

# Design and Analysis of a Game of Monopoly

Aifric Nolan, Aine McKeon, Comfort Dopamu, Mahjabeen Soomro, Rachel white

<b>Product Design</b>	<b>2</b>
Revised Use Case Diagram	2
Refined Class Diagram	4
User Interface MockUps	5
The Main Menu	5
Game Screen	5
Success Screen	6
Client Server Experiments	7
State Machines	7
Sequence Diagrams	8
Object Diagrams	12
Player	12
Square	12
Communication Diagrams	13
Revised Object Diagrams	13
More Refined Class Diagrams	13
Class Skeletons	13
Die Class	13
Class Square	14
Board Class	14
<b>Appendix</b>	<b>15</b>
Team Meetings	15
Meeting 1	15
Sprint Burndown Charts	16
Legend	16
Sprint B	17

# Design and Analysis of a Game of Monopoly

Aifric Nolan, Aine McKeon, Comfort Dopamu, Mahjabeen Soomro, Rachel white

## Product Design

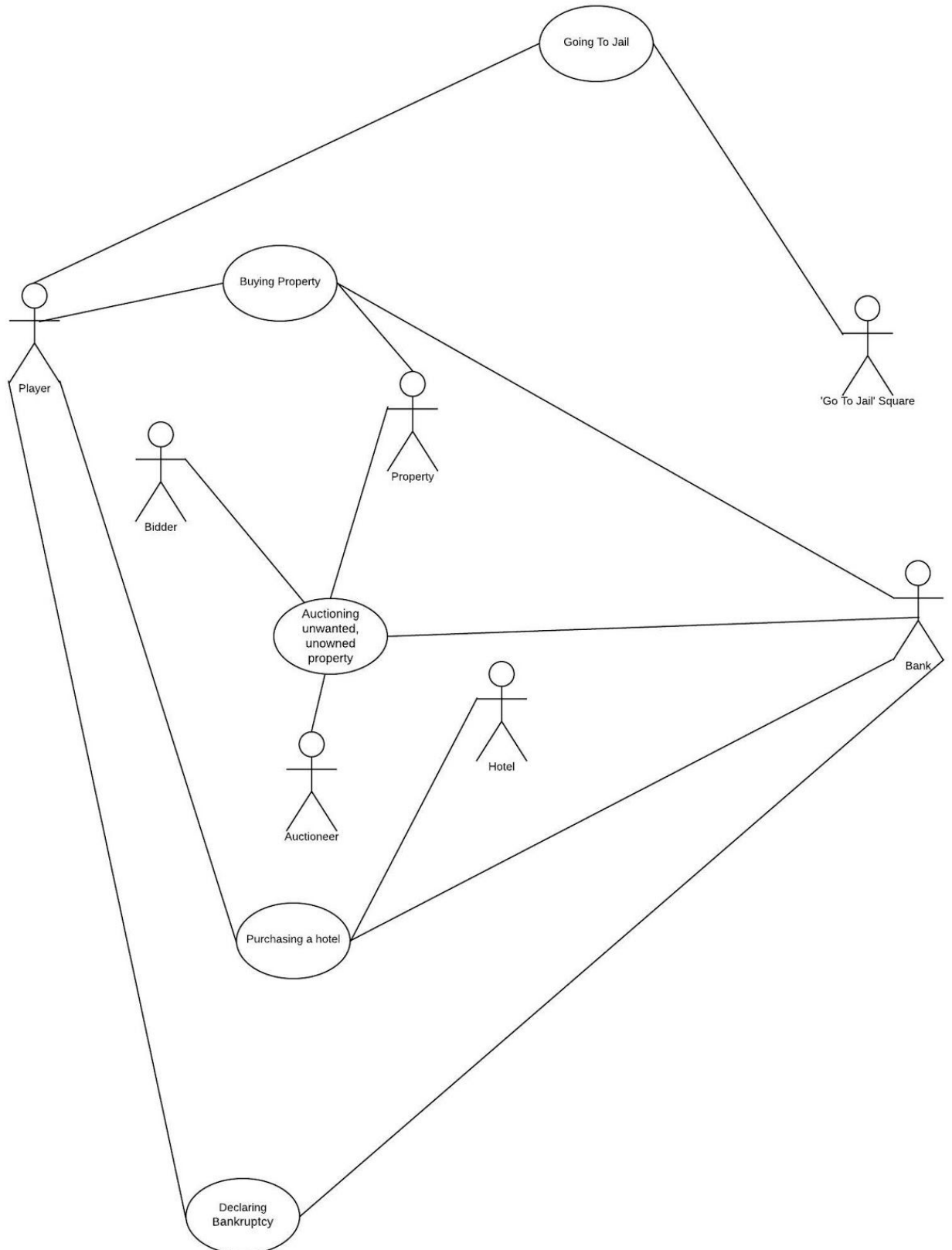
The second stage of the Project is the Product Design. This includes the UI designs, refinements on the classes, object and sequence diagrams. We will also be performing client server tests and providing class skeletons.

