1 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: 40			
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)
Daniel Mason	5	5	5	115
Niamh McLarnon	5	5	5	115
Michal Kennedy	0	0	0	85
Craig Mulligan	1	1	0	85
Charlotte Bisp	1	1	0	85
Omar Basha	1	1	0	85

[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!)*
Daniel Mason	26/03/2021	I agree with the terms and the declaration
Niamh McLarnon	26/03/2021	I agree with the terms and the declaration
Michal Kennedy	26/03/2021	I agree with the terms and the declaration
Charlotte Bisp	26/03/2021	I agree with the terms and the declaration
Omar Basha	26/03/2021	I agree with the terms and the declaration
Craig Mulligan	26/03/2021	I agree with the terms and 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.

The Final PDF Report & The Process

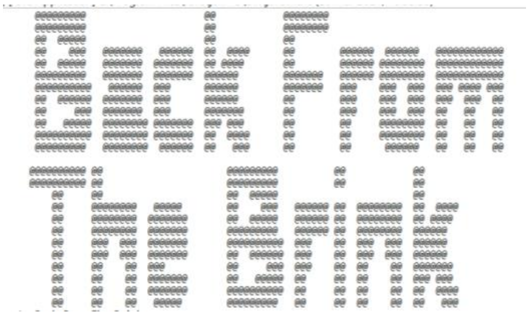
Group 40

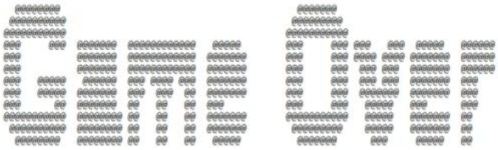
Design Documentation

Text User Interface

[N.M]

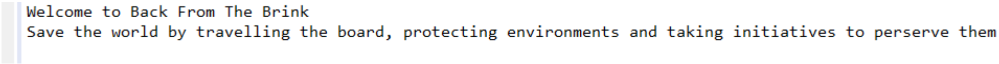
Intro/Outro





We have decided to use ASCII Art for the ‘Back from the Brink’ intro and outro to make the game more eye catching to the players. This will therefore draw in their interest.

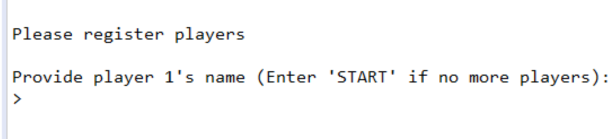
Game Scenario Message



```
Welcome to Back From The Brink
Save the world by travelling the board, protecting environments and taking initiatives to perserve them
```

We have decided to lay the game scenario message (as shown above) as a print statement as it sets the scene for the players, attracting their interest from the start. We have kept it short to keep reading minimal therefore allowing the players to start the game quickly.

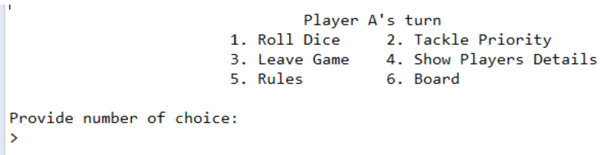
Registering Players Prompt



```
Please register players

Provide player 1's name (Enter 'START' if no more players):
>
```

We have decided to lay the registering players prompt (as shown above) as a simple enter your player’s name in the text box as it is brief and exactly tells the player what they must do to register themselves as a player to start the game.

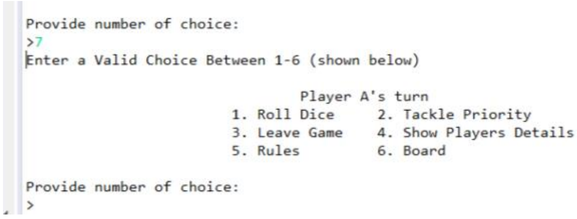
Take Turn Prompt



```
                          Player A's turn
                  1. Roll Dice      2. Tackle Priority
                  3. Leave Game     4. Show Players Details
                  5. Rules          6. Board

Provide number of choice:
>
```

We have decided to lay the enter choice prompt (as shown above) as a switch statement as it clearly lays out the list of options the player to choose from. This avoids any human errors, and it reduces the amount of typing the user has. We have minimised the number of choices the player can chose from as having too many choices may over complicate the game for the player

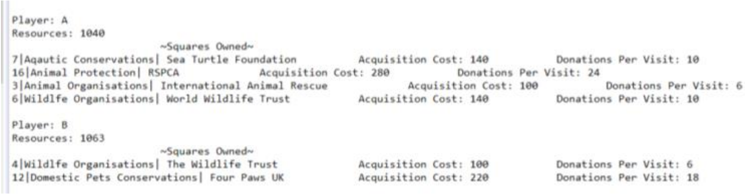
Invalid Option Message



```
Provide number of choice:
>7
Enter a Valid Choice Between 1-6 (shown below)

                         Player A's turn
                 1. Roll Dice      2. Tackle Priority
                 3. Leave Game     4. Show Players Details
                 5. Rules          6. Board

Provide number of choice:
>
```

We have decided to lay the invalid choice prompt (as shown above) as a switch statement as it is concise, and it politely tells the user this is an invalid option. It suggests to them how they can resolve this error. This therefore helps the user to continue playing with the game.

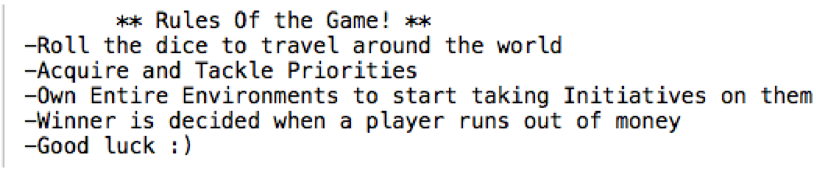
Player Details (Summary)



```
Player: A
Resources: 1040
                        ~Squares Owned~
7|Aqautic Conservations| Sea Turtle Foundation          Acquisition Cost: 140          Donations Per Visit: 10
16|Animal Protection| RSPCA            Acquisition Cost: 280          Donations Per Visit: 24
3|Animal Organisations| International Animal Rescue           Acquisition Cost: 100          Donations Per Visit: 6
6|Wildlfe Organisations| World Wildlife Trust           Acquisition Cost: 140          Donations Per Visit: 10

Player: B
Resources: 1063
                        ~Squares Owned~
4|Wildlfe Organisations| The Wildlife Trust             Acquisition Cost: 100          Donations Per Visit: 6
12|Domestic Pets Conservations| Four Paws UK            Acquisition Cost: 220          Donations Per Visit: 18
```

We have decided to lay the player details summary (as shown above) as a simple print statement. This provides the players with all the relevant details such as player names, player balance and their owned priorities (with their associated costs). We have chosen to show these details as we believe they are the most important and appropriate to the players.

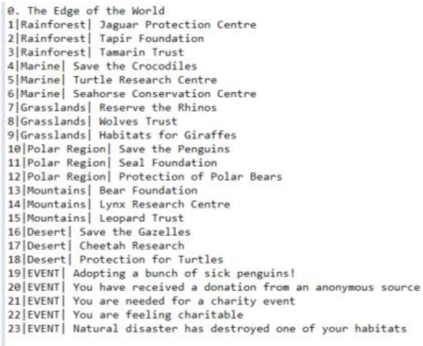
Rules Message



```
       ** Rules Of the Game! **
-Roll the dice to travel around the world
-Acquire and Tackle Priorities
-Own Entire Environments to start taking Initiatives on them
-Winner is decided when a player runs out of money
-Good luck :)
```

We have decided to lay the rules message (as shown above) as a simple print statement as players would not want a long paragraph of rules to try read and understand. It simply and clearly lays out the rules of the game.

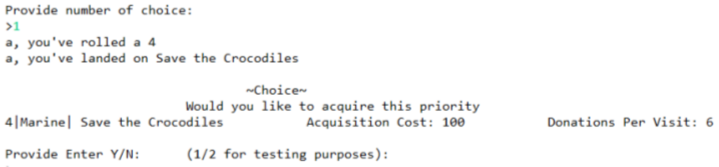
Board Summary



```
0. The Edge of the World
1|Rainforest| Jaguar Protection Centre
2|Rainforest| Tapir Foundation
3|Rainforest| Tamarin Trust
4|Marine| Save the Crocodiles
5|Marine| Turtle Research Centre
6|Marine| Seahorse Conservation Centre
7|Grasslands| Reserve the Rhinos
8|Grasslands| Wolves Trust
9|Grasslands| Habitats for Giraffes
10|Polar Region| Save the Penguins
11|Polar Region| Seal Foundation
12|Polar Region| Protection of Polar Bears
13|Mountains| Bear Foundation
14|Mountains| Lynx Research Centre
15|Mountains| Leopard Trust
16|Desert| Save the Gazelles
17|Desert| Cheetah Research
18|Desert| Protection for Turtles
19|EVENT| Adopting a bunch of sick penguins!
20|EVENT| You have received a donation from an anonymous source
21|EVENT| You are needed for a charity event
22|EVENT| You are feeling charitable
23|EVENT| Natural disaster has destroyed one of your habitats
```

We have decided to include a board summary (as shown above) as a print statement to visualise for the player the board. We have included a list of all the priorities names on the board as well as surprise events.

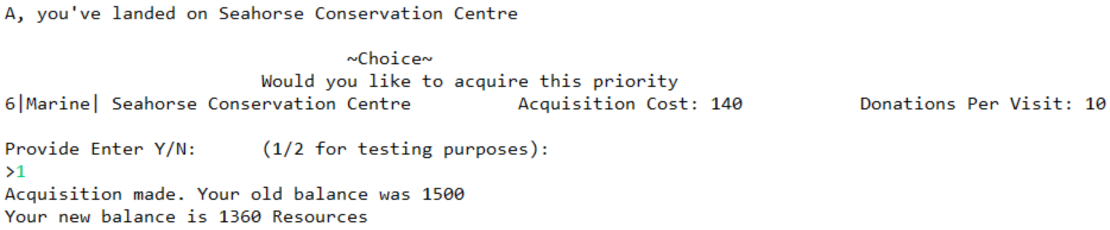
Roll Dice/Move Player Prompt



```
Provide number of choice:
>1
a, you've rolled a 4
a, you've landed on Save the Crocodiles

                                ~Choice~
                        Would you like to acquire this priority
4|Marine| Save the Crocodiles           Acquisition Cost: 100          Donations Per Visit: 6

Provide Enter Y/N:      (1/2 for testing purposes):
```

We have decided to lay the roll dice/move player prompt (as shown above) as it clearly tells the player what they have rolled, where they have landed on the board and it offers them the choice to acquire this priority (with the relevant cost). We have kept the choice simple and the layout minimal.

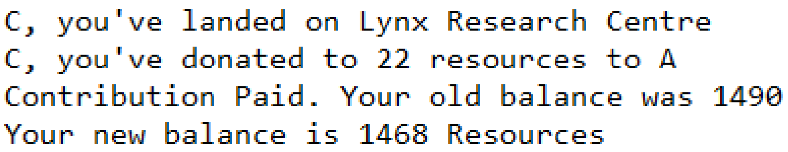
Acquiring Priority Message



```
A, you've landed on Seahorse Conservation Centre

                                ~Choice~
                        Would you like to acquire this priority
6|Marine| Seahorse Conservation Centre          Acquisition Cost: 140          Donations Per Visit: 10

Provide Enter Y/N:      (1/2 for testing purposes):
>1
Acquisition made. Your old balance was 1500
Your new balance is 1360 Resources
```

We have decided to the lay the acquiring priority choice (as shown above) it clearly lays out all the relevant details about the priority the player has landed on such as the action area it belongs to, the name of the priority and the associated costs. It also provides an update to the player on their updated balance

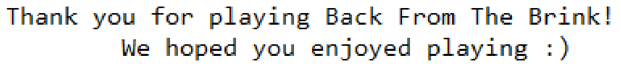
Contribution Transaction Message



```
C, you've landed on Lynx Research Centre
C, you've donated to 22 resources to A
Contribution Paid. Your old balance was 1490
Your new balance is 1468 Resources
```

We have decided to lay the Contribution Transaction message (as shown above) as it tells the player exactly where they have landed, who they are paying and their updated resources amount. It acts as a reminder to the player, of what their resource balance is.

Ending Message

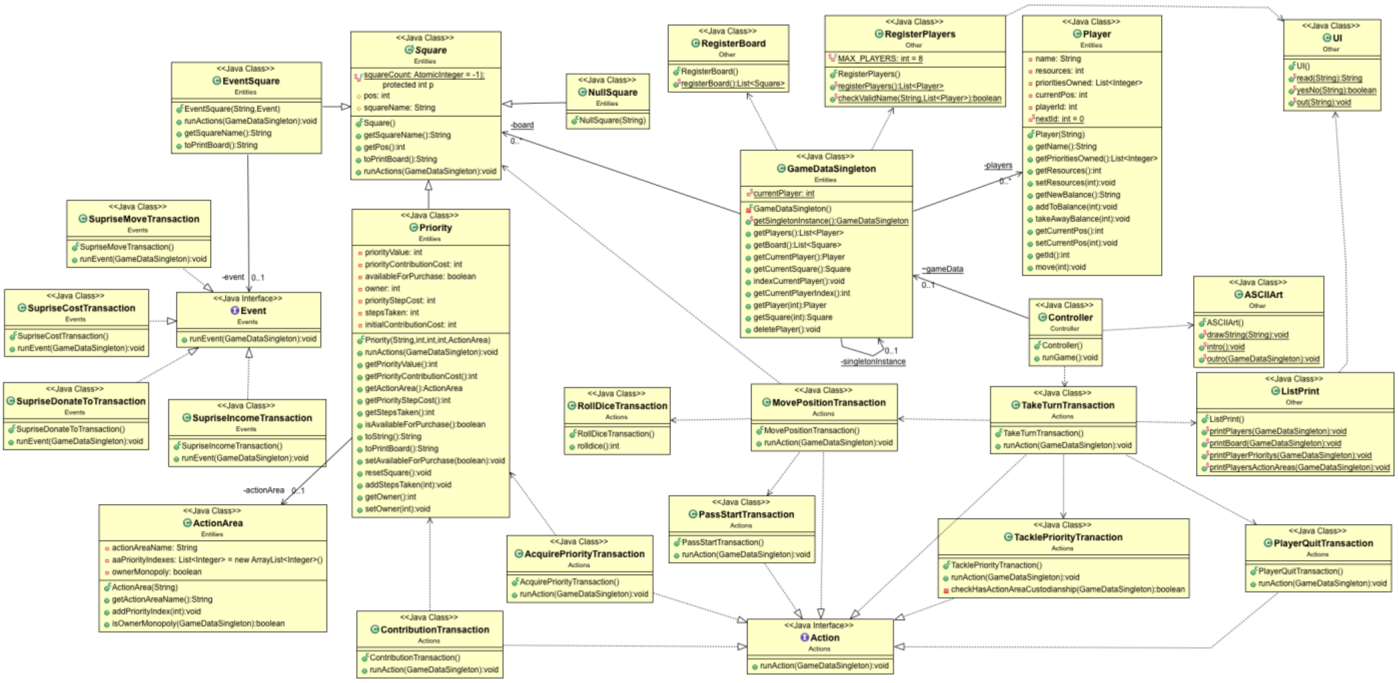


```
Thank you for playing Back From The Brink!
        We hoped you enjoyed playing :)
```

We have decided to lay the ending message (as shown above) as a print statement as it shows appreciation to the player for playing the game encouraging the player to want to play again.

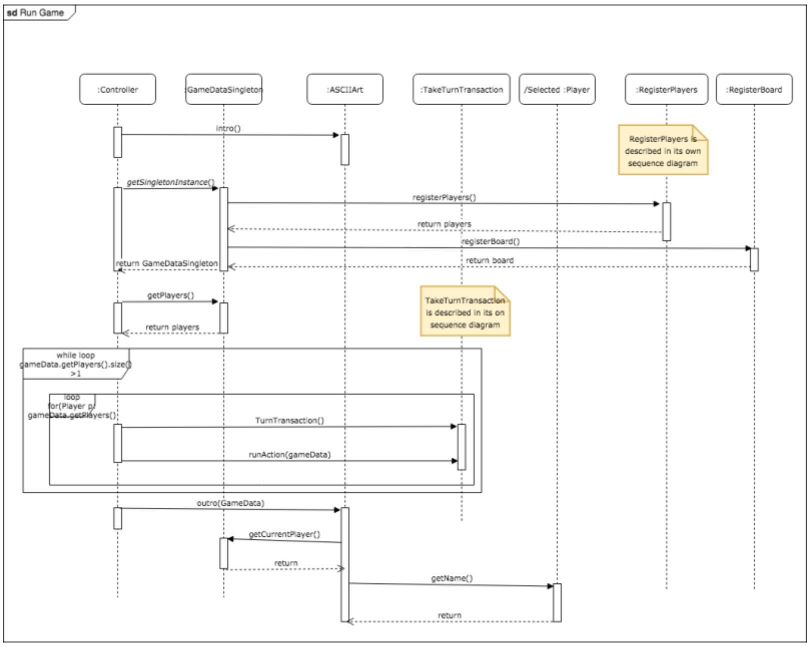
Class Relationship Model and Sequence Diagrams

[D.M]



Our System has over 20 classes. The game is made up of the entities; Player, Square and GameDataSingleton. There are three types of square, each with different runActions() (of which there is 10 implementations). Actions affect the state of the players and board. GameDataSingleton stores the players and the board and instantiates these in RegisterBoard and RegisterPlayers. The game is ran by the controller which performs Player turns till only one is left. Read and ListPrint are for in/outputs.

