Group 41 Semester 2 - PDF Report

‘Keep Cool’

Design Documentation - Documented Design of Text User Interface

Main Menu interface

Upon starting the game , the user will have three options to choose from:
1.) Start Game
2.) Instructions
3.) Exit
The user will enter the choice in the form of an integer input from the keyboard.
The MainMenu can be seen below in Figure1:

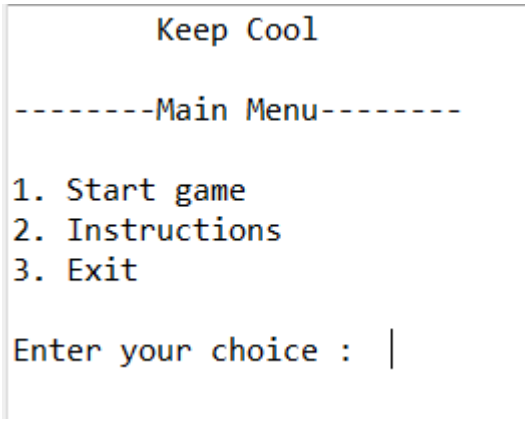


Figure1

We chose this kind of set up for the Main Menu because it was simple and easy to use. We wanted the game to appeal to a wide audience, including those not technology-oriented. Therefore, we ensured the first menu system the user observes does not have an exhaustive list of options to choose from. One of our main goals from the outset was for the game to be simple and easy to use and the first menu should set the tone for this.

Instructions Manual/Menu

It was vital for us when designing the Main Menu system to include an instruction manual at the beginning of the game. This allows the user to read about the background of the game and to get some insight as to what is in store for them. The instructions can also be viewed during the game, at the start of each turn. The instructions options will display a further menu where the user can select the particular area they want to view the instructions for. We decided to have a sub-menu for the instructions as it gives the user the option to choose a section relevant to their problem, rather than displaying the entire instruction manual each time. This would take up a large section of the console up, taking away from the game itself, something we did not want to do. The first image below is of the menu for the instructions from which the player chooses their option. The second image is an example of what is displayed to the user when they choose an option (in this case information about the ‘Go’ square).

Figure2

```
Enter your choice : 2

Keep Cool is a game about the protection of the environment. Within this game the player must
navigate through a board of various biomes, purchasing wildlife sanctuaries and collecting Eco Points to
fuel their eco-friendly escapades. The goal is to gather Eco Points while taking them from other players.
The game ends when the only one player has Eco Points remaining.

INSTRUCTIONS
What would you like to find out more about?

1) Character Selection
2) Beginning the Game
3) Go Square
4) Natural Disaster Square
5) Opportunity Square
6) Airport Square
7) Whale Square
8) Purchasing Wildlife Sanctuaries
9) Going to the Whale
10) Running Out of Funds
11) End Game
12) EXIT INSTRUCTIONS

Enter your choice : 5
The go square is slightly different to all the other squares on the board as the player does not need to
land on the square for the effect to apply. If the player passes over the go square then that player will be
granted 150 eco points.

Would you like to continue reading more instructions? (0 to proceed):
```

As you can see from above, a brief description of the game is printed to the user describing the general synopsis of Keep Cool. The options displayed to the user are displayed and when the user chooses the section relevant to their problem, only information relating to this issue is displayed. As you can see above, the player selects option number 5 and the relevant information regarding this square is displayed. The player then has the option to view other instructions or to continue with the game.

Start Game

When the user starts the game, they are prompted to enter how many players will be playing and to then enter the name for each of the players. After this the first player will be prompted telling them it is their turn. This is all shown in Figure3:

Figure3

```
--------Main Menu--------

1. Start game
2. Instructions
3. Exit

Enter your choice : 1
Enter number of players (2-8) :
Player 1 Enter name : Harry
Player 2 Enter name : Declan
It is Harry's turn.
```

In Figure3, the user starts the game, by choosing '1' from the Main Menu, and then enters that there are two players and subsequently enters the names "Harry" and "Declan".

Player turn

The user's position at the start of the turn will be displayed in the console, in this case, "You are currently on Start". The user is then given a series of options as described in Figure4:

Figure4

```
It is Harry's turn.
You are currently on Start
What will you do?

1) Roll Dice
2) View Owned Land
3) View Player Stats
4) View Board
5) View Instructions
6) Quit Game

Enter your choice :
```

Again, we chose a menu system for the user to point them in the direction of the various possibilities available to them. We believed having a menu-system created a way of displaying the options to the user in a clean and easy to follow manner. The menu is displayed to each player at the start of their turn. Additionally, at the start of each turn the player's position on the board is displayed, allowing them to track where they are on the board.

Rolling Dice

The first option the player has is to roll the dice. Upon entering "1", the roll dice option is carried out and the following is displayed as shown in Figure5. The number rolled is shown ('3), and the player's updated position on the board is then printed enabling the user to follow where they are on the board ("Sea 2"). The player's current balance is also shown and the user has the opportunity to purchase the square if it has not already been purchased. We have made sure to print the players name each time something significant happens within the game to ensure traceability within the game.

Figure5

```
Enter your choice : 1
You have rolled: 3
You are now on Sea 2
Your current balance is: 1500 points.
Do you want to purchase for 100 points? (Type '0' to purchase):
```

We chose the above layout to make it easy for the user to follow what is going on and to enjoy the game. Each piece of feedback is displayed to the user on a new line to make it easier to read. At the end of each player's turn, the following information (what they rolled, their position on the board and current balance is displayed), enabling the player to keep track of their progress within the game. This menu system is displayed to the user upon the start of their turn for the duration of the game, ensuring consistency.

Purchasing a Square

If the user (in this case Harry) decides to purchase the current square, he does so by entering '0'. If the user has sufficient funds, the purchase will be successful and a message is displayed to the user letting them know which square they just bought. The players updated balance will then be displayed.

As you can tell in Figure6, once the turn for the current player is over, it automatically prompts the user telling them whose turn it is now and displays the menu once again. If the next player chooses to roll again the same cycle as described will occur until the player lands on a 'Special' square (Chance Card, Jail etc).

Figure6a

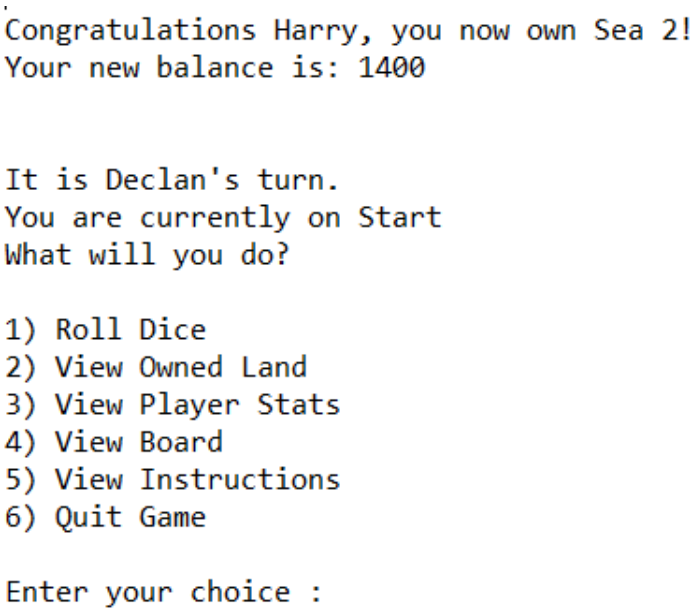
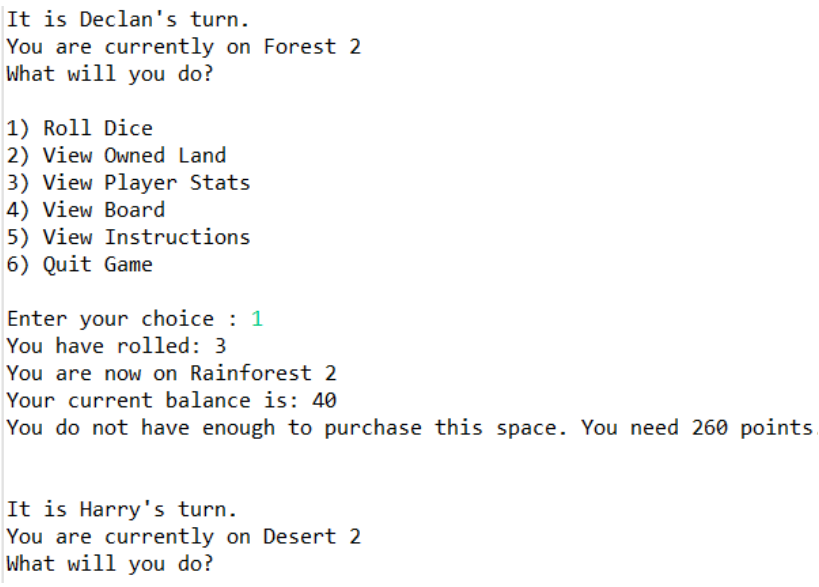


Figure 6b



The above image, *Figure 6b* shows what occurs if a player lands on a square that is available for purchase however, the player does not have sufficient funds to purchase the square in question. As you can see we output the square the player lands on and their current balance, however it is also displayed that they do not have enough eco-points to purchase the square and the amount required is also stated. If this occurs, the game will move onto the next player's turn - in this case Harry's.

View Owned Land

The next menu option is to view the owned land. When it is any players turn they may choose the option 2 from the menu to view the land/squares which they currently own. In the case shown below it is Harry's turn and he decides to view his own land. The output can be seen below in Figure7 where "Sea 2" is returned as this is the only land Harry owns.

Figure 7

```
Enter your choice : 2

Sea 2
```

Viewing player stats

A player during any given turn can choose to 'View player stats'. Upon choosing this option the user is given the option to choose from the current players and view their respective stats. Upon choosing the player of their choice, the stats of the chosen player are displayed. Stats includes, the current balance and the list of properties owned. This is shown in Figure8a and Figure8b respectively.

We included the 'View player stats' option in the menu as a way to keep track of how yourself and the other players are doing in the game. We recognised that since the game is entirely console-based, it can become confusing and hard to keep track of who owns what properties/how many eco-points other players have left etc. We included this function to enable these stats to be checked by any player on their turn and hence to help reduce such confusion.

Figure 8a

```
It is Harry's turn.
You are currently on Sea 2
What will you do?

1) Roll Dice
2) View Owned Land
3) View Player Stats
4) View Board
5) View Instructions
6) Quit Game

Enter your choice : 3

Which player?

1) Harry
2) Declan

Enter your choice : 2
```

Figure 8b

```
Enter your choice : 2
Declan's current balance is: 1380
Declan's list of properties is:

Ocean 1


It is Harry's turn.
You are currently on Sea 2
What will you do?

1) Roll Dice
2) View Owned Land
3) View Player Stats
4) View Board
5) View Instructions
6) Quit Game

Enter your choice :
```

Viewing the board

The player can choose to view the board (choosing option 4) to view the square they are currently on and the next 12 squares ahead of them (limited it to 12 squares so that choosing this option did not display information that takes up the entire console and the player can move a maximum of 12 squares from one roll). The price associated with each square is also displayed, giving the players more information.

We decided to implement the view board functionality all on one line so as to not cluster up the console which we worried would take away from the core functionality of the game. We thought this would be a very helpful feature for the players as it allows them to visualise the board whilst playing the game but never having to leave the console. It also allows them to see where the 'action' squares are located relative to their current position (such as Airport and Opportunity Squares).

The output for the following choice can be seen below in Figure9.

Figure9

```
It is Harry's turn.
You are currently on Sea 2
What will you do?

1) Roll Dice
2) View Owned Land
3) View Player Stats
4) View Board
5) View Instructions
6) Quit Game

Enter your choice : 4
Sea 2(100)      Opportunity     Ocean 1(120)    Ocean 2(100)    Ocean 3(130)    Airport 1       Plains 1(140)   Plains 2(150)   Plains 3(160)   Opportunity     Desert 1(170)   Desert 2(180)   Desert 3(200)
   /\
You are here.
```

Viewing the instructions mid-game

The player also has an option as mentioned above under the instructions section to view the instructions during the game. The player can choose to do this during their turn at any point in the game and if they choose to do so, the same menu as documented and described in *Figure2* is displayed. We decided to give the user the ability to view the instructions not only before starting the game, but also during it in case they encounter any queries when playing the game. Our aim is that this functionality will help improve the player's experience if and when a question arises by answering it for them.

Landing on an opportunity square

When the player lands on an Opportunity square, this is displayed in the console along with a corresponding message about the scenario. The Opportunity squares vary in the actions they cause and each time, the outcome is relayed to the player. Our aim of implementing variability (RNG) to the game was to keep the player guessing what will happen if they indeed land on an Opportunity square and keep them interested. In *FigureX* below Harry rolls and lands on Opportunity and the scenario at hand ("You planted many new trees for the planet") is printed in the console. There will be some kind of action that occurs after landing on the Opportunity square, in this case the player gains 150 eco-points. The player's updated balance is displayed. This can be seen in *Figure 10* below:

Figure10

```
Enter your choice : 1
You have rolled: 3
You are now on Opportunity
Congratulations! You planted many new trees for the planet.
Gain 150 points!
Your new balance is: 1550
```

We aimed to

introduce variability using the Opportunity square and an element of ‘surprise’. In the case above, only one outcome of landing on an Opportunity square is shown, however when playing the game the outcome can either be positive (like above where the player gains eco-points) or negative (the player will experience a negative outcome, for example, losing eco-points). With the variability associated with this square we made sure to lay out what happens upon landing on a square such as this very clear. Firstly, the player is told they have landed on this special square, then text is printed letting the player know what this particular Opportunity square entails and finally, the outcome is printed along with the user's updated balance.

Landing on a square that is already owned by another player

If a player rolls and lands on a square already owned by another player, the game will tell the user what square they are currently on and which other player owns it. The player who landed on this square has eco-points deducted from their total, with their new total displayed. Furthermore, the player who owns the square gains these eco-points and their updated balance is also displayed. This can be observed below in *Figure11* where Declan rolls and lands on Ocean3, a square already owned by the other player Harry.

Figure11

```
It is Declan's turn.
You are currently on Start
What will you do?

1) Roll dice
2) View 'not properties'
3) View player stats
4) Quit game

Enter your choice : 1
You have rolled: 7
You are now on Ocean 3
Owned by: Harry
Declan's new balance: 1370 points.
Harry's new balance: 1500 points.


It is Harry's turn.
You are currently on Ocean 3
What will you do?

1) Roll dice
2) View 'not properties'
3) View player stats
4) Quit game

Enter your choice :
```

When implementing this feature of landing on an already owned property, we made sure to print the updated balance for the two players involved so they can keep up to date with their balance. This feature of our game provides an incentive to buy properties.

Passing ‘Go’

When the user makes a successful trip around the board and passes the ‘Go’ square, the user is rewarded with 150 eco-points. Whenever this takes place, the player is made aware of their new position (in this case it is ‘Go’ itself but could be any square after ‘Go’ and they gain 150 points. This can be seen in *Figure12* where Harry rolls an 11 and as a result moves to the ‘Go’ square from ‘Rainforest 1’.

Figure12

```
It is Harry's turn.
You are currently on Rainforest 1
What will you do?

1) Roll dice
2) View 'not properties'
3) View player stats
4) Quit game

Enter your choice : 1
You have rolled: 11
Go! Collect 150 points.
You are now on Start


It is Declan's turn.
You are currently on Opportunity
What will you do?
```

‘Whale’ square

When the player rolls the dice and is moved to the ‘Whale’ square, they have two options. They can either choose to miss their turn and will be free to mov for their next turn. The second option they have is they can pay 150 eco-points to escape the ‘Whale’ and are moved forwards. We thought giving the player an option here would introduce more variability into the game and further depth. For the choice we used a simple menu system which is easy to read and is clear for the user how to use it. The two options are shown below in *Figure 13:*

Figure13

```
It is Declan's turn.
You are currently on Plains 2
What will you do?

1) Roll Dice
2) View Owned Land
3) View Player Stats
4) View Board
5) View Instructions
6) Quit Game

Enter your choice : 1
You have rolled: 6
You are now on Whale
You are inside the whale!
1.) Miss current turn.
2.) Pay 150 Eco Points?
Please enter 1 or 2:
1
Missed current turn
You are still trapped in the Whale


It is Harry's turn.
```

```
It is Harry's turn.
You are currently on Plains 2
What will you do?

1) Roll Dice
2) View Owned Land
3) View Player Stats
4) View Board
5) View Instructions
6) Quit Game

Enter your choice : 1
You have rolled: 6
You are now on Whale
You are inside the whale!
1.) Miss current turn.
2.) Pay 150 Eco Points?
Please enter 1 or 2:
2
You paid 150 eco points to escape the Whale.
Your current balance is: 810


It is Declan's turn.
```

Purchasing a Sanctuary

When a player lands on a square that they have already purchased they will have the option of purchasing a sanctuary to place on this square. The player will be told as usual what square they have landed on and their balance is printed. The player is then presented with the option to purchase a sanctuary. They can make this decision based on their balance and other factors. If they choose to purchase the sanctuary, it will state the land involved now has a sanctuary on it and how much the sanctuary is worth on top of the cost of the square. When a player lands on this square from now on the price associated with landing on it will have increased by 10% from the original cost. This can be observed in *Figure14* below. Up to 3 sanctuaries can be purchased on any one

square, increasing the cost of other players landing on the square by 50% of the original price. Therefore, purchasing sanctuaries is an efficient way at taking over the board and draining your opponents eco-point balances. We believe this is a straightforward and easy to understand way of introducing another aspect to the game that adds another dimension.