## Revised Use Case Diagram

Previously, our use case diagram had crossing lines which unintentionally created connections that were not there. We have revised our diagram to remove these interconnecting lines.

# Design and Analysis of a Game of Monopoly

Aifric Nolan, Aine McKeon, Comfort Dopamu, Mahjabeen Soomro, Rachel white

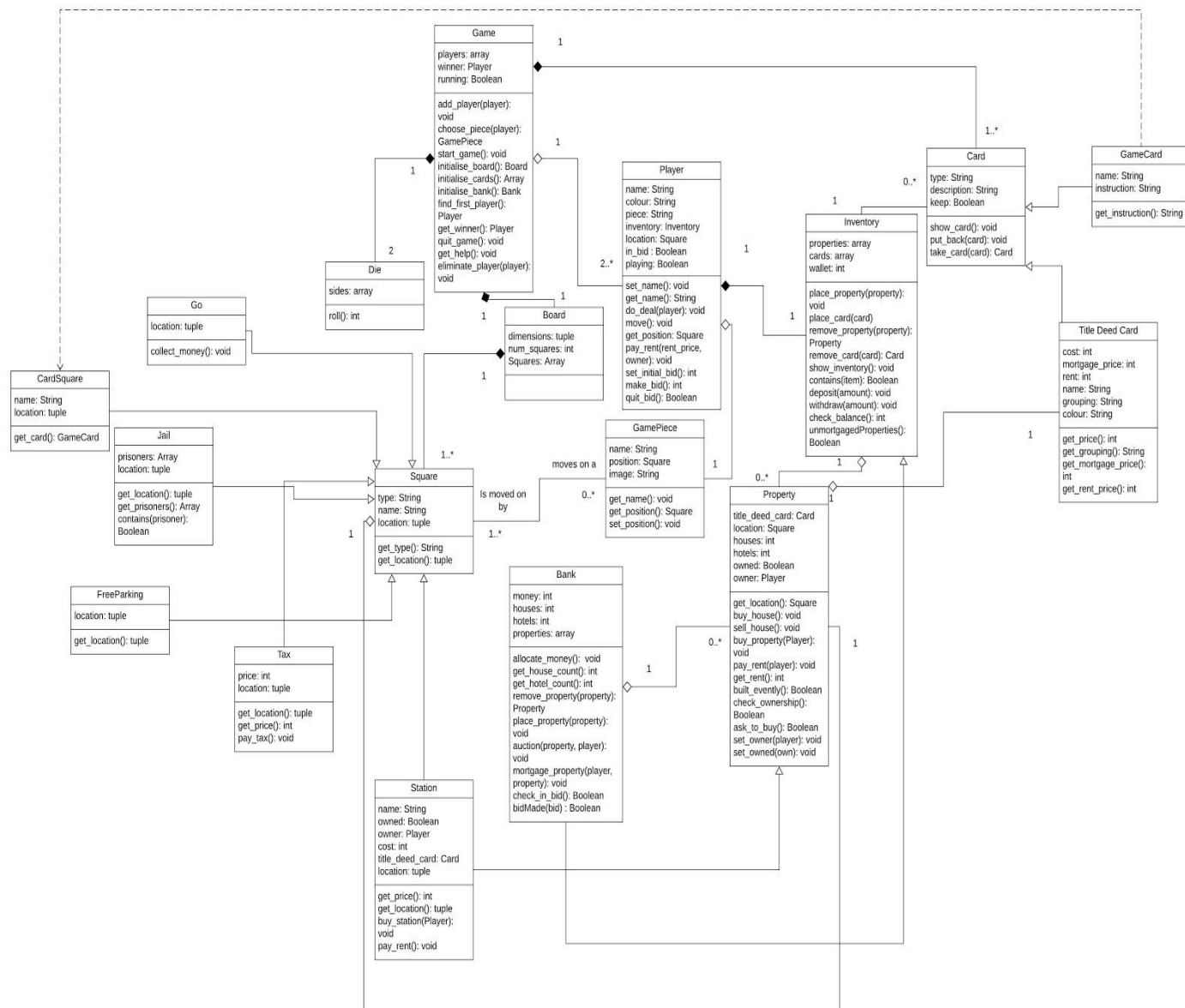


# Design and Analysis of a Game of Monopoly

Aifric Nolan, Aine McKeon, Comfort Dopamu, Mahjabeen Soomro, Rachel white

## Refined Class Diagram

The class diagram has been refined. We have split up some of the classes to make them more manageable and added some more functions to the classes as different flows and operations became more apparent.