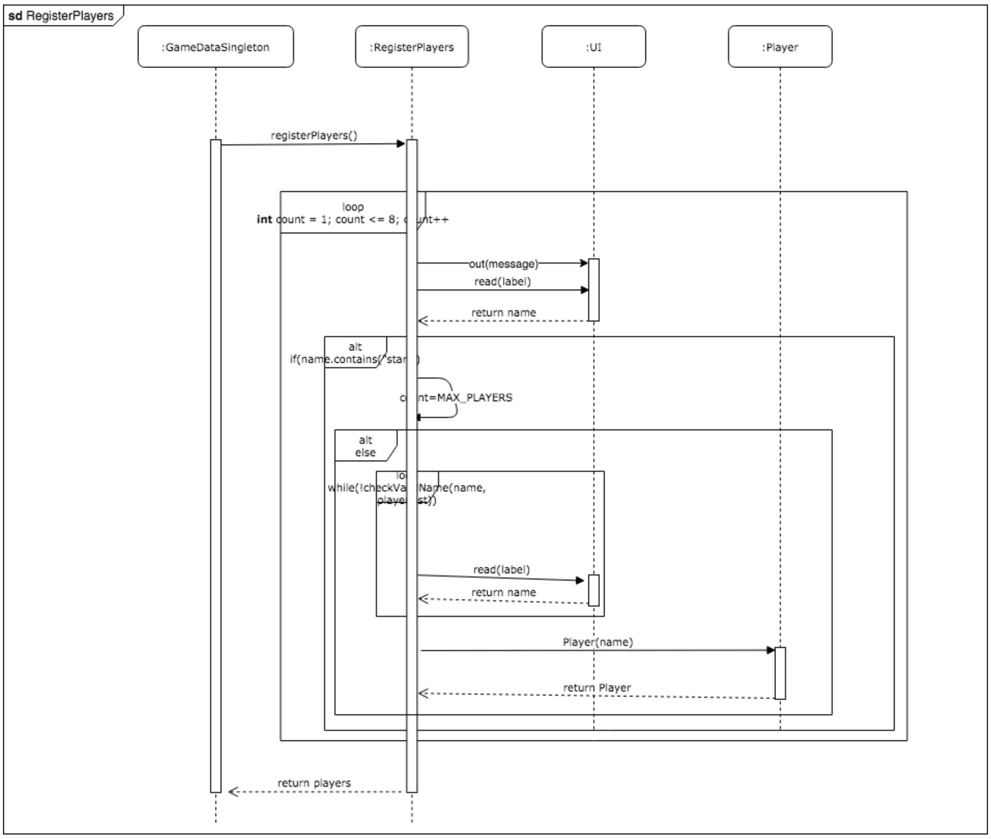
Run Game [D.M]



Commentary

The run game sequence diagram describes the actions the controller takes to allow the game to run

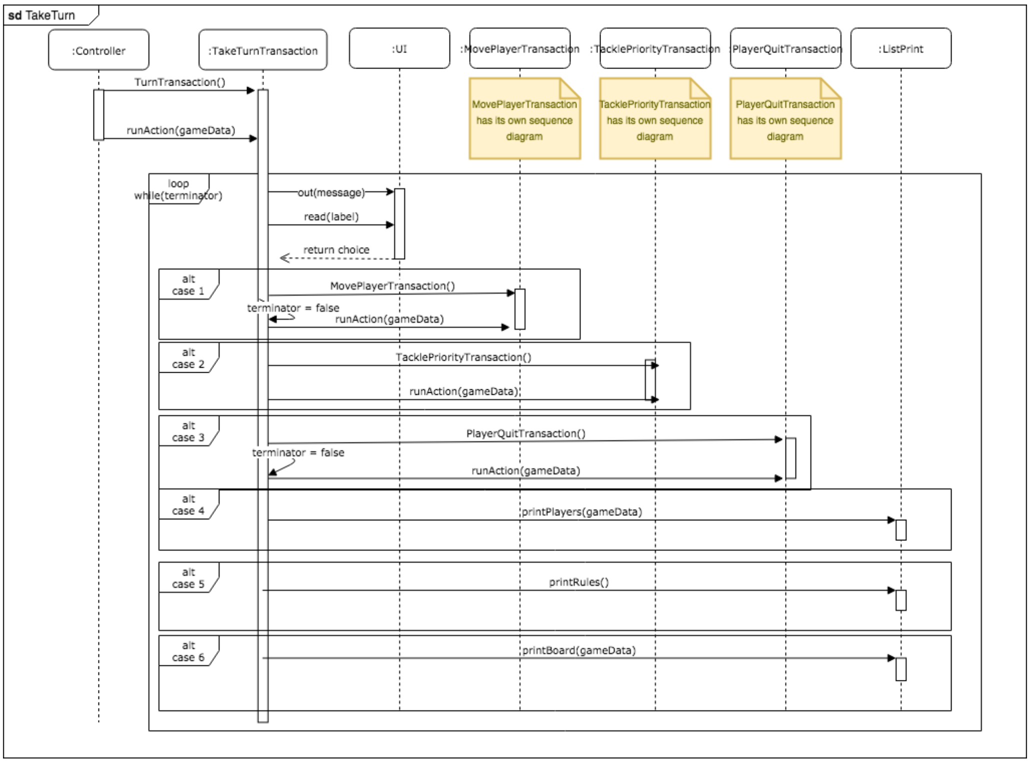
Register Players [D.M]



Commentary

The Register Players sequence diagram describes the program registers players and adds them to Game Data

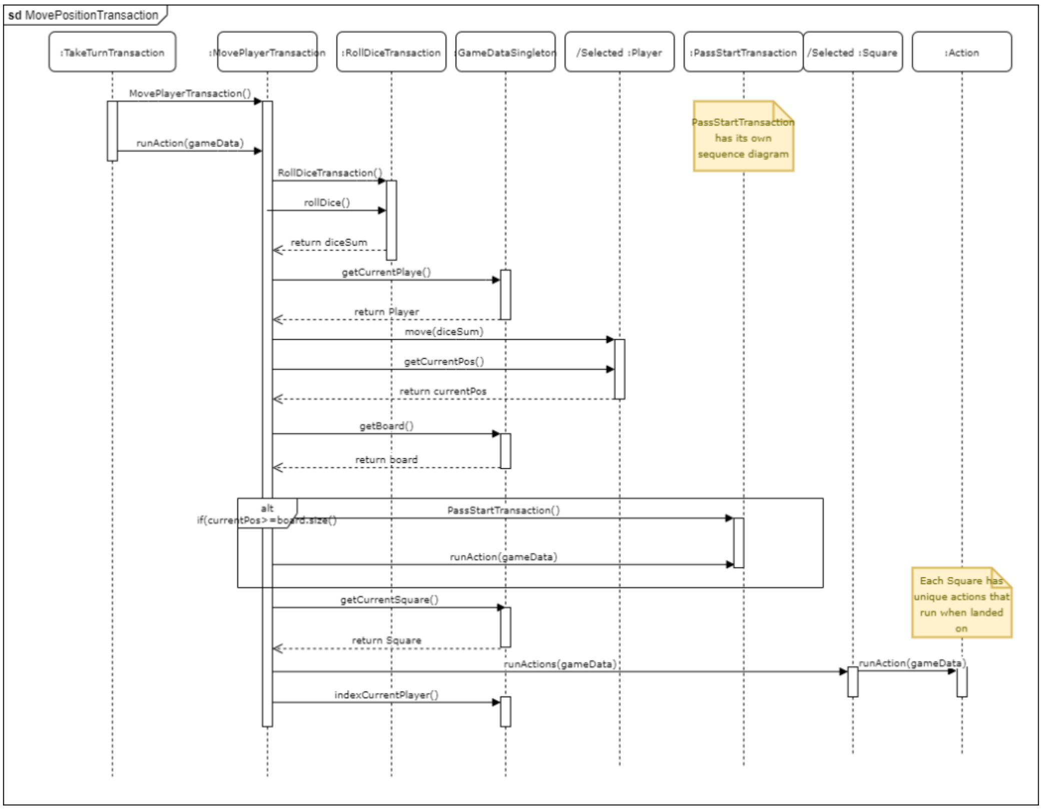
Take Turn [D.M]



Commentary

The Take Turn sequence diagram describes how each player can take their turn

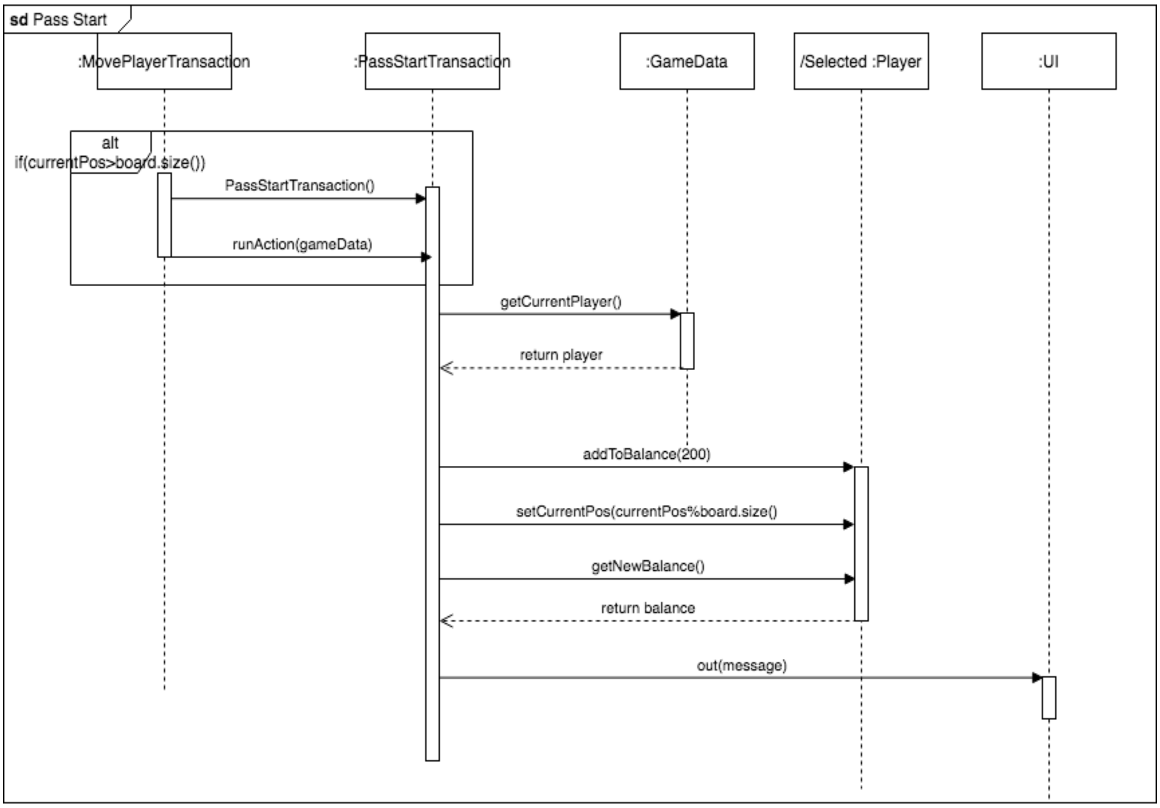
Move Player [N.M]



<u>Commentary</u>

The Move Player sequence diagram describes how the player moves position around the board

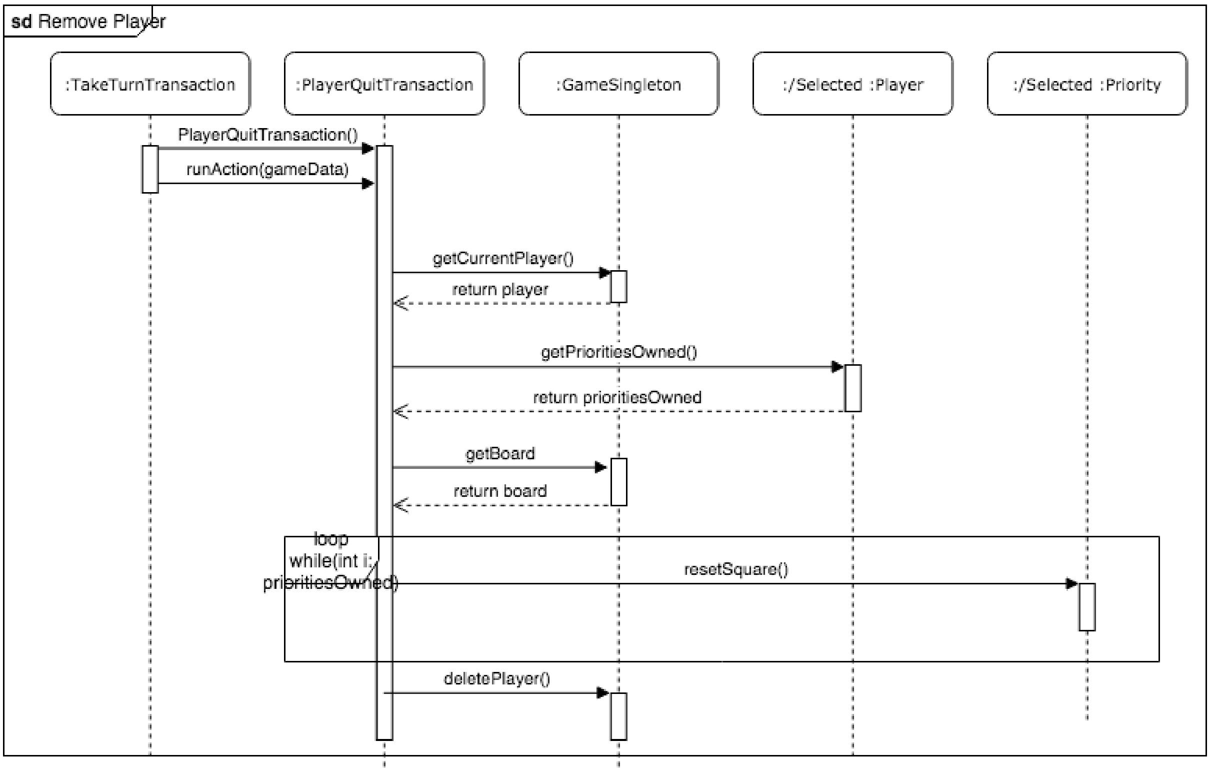
Pass Start [N.M]



<u>Commentary</u>

The Pass Start sequence diagram describes what happens when the player passes the starting square.