Figure14

```
It is Harry's turn.
You are currently on Opportunity
What will you do?

1) Roll Dice
2) View Owned Land
3) View Player Stats
4) View Board
5) View Instructions
6) Quit Game

Enter your choice : 1
You have rolled: 3
You are now on Ocean 3
Owned by: Harry
Your current balance is: 1420 points.
Do you want to purchase a sanctuary for this space for 100 points? (Type '0' to purchase):
0
Ocean 3 now has 1 sanctuaries and is worth 39 points!
Your new balance is: 1320 points.
```

‘Disaster’ square

Another type of special action square within our game is the ‘Disaster’ square. There are two of these on the board and when a player lands on these something negative happens. The player is made aware they have landed on this square and like the Opportunity square, there is a degree of randomness and each time a player lands on one of these squares a different scenario is printed out to them via the console. The outcome of each particular landing on the square is also printed. An example of when a player lands on a ‘Disaster’ square is shown below. In this example the player lands on the square and is told that “Lightning has struck a huge tree down!” and the player has to pay a 150 eco-point fine. The disaster square functionality can be seen below in *Figure 15:*

Figure15

```
Enter your choice : 1
You have rolled: 7
Go! Collect 150 points.
You are now on Natural Disaster
Disaster! Lightning has struck a huge tree down!
Declan pays a fine of 150 points for cost of restoration.


It is Harry's turn.
```

Exiting the game at any stage

At any time during the game, the user can choose to exit and end the game by choosing number 4 on the menu displayed. If there are only two players remaining, the person who is remaining after the person quits in the winner. The last player remaining is the winner and their winning total of eco-points is displayed. In the figure below, Declan had 1350 points and Harry had 1400 eco-points. Harry in this instance of the game was the player who quit and now Declan is the last player remaining and is hence the winner. As you can see the total points Declan won with is displayed and a thank you message is also displayed to the winner and their friends for playing Keep Cool.
This can be seen in *Figure16* below.

Figure16

```
It is Harry 's turn.
You are currently on Sea 3
What will you do?

1) Roll dice
2) View 'not properties'
3) View player stats
4) Quit game

Enter your choice : 4
Declan wins with 1350 points!
To Declan and friends thanks for playing Keep Cool!
```

Winning the game via playthrough (one player with eco-points left)

If the game is down to the last two players and one of them runs out of eco-points, the player with eco-points remaining is crowned the winner. The console will display that the user who ran out of points is now bankrupt and that the other player is the winner with their total eco-points. We chose this way of ending the game, as it is clear who wins and for what reason. Additionally, feedback regarding the total points tally is given, introducing an aspect of replayability where the players can play again to attempt to win by a larger margin. A thank you message for playing will be shown similar to that when the game ends via players quitting. This can be seen in *Figure17* below:

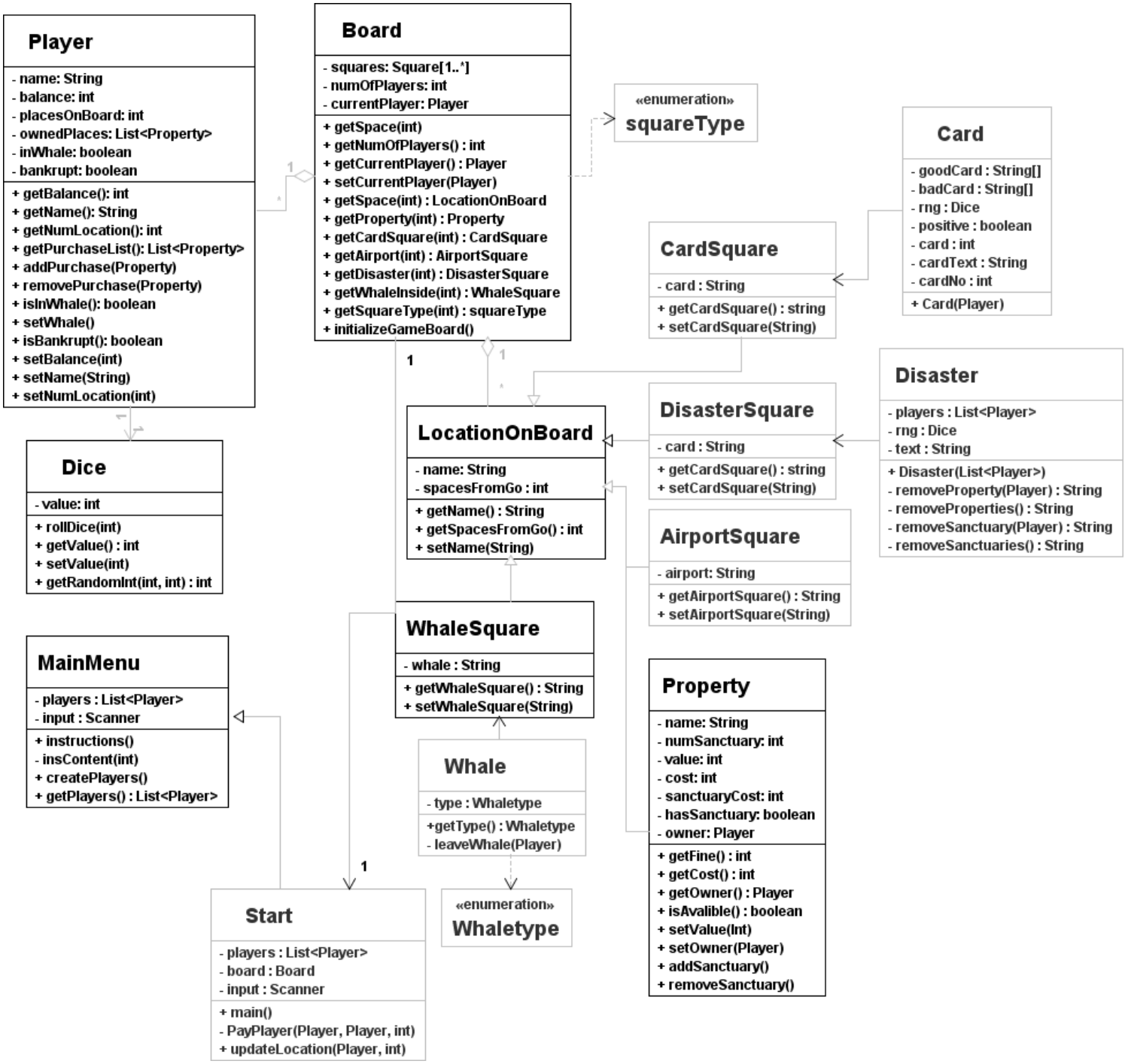
Figure17

```
It is Declan's turn.
You are currently on Sea 2
What will you do?

1) Roll dice
2) View 'not properties'
3) View player stats
4) Quit game

Enter your choice : 1
You have rolled: 12
You are now on Desert 2
Owned by: Harry
Declan's new balance: 0 points.
Harry's new balance: 1270 points.
Declan is bankrupt!
Harry wins with 1270 points!
To Harry and friends thanks for playing Keep Cool!
```

Value-added Features within Keep Cool

We have added different value-added features to enhance the players experience whilst playing our game ‘Keep Cool’. As a group we believe we have produced the required functionality in a manner that demonstrates excellent usability. This has been documented in the above section ‘Documented Design of the Text-User Interface’ where sample prompts and feedback that is relayed to the player is highlighted. Details on this can be read above, however one of the features we have included to demonstrate excellent usability is the menu system of the game. This feature allows the player to choose from a range of options each time it is their turn in the game. The menu gives the player the opportunity to roll the dice, view their own properties, view the properties and stats of the other players, view the board (next 12 squares) and exit the game. The fact the user has a choice in what to do next, gives them control and enables the playing of the game in a smooth and timely manner.
We believe we have developed a novel and coherent Back-from-the-Brink theme based on Monopoly. We have taken the basic features of Monopoly (buying properties) and replaced these with climate- change related squares belonging to different biomes across the world. The aim of our game is to invest in different biomes and increase the eco-points you have. Sanctuaries can be bought on the biomes which further improve the climate and hence increase the player’s eco-points. There are however negative squares which can reduce eco-points, these include the ‘Whale’ square and the ‘Opportunity’ squares. The ‘Whale’ square is based off the ‘Jail’ in Monopoly and rather than chance cards we have ‘Opportunity’ cards. These cards cause the player who lands on them to experience something unique and climate-based within the game. As in Monopoly, the last player with points wins the game.
Within the game there are varied action squares. These can come under the ‘Opportunity’ squares. When landing on these there is a chance that the result will be positive or negative for the user. When a player lands on one of these squares, an accompanying short sentence will be displayed to the user, telling them the outcome of what has happened. Some of these outcomes will cause the user to gain points and some to lose points. The probability of these is not equal, introducing some variability into the game for the user. This means that the game can be particularly punishing for certain players and not as punishing for others. The sanctuaries also introduce more variation in the squares. When a square becomes a sanctuary, the price other players have to pay if they land on them is increased. The ‘Airport’, ‘Whale’ and ‘Disaster’ squares are also action areas that have unique functionality within the game. For example, the ‘Airport’ square requires a fee to be paid and the user is transported somewhere around the board. the ‘D
There are special squares within ‘Keep Cool’ relevant to the climate theme of the game. These have been mentioned above, and include the ‘Airport’, ‘Whale’ and ‘Disaster’ squares. These all vary from the squares that are seen within the classic Monopoly game and are unique to ‘Keep Cool’. These are relevant to the theme with all three having some impact on the state of events within the game. For example, the ‘Airport’ square requires a fee to be paid (as flying is bad for the climate) and the player is transported somewhere around the board. The ‘Disaster’ square is a special square which when landed on has a chance of causing the player to lose some sanctuary and/or properties.
Surprise costs are also a part of ‘Keep Cool’. There are expected costs for purchasing squares and for landing on another player's already owned square. However, there are additional surprise costs for example, landing on an ‘Opportunity’ square can entail a cost (or a gain in points), the ‘Airport’ square can also entail a hidden cost. The ‘Natural Disaster’ square can be very detrimental to the player and have an indirect cost due to losing properties or sanctuaries.
When the game ends, either by players quitting or by a player winning, within the console there is feedback, congratulating the winner and thanking the players for playing the game.

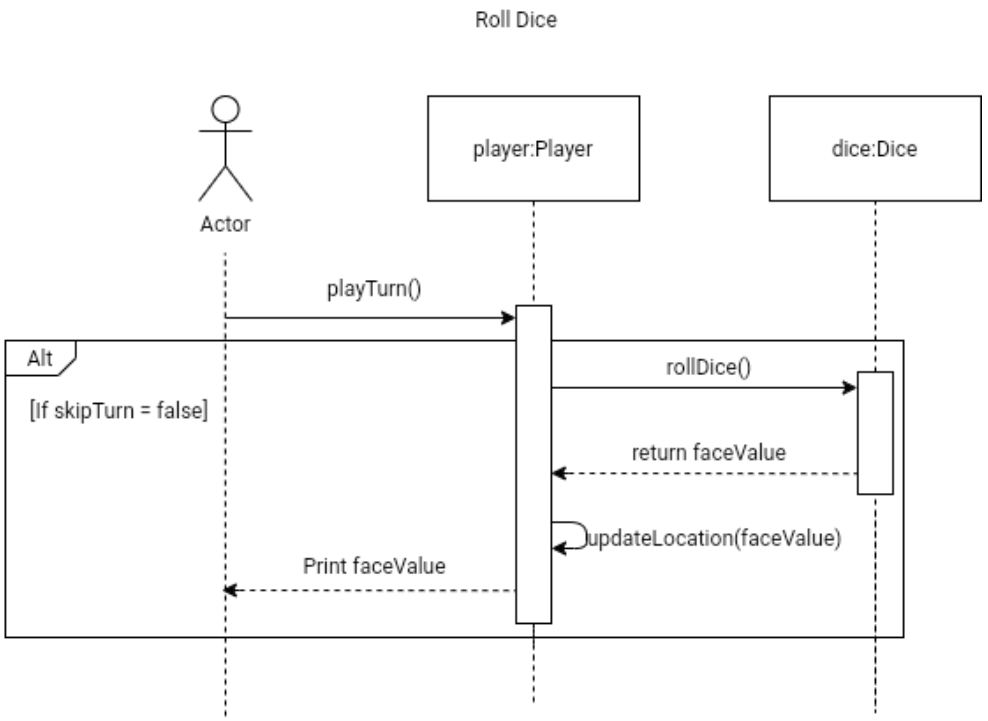
Class Relationship Model:



This is the class diagram, in Unified Modeling Language (UML), that describes the structure of our game system. It shows the system's classes, their attributes, operations, and the relationships among objects.

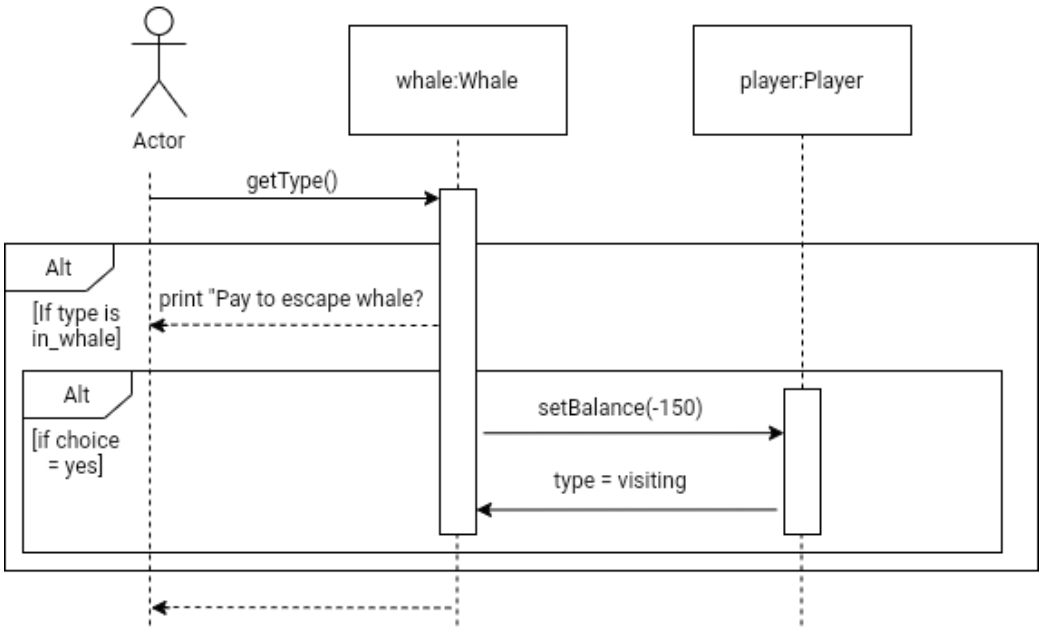
Sequence Diagrams:

Roll Dice



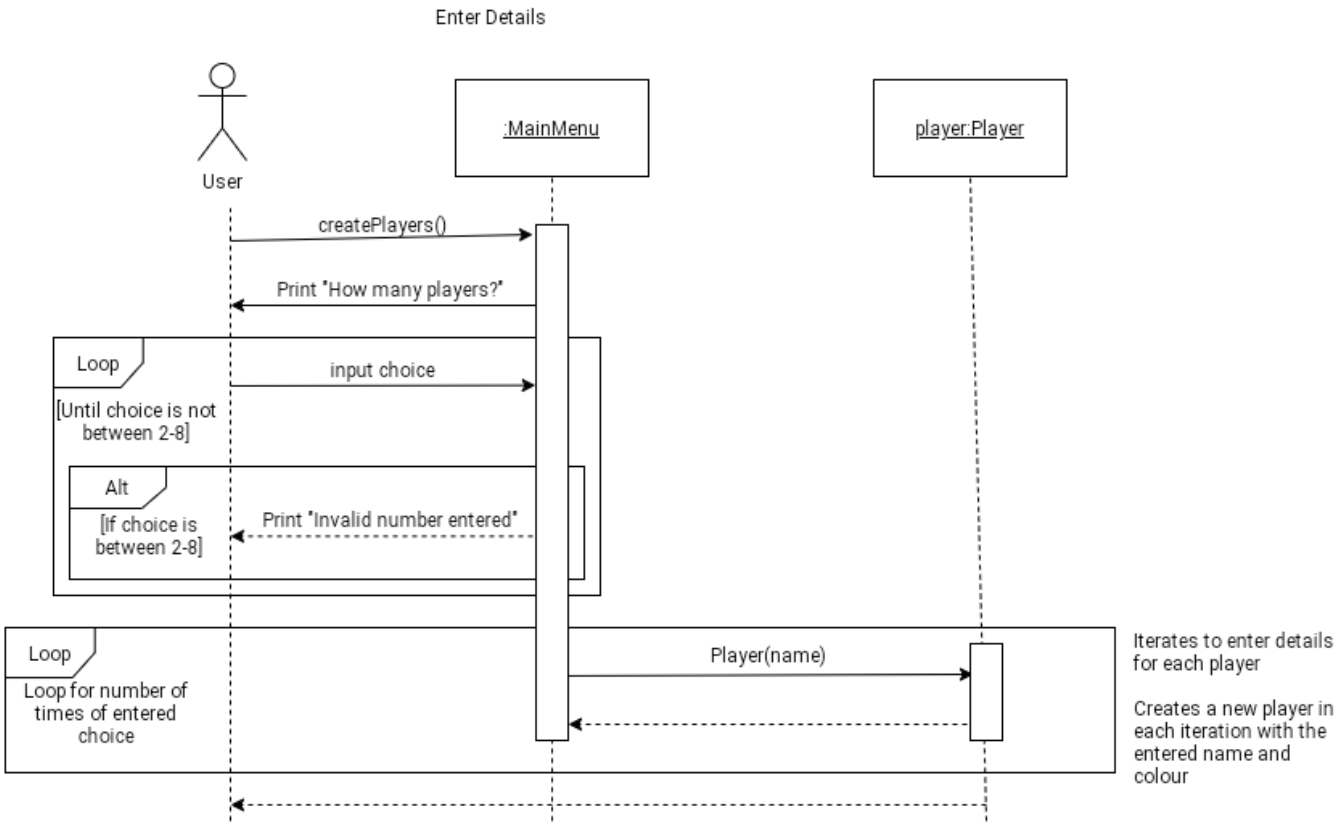
This is the Roll Dice sequence diagram showing the objects involved in the operation and the sequence of messages exchanged between the various objects needed to perform the functionality of the operation. Here, when it is the player's turn the rollDice() method is called, the number rolled is returned and the updateLocation() method is used, updating the player's location on the board. The outcome of the roll of the dice is then printed for the player to see.

Get out of Whale



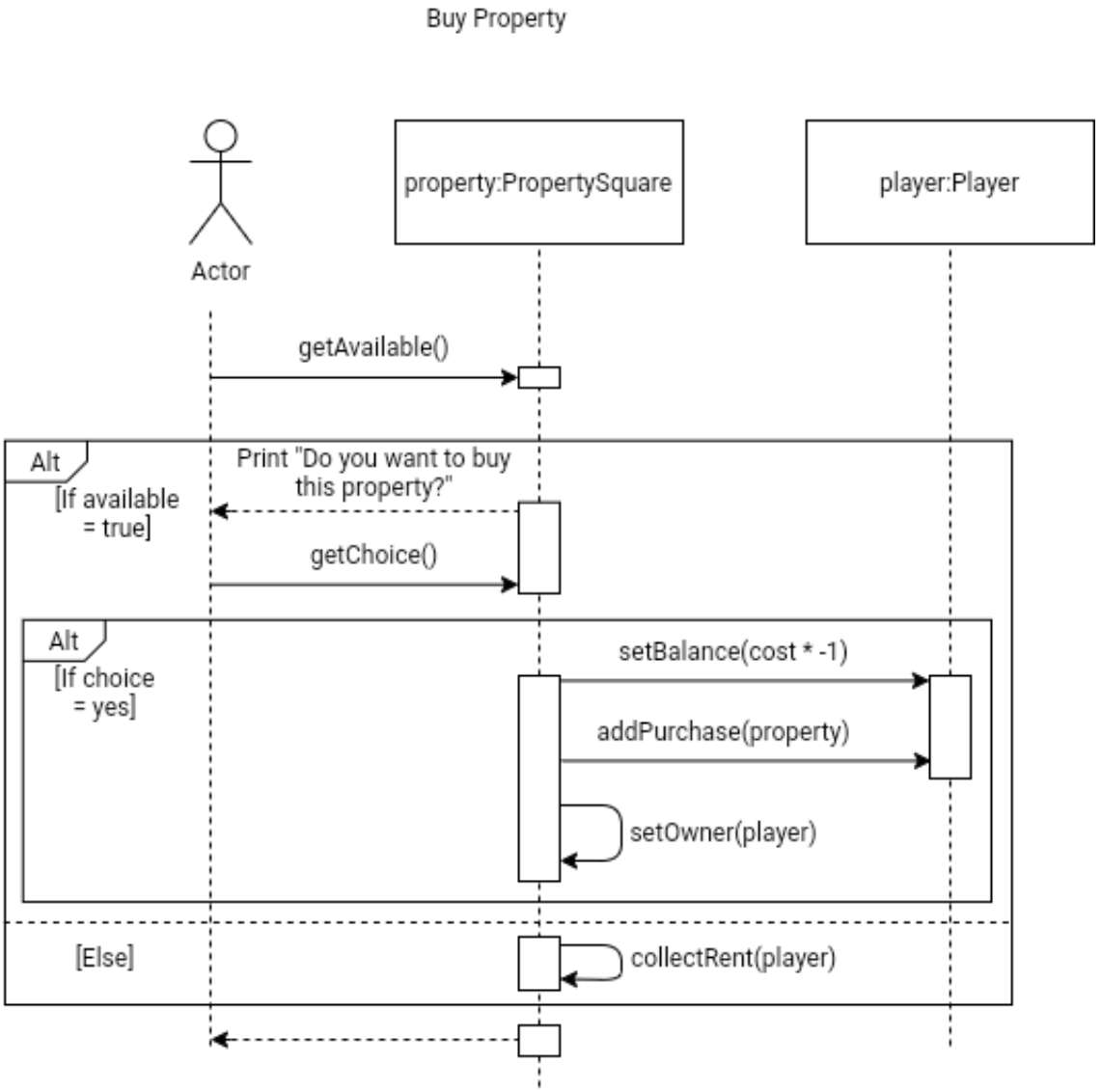
The Get out of Whale sequence diagram above shows the objects and messages involved when the player goes to the Whale square. When the player lands on this square, they have the option to either pay to escape the Whale or they can refuse to pay and instead miss a go. If they choose to pay the fine, 150 eco-points is deducted from their balance.

Enter Details



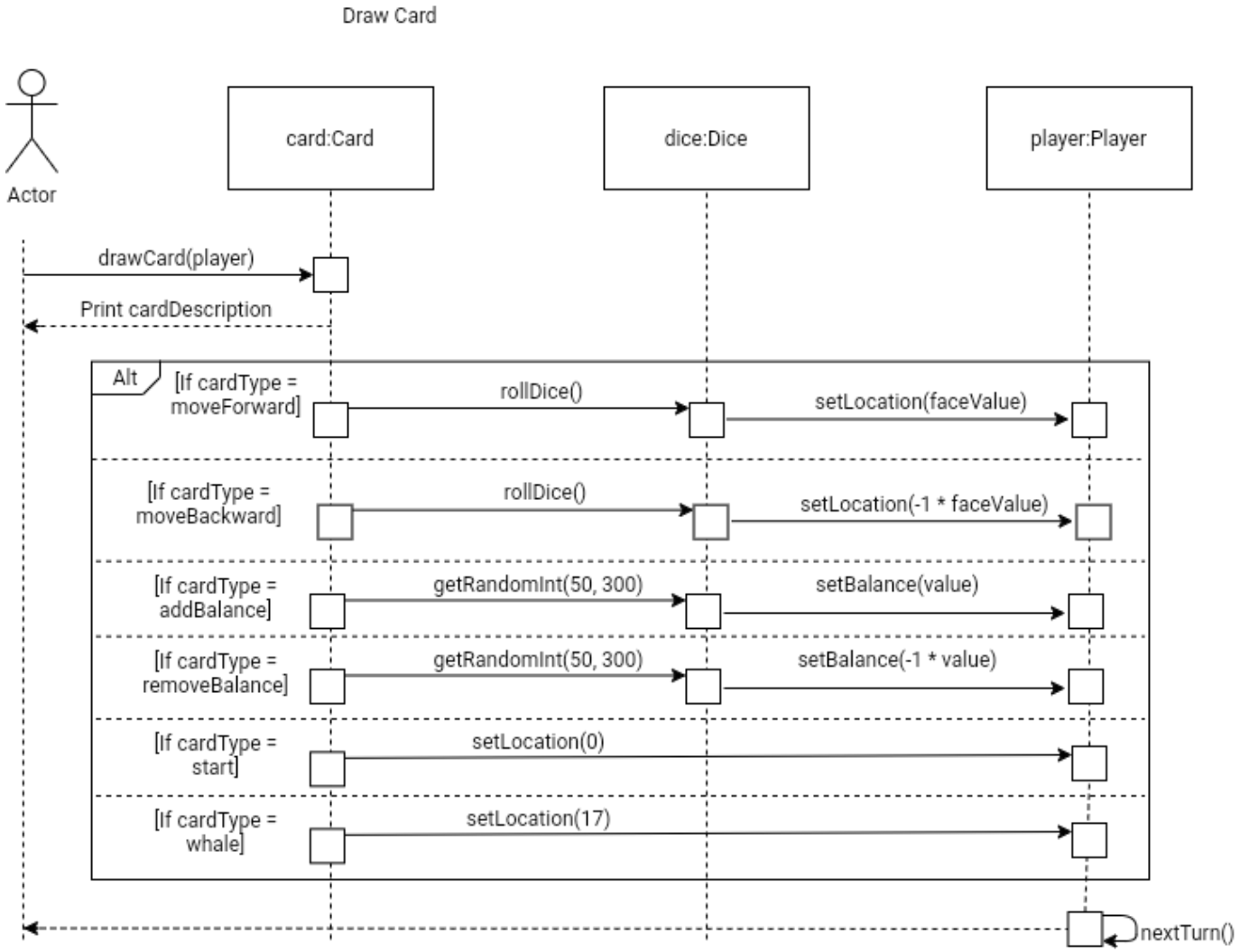
The Enter Details sequence diagram shows the objects and messages involved when the player enters their details in the game. Initially, the user executes the createPlayer() method where they choose the number of players (there is a loop here that executes until the user enters an appropriate integer choice between 2-8). Depending on the number of players another loop executes (one for each player in the game) where the user then enters the details of each player on each iteration.

Buy Property



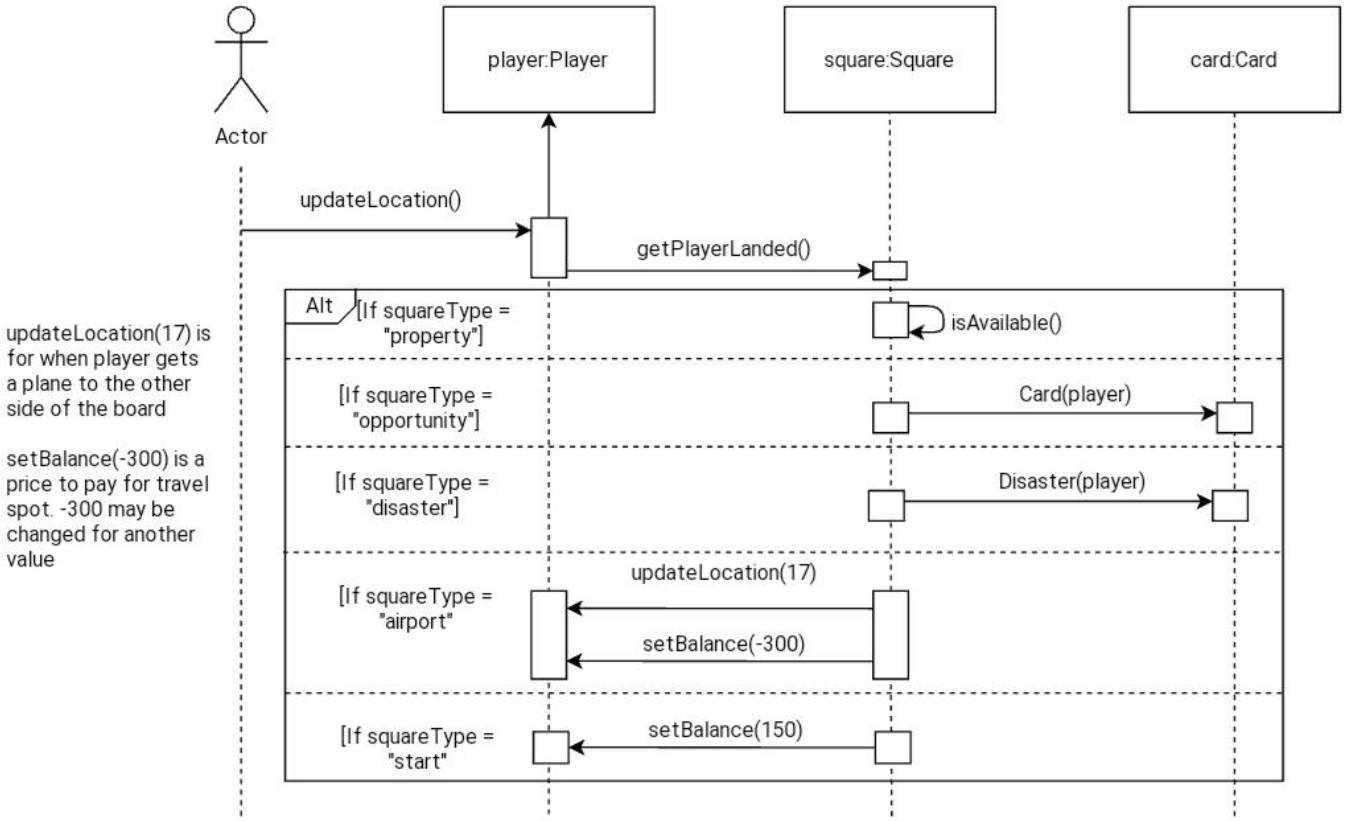
Property sequence diagram above shows the objects and messages involved when the player purchases a square on the board. Firstly, the getAvailable() method is called and if this returns that the square is available for purchase the user is given the choice as to whether they wish to purchase the square or not. getChoice() is used to gather the response from the player. If the user chooses to purchase the square, the price of the square is deducted from the players balance and the properties added to the players list of owned properties using the addPurchase(property) method. The owner of this square is then set using setOwner(player). On the other hand, if the square is not available to purchase, the collectRent(player) method is called which deducts the rent value of the property from the balance of the player who landed on the square.