# Design and Analysis of a Game of Monopoly

Aifric Nolan, Aine McKeon, Comfort Dopamu, Mahjabeen Soomro, Rachel white

## User Interface MockUps

### The Main Menu

The game will open on the below screen. This is the Home Menu. From here, the user can choose the start the game or learn how to play from the Help Menu.

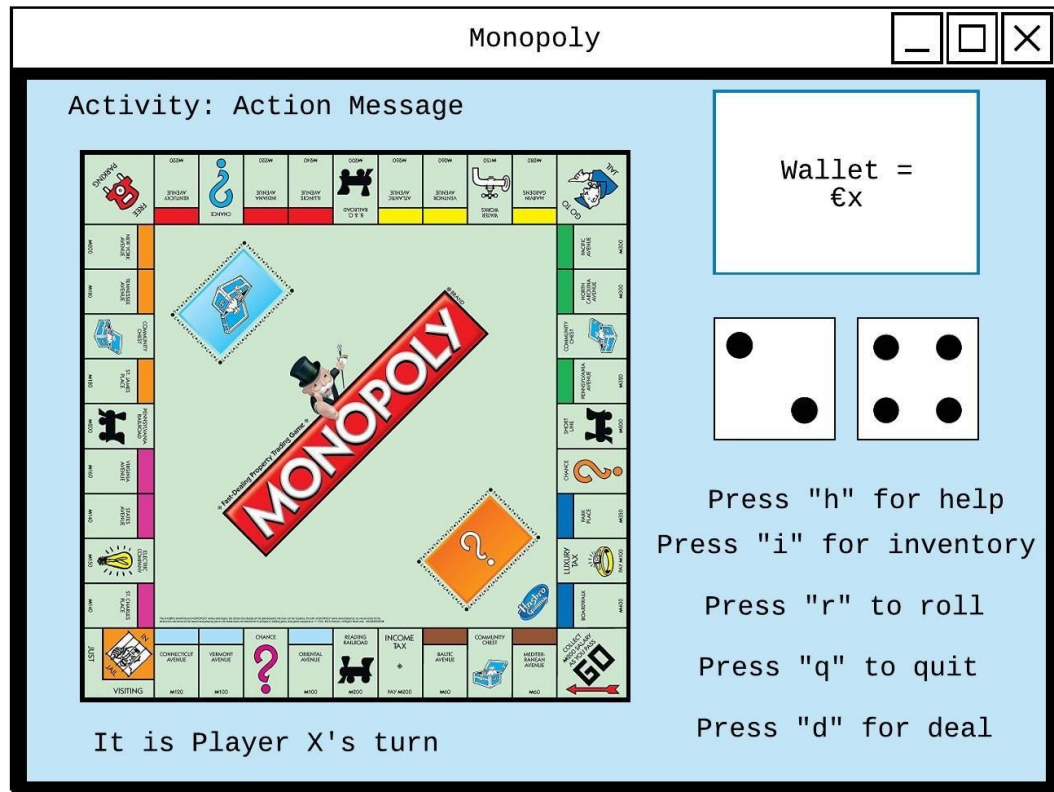


### Game Screen

This is the main screen. It is where the main gameplay occurs. From this screen, the user clicks to roll, moves around the board and performs regular game options.

# Design and Analysis of a Game of Monopoly

Aifric Nolan, Aine McKeon, Comfort Dopamu, Mahjabeen Soomro, Rachel white



## Success Screen

When a player has won a game, a success screen will be shown before the screen switches back to the main screen again.