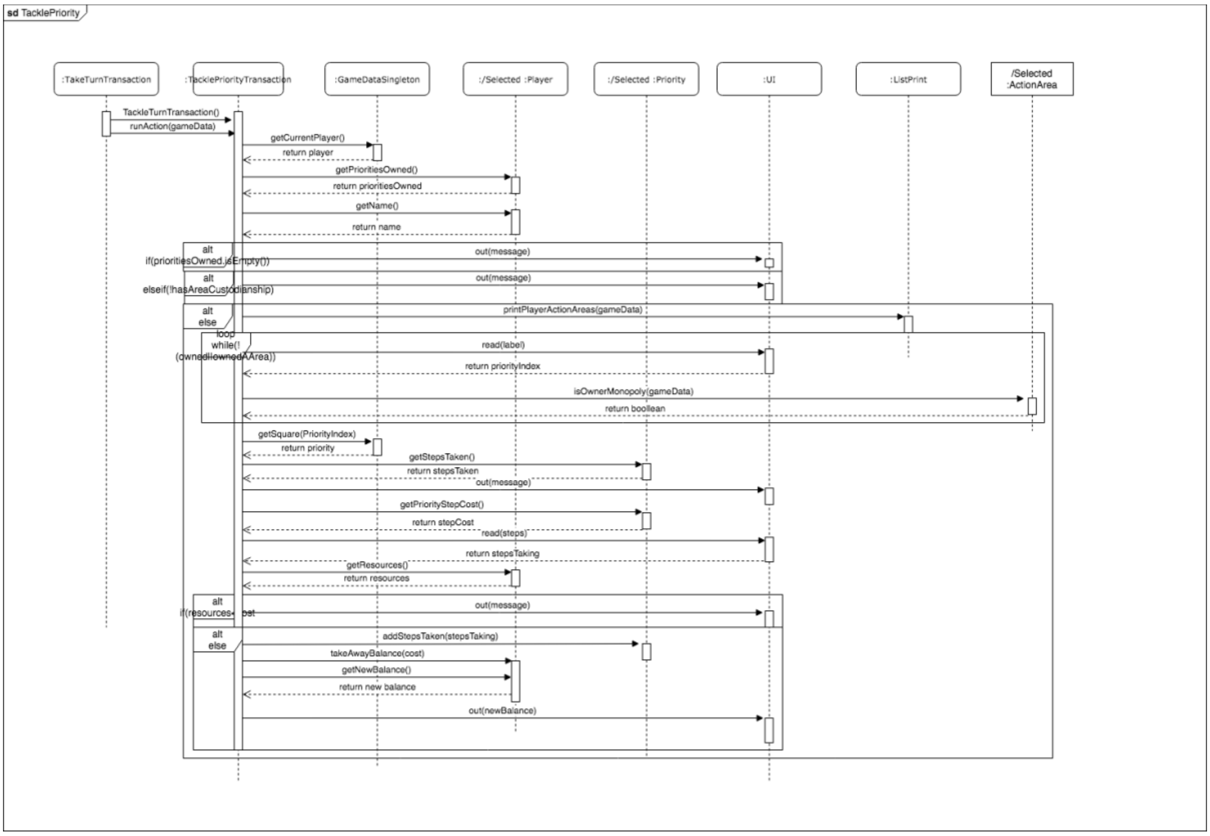
Player Quit [N.M]



Commentary

The Player Quit sequence diagram describes how the player can remove themselves from the game during their turn

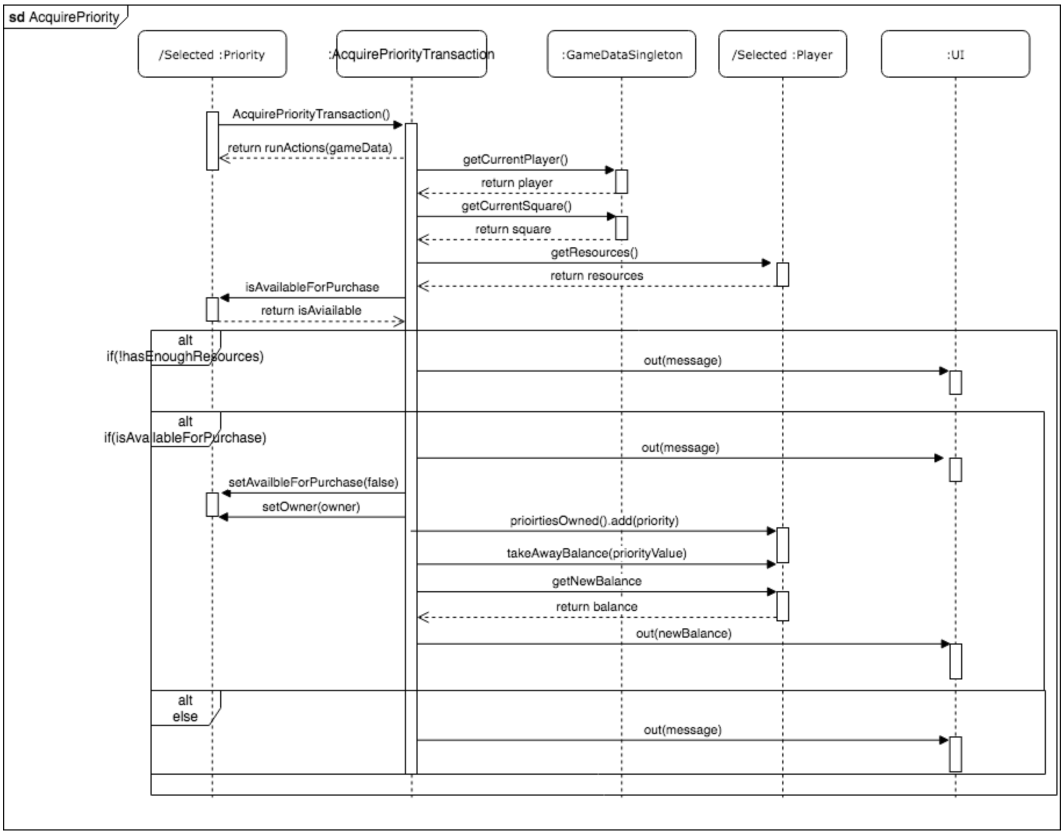
Tackle Priority [D.M]



Commentary

The Tackle Priority sequence diagram describes how the player is able to select a priority they own (during their turn) and purchases upgrades for that priority

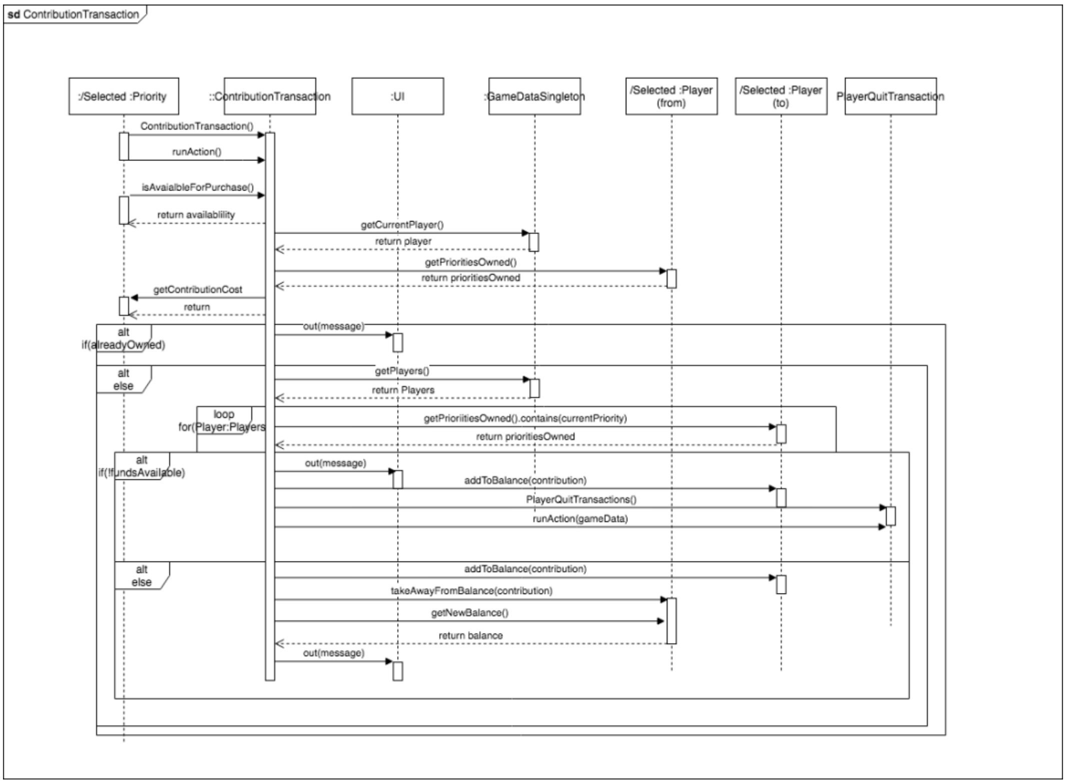
Acquire Priority [D.M]



Commentary

The Acquire Priority sequence diagram describes how the player is able to purchases a priority

Contribution [D.M]



Commentary

The Contribution Transaction sequence diagram describes how the player is to pay another player whose priority they have landed on

Final Game Layout

[N.M]

Action Areas

Rainforest

(1) Jaguar Protection Centre

(2) Tapir Foundation

(3) Tamarin Trust

Marine

(4) Save the Crocodiles

(5) Turtle Research Centre

(6) Seahorse Conservation Centre

Grasslands

(7) Reserve the Rhinos

(8) Wolves Trust

(9) Habitats for Giraffes

Polar Regions

(10) Save the Penguins

(11) Seal Foundation

Mountains

(13) Bear Foundation

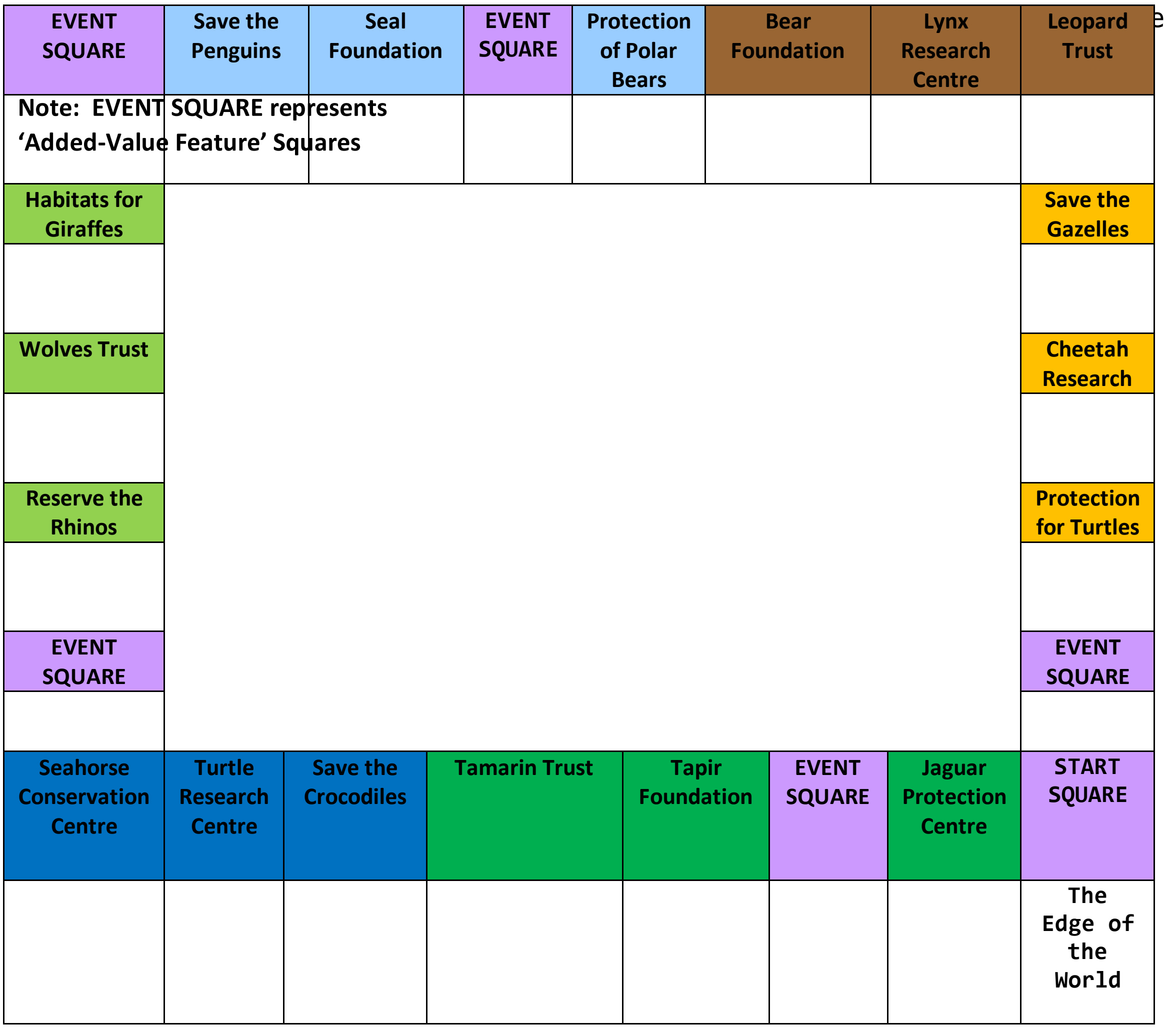
(14) Lynx Research Centre

Desert

(16) Save the Gazelles

(17) Cheetah Research

es



Note: EVENT SQUARE represents 'Added-Value Feature' Squares

Square 0 “The Edge of the World”	Square 1 Colour Dark Green “Jaguar Protection Centre”	Square 2 Colour Purple Surprise Event.	Square 3 Colour Dark Green “Tapir Foundation”	Square 4 Colour Dark Green “Save the Crocodiles”	Square 5 Colour Dark Blue “World Wildlife Trust”
Collect 200 Ɍ	Purchase cost 60R Step Cost 30R Contribution Costs: 2R 1Step(10R), 2Step(30R), 3Step(90R) Maj Step cost 250R	Potential Events Surprise Costs Surprise Donation Surprise Income Surprise Move	Purchase cost 60R Step Cost 30R Contribution Costs: 4R 1Step(20R), 2Step(60R), 3Step(180R) Maj Step cost 450R	Purchase cost 100R Step Cost 50R Contribution Costs: 6R 1Step(30R), 2Step(90R), 3Step(270R) Maj Step cost 550R	Purchase cost 100R Step Cost 50R Contribution Costs: 6R 1Step(30R), 2Step(90R), 3Step(270R) Maj Step cost 550R

Square 6 Colour Dark Blue “Turtle Research Centre”	Square 7 Colour Dark Blue “Seahorse Conservation Centre”	Square 8 Colour Purple Surprise Event.	Square 9 Colour Green “Sea Turtle Foundation”	Square 10 Colour Green “Reserve the Rhinos”	Square 11 Colour Green “Wolves Trust”
Purchase cost 120R Step Cost 60R Contribution Costs: 8R 1Step(40R), 2Step(100R), 3Step(3000R) Maj Step cost 600R	Purchase cost 140R Step Cost 70R Contribution Costs: 10R 1Step(50R), 2Step(150R), 3Step(450R) Maj Step cost 750R	Potential Events Surprise Costs Surprise Donation Surprise Income Surprise Move	Purchase cost 140R Step Cost 70R Contribution Costs: 10R 1Step(50R), 2Step(150R), 3Step(4500R) Maj Step cost 750R	Purchase cost 160R Step Cost 80R Contribution Costs: 12R 1Step(60R), 2Step(180R), 3Step(500R) Maj Step cost 900R	Purchase cost 180R Step Cost 90R Contribution Costs: 14R 1Step(70R), 2Step(200R), 3Step(550R) Maj Step cost 950R

Square 12 Colour Green “Habitats for Giraffes”	Square 13 Colour Purple Surprise Event.	Square 14 Colour Blue “Save the Penguins”	Square 15 Colour Blue “Seal Foundation”	Square 16 Colour Purple Surprise Event.	Square 17 Colour Blue “Protection of Polar Bears”
Purchase cost 180R Step Cost 90R Contribution Costs: 14R 1Step(70R), 2Step(200R), 3Step(550R) Maj Step cost 950R	Potential Events Surprise Costs Surprise Donation Surprise Income Surprise Move	Purchase cost 200R Step Cost 100R Contribution Costs: 16R 1Step(80R), 2Step(220R), 3Step(600R) Maj Step cost 1000R	Purchase cost 220R Step Cost 110R Contribution Costs: 18R 1Step(90R), 2Step(250R), 3Step(700R) Maj Step cost 1050R	Potential Events Surprise Costs Surprise Donation Surprise Income Surprise Move	Purchase cost 220R Step Cost 120R Contribution Costs: 18R 1Step(90R), 2Step(250R), 3Step(700R) Maj Step cost 1050R

Square 18 Colour Brown “Bear Foundation”	Square 19 Colour Brown “Lynx Research Centre”	Square 20 Colour Blue “Leopard Trust”	Square 21 Colour Yellow “Save the Gazelles”	Square 22 Colour Yellow “Cheetah Research”	Square 23 Colour Yellow “Protection for Turtles”	Square 24 Colour Purple Surprise Event.
Purchase cost 240R Step Cost 120R Contribution Costs: 20R 1Step(100R), 2Step(300R), 3Step(750R) Maj Step cost 1100R	Purchase cost 260R Step Cost 130R Contribution Costs: 22R 1Step(110R), 2Step(330R), 3Step(800R) Maj Step cost 1150R	Purchase cost 280R Step Cost 140R Contribution Costs: 24R 1Step(120R), 2Step(360R), 3Step(850R) Maj Step cost 1200R	Purchase cost 280R Step Cost 140R Contribution Costs: 24R 1Step(120R), 2Step(360R), 3Step(850R) Maj Step cost 1200R	Purchase cost 280R Step Cost 140R Contribution Costs: 24R 1Step(120R), 2Step(360R), 3Step(850R) Maj Step cost 1200R	Purchase cost 300R Step Cost 150R Contribution Costs: 26R 1Step(130R), 2Step(380R), 3Step(900R) Maj Step cost 1250R	Potential Events Surprise Costs Surprise Donation Surprise Income Surprise Move