Draw Card



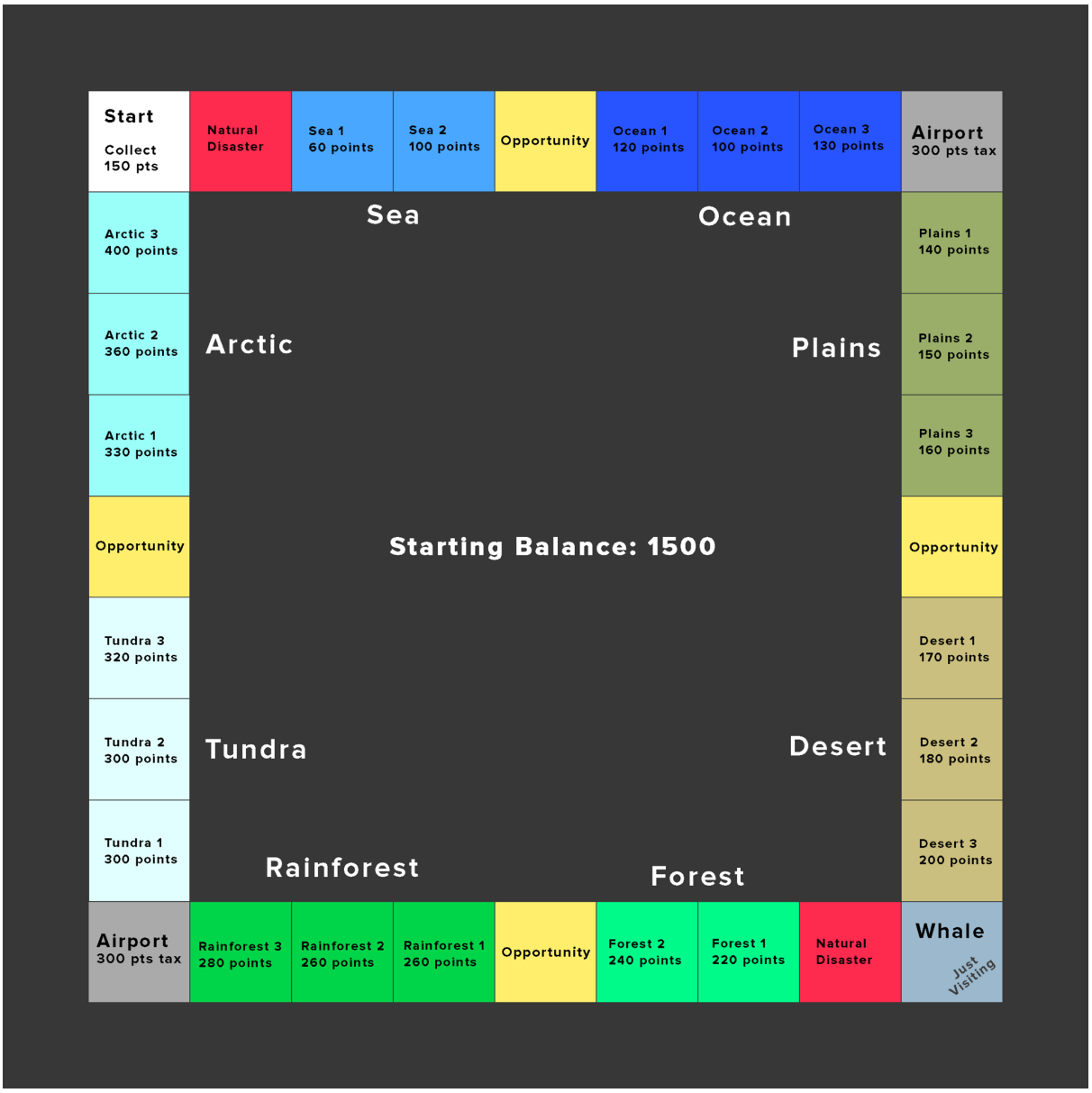
The Draw Card sequence diagram above shows the objects and messages involved when the player lands on one of the squares on the board which draws a card. The drawCard(player) method is called and the cardDescription is printed. There are many different variations of cards that can be drawn (cardType). It can be a moveForward type where the rollDice() method is executed and the player's location is updated using setLocation(faceValue). It could also be a moveBackward card type where the same thing happens however, the player moves backwards spaces using setLocation(-1* faceValue). The cardType could be addBalance or removeBalance where the getRandomInt(50, 300) method is called returning a random integer between these values and the setBalance(value) and setBalance(-1 * value) methods are used to update the balance of the player. The cardType can be start where the setLocation(0) method is called and the player is moved to the start square on the board. Finally, the cardType can be whale, where the setLocation(17) method is executed and the player is moved to the whale square. After this the nextTurn() method is called, moving the game onto the next player.

Move



The Move sequence diagram above shows the objects and messages involved when the player moves on the board. Upon the updateLocation() method being called, the player's location on the board is updated. If the squareType is property and the getPlayerLanded() method is called the system then checks if the square is available using isAvailable(). If the squareType is an opportunity the Card(player) is used. Likewise if the squareType is a disaster, the Disaster(player) method is used. If the squareType is airport updateLocation(17) is called and the setBalance(-300) is called as the airport square costs 300 eco-points. Lastly, if the squareType is a start the setBalance(150) is called as the player passes 'Go' and therefore they gain 150 eco-points.

Final Game Layout:



Above is our draft game layout where all of the squares within our game is shown. You can see the different biomes - Sea, Ocean, Plains, Desert, Forest, Rainforest, Tundra and Arctic. Additionally, we have the Start, Natural Disaster, Opportunity, Airport and Whale squares. These are all different 'Action Areas' which when the player lands on these different outcomes occur. Some can reward the player with eco-points (positive outcome), others will have a negative effect on the player and deduct eco-points. Some squares (Opportunity) can have either a positive or negative effect - there is a degree of randomness within the game to keep the user interested.

Test Plan Snippet (See Appendix for the full test plan)

	A	B	C	D	E	F	G	H	I	J
1	**Test case ID**	**Objective**	**Preconditions**	**Test data**	**Expected Result**	**Test completion date**	**Status**	**Tester**	**Defect ID**	**Comments**
2	TCase_1	Check if the code runs without any errors	N/A	N/A	The code successfully executed without errors	3/8/2021	Passed	Joel Donaldson		
3	TCase_2	Check if the exit option on the main menu ends the code	User must be on the main menu	3	The code will successfully end without error	3/8/2021	Passed	Joel Donaldson		
4	TCase_3	Check if the instructions options on the menu correctly opens the instructions dialogue	User must be on the main menu	2	When 2 is entered into the console the user wil be navigated to the instructions menu menu	3/8/2021	Passed	Joel Donaldson		
5	TCase_4	Check if the game initiates the character creation section when start game is selected	User must be on the main menu	1	When 1 is entered into the console the user will be taken to the character creation menu	3/8/2021	Passed	Joel Donaldson		
6	TCase_5	Check if the code responds correctly to the player entering 1 as the number of players in the game	User must be on the character creation menu	1	When 1 is entered into the console the user will be informed to enter a valid number of players instead	3/8/2021	Passed	Joel Donaldson		
7	TCase_6	Check if the code responds correctly to the player entering a number over 8 as the number of players in the game	User must be on the character creation menu	12	When 12 is entered into the console the user will be informed to enter a valid number of players instead	3/8/2021	Passed	Joel Donaldson		
8	TCase_7	Check if the code responds correctly to the player entering a number between 2 and 8 inclusive as the number of players in the game	User must be on the character creation menu	4	When 4 is entered into the console the user will be taken to the character naming section of the character creation menu	3/8/2021	Passed	Matthew Creighton		

Above you can see a snippet of the test plan (see full test plan in the appendix). There is a Test Case ID for each individual test and the object of the test is stated in each case. The preconditions for each test case is shown, along with the test data. The expected result from the test is displayed along with the test completion data, whether the test passed or failed and who the test was carried out by. There is also a column for DefectID and any comments for if the test failed. In the above snippet of the test plan these two columns were not required, however if you look at the full test plan there are examples where these columns are used.

Regarding the test plan we tried to test a wide range of features within Keep Cool to ensure smooth play through for users and to reduce the number of bugs within the game. Additionally, on top of the test plan shown. JUnit testing was performed on certain aspects of the core functionality. Evidence of this testing is displayed within the appendix of the report.

Appendix

Full test-plan:

Test case ID	Objective	Preconditions	Test data	Expected Result	Test completion date	Status	Tester	Defect ID	Comments
TCase_1	Check if the code runs without any errors	N/A	N/A	The code successfully executed without errors	3/8/2021	Passed	Joel Donaldson		
TCase_2	Check if the exit option on the main menu ends the code	User must be on the main menu	3	The code will successfully end without error	3/8/2021	Passed	Joel Donaldson		
TCase_3	Check if the instructions options on the menu correctly opens the instructions dialogue	User must be on the main menu	2	When 2 is entered into the console the user wil be navigated to the instructions menu menu	3/8/2021	Passed	Joel Donaldson		
TCase_4	Check if the game initiates the character creation section when start game is selected	User must be on the main menu	1	When 1 is entered into the console the user will be taken to the character creation menu	3/8/2021	Passed	Joel Donaldson		
TCase_5	Check if the code responds correctly to the player entering 1 as the number of players in the game	User must be on the character creation menu	1	When 1 is entered into the console the user will be informed to enter a valid number of players instead	3/8/2021	Passed	Joel Donaldson		
TCase_6	Check if the code responds correctly to the player entering a number over 8 as the number of players in the game	User must be on the character creation menu	12	When 12 is entered into the console the user will be informed to enter a valid number of players instead	3/8/2021	Passed	Joel Donaldson		
TCase_7	Check if the code responds correctly to the player entering a number between 2 and 8 inclusive as the number of players in the game	User must be on the character creation menu	4	When 4 is entered into the console the user will be taken to the character naming section of the character creation menu	3/8/2021	Passed	Matthew Creighton		
TCase_8	Check if the code responds correctly to the player entering an erronious player name	User must be on the character creation menu	Al3x&nd3r	When a player enters an erronious player name the console	3/8/2021	Passed	Matthew Creighton		
TCase_9	Check if the code responds correctly to the player entering an extreme player name	User must be on the character creation menu	Wolfeschlegelsteinhausenbergerdorff	When a player enter an extreme name the name is accepted and the next character is created on the case of the last character the game will begin	3/8/2021	Passed	Matthew Creighton		
TCase_10	Check if the code correctly creates a player when the player enters a suitable player name	User must be on the character creation menu	Alexander	When the player enters a regular name, the name is accepted and the next character is created on the case of the last character the game will begin	3/8/2021	Passed	Matthew Creighton		
TCase_11	Check if the code responds correcly when the user enters a string or other input type that is not of type integer when they are on the first menu within the game	User must be on the Start Game menu	]	When the player enters], the system should let them know this is not a valid input type and they have to input an integer between 1 and 3. The system should not crash	3/8/2021	Passed	Ryan Craig		
TCase_12	Check the code responds correctly and does not crash when the user enters an integer value that is not between the range 1 and 3 (the accepted inputs) on the Start Game menu	User must be on the Start Game menu	55	When the player enters 55 which is well outside the range of integers they can choose from, the system should tell them to enter a suitable integer value between 1 and 3 and the system should not crash	3/8/2021	Passed	Jack Cosby		
TCase_13	Check if the code responds correctly responds when the player enters in an integer that is outside the range of integers that they can choose from in the menu when the game has started	User must be on the main menu	25	When the player enters 25 which is well outside the range of integers they can choose from, the system should tell them to enter a suitable integer value between 1 and 6 and the system should not crash	3/8/2021	Passed	James Cassidy		

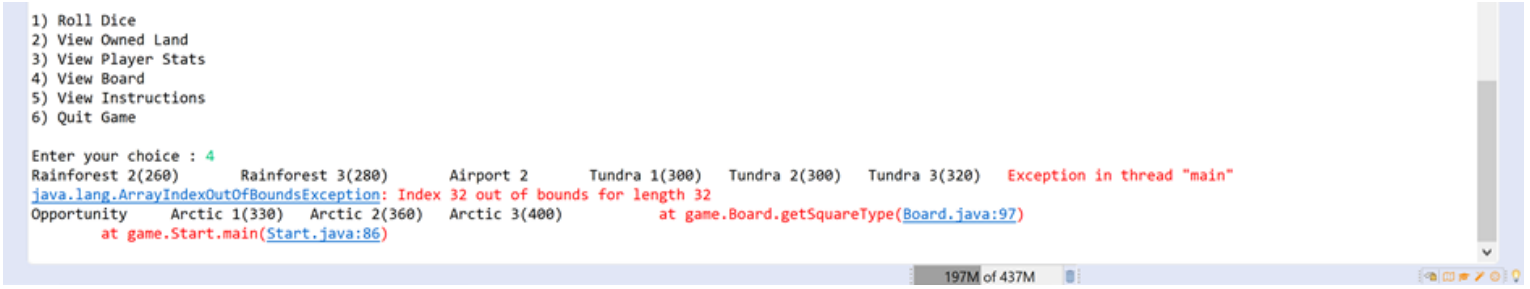
	A	B	C	D	E	F	G	H	I	J
15	TCase_14	Check if the code responds correctly when a player enters in a word instead of a integer input when choosing from the main menu system when the game has started	User must be on the main menu	Hello	When "Hello" is entered the system simply asks for a valid input integer between 1 and 6 and the system does not crash	3/8/2021	Passed	Darryl Donnely		
16	TCase_15	Check if the dice roll function executes correctly when the "roll dice" option is selected	User must be on the game menu	1	The Dice is rolled and the player moves the correct number of spaces	3/8/2021	Passed	Matthew Creighton		
17	TCase_16	Check if the "View Board" operates correctly	User must be on the game menu	4	The player board is opended and shows the sqare the player is on as well as the 12 next squares	3/8/2021	Failed	Matthew Creighton	DCase_01	There was an exception in 'main' error here and the error message said there was an array out of bounds exception
18	TCase_17	Check if the player stats dialogue opens when the player selects the "view player stats" option	User must be on the game menu	3	The player stats dialogue opens and displays up to date and accurate information	3/8/2021	Passed	Ryan Craig		
19	TCase_18	Check if the player is removed ends when the player selects the "Exit" option	User must be on the game menu	6	The code is executed without errors and the game continues with the remaining players	3/8/2021		Ryan Craig		
20	TCase_19	Check that when a player leaves the game, the land they owned is available for other players to purchase	User must be on the game menu, must own some land and must quit the game	6	The code is executed without errors and the game continues with the remaining players. The land that the player who left owned is now available to purchase if landed on by another player			Ryan Craig		
21	TCase_20	Check if the game ends when there are only two players remaining and one player selects the "exit" option	User must be on the game menu	6	The code is executed correctly and the game ends and the last remaining player is the winner			Ryan Craig		

	A	B	C	D	E	F	G	H	I	J
22	TCase_21	Check if the code responds correctly to the player entering an erronious number (decimal number) as an option in the game menu	User must be on the game menu	1.3	The code will ask to input a valid number	3/8/2021	FAILED	Ryan Craig	DCase_02	The main menu system through an exception in the 'main' and the game crashed upon entereing a decimal number
23	TCase_22	Check if the code responds correctly to the player viewing their owned biomes when the playes owns none	User must be on the game menu. User must currently own no biomes	2	The console will inform the player that they do not own any biomes	3/8/2021	Passed	Ryan Craig		
24	TCase_23	Check if the code responds correctly to the player viewing their owned biomes when the playes owns at least one	User must be on the game menu. User must own at least one biome	2	The console will inform the player of the biomes that they own	3/8/2021	Passed	Jack Cosby		
25	TCase_24	Check if the "view instructions" feature works when selected from the player menu (within the game)	User must be on the game menu	5	The player is navigated to the instructions menu and they can choose from there which instructions they want to view	3/8/2021	Passed	Jack Cosby		
26	TCase_25	Check if the player menu correctly reopens after instructions are viewed	User must be on the instructions	12	The player is navigated back to the game menu without missing their turn	3/8/2021	Passed	Jack Cosby		
27	TCase_26	Opportunity Card 1: Add/Remove eco points from player works as intended	Player must have first landed on the opportunity card square	N/A	The eco points are added/removed from the users balance and the updated balance printed showing the updated amount	3/8/2021	Passed	Jack Cosby		
28	TCase_27	Opportunity Card 2: Moving the player spaces works as intended	User must have first landed on the opportunity card square	N/A	The user is told how many spaces they are being moved and they are move the correct number of spaces on the board	3/8/2021	Passed	Jack Cosby		
29	TCase_28	Opportunity Card 3: Advance to Go/whale works as intended	User must have first landed on the opportunity card square	N/A	The user is taken directly to the go/whale square	3/8/2021	Passed	Jack Cosby		

