, I just draw over the green with blue and switch the starting tile |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# Evaluation

*Sprint reflection and summary*

This sprint has gone very well, I have finished my game in the time I had. It is working well and I have tested for all errors that I can find. I managed my time well and finished the bulk of the work early enough to get good feedback and test everything I needed to.

I am happy with how this final sprint tailored to end users’ needs, with feedback from possible users changing some plans I had made. I believe this final sprint has addresses implications well.

*What major changes and achievements did you complete in this sprint?*

This sprint was all about finishing the game, I added in the process for alternating between players, created a second board for the second player. Players can place their own ships and shoot at the other player’s board without any issues. I then processed the players shots and checked whether they hit a ship or not. If they did, I checked whether this shot would sink a ship. If it did, then I checked to see if it was their opponents final ship, if it was, they won the game.

*Provide evidence (screenshot / photo) of your project management tool(s) being used to manage the development of your outcome at the end of your sprint here*

*Graphical user interface, application

Description automatically generated*

Project Summary

* ***addressing relevant implications.***

*How did you address the relevant implications in the development of this outcome?*

For my project I decided that aesthetics, functionality, and usability implications were the most important.

Functionality implications were important because the game must work properly to be of any interest to anyone. I needed to make sure that there were no errors and that nothing could break the game. To address this, I made sure to test every possible input to make sure there were not any fatal errors that could occur. I also made sure to use github for good version control so that if I had an unfixable error then I could fall back on a previous version without losing too much time. By controlling these factors, I have managed to develop a very functional response with no chance of anything such as invalid input breaking the program. I have made sure that the user is not able to do something to break the code, and if they do I have tried to put a message detailing what they should actually be doing, as well as making sure the error doesn’t stop anything from working.

Usability implications were important because I wanted my game to be accessible for anyone. And if it had been too complex to use, or just not clear how to play, then it would not have been playable for a lot of people. Therefore, to address this I made sure that I was checking every change with end users so that I could specifically develop it for them. I also made sure that the program was laid out in a way that was easy to understand what was happening. I have also added a help button which gives advice for how to use the game based off what stage they are at. This means that there’s not point at which the user won’t know how to play, which makes it far more usable.

Aesthetics implications were important

* ***synthesising information gained from the planning, testing and trialling of components***

*How did you use the tools, techniques and process of each sprint inform the development of this outcome?*

* ***discussing how this information led to the development of a high-quality digital technologies outcome.***

*How did the process help to shape the development of your outcome? Provide evidence.*