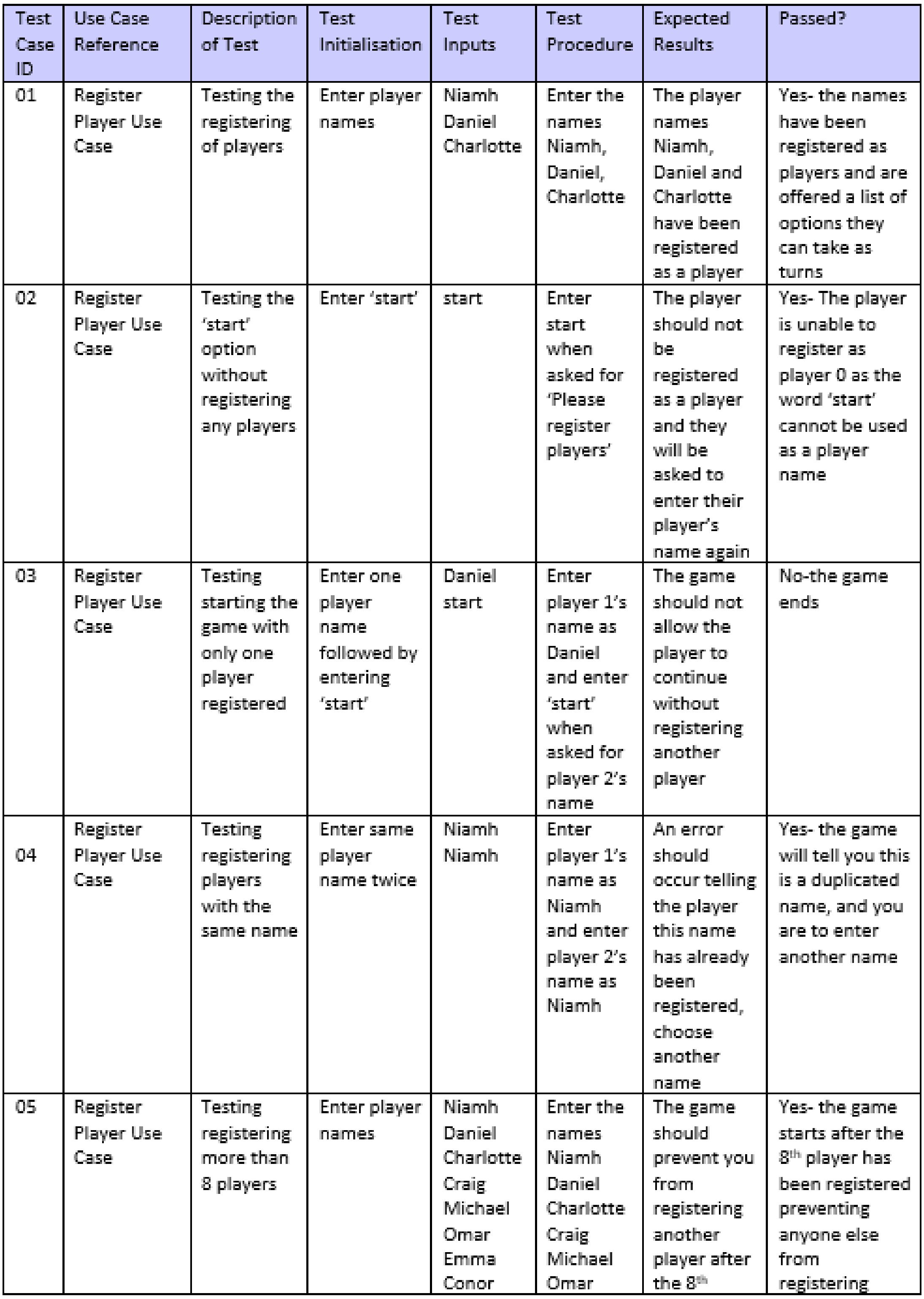
Implementation-Related Documentation

Test Plan

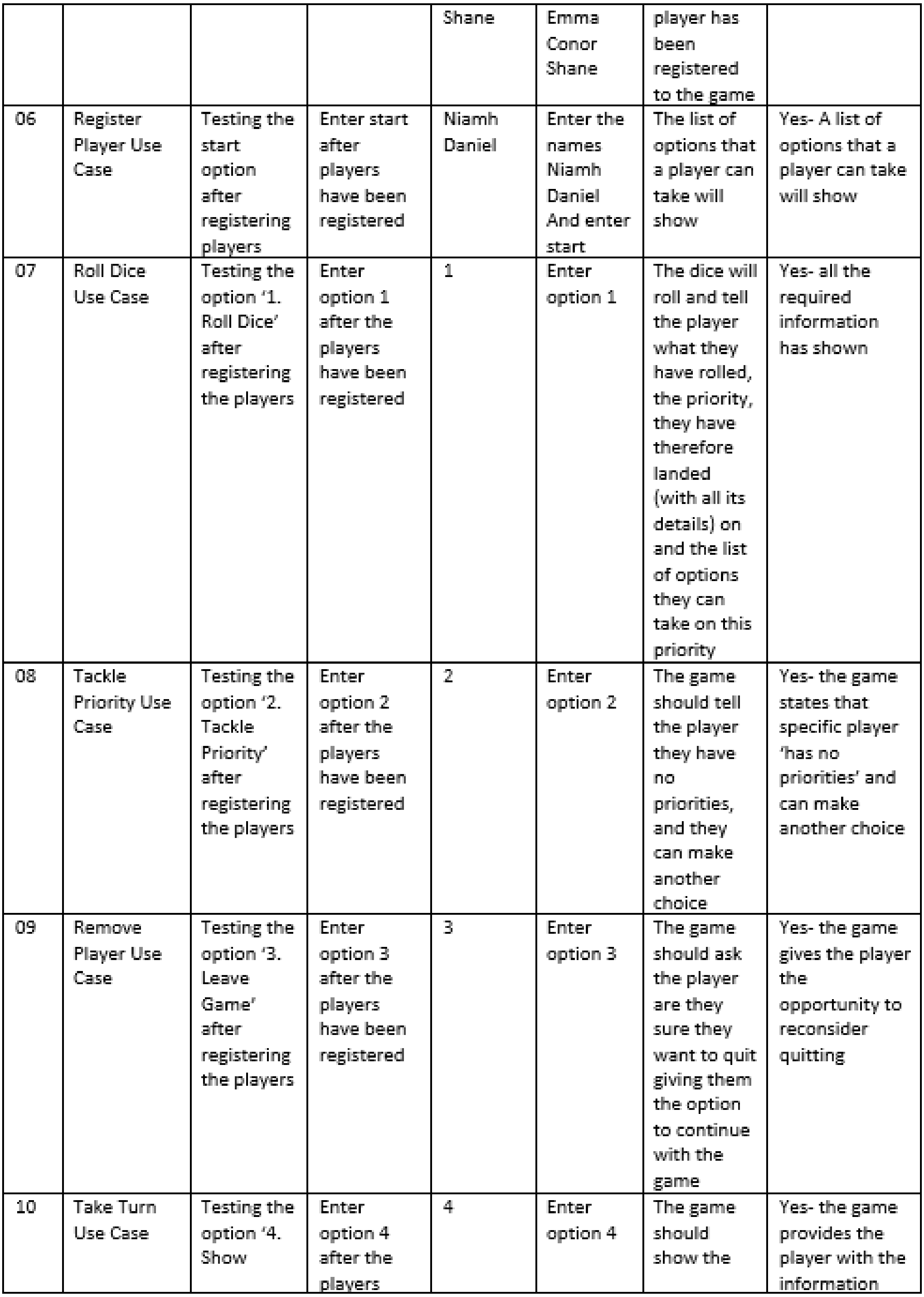
[N.M]

Test Acceptance

We have tested all key elements of system functionality through acceptance testing and Junit testing. Shown below is a sample of our acceptance test document.

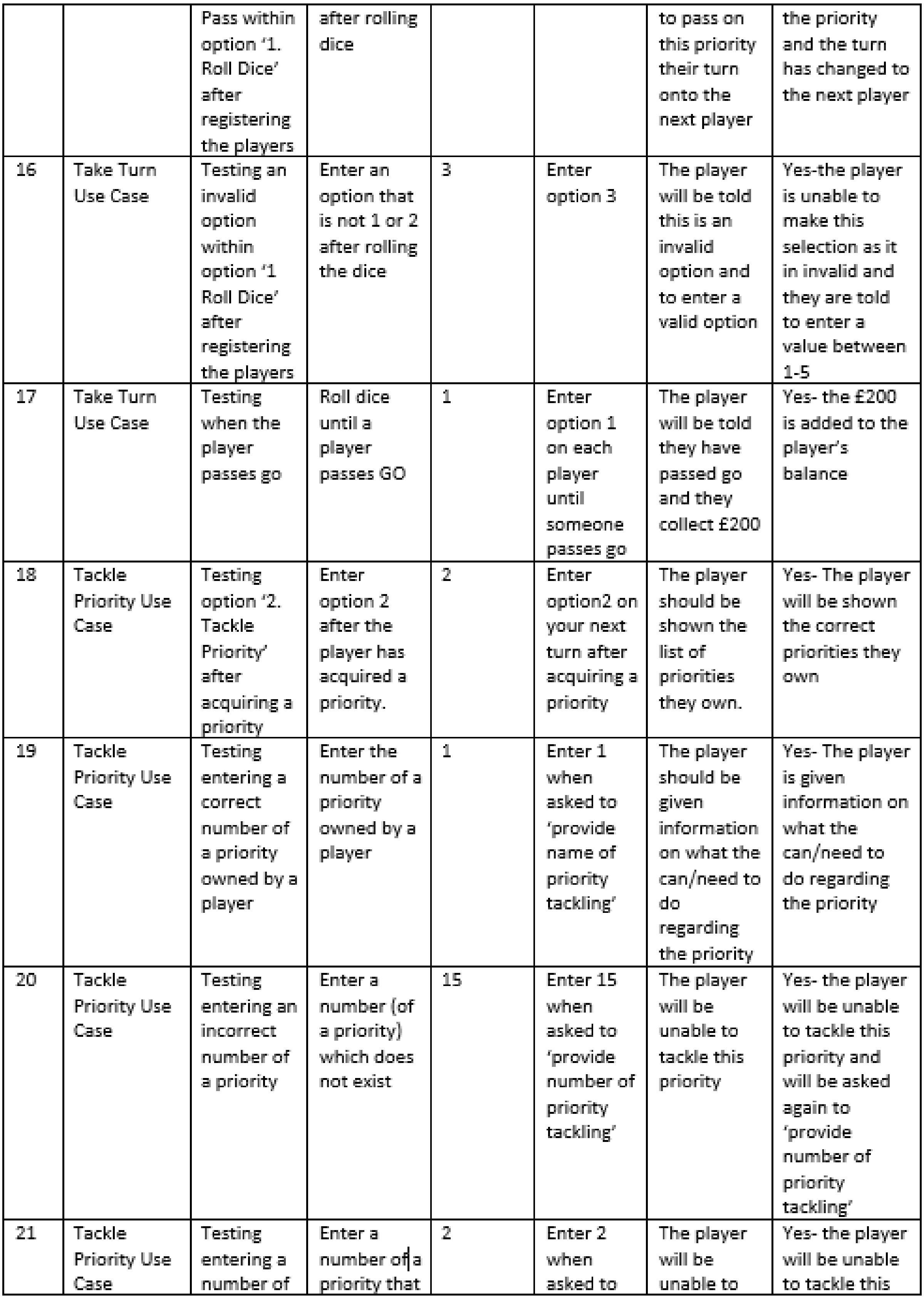


Test Case ID	Use Case Reference	Description of Test	Test Initialisation	Test Inputs	Test Procedure	Expected Results	Passed?
01	Register Player Use Case	Testing the registering of players	Enter player names	Niamh Daniel Charlotte	Enter the names Niamh, Daniel, Charlotte	The player names Niamh, Daniel and Charlotte have been registered as a player	Yes- the names have been registered as players and are offered a list of options they can take as turns
02	Register Player Use Case	Testing the 'start' option without registering any players	Enter 'start'	start	Enter start when asked for 'Please register players'	The player should not be registered as a player and they will be asked to enter their player's name again	Yes- The player is unable to register as player 0 as the word 'start' cannot be used as a player name
03	Register Player Use Case	Testing starting the game with only one player registered	Enter one player name followed by entering 'start'	Daniel start	Enter player 1's name as Daniel and enter 'start' when asked for player 2's name	The game should not allow the player to continue without registering another player	No-the game ends
04	Register Player Use Case	Testing registering players with the same name	Enter same player name twice	Niamh Niamh	Enter player 1's name as Niamh and enter player 2's name as Niamh	An error should occur telling the player this name has already been registered, choose another name	Yes- the game will tell you this is a duplicated name, and you are to enter another name
05	Register Player Use Case	Testing registering more than 8 players	Enter player names	Niamh Daniel Charlotte Craig Michael Omar Emma Conor	Enter the names Niamh Daniel Charlotte Craig Michael Omar	The game should prevent you from registering another player after the 8th	Yes- the game starts after the 8th player has been registered preventing anyone else from registering

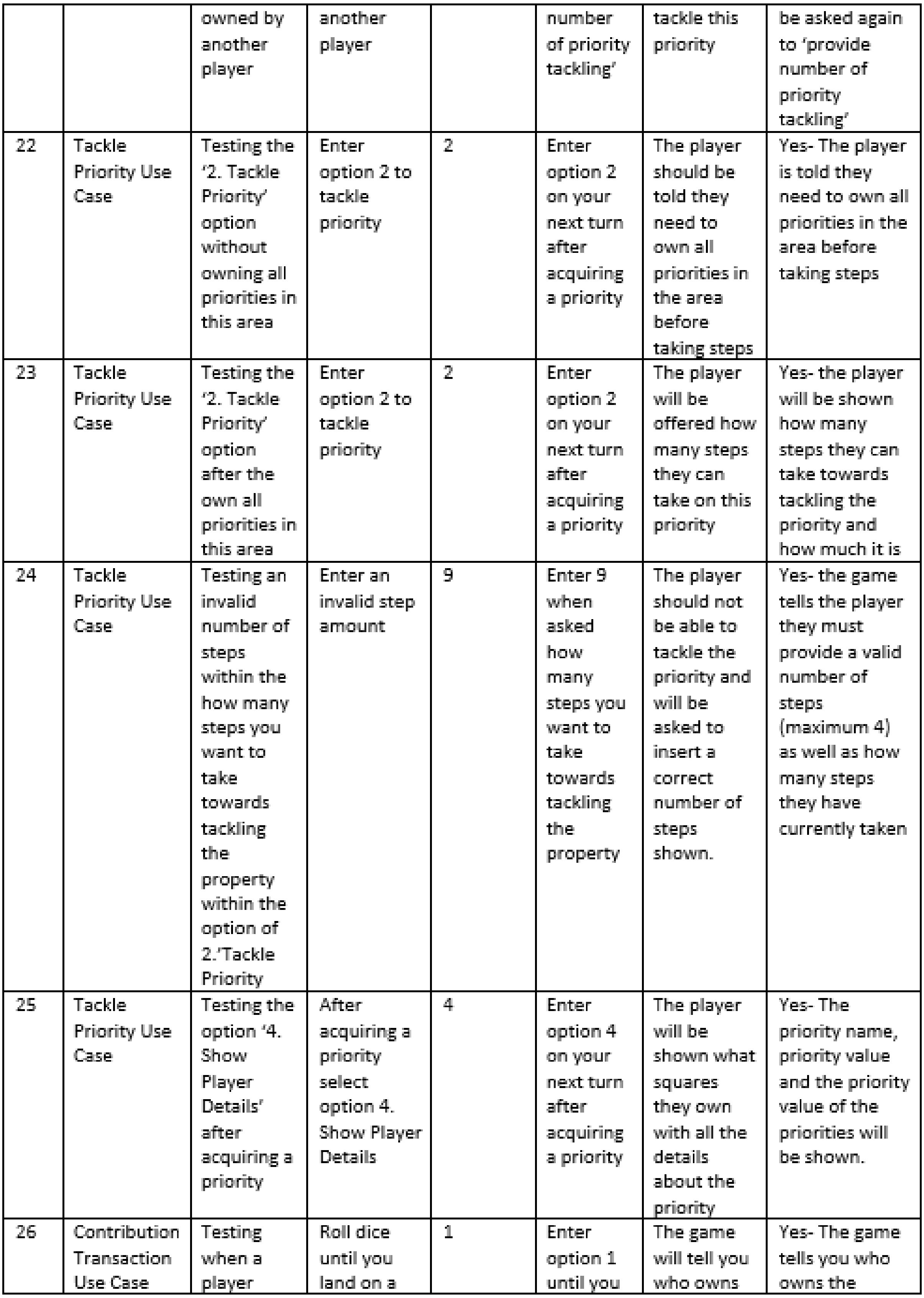


				Shane	Emma Conor Shane	player has been registered to the game	
06	Register Player Use Case	Testing the start option after registering players	Enter start after players have been registered	Niamh Daniel	Enter the names Niamh Daniel And enter start	The list of options that a player can take will show	Yes- A list of options that a player can take will show
07	Roll Dice Use Case	Testing the option '1. Roll Dice' after registering the players	Enter option 1 after the players have been registered	1	Enter option 1	The dice will roll and tell the player what they have rolled, the priority, they have therefore landed (with all its details) on and the list of options they can take on this priority	Yes- all the required information has shown
08	Tackle Priority Use Case	Testing the option '2. Tackle Priority' after registering the players	Enter option 2 after the players have been registered	2	Enter option 2	The game should tell the player they have no priorities, and they can make another choice	Yes- the game states that specific player 'has no priorities' and can make another choice
09	Remove Player Use Case	Testing the option '3. Leave Game' after registering the players	Enter option 3 after the players have been registered	3	Enter option 3	The game should ask the player are they sure they want to quit giving them the option to continue with the game	Yes- the game gives the player the opportunity to reconsider quitting
10	Take Turn Use Case	Testing the option '4. Show	Enter option 4 after the players	4	Enter option 4	The game should show the	Yes- the game provides the player with the information