	A	B	C	D	E	F	G	H
30	TCase_29	Player stays in the 'Whale' square until they pay the fine to be released	User must first have been sent to the 'Whale' by any effect	N/A	The user pays the 150 eco-point fine and the cost of leaving the 'Whale' square is deducted from their balance and the updated balance is printed	3/8/2021	Passed	Darryl Donnely
31	TCase_30	Player is removed from game when they run out of currency	The user must have run out of currency	N/A	The player is removed from the game and their assets are reset, available to be purchased by other players	3/8/2021	Passed	Darryl Donnely
32	TCase_31	Last player left is declared the winner	All users other than one must have run out of eco points/quit the game	N/A	The last player is named as the winner and the game ends	3/8/2021	Passed	Darryl Donnely
33	TCase_32	Disaster card 1: Removing a random property from a single player	User must first land on the disaster square	N/A	The player who has landed on the disaster square will have one of their properties taken away at random. They will be notified of this and the game will continue	3/8/2021	Passed	Darryl Donnely
34	TCase_33	Disaster card 2: Removing a random property from all players that own at least one	User must first land on the disaster square	N/A	All players will have a property taken away at random. they will be notified of theis and the game will continue	3/8/2021	Passed	James Cassidy
35	TCase_34	Disaster card 3: Removing a random sanctuary from a single player	User must first land on the disaster square	N/A	The player who has landed on the disaster square will have one of their sanctuary taken away at random. They will be notified of this and the game will continue	3/8/2021	Passed	James Cassidy
36	TCase_35	Disaster card 4: Removing a random sanctuary from all players that own at least one	User must first land on the disaster square	N/A	All players will have a sanctuary taken away at random. They will be notified of this and the game will continue	3/8/2021	Passed	James Cassidy
37	TCase_36	Check if the player can choose the option "View Player Stats" and then select a player within the game to view their stats	User must be on the main menu	N/A	The stats of the player chosen is shown, including their balance, position on the board and which sqaures they own	3/12/2021	Passed	James Cassidy

Failed tests:

DefectID: DCase_01



The above screenshot is of defect DCase_01 which corresponds to TCase_16. As you can see there is a Java array out of bounds error produced when the player selects the option “View Board”. This error was unknown to us prior to testing and since we were able to identify the error, we have gone back to the code to fix the bug. Below is another screenshot of the above “View Board” option working as intended, showing the squares ahead of the player (some are cut off due to the screenshot):

```
It is Harry's turn.
You are currently on Plains 2
What will you do?

1) Roll Dice
2) View Owned Land
3) View Player Stats
4) View Board
5) View Instructions
6) Quit Game

Enter your choice : 4
Plains 2(150)   Plains 3(160)   Opportunity     Desert 1(170)   Desert 2(180)   Desert 3(200)   Whale
    /\
You are here.
```

DefectID: DCase_02

```
<terminated> Start [Java Application] C:\Program Files\Java\jdk-12.0.2\bin\javaw.exe (1
3. Exit

Enter your choice : 1.3
Exception in thread "main" java.util.InputMismatchException
        at java.base/java.util.Scanner.throwFor(Scanner.java:939)
        at java.base/java.util.Scanner.next(Scanner.java:1594)
        at java.base/java.util.Scanner.nextInt(Scanner.java:2258)
        at java.base/java.util.Scanner.nextInt(Scanner.java:2212)
        at game.MainMenu.mainMenu(MainMenu.java:31)
        at game.Start.main(Start.java:18)
```

The above screenshot is of DCase_02 which corresponds to TCase_21. As you can see the system has crashed upon the user entering the decimal number “1.3”. This is not of the integer type required and hence the program crashed. The testing of our game allowed us to detect this bug and we have gone back to fix the issue. The updated outcome is shown below:

```
        Keep Cool

--------Main Menu--------

1. Start game
2. Instructions
3. Exit

Enter your choice : 1.3
You must enter a number between 1-3.

        Keep Cool

--------Main Menu--------

1. Start game
2. Instructions
3. Exit

Enter your choice :
```

Future opportunities to implement secure system features:

Implementing a username and password system for the players where players are required to create a username and password in order to create an account. There could be a database in the background (something like firebase could be used) where the different player information is stored.

Having a number of different permissions may improve security - us the developers could have read, write and execute permissions. However, players of the game would only have execute permissions, allowing them to play the game as intended but having certain restrictions (do not allow them to be able to write/change code).

Threat Modeling Analysis is a tool which us developers could implement which would allow the strengths and weaknesses of the system to be identified.

Appendix - Meeting Minutes

Minutes for CSC2058 Project 41 Week commencing 11/01/21 Date of this minute 15/01/21

The following team members were present on Teams (if not Teams, indicate platform) when minutes were discussed:

Name (printed/typed)	Signature (agreed bitmap or initials)
Ryan Craig	RC
James Cassidy	JC
Matthew Creighton	MC
Darryl Donnely	DD
Jack Cosby	JC
Joel Donaldson	JD

Task Reporting (Briefly list the progress for each team member in the last week.*)

Name (1): Ryan Craig
- Review feedback from Semester 1 Deliverables

Name (2): James Cassidy
- Review feedback from Semester 1 Deliverables

Name (3): Matthew Creighton
- Review feedback from Semester 1 Deliverables

Name (4): Darryl Donnely
- Review feedback from Semester 1 Deliverables

Name (5): Jack Cosby
- Review feedback from Semester 1 Deliverables

Name (6): Joel Donaldson
- Review feedback from Semester 1 Deliverables

Actions Planned (Briefly list the actions required of each team member for the next week.)

Name (1): Ryan Craig
- Think of any changes we need to make after consulting the feedback and report back to group in next meeting

Name (2): James Cassidy
- Think of any changes we need to make after consulting the feedback and report back to group in next meeting

Name (3): Matthew Creighton
- Think of any changes we need to make after consulting the feedback and report back to group in next meeting

Name (4): Darryl Donnely
- Think of any changes we need to make after consulting the feedback and report back to group in next meeting

Name (5): Jack Cosby
- Think of any changes we need to make after consulting the feedback and report back to group in next meeting

Name (6): Joel Donaldson
- Think of any changes we need to make after consulting the feedback and report back to group in next meeting

Minutes for CSC2058 Project 41 Week commencing 18/01/21 Date of this minute 22/01/21

The following team members were present on Teams (if not Teams, indicate platform) when minutes were discussed:

Name (printed/typed)	Signature (agreed bitmap or initials)
Ryan Craig	RC
James Cassidy	JC
Matthew Creighton	MC
Darryl Donnely	DD
Jack Cosby	JC
Joel Donaldson	JD

Task Reporting (Briefly list the progress for each team member in the last week.*)

Name (1): Ryan Craig
- Review feedback from Semester 1 Deliverables and gather any changes needed to class relationship diagram and sequence diagrams etc.

Name (2): James Cassidy
- Review feedback from Semester 1 Deliverables and gather any changes needed to class relationship diagram and sequence diagrams etc.

Name (3): Matthew Creighton
- Review feedback from Semester 1 Deliverables and gather any changes needed to class relationship diagram and sequence diagrams etc.

Name (4): Darryl Donnely
- Review feedback from Semester 1 Deliverables and gather any changes needed to class relationship diagram and sequence diagrams etc.

Name (5): Jack Cosby
- Review feedback from Semester 1 Deliverables and gather any changes needed to class relationship diagram and sequence diagrams etc.

Name (6): Joel Donaldson
- Review feedback from Semester 1 Deliverables and gather any changes needed to class relationship diagram and sequence diagrams etc.

Actions Planned (Briefly list the actions required of each team member for the next week.)

Name (1): Ryan Craig
- As a group carry out variable name changes and other small changes as indicated by our group feedback.

Name (2): James Cassidy
- As a group carry out variable name changes and other small changes as indicated by our group feedback.

Name (3): Matthew Creighton
- As a group carry out variable name changes and other small changes as indicated by our group feedback.

Name (4): Darryl Donnely
- As a group carry out variable name changes and other small changes as indicated by our group feedback.

Name (5): Jack Cosby
- As a group carry out variable name changes and other small changes as indicated by our group feedback.

Name (6): Joel Donaldson

- As a group carry out variable name changes and other small changes as indicated by our group feedback.

Minutes for CSC2058 Project 41 Week commencing 25/01/21 Date of this minute 29/01/21

The following team members were present on Teams (if not Teams, indicate platform) when minutes were discussed:

Name (printed/typed)	Signature (agreed bitmap or initials)
Ryan Craig	RC
James Cassidy	JC
Matthew Creighton	MC
Darryl Donnely	DD
Jack Cosby	JC
Joel Donaldson	JD

Task Reporting (Briefly list the progress for each team member in the last week.*)

Name (1): Ryan Craig
- Completed changes from feedback.

Name (2): James Cassidy
- Completed changes from feedback.

Name (3): Matthew Creighton
- Completed changes from feedback.

Name (4): Darryl Donnely
- Completed changes from feedback.

Name (5): Jack Cosby
- Completed changes from feedback.

Name (6): Joel Donaldson
- Completed changes from feedback.

*Printouts giving an overview of interim deliverables may be added as a supplement to these minutes.

Actions Planned (Briefly list the actions required of each team member for the next week.)

Name (1): Ryan Craig
- Think about next steps required in the project and split up into pairs to begin developing code on different sections.

Name (2): James Cassidy
- Think about next steps required in the project and split up into pairs to begin developing code on different sections.

Name (3): Matthew Creighton
- Think about next steps required in the project and split up into pairs to begin developing code on different sections.

Name (4): Darryl Donnely
- Think about next steps required in the project and split up into pairs to begin developing code on different sections.

Name (5): Jack Cosby

- Think about next steps required in the project and split up into pairs to begin developing code on different sections.

Name (6): Joel Donaldson

- Think about next steps required in the project and split up into pairs to begin developing code on different sections.

Minutes for CSC2058 Project 41 Week commencing 01/02/21 Date of this minute 05/02/21

The following team members were present on Teams (if not Teams, indicate platform) when minutes were discussed:

Name (printed/typed)	Signature (agreed bitmap or initials)
Ryan Craig	RC
James Cassidy	JC
Matthew Creighton	MC
Darryl Donnely	DD
Jack Cosby	JC
Joel Donaldson	JD

Task Reporting (Briefly list the progress for each team member in the last week.*)

Name (1): Ryan Craig
- Began working on the Design Documentation, documenting the text-user interface.

Name (2): James Cassidy
- Began working on the Airport Square java file.

Name (3): Matthew Creighton
- Began working on the Wildlife Sanctuaries

Name (4): Darryl Donnely
- Began developing the main menu system with Jack - added more options from the user to choose from.

Name (5): Jack Cosby
- Continued developing the main menu system with Darryl - 'View Player Stats is now an option

Name (6): Joel Donaldson
- Began writing the Instruction manual for the game.

*Printouts giving an overview of interim deliverables may be added as a supplement to these minutes.

Actions Planned (Briefly list the actions required of each team member for the next week.)

Name (1): Ryan Craig
- Continue with Design Documentation and continually update as the game itself undergoes changes (pulling the changes made from GIT)

Name (2): James Cassidy
- Further develop the Airport Square function and implement this into the game.

Name (3): Matthew Creighton
- Further develop the Wildlife Sanctuaries and implement this into the game.

Name (4): Darryl Donnely
- Begin working on the "Land" list.

Name (5): Jack Cosby

- Work on getting the game to finish when there is only one player left and removing players as they have no eco-points left.

Name (6): Joel Donaldson

- Continue working on the Instructions manual - plan to have a detailed manual that can be viewed within the browser and a shorter, more concise manual viewable within the game itself.

Minutes for CSC2058 Project 41 Week commencing 08/02/21 Date of this minute 12/02/21

The following team members were present on Teams (if not Teams, indicate platform) when minutes were discussed:

Name (printed/typed)	Signature (agreed bitmap or initials)
Ryan Craig	RC
James Cassidy	JC
Matthew Creighton	MC
Darryl Donnely	DD
Jack Cosby	JC
Joel Donaldson	JD

Task Reporting (Briefly list the progress for each team member in the last week.*)

Name (1): Ryan Craig
- Design documentation progress and team minutes.

Name (2): James Cassidy
- Tidied up the airport square functionality.

Name (3): Matthew Creighton
- Put together the functionality of the wildlife sanctuaries with Darryl.

Name (4): Darryl Donnely
- Developed the different “Land” squares functionality.

Name (5): Jack Cosby
- Added functionality that the game now ends when there is a clear winner - only one person left with eco-points. Furthemore, when a player quits the game, the game ends and the person with the most points is the winner.

Name (6): Joel Donaldson
- Instruction manual finished and implemented into the game.

*Printouts giving an overview of interim deliverables may be added as a supplement to these minutes.

Actions Planned (Briefly list the actions required of each team member for the next week.)

Name (1): Ryan Craig
- Continue with Design Documentation and continually update as the game itself undergoes changes (pulling the changes made from GIT). Also, begin work on a burndown chart that will help the team see where we are at in terms of meeting the deadlines set for ourselves.

Name (2): James Cassidy
- Started to look at testing different features and inputs, making a note of areas we need to protect from invalid input,

Name (3): Matthew Creighton
- Start work on the opportunity cards and their implementation.

Name (4): Darryl Donnely

- Add functionality when the user passes go and to develop the Airport java file.

Name (5): Jack Cosby
- Continue to add functionality for enabling players to go bankrupt and for the game to finish under appropriate circumstances.

Name (6): Joel Donaldson
- Begin researching the test plan and looking at potential test cases with James.

Minutes for CSC2058 Project 41 Week commencing 15/02/21 Date of this minute 19/02/21

The following team members were present on Teams (if not Teams, indicate platform) when minutes were discussed:

Name (printed/typed)	Signature (agreed bitmap or initials)
Ryan Craig	RC
James Cassidy	JC
Matthew Creighton	MC
Darryl Donnely	DD
Jack Cosby	JC
Joel Donaldson	JD

Task Reporting (Briefly list the progress for each team member in the last week.*)

Name (1): Ryan Craig
- Continued with design documentation and began work on creating the burndown chart.

Name (2): James Cassidy
- Gathered a list of areas that require further development to ensure the game does not crash upon entering invalid inputs.

Name (3): Matthew Creighton
- Made progress on the cards within the game - created a switch statement with the different card options.

Name (4): Darryl Donnely
- The game now has functionality for passing “Go” (user is rewarded with points) and the airport squares function as intended.

Name (5): Jack Cosby
- Players can now go bankrupt, leave the game and win the game.

Name (6): Joel Donaldson
- Researched the best format for our test plan and began to write it up.

*Printouts giving an overview of interim deliverables may be added as a supplement to these minutes.

Actions Planned (Briefly list the actions required of each team member for the next week.)

Name (1): Ryan Craig
- Make any more changes to the design documentation and team minutes required. Finish the burndown chart.

Name (2): James Cassidy
- Plan the ‘Whale’ square java file and its implementation

Name (3): Matthew Creighton
- Re-work the instructions manual and create a menu the user can choose from to view relevant instructions.

Name (4): Darryl Donnely

- Began making the game more secure by adding checks to prevent the game crashing upon the user entering invalid inputs. Working with Matthew to introduce some variability (RNG) to the card functionality with different chances of drawing certain cards.

Name (5): Jack Cosby
- Develop any remaining java files necessary to improve functionality of the game

Name (6): Joel Donaldson
- Continue with the test plan and start looking at JUnit testing.

Minutes for CSC2058 Project 41 Week commencing 22/02/21 Date of this minute 26/02/21

The following team members were present on Teams (if not Teams, indicate platform) when minutes were discussed:

Name (printed/typed)	Signature (agreed bitmap or initials)
Ryan Craig	RC
James Cassidy	JC
Matthew Creighton	MC
Darryl Donnely	DD
Jack Cosby	JC
Joel Donaldson	JD

Task Reporting (Briefly list the progress for each team member in the last week.*)

Name (1): Ryan Craig
- Completed burndown chart and planned out documentation for all value-added features.

Name (2): James Cassidy
- Began working on the 'Whale' square.

Name (3): Matthew Creighton
- Made changes to the instructions - created a menu to give the user a choice to choose the relevant area they need instructions for.

Name (4): Darryl Donnely
- The opportunity square now has a certain amount of variability associated with it - certain chance for positive outcomes and certain chance for negative outcomes.

Name (5): Jack Cosby
- Further developed the ability to 'view' the board in-game. The user can now view the next 12 squares ahead of the position they are currently at.

Name (6): Joel Donaldson
- Continued to develop the various different tests.

*Printouts giving an overview of interim deliverables may be added as a supplement to these minutes.

Actions Planned (Briefly list the actions required of each team member for the next week.)

Name (1): Ryan Craig
- Write-up section on value-added features and push an extra section of code that congratulates and thanks the players for playing Keep Cool.

Name (2): James Cassidy
- Developed the 'Whale' square functionality, ready to commit to GIT.

Name (3): Matthew Creighton
- Shortened down the instructions to be more concise and began to create a menu-system for the player to choose what instructions they want to view.

Name (4): Darryl Donnely
- Help James working on the 'Whale' square functionality.

Name (5): Jack Cosby

- Fixing some bugs within the view board function that occur when the squares ahead of the player passes the ‘Go’ square.

Name (6): Joel Donaldson

- Continue with the testing documentation.