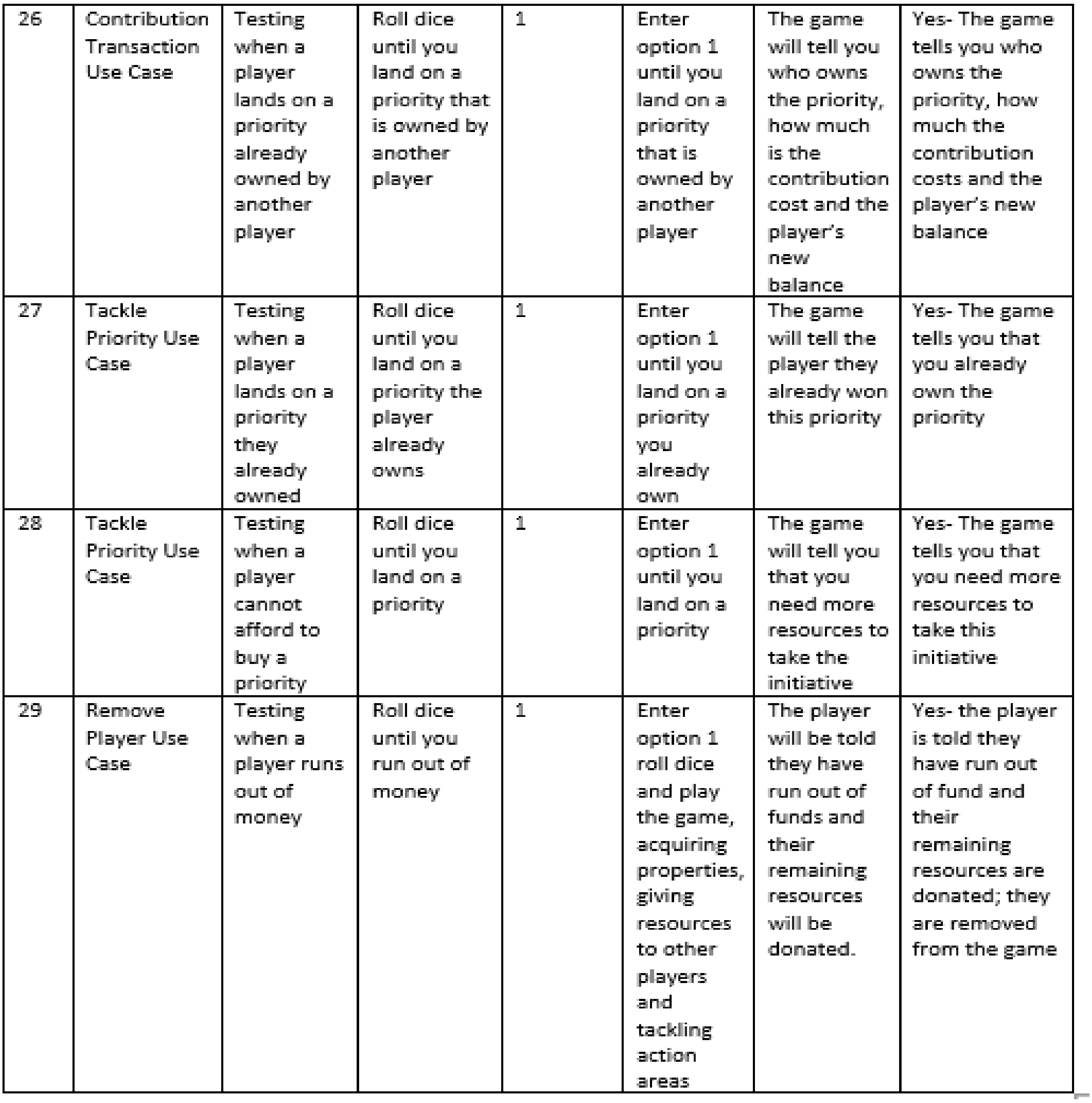
		Player Details' after registering the players	have been registered			player's details i.e. Name Balance Priorities own (if any), and they can make another choice	required and they can make another choice
11	Take Turn Use Case	Testing the option '5. Rules' after registering the players	Enter option 5 after the players have been registered	5	Enter option 5	The game should provide the player with a short set of rules	Yes- the game provides the player with the list of rules
12	Take Turn Use Case	Testing the option '6 Board' after registering the players	Enter option 6 after the players have been registered	6	Enter option 6	The game should provide the player with an overview of the board	Yes- the game provides the player with an overview of the board
13	Take Turn Use Case	Testing an invalid option choice i.e., a choice number that is not between 1-4 when it is a player's turn	Enter an option that is not between 1-4 after the players have been registered	5	Enter an invalid option of 5	The game should tell the player this is an invalid choice and allow them to enter a valid choice	Yes- the game tells the player to enter a valid choice
14	Tackle Priority Use Case	Testing the option '1. Acquire Priority' (with sufficient funds) within option '1. Roll Dice' after registering the players	Enter option 1 after rolling dice	1	Enter option 1	The player will have acquired the priority as they have sufficient funds, and their new balance will be shown. The player turn count will increase	Yes- The player has acquired the priority and their new balance is shown. The next player can now take their turn
15	Take Turn Use Case	Testing the option '2.	Enter option 2	2	Enter option 2	The player will be able	Yes- the player can pass on the



		Pass within option '1. Roll Dice' after registering the players	after rolling dice			to pass on this priority their turn onto the next player	the priority and the turn has changed to the next player
16	Take Turn Use Case	Testing an invalid option within option '1 Roll Dice' after registering the players	Enter an option that is not 1 or 2 after rolling the dice	3	Enter option 3	The player will be told this is an invalid option and to enter a valid option	Yes-the player is unable to make this selection as it in invalid and they are told to enter a value between 1-5
17	Take Turn Use Case	Testing when the player passes go	Roll dice until a player passes GO	1	Enter option 1 on each player until someone passes go	The player will be told they have passed go and they collect £200	Yes- the £200 is added to the player's balance
18	Tackle Priority Use Case	Testing option '2. Tackle Priority' after acquiring a priority	Enter option 2 after the player has acquired a priority.	2	Enter option2 on your next turn after acquiring a priority	The player should be shown the list of priorities they own.	Yes- The player will be shown the correct priorities they own
19	Tackle Priority Use Case	Testing entering a correct number of a priority owned by a player	Enter the number of a priority owned by a player	1	Enter 1 when asked to 'provide name of priority tackling'	The player should be given information on what the can/need to do regarding the priority	Yes- The player is given information on what the can/need to do regarding the priority
20	Tackle Priority Use Case	Testing entering an incorrect number of a priority	Enter a number (of a priority) which does not exist	15	Enter 15 when asked to 'provide number of priority tackling'	The player will be unable to tackle this priority	Yes- the player will be unable to tackle this priority and will be asked again to 'provide number of priority tackling'
21	Tackle Priority Use Case	Testing entering a number of	Enter a number of a priority that	2	Enter 2 when asked to	The player will be unable to	Yes- the player will be unable to tackle this



		owned by another player	another player		number of priority tackling'	tackle this priority	be asked again to 'provide number of priority tackling'
22	Tackle Priority Use Case	Testing the '2. Tackle Priority' option without owning all priorities in this area	Enter option 2 to tackle priority	2	Enter option 2 on your next turn after acquiring a priority	The player should be told they need to own all priorities in the area before taking steps	Yes- The player is told they need to own all priorities in the area before taking steps
23	Tackle Priority Use Case	Testing the '2. Tackle Priority' option after the own all priorities in this area	Enter option 2 to tackle priority	2	Enter option 2 on your next turn after acquiring a priority	The player will be offered how many steps they can take on this priority	Yes- the player will be shown how many steps they can take towards tackling the priority and how much it is
24	Tackle Priority Use Case	Testing an invalid number of steps within the how many steps you want to take towards tackling the property within the option of 2.'Tackle Priority	Enter an invalid step amount	9	Enter 9 when asked how many steps you want to take towards tackling the property	The player should not be able to tackle the priority and will be asked to insert a correct number of steps shown.	Yes- the game tells the player they must provide a valid number of steps (maximum 4) as well as how many steps they have currently taken
25	Tackle Priority Use Case	Testing the option '4. Show Player Details' after acquiring a priority	After acquiring a priority select option 4. Show Player Details	4	Enter option 4 on your next turn after acquiring a priority	The player will be shown what squares they own with all the details about the priority	Yes- The priority name, priority value and the priority value of the priorities will be shown.
26	Contribution Transaction Use Case	Testing when a player	Roll dice until you land on a	1	Enter option 1 until you	The game will tell you who owns	Yes- The game tells you who owns the



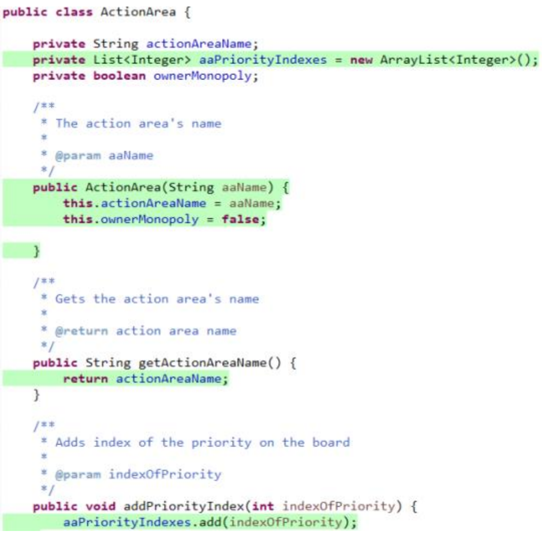
26	Contribution Transaction Use Case	Testing when a player lands on a priority already owned by another player	Roll dice until you land on a priority that is owned by another player	1	Enter option 1 until you land on a priority that is owned by another player	The game will tell you who owns the priority, how much is the contribution cost and the player's new balance	Yes- The game tells you who owns the priority, how much the contribution costs and the player's new balance
27	Tackle Priority Use Case	Testing when a player lands on a priority they already owned	Roll dice until you land on a priority the player already owns	1	Enter option 1 until you land on a priority you already own	The game will tell the player they already won this priority	Yes- The game tells you that you already own the priority
28	Tackle Priority Use Case	Testing when a player cannot afford to buy a priority	Roll dice until you land on a priority	1	Enter option 1 until you land on a priority	The game will tell you that you need more resources to take the initiative	Yes- The game tells you that you need more resources to take this initiative
29	Remove Player Use Case	Testing when a player runs out of money	Roll dice until you run out of money	1	Enter option 1 roll dice and play the game, acquiring properties, giving resources to other players and tackling action areas	The player will be told they have run out of funds and their remaining resources will be donated.	Yes- the player is told they have run out of fund and their remaining resources are donated; they are removed from the game

Junit Testing

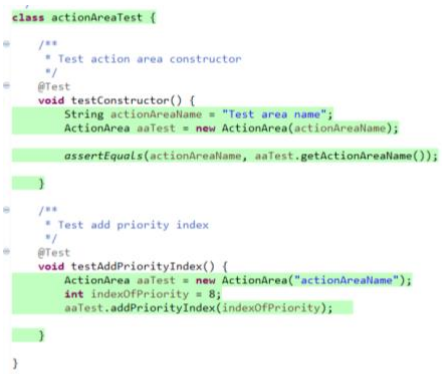
[N.M; D.M]

Shown below are some sample screenshots of our Junit testing.

ActionArea Junit Test

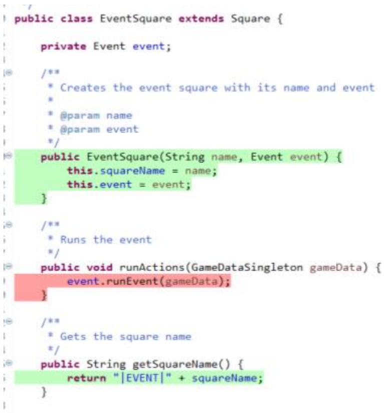


```java
public class ActionArea {

    private String actionAreaName;
    private List<Integer> aaPriorityIndexes = new ArrayList<Integer>();
    private boolean ownerMonopoly;

    /**
     * The action area's name
     *
     * @param aaName
     */
    public ActionArea(String aaName) {
        this.actionAreaName = aaName;
        this.ownerMonopoly = false;

    }

    /**
     * Gets the action area's name
     *
     * @return action area name
     */
    public String getActionAreaName() {
        return actionAreaName;
    }

    /**
     * Adds index of the priority on the board
     *
     * @param indexOfPriority
     */
    public void addPriorityIndex(int indexOfPriority) {
        aaPriorityIndexes.add(indexOfPriority);
```

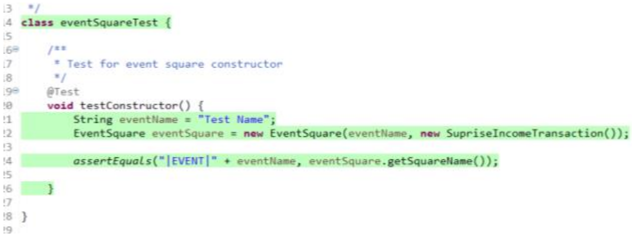


```java
class actionAreaTest {

    /**
     * Test action area constructor
     */
    @Test
    void testConstructor() {
        String actionAreaName = "Test area name";
        ActionArea aaTest = new ActionArea(actionAreaName);

        assertEquals(actionAreaName, aaTest.getActionAreaName());

    }

    /**
     * Test add priority index
     */
    @Test
    void testAddPriorityIndex() {
        ActionArea aaTest = new ActionArea("actionAreaName");
        int indexOfPriority = 8;
        aaTest.addPriorityIndex(indexOfPriority);

    }

}
```

EventSquare Junit Test

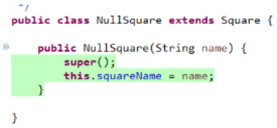


```java
public class EventSquare extends Square {

    private Event event;

    /**
     * Creates the event square with its name and event
     *
     * @param name
     * @param event
     */
    public EventSquare(String name, Event event) {
        this.squareName = name;
        this.event = event;
    }

    /**
     * Runs the event
     */
    public void runActions(GameDataSingleton gameData) {
        event.runEvent(gameData);
    }

    /**
     * Gets the square name
     */
    public String getSquareName() {
        return "|EVENT|" + squareName;
    }
```

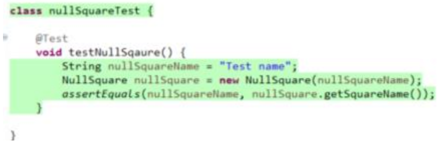


```java
 */
class eventSquareTest {

    /**
     * Test for event square constructor
     */
    @Test
    void testConstructor() {
        String eventName = "Test Name";
        EventSquare eventSquare = new EventSquare(eventName, new SupriseIncomeTransaction());

        assertEquals("|EVENT|" + eventName, eventSquare.getSquareName());

    }

}
```

NullSquare Junit Test

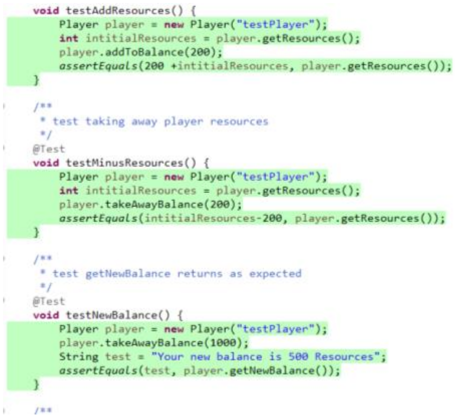


```java
 */
public class NullSquare extends Square {

    public NullSquare(String name) {
        super();
        this.squareName = name;
    }

}
```

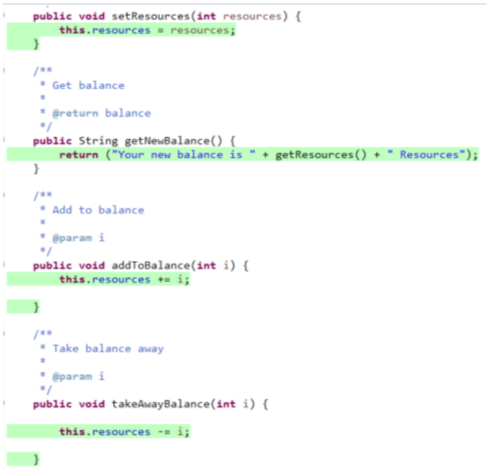


```java
class nullSquareTest {

    @Test
    void testNullSqaure() {
        String nullSquareName = "Test name";
        NullSquare nullSquare = new NullSquare(nullSquareName);
        assertEquals(nullSquareName, nullSquare.getSquareName());
    }

}
```

Player Junit Test

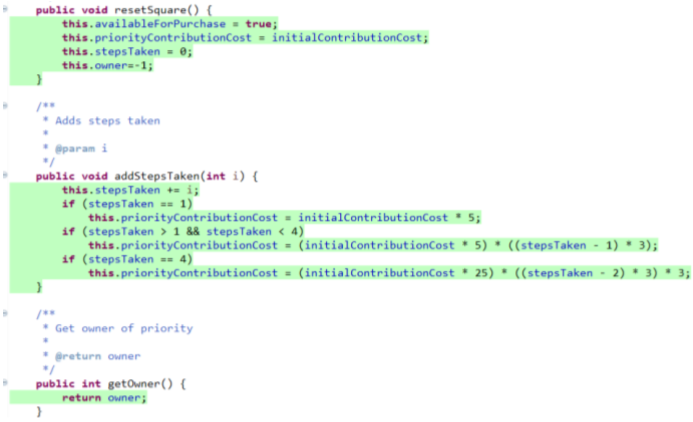


```
    void testAddResources() {
        Player player = new Player("testPlayer");
        int intitialResources = player.getResources();
        player.addToBalance(200);
        assertEquals(200 +intitialResources, player.getResources());
    }

    /**
     * test taking away player resources
     */
    @Test
    void testMinusResources() {
        Player player = new Player("testPlayer");
        int intitialResources = player.getResources();
        player.takeAwayBalance(200);
        assertEquals(intitialResources-200, player.getResources());
    }

    /**
     * test getNewBalance returns as expected
     */
    @Test
    void testNewBalance() {
        Player player = new Player("testPlayer");
        player.takeAwayBalance(1000);
        String test = "Your new balance is 500 Resources";
        assertEquals(test, player.getNewBalance());
    }

    /**
```

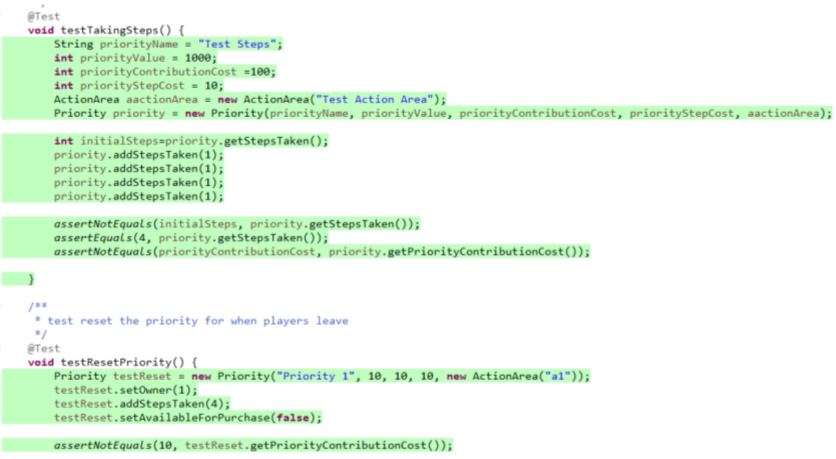


```
    public void setResources(int resources) {
        this.resources = resources;
    }

    /**
     * Get balance
     *
     * @return balance
     */
    public String getNewBalance() {
        return ("Your new balance is " + getResources() + " Resources");
    }

    /**
     * Add to balance
     *
     * @param i
     */
    public void addToBalance(int i) {
        this.resources += i;

    }

    /**
     * Take balance away
     *
     * @param i
     */
    public void takeAwayBalance(int i) {

        this.resources -= i;

    }
```

Priority Junit Test



```
    public void resetSquare() {
        this.availableForPurchase = true;
        this.priorityContributionCost = initialContributionCost;
        this.stepsTaken = 0;
        this.owner=-1;
    }

    /**
     * Adds steps taken
     *
     * @param i
     */
    public void addStepsTaken(int i) {
        this.stepsTaken += i;
        if (stepsTaken == 1)
            this.priorityContributionCost = initialContributionCost * 5;
        if (stepsTaken > 1 && stepsTaken < 4)
            this.priorityContributionCost = (initialContributionCost * 5) * ((stepsTaken - 1) * 3);
        if (stepsTaken == 4)
            this.priorityContributionCost = (initialContributionCost * 25) * ((stepsTaken - 2) * 3) * 3;
    }

    /**
     * Get owner of priority
     *
     * @return owner
     */
    public int getOwner() {
        return owner;
    }
```



```
    @Test
    void testTakingSteps() {
        String priorityName = "Test Steps";
        int priorityValue = 1000;
        int priorityContributionCost =100;
        int priorityStepCost = 10;
        ActionArea aactionArea = new ActionArea("Test Action Area");
        Priority priority = new Priority(priorityName, priorityValue, priorityContributionCost, priorityStepCost, aactionArea);

        int initialSteps=priority.getStepsTaken();
        priority.addStepsTaken(1);
        priority.addStepsTaken(1);
        priority.addStepsTaken(1);
        priority.addStepsTaken(1);

        assertNotEquals(initialSteps, priority.getStepsTaken());
        assertEquals(4, priority.getStepsTaken());
        assertNotEquals(priorityContributionCost, priority.getPriorityContributionCost());

    }

    /**
     * test reset the priority for when players leave
     */
    @Test
    void testResetPriority() {
        Priority testReset = new Priority("Priority 1", 10, 10, 10, new ActionArea("a1"));
        testReset.setOwner(1);
        testReset.addStepsTaken(4);
        testReset.setAvailableForPurchase(false);

        assertNotEquals(10, testReset.getPriorityContributionCost());
```

Appendix

Adherence to Process

(Submission 2 Team Meetings)

Minutes for CSC2058 Project: 25 Minutes **Week commencing:** 22/3/2021

Date of this minute: 26/3/2021

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)
Daniel Mason	DM
Niamh McLarnon	NML
Craig Mulligan	CM
Michael Kennedy	MK
Omar Ahmed Hassan Abdelfattah Bashah	OB
Charlotte Bisp	CB

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

PEER REVIEW

Name (Daniel):

- Completed code
- Completed class diagram
- Completed sequence diagrams
- Completed Junit testing

Name (Niamh):

- Completed text user interface
- Completed testing
- Completed Junit testing
- Completed pdf report

Name (Craig):

- Attended peer review
- Checked over gitlab

Name (Michael):

- Attended peer review
- Checked over gitlab

Name (Omar):

- Attended peer review
- Checked over gitlab

Name (Charlotte):

- Attended peer review
- Checked over gitlab

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

Name (Daniel):

- Create demo video
- Submit demo video
- Submit code

Name (Niamh):

- Submit pdf

Name (Craig):

Name (Michael):

Name (Omar):

Name (Charlotte):

Minutes for CSC2058 Project: 25 Minutes **Week commencing:** 25/1/2021

Date of this minute: 25/1/2021

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)
Daniel Mason	DM
Niamh McLarnon	NML
Craig Mulligan	CM
Michael Kennedy	MK
Omar Ahmed Hassan Abdelfattah Bashah	OB
Charlotte Bisp	CB

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

Name (Daniel):
- Reviewed Feedback for Submission 1
- Video call meeting with Ian to review submission 1

Name (Niamh):
- Contacted Ian for video call feedback on Submission 1 feedback
- Reviewed feedback for Submission 1
- Video call meeting with Ian to review submission 1

Name (Craig):

Name (Michael):

Name (Omar):

Name (Charlotte):

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

Name (Daniel):
- Draft layout for Final PDF report

Name (Niamh):
- Planned timetable for completing submission 2 deliverables

Name (Craig):

Name (Michael):

Name (Omar):

Name (Charlotte):

Minutes for CSC2058 Project: 25 Minutes **Week commencing:** 1/2/2021

Date of this minute: 1/2/2021

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)
Daniel Mason	DM
Niamh McLarnon	NML
Craig Mulligan	CM
Michael Kennedy	MK
Omar Ahmed Hassan Abdelfattah Bashah	OB
Charlotte Bisp	CB

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

Name (Daniel):
- Created pdf file for finale report

Name (Niamh):
- Went over code and spec and made notes on what needs done for submission 2

Name (Craig):

Name (Michael):

Name (Omar):

Name (Charlotte):

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

Name (Daniel):
- Look at what needs to be done on code for submission 2
- Cleaned Repository

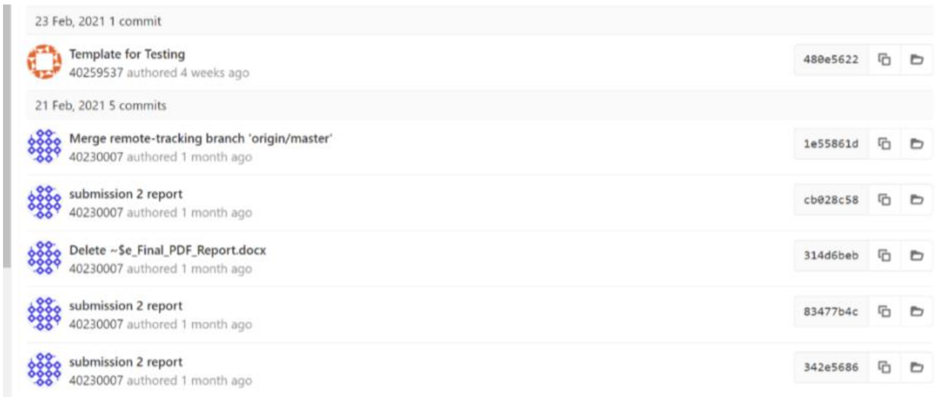
Name (Niamh):
- Plan what value-added features there will be
- Created template for testing

Name (Craig):

Name (Michael):

Name (Omar):

Name (Charlotte):



23 Feb, 2021 1 commit

Template for Testing
40259537 authored 4 weeks ago
480e5622

21 Feb, 2021 5 commits

Merge remote-tracking branch 'origin/master'
40230007 authored 1 month ago
1e55861d

submission 2 report
40230007 authored 1 month ago
cb028c58

Delete ~$e_Final_PDF_Report.docx
40230007 authored 1 month ago
314d6beb

submission 2 report
40230007 authored 1 month ago
83477b4c

submission 2 report
40230007 authored 1 month ago
342e5686

- **Minutes for CSC2058 Project:** 25 Minutes **Week commencing:** 8/2/2021
- **Date of this minute:** N/A

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

Name (Daniel):

- Continued to work on code

Name (Niamh):

- Started test plan

Name (Craig):

Name (Michael):

Name (Omar):

Name (Charlotte):

N/A no meeting this week

Minutes for CSC2058 Project: 25 Minutes **Week commencing:** 15/2/2021

Date of this minute: 15/2/2021

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)
Daniel Mason	DM
Niamh McLarnon	NML
Craig Mulligan	CM
Michael Kennedy	MK
Omar Ahmed Hassan Abdelfattah Bashah	OB
Charlotte Bisp	CB

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

Name (Daniel):

- Worked on code (add more of the basic functionalities, close to completing all basic functionalities)

Name (Niamh):

- Started working on testing documentation
- Minor updates to code

Name (Craig):

Name (Michael):

Name (Omar):

Name (Charlotte):

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

Name (Daniel):

- Finish basic functionalities of code
- Make action areas work

Name (Niamh):

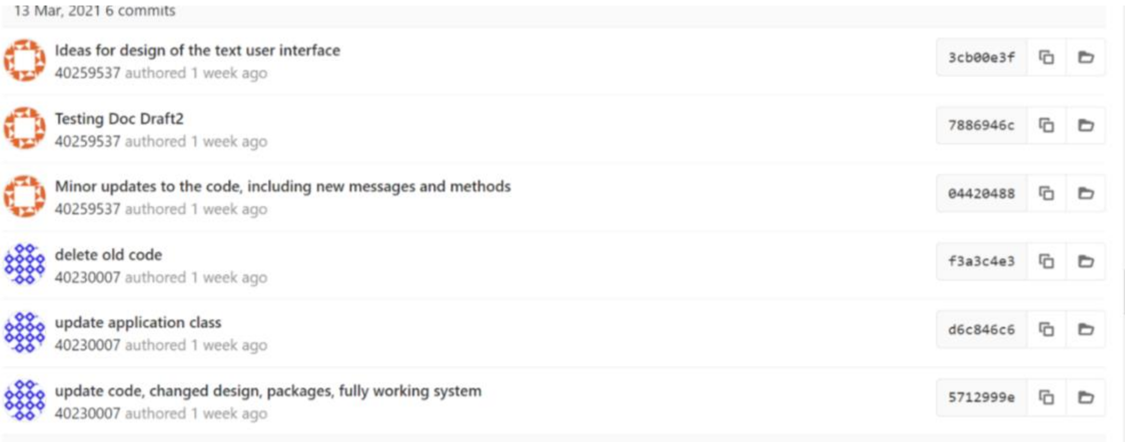
- Start Text user interface
- Continue on testing documentation

Name (Craig):

Name (Michael):

Name (Omar):

Name (Charlotte):



13 Mar, 2021 6 commits

Ideas for design of the text user interface
40259537 authored 1 week ago
3cb00e3f

Testing Doc Draft2
40259537 authored 1 week ago
7886946c

Minor updates to the code, including new messages and methods
40259537 authored 1 week ago
04420488

delete old code
40230007 authored 1 week ago
f3a3c4e3

update application class
40230007 authored 1 week ago
d6c846c6

update code, changed design, packages, fully working system
40230007 authored 1 week ago
5712999e

- **Minutes for CSC2058 Project:** 25 Minutes **Week commencing:** 22/2/2021

- **Date of this minute:** 22/2/2021

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)
Daniel Mason	DM
Niamh McLarnon	NML
Craig Mulligan	CM
Michael Kennedy	MK
Omar Ahmed Hassan Abdelfattah Bashah	OB
Charlotte Bisp	CB

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

Name (Daniel):
- Worked on the working system

Name (Niamh):
- Continued to work on testing

Name (Craig):

Name (Michael):

Name (Omar):

Name (Charlotte):

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

Name (Daniel):
- Continue code

Name (Niamh):
- Continue to work on testing

Name (Craig):

Name (Michael):

Name (Omar):

Name (Charlotte):

- **Minutes for CSC2058 Project:** 25 Minutes **Week commencing:** 1/3/2021

- **Date of this minute:** 1/3/2021

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)
Daniel Mason	DM
Niamh McLarnon	NML
Craig Mulligan	CM
Michael Kennedy	MK
Omar Ahmed Hassan Abdelfattah Bashah	OB
Charlotte Bisp	CB

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

Name (Daniel):
- Worked on code
- Cleaned repository

Name (Niamh):
- Worked on testing and text user interface

Name (Craig):

Name (Michael):

Name (Omar):

Name (Charlotte):