Minutes for CSC2058 Project 41 Week commencing 01/03/21 Date of this minute 05/03/21

The following team members were present on Teams (if not Teams, indicate platform) when minutes were discussed:

Name (printed/typed)	Signature (agreed bitmap or initials)
Ryan Craig	RC
James Cassidy	JC
Matthew Creighton	MC
Darryl Donnely	DD
Jack Cosby	JC
Joel Donaldson	JD

Task Reporting (Briefly list the progress for each team member in the last week.*)

Name (1): Ryan Craig
- Completed burndown chart and planned out documentation for all value-added features.

Name (2): James Cassidy
- Began working on the 'Whale' square.

Name (3): Matthew Creighton
- Made changes to the instructions - created a menu to give the user a choice to choose the relevant area they need instructions for.

Name (4): Darryl Donnely
- The opportunity square now has a certain amount of variability associated with it - certain chance for positive outcomes and certain chance for negative outcomes.

Name (5): Jack Cosby
- Further developed the ability to 'view' the board in-game. The user can now view the next 12 squares ahead of the position they are currently at.

Name (6): Joel Donaldson
- Continued to develop the various different tests.

*Printouts giving an overview of interim deliverables may be added as a supplement to these minutes.

Actions Planned (Briefly list the actions required of each team member for the next week.)

Name (1): Ryan Craig
- Write-up section on value-added features and push an extra section of code that congratulates and thanks the players for playing Keep Cool.

Name (2): James Cassidy
- Develop the 'Whale' square functionality, ready to commit to GIT.

Name (3): Matthew Creighton
- Shortened down the instructions to be more concise and began to create a menu-system for the player to choose what instructions they want to view.

Name (4): Darryl Donnely
- Help James working on the 'Whale' square functionality.

Name (5): Jack Cosby

- Fixing some bugs within the ‘View Board’ function that occur when the squares ahead of the player passes the ‘Go’ square.

Name (6): Joel Donaldson

- Finish the testing document.

Minutes for CSC2058 Project 41 Week commencing 08/03/21 Date of this minute 12/03/21

The following team members were present on Teams (if not Teams, indicate platform) when minutes were discussed:

Name (printed/typed)	Signature (agreed bitmap or initials)
Ryan Craig	RC
James Cassidy	JC
Matthew Creighton	MC
Darryl Donnely	DD
Jack Cosby	JC
Joel Donaldson	JD

Task Reporting (Briefly list the progress for each team member in the last week.*)

Name (1): Ryan Craig
- Wrote up about the value-added features added to our game and explained the rationale behind why we implemented them.

Name (2): James Cassidy
- 'Whale' square works - now has functionality similar to 'Jail' square in monopoly.

Name (3): Matthew Creighton
- Made the instructions more concise and ensured the instructions terminology matches up with the terminology used in the game itself.

Name (4): Darryl Donnely
- Aided James is the functionality of the 'Whale' square.

Name (5): Jack Cosby
- Fixed bugs on the 'View Board' option on the menu.

Name (6): Joel Donaldson
- Finished up the main test plan.

*Printouts giving an overview of interim deliverables may be added as a supplement to these minutes.

Actions Planned (Briefly list the actions required of each team member for the next week.)

Name (1): Ryan Craig
- Go over the test plan and include the screenshots for the tests that initially failed during testing. Also then include screenshots of the same error not occurring after we fixed the issue.

Name (2): James Cassidy
- Begin working on the class relationship model from semester 1 and update accordingly with the features we have since added.

Name (3): Matthew Creighton
- Went through the code and java files ensuring that the naming of classes etc match up with the names used within the documentation,

Name (4): Darryl Donnely
- Begin developing a new square called 'Disaster' square which when landed on causes something detrimental to occur to the player/all players.

Name (5): Jack Cosby

- Continue developing ‘Wildlife Sanctuaries’ as there was missing functionality here

Name (6): Joel Donaldson

- Begin with generating some JUnit testing.

Minutes for CSC2058 Project 41 Week commencing 15/03/21 Date of this minute 19/03/21

The following team members were present on Teams (if not Teams, indicate platform) when minutes were discussed:

Name (printed/typed)	Signature (agreed bitmap or initials)
Ryan Craig	RC
James Cassidy	JC
Matthew Creighton	MC
Darryl Donnely	DD
Jack Cosby	JC
Joel Donaldson	JD

Task Reporting (Briefly list the progress for each team member in the last week.*)

Name (1): Ryan Craig
- Got together the screenshots or error messages we got during testing and the updated screenshots showing the error messages no longer occurring.

Name (2): James Cassidy
- Started work on the class relationship model, updating it as required.

Name (3): Matthew Creighton
- Went through code and documentation ensuring that all of our names matched up.

Name (4): Darryl Donnely
- 'Disaster' square now functions with it causing random negative effects when a player lands on it. The outcome of landing on the square will be different for each player.

Name (5): Jack Cosby
- Added more functionality to the 'Wildlife Sanctuaries' - when a player lands on a piece of land that they already own, they have the chance to purchase it again and place a 'Wildlife Sanctuary' on it. This will make the cost of other players landing on this square increase.

Name (6): Joel Donaldson
- Started some JUnit tests.

*Printouts giving an overview of interim deliverables may be added as a supplement to these minutes.

Actions Planned (Briefly list the actions required of each team member for the next week.)

Name (1): Ryan Craig
- Put together the final test plan, including evidence of finding bugs within the game and then us going back and fixing these bugs.
- Record section for group demo.

Name (2): James Cassidy
- Updated the class relationship model to represent the game in its current state.
- Record section for group demo.

Name (3): Matthew Creighton
- Help Joel go through some JUnit testing and start creating some JUnit test cases.
- Record section for group demo.

Name (4): Darryl Donnely
- Finish off the “Disaster’ square and begin looking at the sequence diagrams from semester 1 and identify any new diagrams required.
- Record section for group demo.

Name (5): Jack Cosby
- Finish developing the ‘Wildlife Sanctuaries’ square which should round up the code development of the game.
- Record section for group demo.

Name (6): Joel Donaldson
- Continue with JUnit testing - working with Matthew.
- Record section for group demo.

Minutes for CSC2058 Project 41 Week commencing 22/03/21 Date of this minute 25/03/21

The following team members were present on Teams (if not Teams, indicate platform) when minutes were discussed:

Name (printed/typed)	Signature (agreed bitmap or initials)
Ryan Craig	RC
James Cassidy	JC
Matthew Creighton	MC
Darryl Donnely	DD
Jack Cosby	JC
Joel Donaldson	JD

Task Reporting (Briefly list the progress for each team member in the last week.*)

Name (1): Ryan Craig
- Finished test plan and assembled final PDF document together - combining team minutes, GIT screenshots, testing etc.
- Recorded section for group demo.

Name (2): James Cassidy
- Finished the updated class relationship diagram.
- Recorded section for group demo.

Name (3): Matthew Creighton
- Finished the JUnit testing and recorded screenshots of this.
- Recorded section for group demo.

Name (4): Darryl Donnely
- Finished the sequence diagrams and included these in the final report.
- Recorded section for group demo.

Name (5): Jack Cosby
- Finished tidying up the java classes so the game is in its final form.
- Recorded section for group demo.

Name (6): Joel Donaldson
- Recorded section for group demo.
- Assembled the entire group demo together as well as the corresponding audio recordings from the group, ready to be uploaded.

*Printouts giving an overview of interim deliverables may be added as a supplement to these minutes.

Actions Planned (Briefly list the actions required of each team member for the next week.)

Name (1): Ryan Craig
- Final check over everything prior to submission.

Name (2): James Cassidy
- Final check over everything prior to submission.

Name (3): Matthew Creighton
- Final check over everything prior to submission.

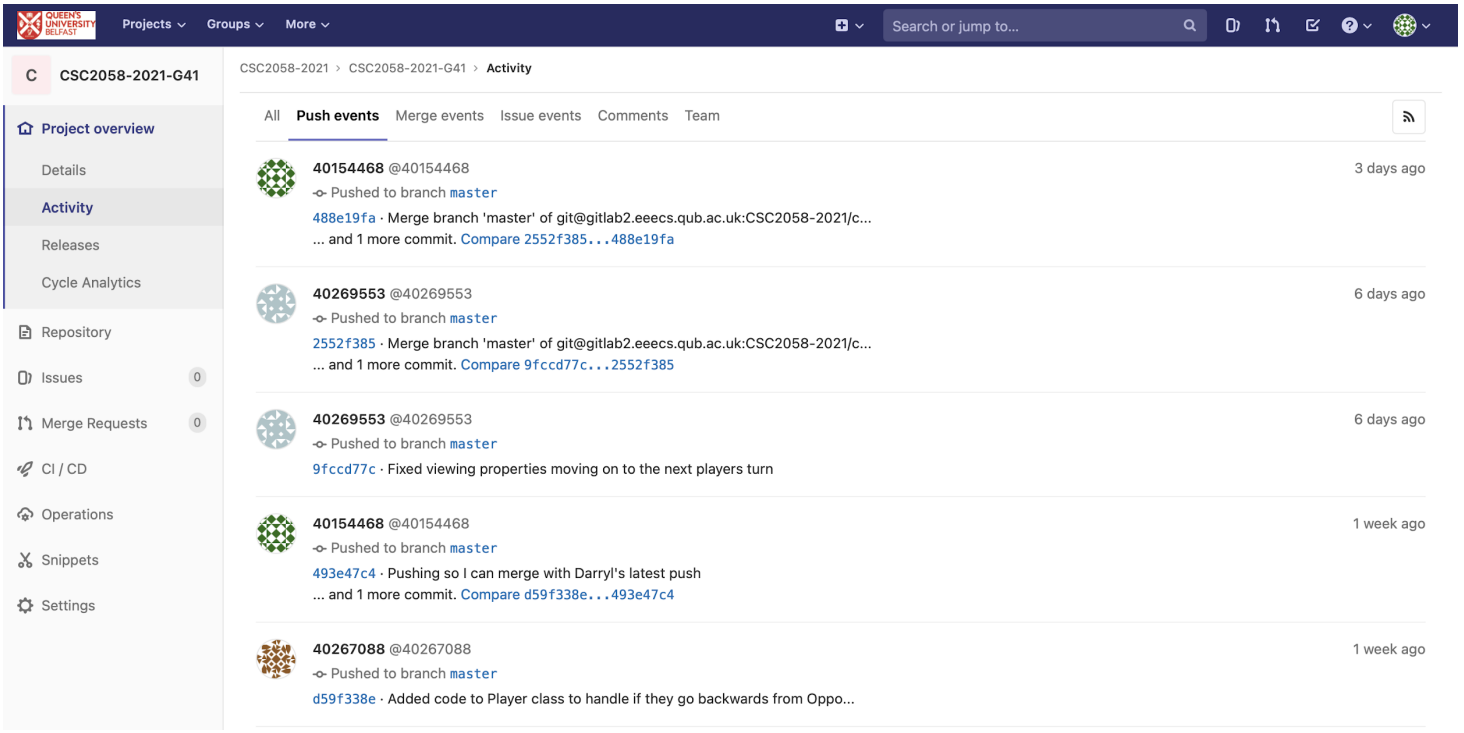
Name (4): Darryl Donnely
- Final check over everything prior to submission.

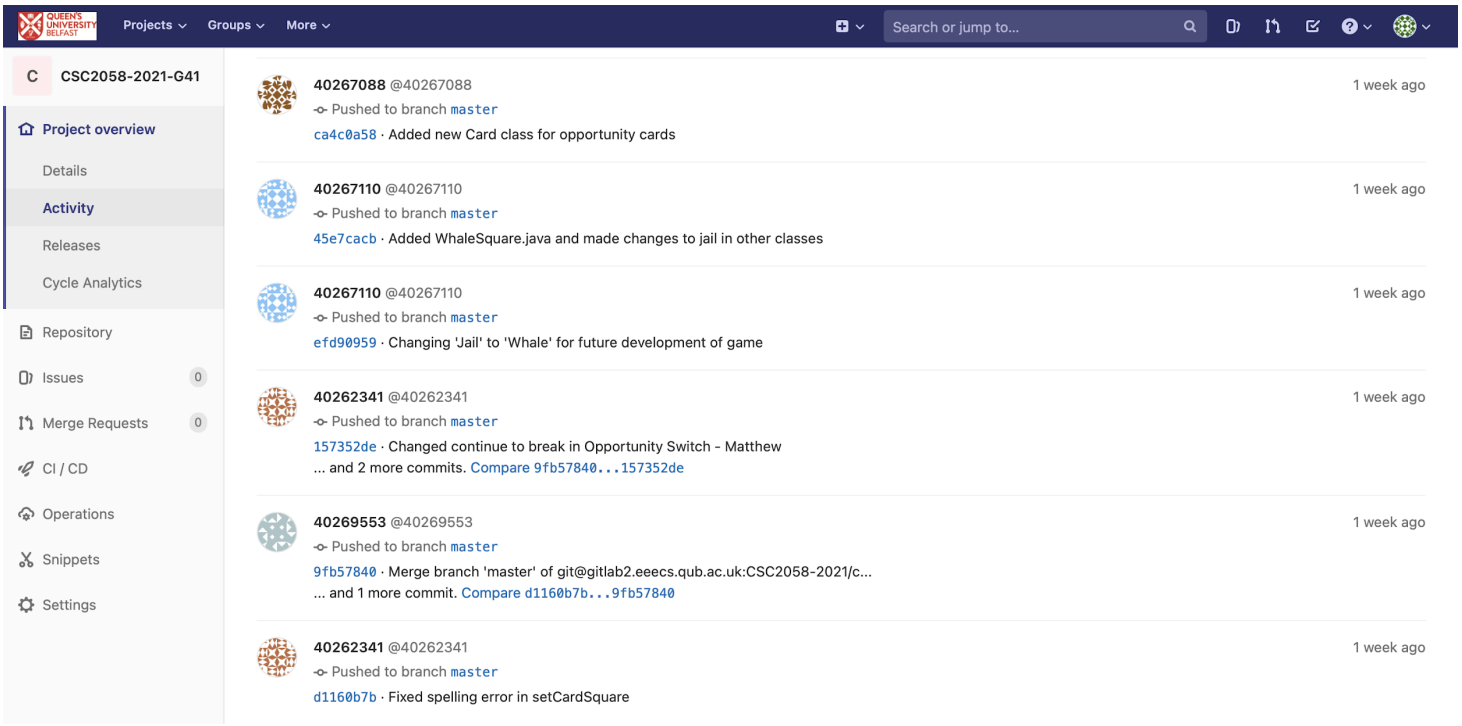
Name (5): Jack Cosby
- Final check over everything prior to submission.

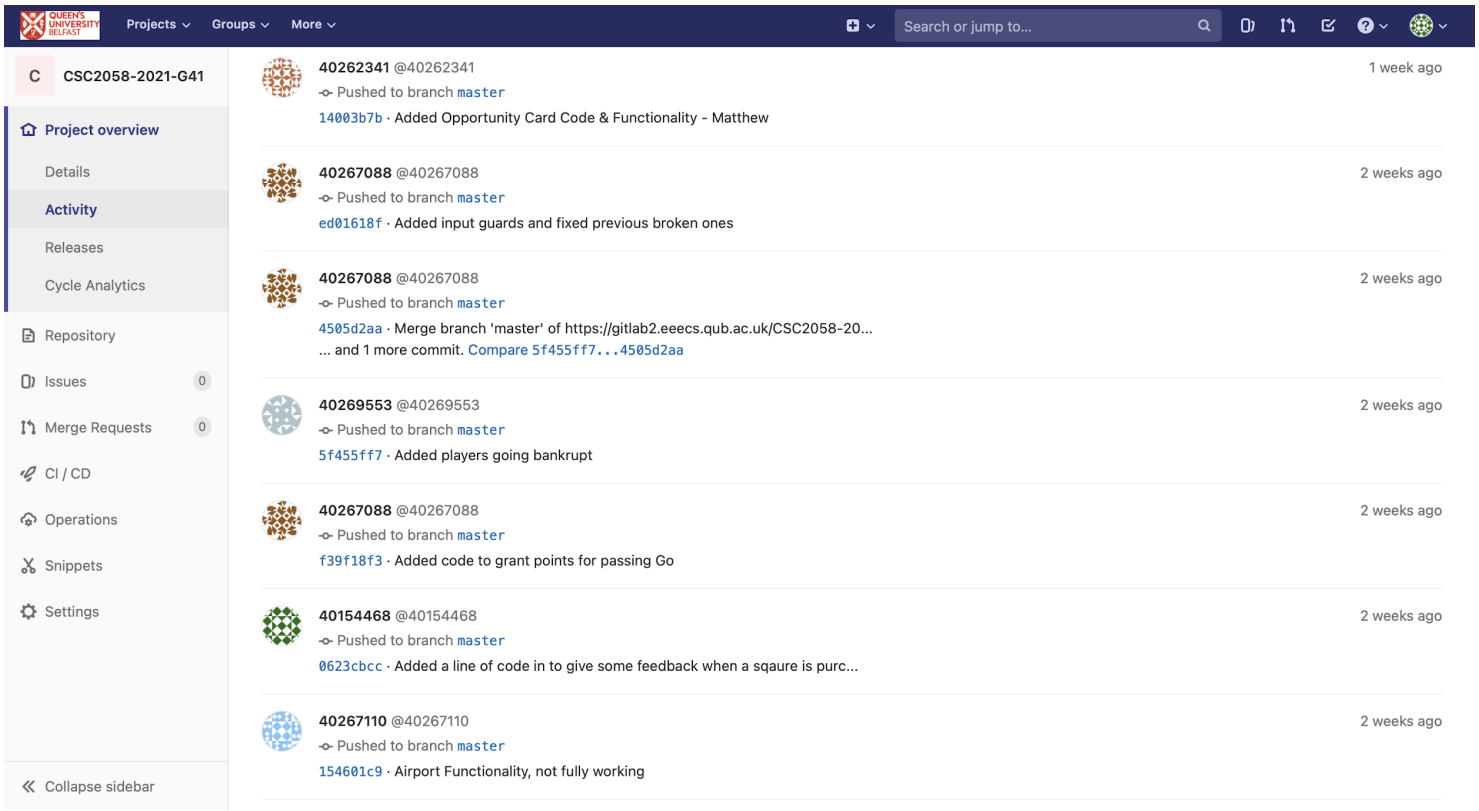
Name (6): Joel Donaldson
- Final check over everything prior to submission.

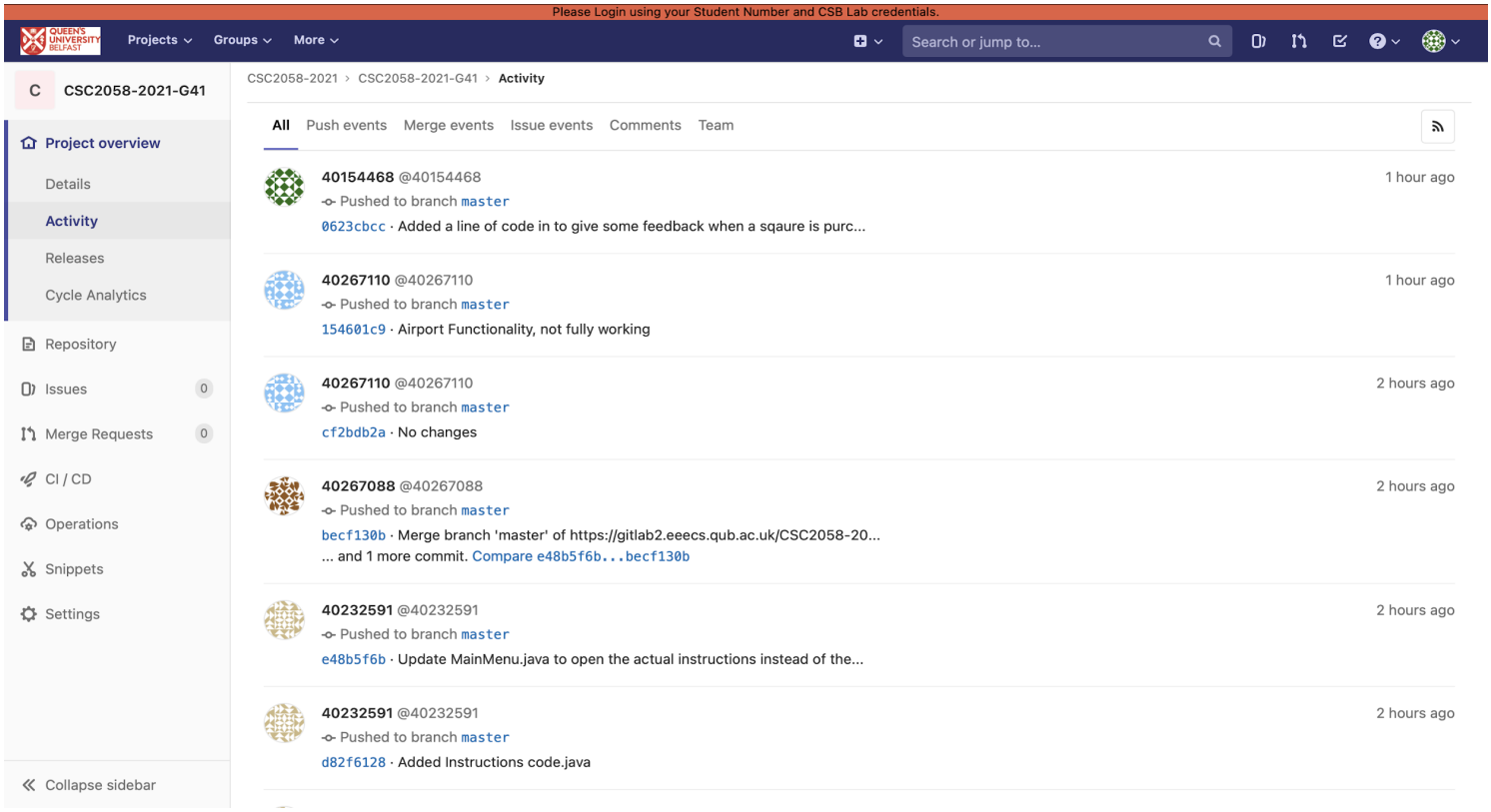
Appendix - Evidence of Adherence to Process

Evidence of populated GIT repository





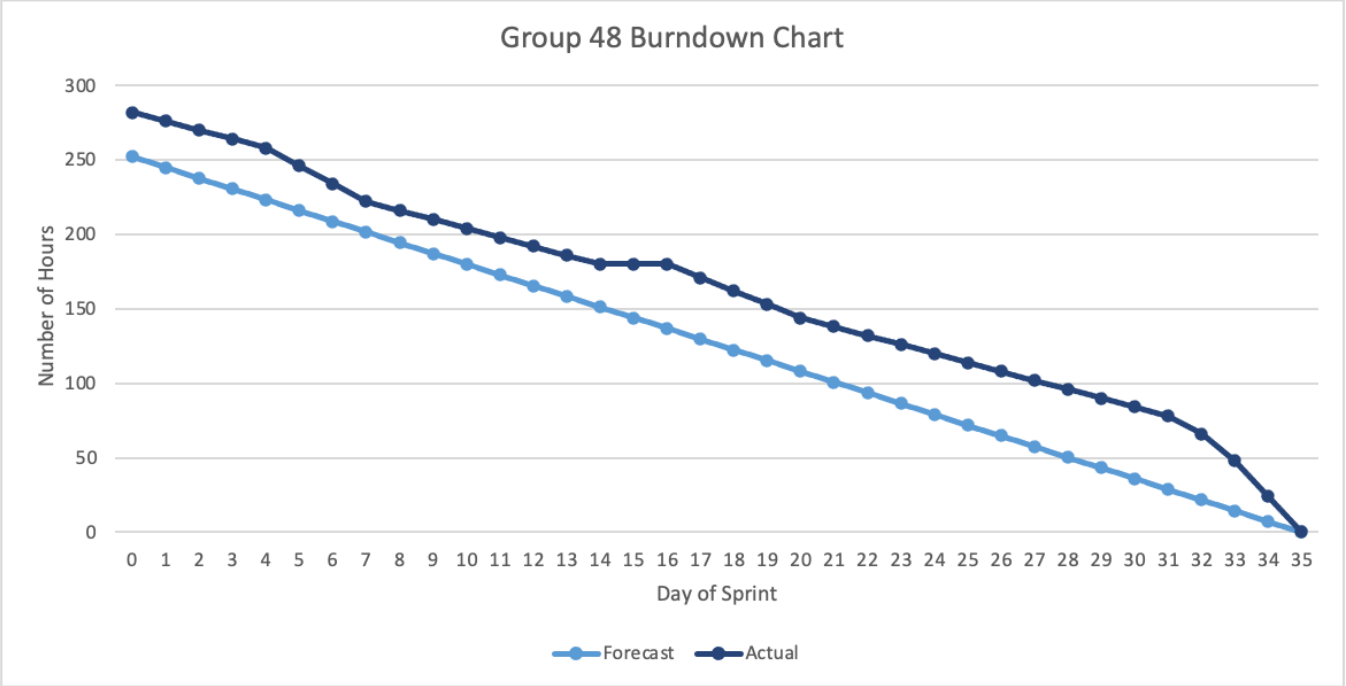




Above is a selection of screenshots that were taken from our groups GIT repository. These show that each group member was involved in pushing and pulling code to and from the repository.

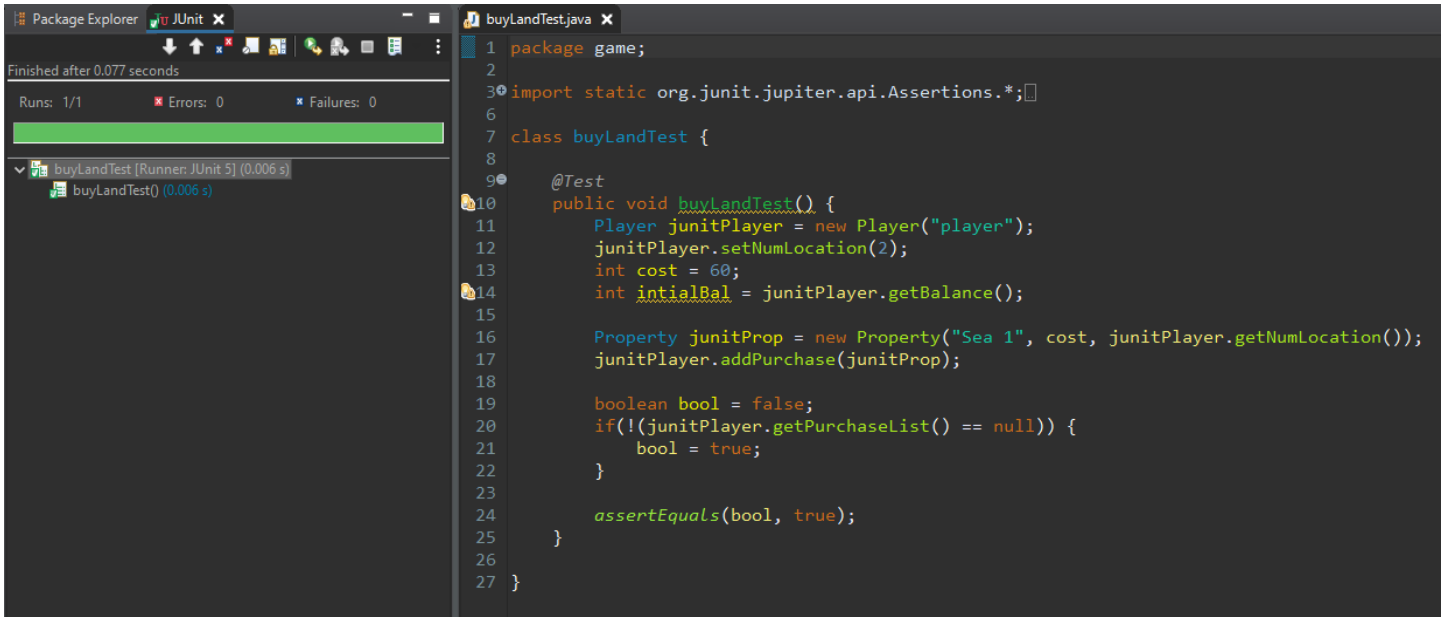
Good project management

To demonstrate good project management and adherence to process, we decided a burndown chart would be beneficial to monitor our progress and to help estimate work remaining. To create the burndown chart I used Excel to perform the calculations. This involved estimating the number of hours available to each group member and the productivity of each member also analysed. We are aware that burndown charts are typically used for shorter sprints (1-2 weeks) however, we created the chart in the time period of 35 days leading up to the deadline for the submission. This was the period of time where our group was able to work as efficiently as possible on the project. The chart acted as a means of keeping the team motivated as we could see as we approached the project deadline how much more work was required. The team was able to remain motivated right up until the submission date.

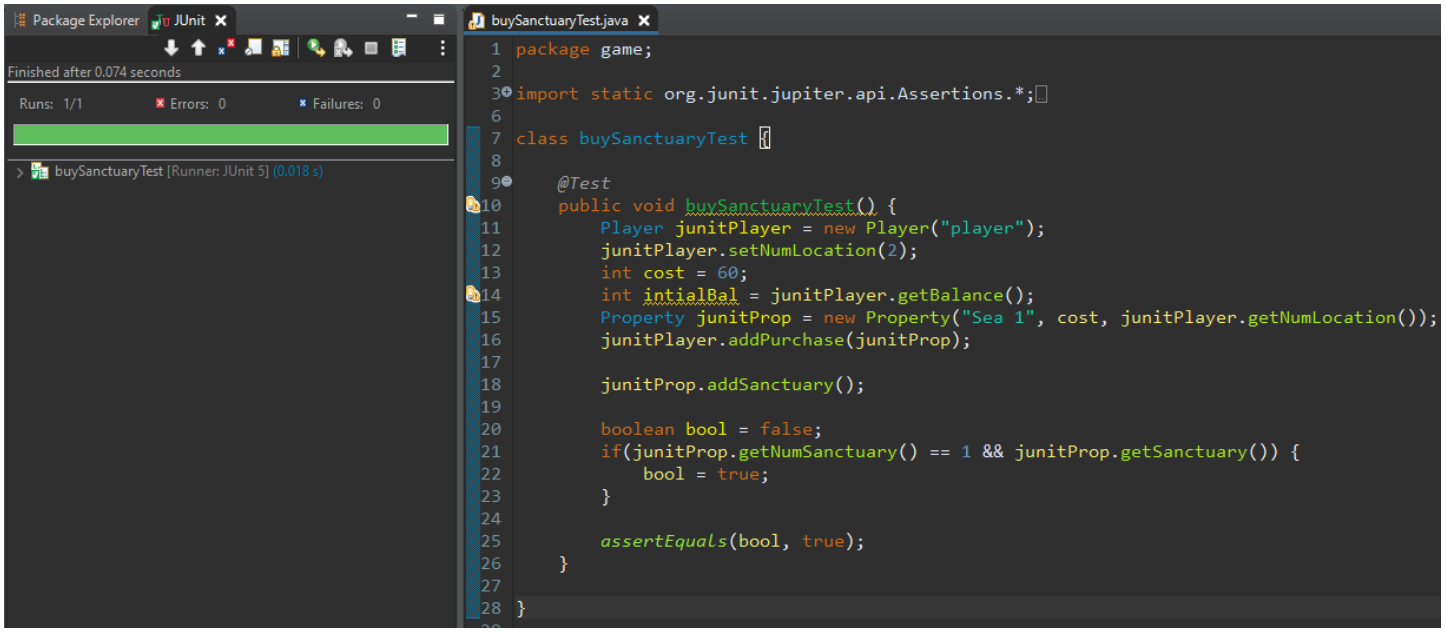


JUnit testing screenshots

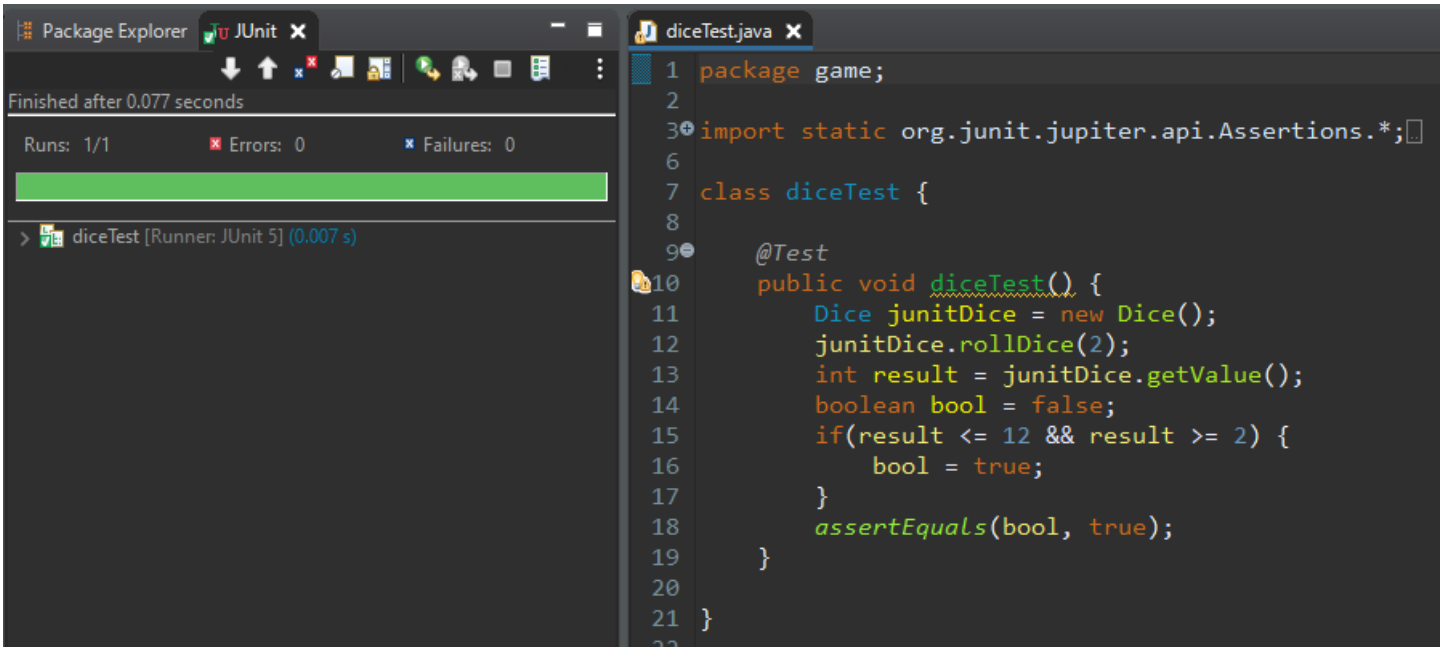
buyLandTest - This JUnit tests the purchase of an area on the board, checking if the player now has this in their inventory.