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

Name (Daniel):
- Continue to work on code

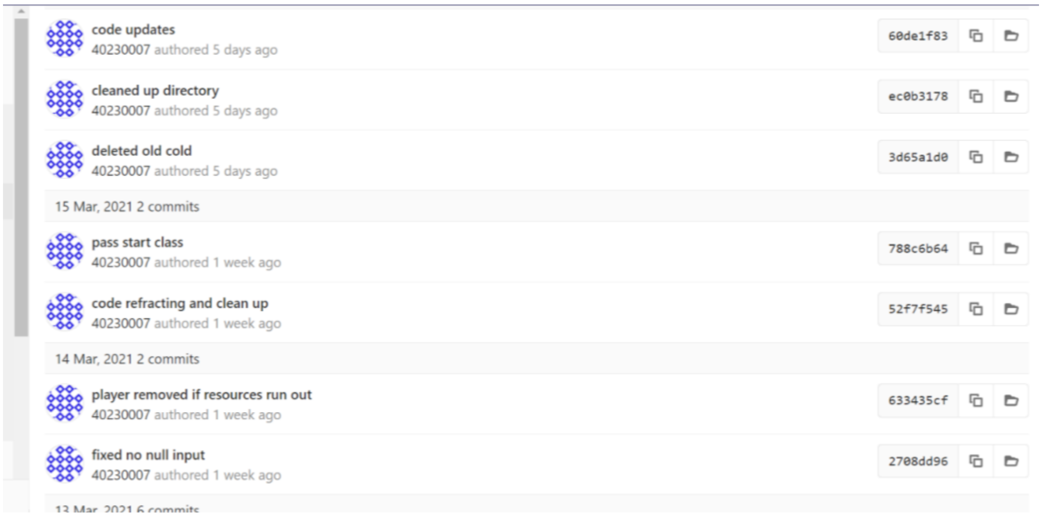
Name (Niamh):
- Continue working on testing

Name (Craig):

Name (Michael):

Name (Omar):

Name (Charlotte):



code updates
40230007 authored 5 days ago 60de1f83

cleaned up directory
40230007 authored 5 days ago ec0b3178

deleted old cold
40230007 authored 5 days ago 3d65a1d0

15 Mar, 2021 2 commits

pass start class
40230007 authored 1 week ago 788c6b64

code refracting and clean up
40230007 authored 1 week ago 52f7f545

14 Mar, 2021 2 commits

player removed if resources run out
40230007 authored 1 week ago 633435cf

fixed no null input
40230007 authored 1 week ago 2708dd96

13 Mar, 2021 6 commits

Minutes for CSC2058 Project: 25 Minutes **Week commencing:** 8/3/2021

Date of this minute: 8/3/2021

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)
Daniel Mason	DM
Niamh McLarnon	NML
Craig Mulligan	CM
Michael Kennedy	MK
Omar Ahmed Hassan Abdelfattah Bashah	OB
Charlotte Bisp	CB

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

Name (Daniel):
- Worked on code
- Applied SOLID to code
- Updated PDF document

Name (Niamh):
- Uploaded more testing
- Updated Game Layout

Name (Craig):

Name (Michael):

Name (Omar):

Name (Charlotte):

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

Name (Daniel):
- Continue to work on code

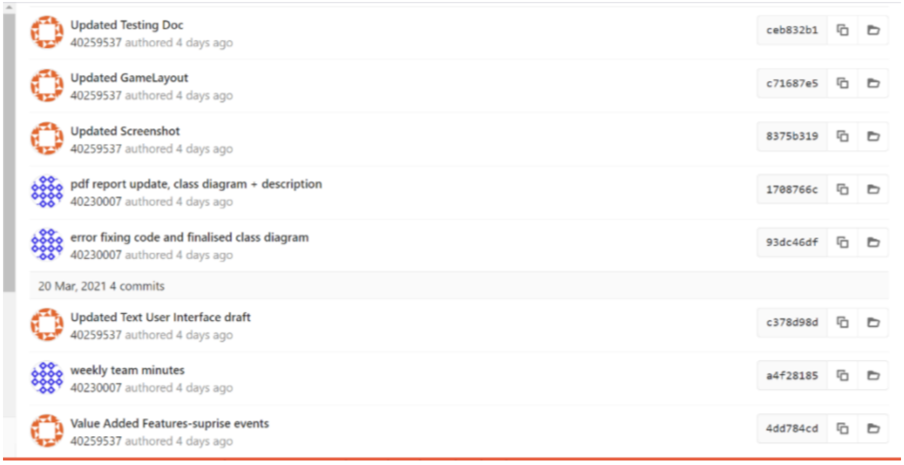
Name (Niamh):
- Continue to work testing
- Work on code

Name (Craig):

Name (Michael):

Name (Omar):

Name (Charlotte):



Updated Testing Doc
40259537 authored 4 days ago
ceb832b1

Updated GameLayout
40259537 authored 4 days ago
c71687e5

Updated Screenshot
40259537 authored 4 days ago
8375b319

pdf report update, class diagram + description
40230007 authored 4 days ago
1708766c

error fixing code and finalised class diagram
40230007 authored 4 days ago
93dc46df

20 Mar, 2021 4 commits

Updated Text User Interface draft
40259537 authored 4 days ago
c378d98d

weekly team minutes
40230007 authored 4 days ago
a4f28185

Value Added Features-suprise events
40259537 authored 4 days ago
4dd784cd

Minutes for CSC2058 Project: 25 Minutes **Week commencing:** 15/3/2021

Date of this minute: 15/3/2021

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)
Daniel Mason	DM
Niamh McLarnon	NML
Craig Mulligan	CM
Michael Kennedy	MK
Omar Ahmed Hassan Abdelfattah Bashah	OB
Charlotte Bisp	CB

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

Name (Daniel):

- Started class diagram
- Started use case realisations

Name (Niamh):

- Worked on text user interface
- Worked on testing

Name (Craig):

Name (Michael):

Name (Omar):

Name (Charlotte):

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

Name (Daniel):

- Finalise code, diagrams

Name (Niamh):

- Finalise pdf report

Name (Craig):

Name (Michael):

Name (Omar):

Name (Charlotte):

(Submission 1 Team Meetings)

Minutes for CSC2058 Project: 20 Minutes **Week commencing:** 12/10/2020

Date of this minute: 13/10/2020

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)
Daniel Mason	DM
Niamh McLarnon	NML
Craig Mulligan	CM
Michael Kennedy	MK
Omar Ahmed Hassan Abdelfattah Bashah	OB
Charlotte Bisp	CB

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

Name (Daniel):

- Introduction
- Strengths & Weaknesses

Name (Niamh):

- Introduction
- Strengths & Weaknesses

Name (Craig):

- Introduction
- Strengths & Weaknesses

Name (Michael):

- Introduction
- Strengths & Weaknesses

Name (Omar):

- Introduction
- Strengths & Weaknesses

Name (Charlotte):

- Introduction
- Strengths & Weaknesses

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

Name (Daniel):

- Read Project Requirements
- Look at Week 1-3 module notes
- Read Activity Plan

Name (Niamh):

- Read Project Requirements
- Look at Week 1-3 module notes
- Read Activity Plan

Name (Craig):

- Read Project Requirements
- Look at Week 1-3 module notes
- Read Activity Plan

Name (Michael):

- Read Project Requirements
- Look at Week 1-3 module notes
- Read Activity Plan

Name (Omar):

- Read Project Requirements
- Look at Week 1-3 module notes
- Read Activity Plan

Name (Charlotte):

- Read Project Requirements
- Look at Week 1-3 module notes
- Read Activity Plan

Minutes for CSC2058 Project: 25 Minutes **Week commencing:** 19/10/2020

Date of this minute: 22/10/2020

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)
Daniel Mason	DM
Niamh McLarnon	NML
Craig Mulligan	CM
Michael Kennedy	MK
Omar Ahmed Hassan Abdelfattah Bashah	OB
Charlotte Bisp	CB

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

Name (Daniel):

- Created 1st Draft Use Case Diagrams
- Started Code

Name (Niamh):

- Created 1st Draft Gantt Chart

Name (Craig):

- Read Project Requirements & informed group of deliverables

Name (Michael):

- Created 1st Draft of Class Diagram

Name (Omar):

- Read Project Requirements & looked at full module notes and recommended texts to date

Name (Charlotte):

- Organised team meeting

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

Name (Daniel):

- Review weeks deliverables from other team members & prepare feedback
- Develop Code

Name (Niamh):

- Review weeks deliverables from other team members & prepare feedback
- Create initial Use Case Descriptions deliverable

Name (Craig):

- Review weeks deliverables from other team members & prepare feedback
- Assist with code

Name (Michael):

- Review weeks deliverables from other team members & prepare feedback
- Continue to work on Class diagram

Name (Omar):

- Review weeks deliverables from other team members & prepare feedback
- Install IDE and learn Java code

Name (Charlotte):

- Review weeks deliverables from other team members & prepare feedback
- Assist with tasks

Minutes for CSC2058 Project: 25 Minutes

Week commencing: 26/10/2020

Date of this minute: 26/10/2020

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)
Daniel Mason	DM
Niamh McLarnon	NML
Craig Mulligan	CM
Michael Kennedy	MK
Omar Ahmed Hassan Abdelfattah Bashah	OB
Charlotte Bisp	CB

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

Name (Daniel):

- Contacted Queens to get GitLab repository given to team
- Committed files to Gitlab
- Worked on code
- Corrected Use Case Diagrams

Name (Niamh):

- Committed files to Gitlab
- Began 1st Draft of Use Case Descriptions deliverables

Name (Craig):

- Assisted with code
- Helped organise GitLab repository

Name (Michael):

- Worked on Class Diagram Deliverable

Name (Omar):

- Provided feedback/corrections for others deliverables and his java progress

Name (Charlotte):

- Organised team meeting
- Provided feedback/corrections for others deliverables

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

READING WEEK

Minutes for CSC2058 Project: 20 Minutes **Week commencing:** 09/11/2020

Date of this minute: 09/11/2020

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)
Daniel Mason	DM
Niamh McLarnon	NML
Craig Mulligan	CM
Michael Kennedy	MK
Omar Ahmed Hassan Abdelfattah Bashah	OB
Charlotte Bisp	CB

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

Name (Daniel):

- Worked on code
- Committed Weekly Team Minutes Documents
- Organised google meet to screen share code

Name (Niamh):

- Worked on Use Case Descriptions

Name (Craig):

- Gave feedback on code

Name (Michael):

- Looked over use case specification

Name (Omar):

- Provided feedback on project deliverables

Name (Charlotte):

- Organised team meeting

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

Name (Daniel):

- Finalise Use Case Diagrams
- Complete functioning code

Name (Niamh):

- Finalise Use Case Descriptions

Name (Craig):

- Review weeks deliverables from other team members & prepare feedback

Name (Michael):

- Review weeks deliverables from other team members & prepare feedback

Name (Omar):

- Review weeks deliverables from other team members & prepare feedback

Name (Charlotte):

- Review weeks deliverables from other team members & prepare feedback

Minutes for CSC2058 Project: 20 Minutes **Week commencing:** 16/11/2020

Date of this minute: 16/11/2020

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)
Daniel Mason	DM
Niamh McLarnon	NML
Craig Mulligan	CM
Michael Kennedy	MK
Omar Ahmed Hassan Abdelfattah Bashah	OB
Charlotte Bisp	CB

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

Name (Daniel):

- Worked on code

Name (Niamh):

- Worked on Gantt Chart

Name (Craig):

- Worked on use case realisations

Name (Michael):

- Gave feedback on deliverables

Name (Omar):

- Corrected Use Case Diagrams

Name (Charlotte):

- Created 1st draft game layout

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

Name (Daniel):

- Finalise code

Name (Niamh):

- Finalise Use Case Descriptions & Gantt Chart

Name (Craig):

- Create 1st draft Use Case Realisation

Name (Michael):

- Review weeks deliverables from other team members & prepare feedback

Name (Omar):

- Finalise Use Case Diagrams
- Create 1st Draft PDF report

Name (Charlotte):

- Finalise 1st Draft Game layout

Minutes for CSC2058 Project: 15 Minutes **Week commencing:** 23/11/2020

Date of this minute: 23/11/2020

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)
Daniel Mason	DM
Niamh McLarnon	NML
Craig Mulligan	CM
Michael Kennedy	MK
Omar Ahmed Hassan Abdelfattah Bashah	OB
Charlotte Bisp	CB

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

Name (Daniel):

- Continued work on code

Name (Niamh):

- Final draft of Gantt chart completed
- Use case descriptions completed

Name (Craig):

- Created first draft of use case realisations

Name (Michael):

- Delivered feedback on some of the new content on GitLab

Name (Omar):

- Provided feedback/corrections for use case diagrams

Name (Charlotte):

- Completed the game layout draft

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

Name (Daniel):

- Continue code to prep for video demo

Name (Niamh):

- Review weeks deliverables from other team members & prepare feedback

Name (Craig):

- Review and correct use case realisations.

Name (Michael):

- Review weeks deliverables from other team members & prepare feedback

Name (Omar):

- Review weeks deliverables from other team members & prepare feedback

Name (Charlotte):

- Review and correct game layout if necessary

Minutes for CSC2058 Project: 25 Minutes **Week commencing:** 30/11/2020

Date of this minute: 30/11/2020

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)
Daniel Mason	DM
Niamh McLarnon	NML
Craig Mulligan	CM
Michael Kennedy	MK
Omar Ahmed Hassan Abdelfattah Bashah	OB
Charlotte Bisp	CB

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

Name (Daniel):

- Continued code (ready for demo)
- Created use case realisations
- Created interim demo

Name (Niamh):

- Reviewed and corrected game layout
- Corrected realisations
- Helped with interim demo

Name (Craig):

- Reviewed code and provided feedback

Name (Michael):

- Reviewed deliverables and provided feedback

Name (Omar):

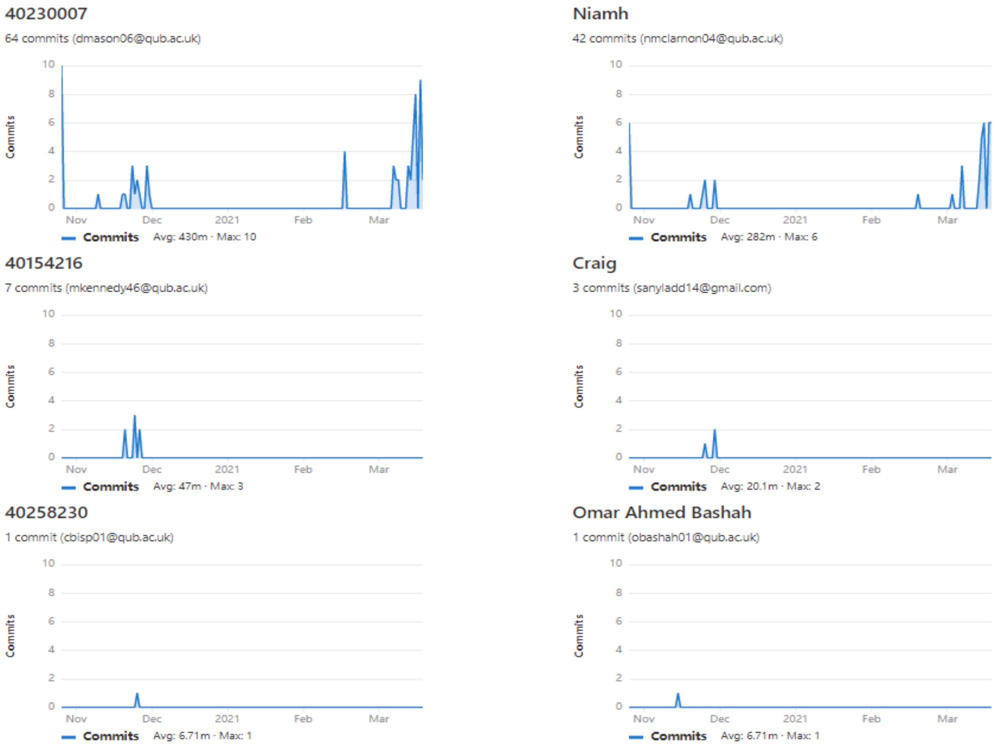
- Reviewed deliverables and provided feedback

Name (Charlotte):

- Reviewed deliverables and provided feedback

GitLab’s Activity Record

Our groups activity can be seen from our GitLab activity log and the team meeting (as seen above).



40230007

64 commits (dmason06@qub.ac.uk)

Commits Avg: 430m · Max: 10

Niamh

42 commits (nmclarnon04@qub.ac.uk)

Commits Avg: 282m · Max: 6

40154216

7 commits (mkennedy46@qub.ac.uk)

Commits Avg: 47m · Max: 3

Craig

3 commits (sanyladd14@gmail.com)

Commits Avg: 20.1m · Max: 2

40258230

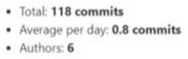
1 commit (cbisp01@qub.ac.uk)

Commits Avg: 6.71m · Max: 1

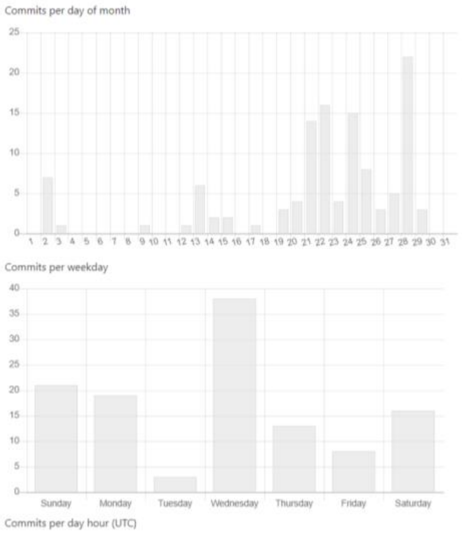
Omar Ahmed Bashah

1 commit (obashah01@qub.ac.uk)

Commits Avg: 6.71m · Max: 1



- Total: **118 commits**
- Average per day: **0.8 commits**
- Authors: **6**



Commits per day of month

Commits per weekday

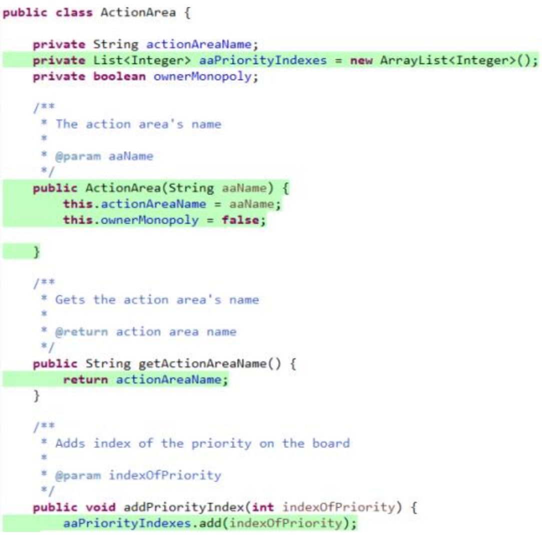
Commits per day hour (UTC)

Junit Test Runs

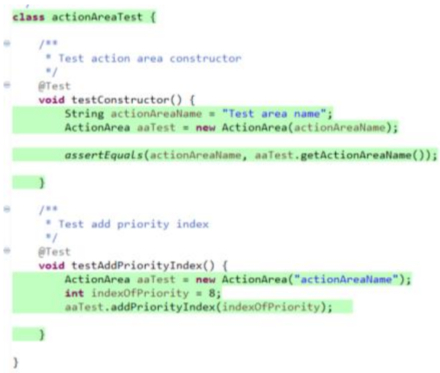
[N.M; D.M]

Shown below are some sample screenshots of our Junit testing.

ActionArea Junit Test

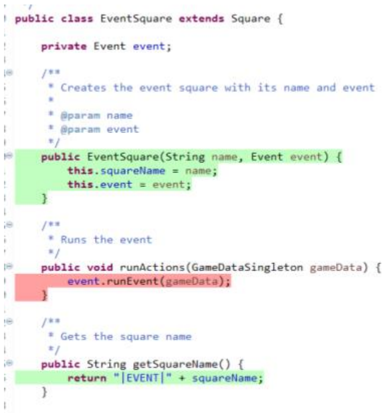


```
public class ActionArea {

    private String actionAreaName;
    private List<Integer> aaPriorityIndexes = new ArrayList<Integer>();
    private boolean ownerMonopoly;

    /**
     * The action area's name
     *
     * @param aaName
     */
    public ActionArea(String aaName) {
        this.actionAreaName = aaName;
        this.ownerMonopoly = false;

    }

    /**
     * Gets the action area's name
     *
     * @return action area name
     */
    public String getActionAreaName() {
        return actionAreaName;
    }

    /**
     * Adds index of the priority on the board
     *
     * @param indexOfPriority
     */
    public void addPriorityIndex(int indexOfPriority) {
        aaPriorityIndexes.add(indexOfPriority);
```

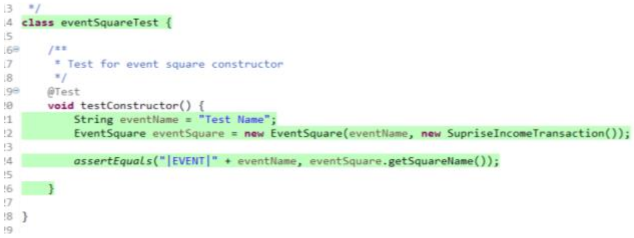


```
class actionAreaTest {

    /**
     * Test action area constructor
     */
    @Test
    void testConstructor() {
        String actionAreaName = "Test area name";
        ActionArea aaTest = new ActionArea(actionAreaName);

        assertEquals(actionAreaName, aaTest.getActionAreaName());

    }

    /**
     * Test add priority index
     */
    @Test
    void testAddPriorityIndex() {
        ActionArea aaTest = new ActionArea("actionAreaName");
        int indexOfPriority = 8;
        aaTest.addPriorityIndex(indexOfPriority);

    }

}
```

EventSquare Junit Test

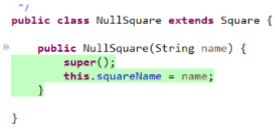


```
 */
public class EventSquare extends Square {

    private Event event;

    /**
     * Creates the event square with its name and event
     *
     * @param name
     * @param event
     */
    public EventSquare(String name, Event event) {
        this.squareName = name;
        this.event = event;
    }

    /**
     * Runs the event
     */
    public void runActions(GameDataSingleton gameData) {
        event.runEvent(gameData);
    }

    /**
     * Gets the square name
     */
    public String getSquareName() {
        return "|EVENT|" + squareName;
    }
```

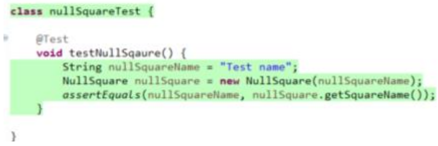


```
 */
class eventSquareTest {

    /**
     * Test for event square constructor
     */
    @Test
    void testConstructor() {
        String eventName = "Test Name";
        EventSquare eventSquare = new EventSquare(eventName, new SupriseIncomeTransaction());

        assertEquals("|EVENT|" + eventName, eventSquare.getSquareName());

    }

}
```

NullSquare Junit Test

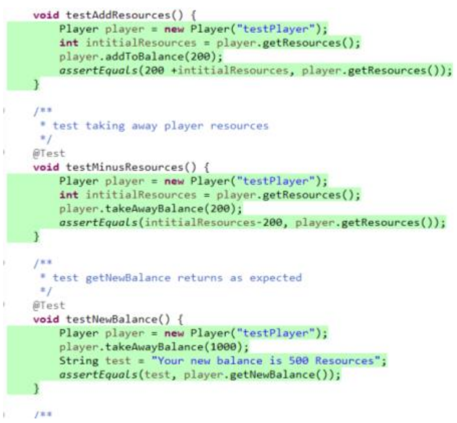


```
 */
public class NullSquare extends Square {

    public NullSquare(String name) {
        super();
        this.squareName = name;
    }

}
```

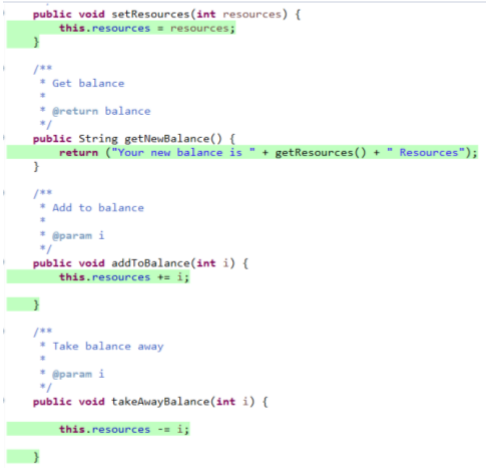


```
class nullSquareTest {

    @Test
    void testNullSqaure() {
        String nullSquareName = "Test name";
        NullSquare nullSquare = new NullSquare(nullSquareName);
        assertEquals(nullSquareName, nullSquare.getSquareName());
    }

}
```

Player Junit Test

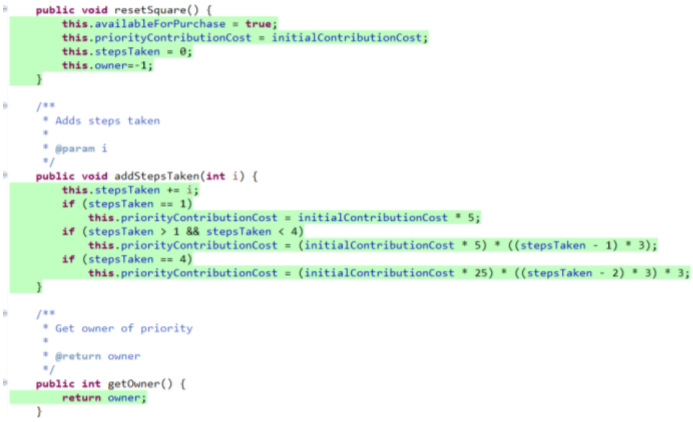


```
    void testAddResources() {
        Player player = new Player("testPlayer");
        int intitialResources = player.getResources();
        player.addToBalance(200);
        assertEquals(200 +intitialResources, player.getResources());
    }

    /**
     * test taking away player resources
     */
    @Test
    void testMinusResources() {
        Player player = new Player("testPlayer");
        int intitialResources = player.getResources();
        player.takeAwayBalance(200);
        assertEquals(intitialResources-200, player.getResources());
    }

    /**
     * test getNewBalance returns as expected
     */
    @Test
    void testNewBalance() {
        Player player = new Player("testPlayer");
        player.takeAwayBalance(1000);
        String test = "Your new balance is 500 Resources";
        assertEquals(test, player.getNewBalance());
    }

    /**
```

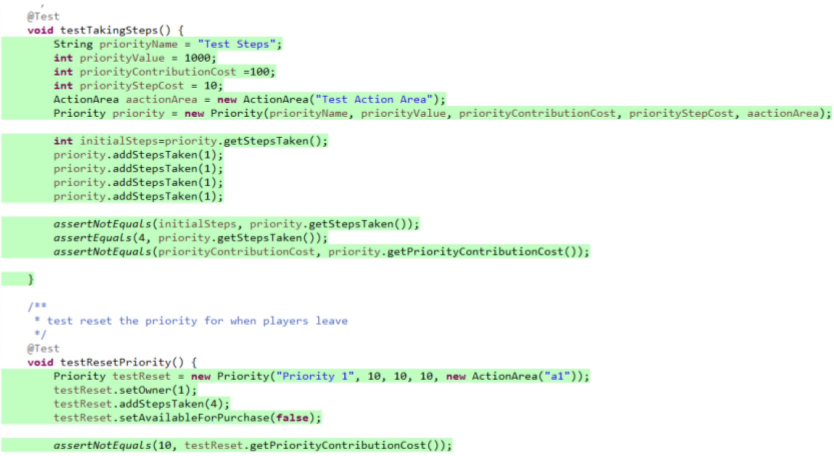


```
    public void setResources(int resources) {
        this.resources = resources;
    }

    /**
     * Get balance
     *
     * @return balance
     */
    public String getNewBalance() {
        return ("Your new balance is " + getResources() + " Resources");
    }

    /**
     * Add to balance
     *
     * @param i
     */
    public void addToBalance(int i) {
        this.resources += i;

    }

    /**
     * Take balance away
     *
     * @param i
     */
    public void takeAwayBalance(int i) {

        this.resources -= i;

    }
```

Priority Junit Test



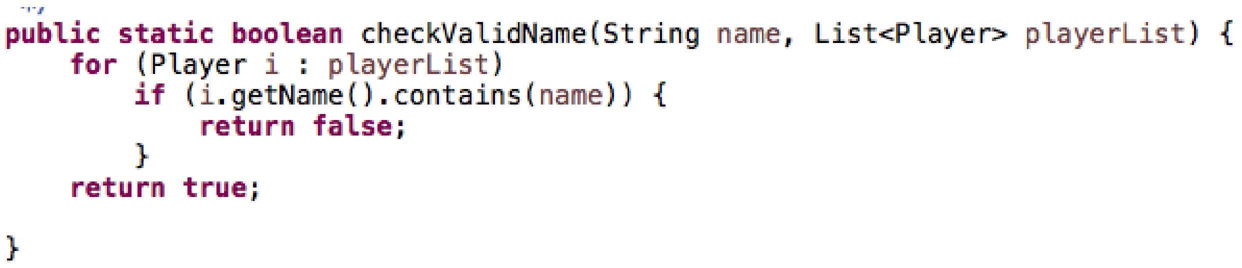
```
    public void resetSquare() {
        this.availableForPurchase = true;
        this.priorityContributionCost = initialContributionCost;
        this.stepsTaken = 0;
        this.owner=-1;
    }

    /**
     * Adds steps taken
     *
     * @param i
     */
    public void addStepsTaken(int i) {
        this.stepsTaken += i;
        if (stepsTaken == 1)
            this.priorityContributionCost = initialContributionCost * 5;
        if (stepsTaken > 1 && stepsTaken < 4)
            this.priorityContributionCost = (initialContributionCost * 5) * ((stepsTaken - 1) * 3);
        if (stepsTaken == 4)
            this.priorityContributionCost = (initialContributionCost * 25) * ((stepsTaken - 2) * 3) * 3;
    }

    /**
     * Get owner of priority
     *
     * @return owner
     */
    public int getOwner() {
        return owner;
    }
```



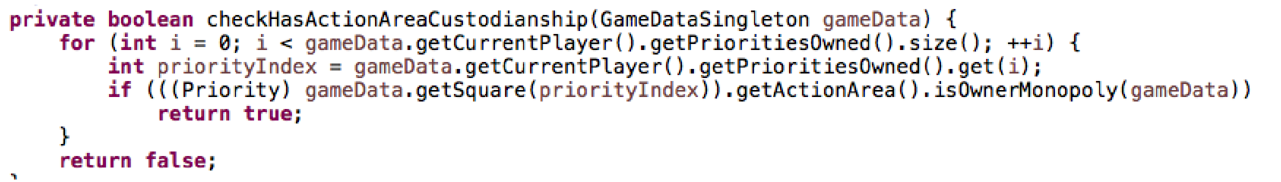
```
    @Test
    void testTakingSteps() {
        String priorityName = "Test Steps";
        int priorityValue = 1000;
        int priorityContributionCost =100;
        int priorityStepCost = 10;
        ActionArea aactionArea = new ActionArea("Test Action Area");
        Priority priority = new Priority(priorityName, priorityValue, priorityContributionCost, priorityStepCost, aactionArea);

        int initialSteps=priority.getStepsTaken();
        priority.addStepsTaken(1);
        priority.addStepsTaken(1);
        priority.addStepsTaken(1);
        priority.addStepsTaken(1);

        assertNotEquals(initialSteps, priority.getStepsTaken());
        assertEquals(4, priority.getStepsTaken());
        assertNotEquals(priorityContributionCost, priority.getPriorityContributionCost());

    }

    /**
     * test reset the priority for when players leave
     */
    @Test
    void testResetPriority() {
        Priority testReset = new Priority("Priority 1", 10, 10, 10, new ActionArea("a1"));
        testReset.setOwner(1);
        testReset.addStepsTaken(4);
        testReset.setAvailableForPurchase(false);

        assertNotEquals(10, testReset.getPriorityContributionCost());
```

Secure System features

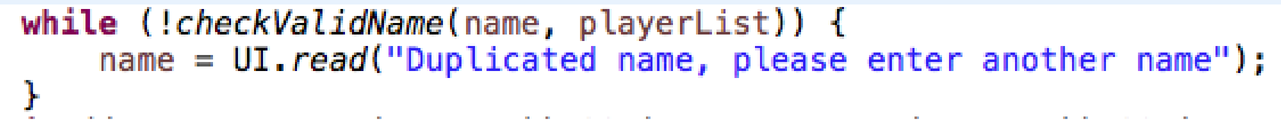
We used various security features, we have all attributes private with getters and setters for relevant fields. Here is some snippets of code of checks within the system:



```
public static boolean checkValidName(String name, List<Player> playerList) {
    for (Player i : playerList)
        if (i.getName().contains(name)) {
            return false;
        }
    return true;

}
```



```
private boolean checkHasActionAreaCustodianship(GameDataSingleton gameData) {
    for (int i = 0; i < gameData.getCurrentPlayer().getPrioritiesOwned().size(); ++i) {
        int priorityIndex = gameData.getCurrentPlayer().getPrioritiesOwned().get(i);
        if (((Priority) gameData.getSquare(priorityIndex)).getActionArea().isOwnerMonopoly(gameData))
            return true;
    }
    return false;
```



```
while (!checkValidName(name, playerList)) {
    name = UI.read("Duplicated name, please enter another name");
}
```

Future Opportunities

We have considered some secure features we would implement in the future.

- Adding a save game function- This would allow players to save their progress and continue with the game at a later stage.
- Adding a leader board- This would allow players to compete against each other (and previous gameplays involving themselves) to try place highly on the leader board.
- Timed gameplays- For example, playing a quick round of a set twenty-minute game or having a timer record their gameplay times. This would allow the players to try maximise their playing time effectively to try own and develop as many action areas as they can.
- Difficulty- The game will have various difficulties to choose from when launched and these will affect the Resources, Squares and Events and make them harsher on players.