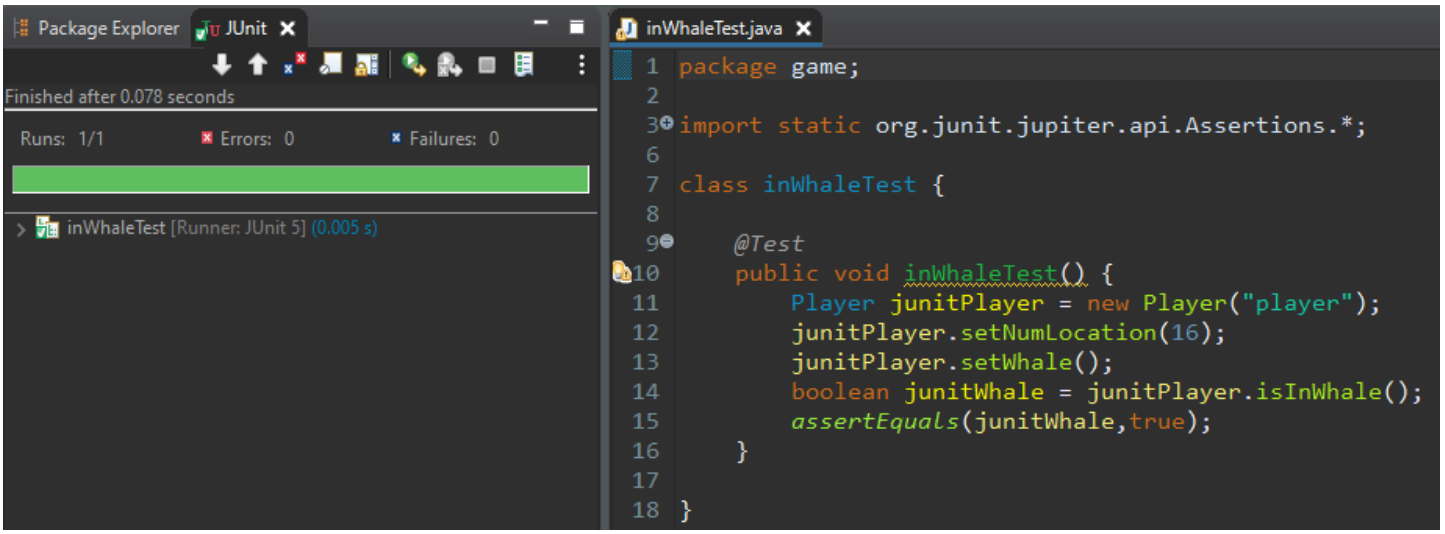
buySanctuaryTest - This JUnit tests the purchase of a Sanctuary on an area of Land, checking if the player now has this in their inventory.



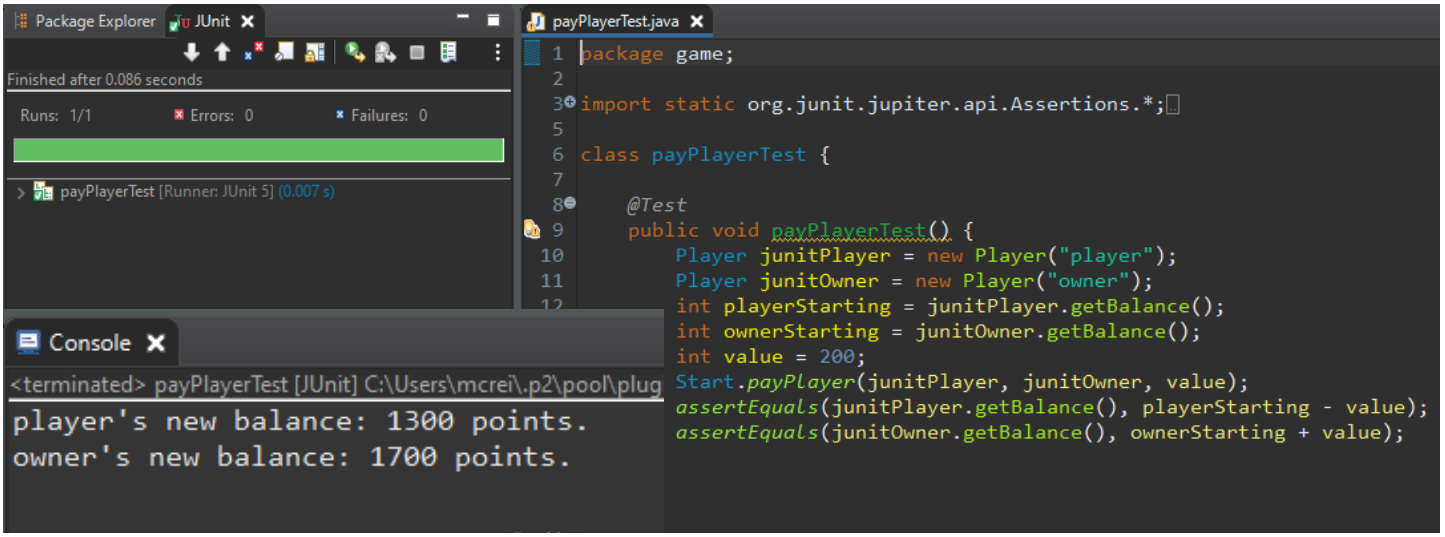
diceTest - This JUnit tests the rolling of 2 dice, and if the result from rollDice equals a number between 2 and 12.



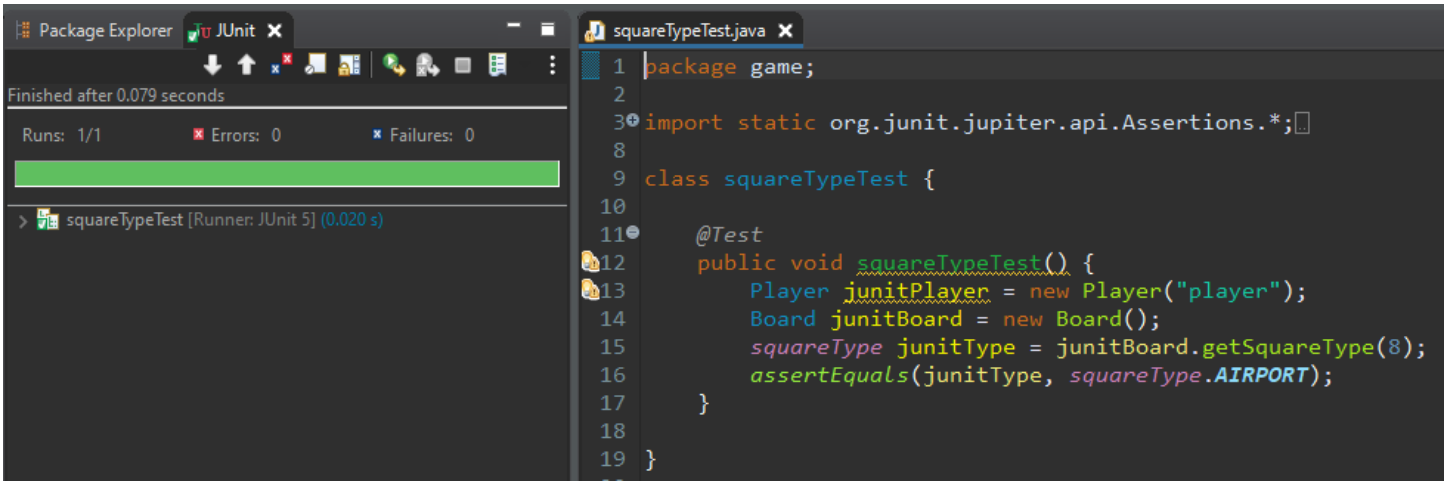
inWhaleTest - This JUnit tests if the player has been set to the Whale and then if their current state is inWhale.



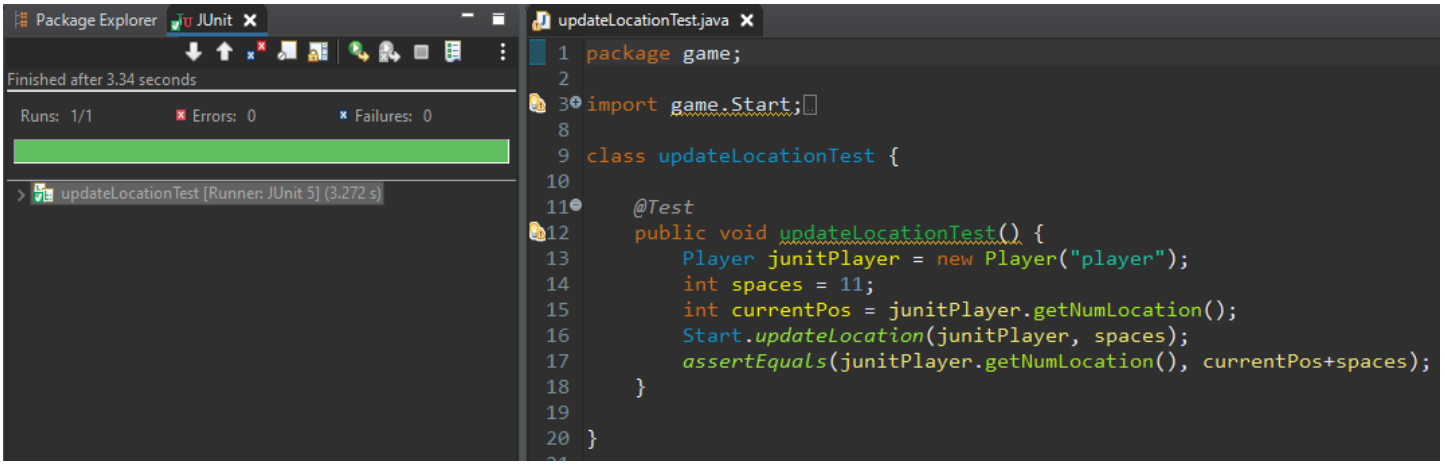
payPlayerTest - This JUnit tests the scenario where a player would land on another owned area on the board. This test checks if players balances and the transaction of payPlayer functions as intended.



squareTypeTest - This JUnit tests if the current number location of the player on the board equals the intended type of square. Airport 1 was used in this test example.



updateLocationTest - This JUnit test simulates a dice roll and a player move. This checks if updateLocation successfully moves the player the number of spaces.



CSC2058 Peer Assessment 2: Back from the Brink

This Assessment Document is intended to provide you and your assessor with an overview of each group member's involvement in the delivery of the CSC2058 Project.
Each group should complete one Assessment Document and its content must be agreed by all group members. The completed form should be included **at the start** of your group's PDF report. ***Don't forget to fill in the Group Number.***
There are two main parts to the Assessment Document – the Evaluation and the Declaration. Both parts must be completed – otherwise your group's report will not be marked. Arrange a group meeting to discuss the evaluation, and **see the note below**!

Evaluation	Group Number: 41			
Name	Contribution to team-working and motivation[1]	Contribution to PDF Report 2[1,2]	Contribution to the Working System[1,2]	**Peer Score** (Range 85 – 115)
Matthew Creighton	*5*	*5*	*5*	*115*
Joel Donaldson	*5*	*5*	*5*	*115*
Ryan Craig	*5*	*5*	*5*	*115*
James Cassidy	*5*	*5*	*5*	*115*
Jack Cosby	*5*	*5*	*5*	*115*
Darryl Donnelly	*5*	*5*	*5*	*115*

[1]Values for contribution: 1 = Minimal Contribution; 2 = Reasonable Contribution; 3 = Good Contribution; 4 = Very Good Contribution; 5 = Excellent Contribution
[2]This value should consider contributions in the round – direct contributions to required deliverables, and contributions that have made the deliverables possible.

Declaration

"I declare that I have read the Queen's University regulations on plagiarism, and that any contribution I have made to the attached submission is my own original work, except for any elements that I have clearly attributed to third parties. I understand that this submission will be subject to an electronic test for plagiarism and will also be subject to the University's regulations concerning late submission if it is received after the deadline."

Name	Date	Confirmation *(use the words shown in the example below!)*
Matthew Creighton	*18/03/21*	*I agree to the terms of the declaration*
Joel Donaldson	*18/03/21*	*I agree to the terms of the declaration*
Ryan Craig	*18/03/21*	*I agree to the terms of the declaration*
James Cassidy	*18/03/21*	*I agree to the terms of the declaration*
Jack Cosby	*18/03/21*	*I agree to the terms of the declaration*
Darryl Donnelly	*18/03/21*	*I agree to the terms of the declaration*

A note on the Evaluation:
Complete **all** the columns in the Evaluation Table. The Contribution columns are intended to help team members quantify each other's input to the project, before they award agreed **Peer Scores**. There will not necessarily be a precise correlation between the Peer Score and the Contribution values. However, high Contribution values, as an indicator of the importance of the team member's work to the success of the project, should normally result in a high Peer Score for a team member. Likewise a low Peer Score would be the expected outcome if Contribution values are low. Students who have made a high-value Contribution in all three contribution categories (e.g. 5,5,5) should expect to receive a higher Peer Score than students who have made a lower-value Contribution in one or more categories (e.g. 5,5,3).

If, having reviewed the Contribution values, the team agrees that Team Member 1 made a minimal contribution overall, a Peer Score of 85 would be appropriate for Team Member 1. If Team Member 1's contribution was excellent (critical to the success of the project in all areas of engagement), consider a peer score of 115. If Team Member 1 made a generally good contribution, doing what was expected of them, they could expect to receive a Peer Score of 100. It may be that a team member (for whatever reason) has disengaged from the project entirely, and in such circumstances a Peer Mark of 0 may be acceptable. ***Please inform the module Lecturer if a team member has left your group or has ceased to play an active role in the group.***

Each team member's overall score for the Semester 2 deliverable will be calculated according to the formula alongside, where S_i is Team Member *i*'s overall score, P_i is the Peer Score received by Team Member *i*, N is the number of members in the team, and M is the raw mark awarded to the deliverable by the assessor.

$$S_i = \frac{P_i}{\frac{1}{N}\sum_{j=1}^{N} P_j} \times M$$

Any Peer Score within the range 85 – 115 will normally be accepted by the module Lecturer. **However, students are expected to award a range of marks within a team: it is very unusual in a project for everyone to display exactly the same level of ability and commitment, and the Peer Scores should reflect this.** Be fair: be prepared to recognise someone who has adopted a leading role in the project, and acknowledge the fact that some contributions will be weaker than others. Uniform marks, or marks outside the range 85 – 115, may require that the Team discuss its decision with the module Lecturer, in order to agree a fair distribution of marks. Throughout the project, team members should use appropriately named folders in GitLab to help them co-ordinate their work and maintain a record of their contributions. Where team members cannot agree a distribution, or the distribution is unreasonable, the module Lecturer's judgement will be final.