# Design and Analysis of a Game of Monopoly

Aifric Nolan, Aine McKeon, Comfort Dopamu, Mahjabeen Soomro, Rachel white



## Client Server Experiments

Blah blah blah

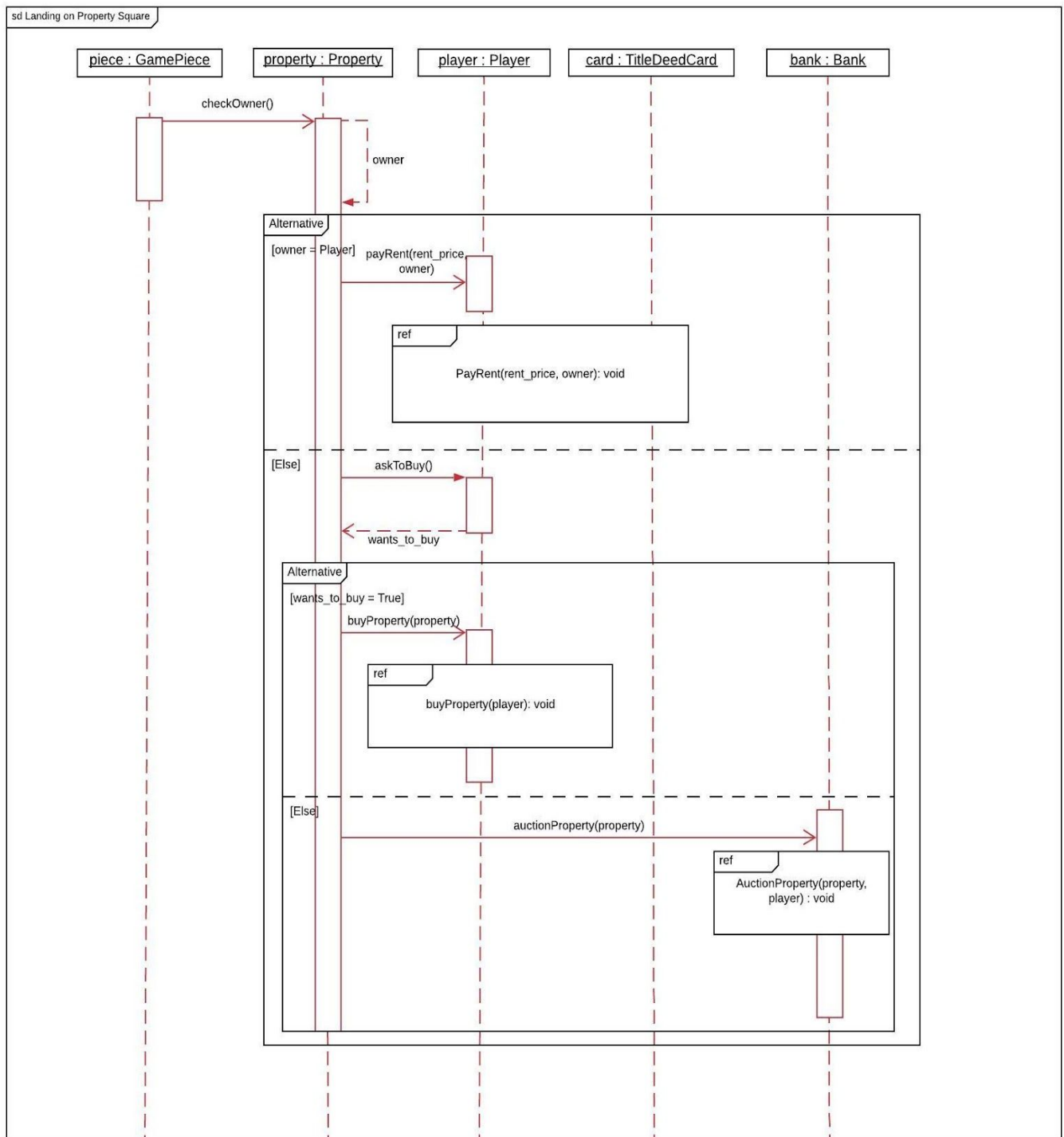
## State Machines

Blah blah blah

# Design and Analysis of a Game of Monopoly

Aifric Nolan, Aine McKeon, Comfort Dopamu, Mahjabeen Soomro, Rachel white

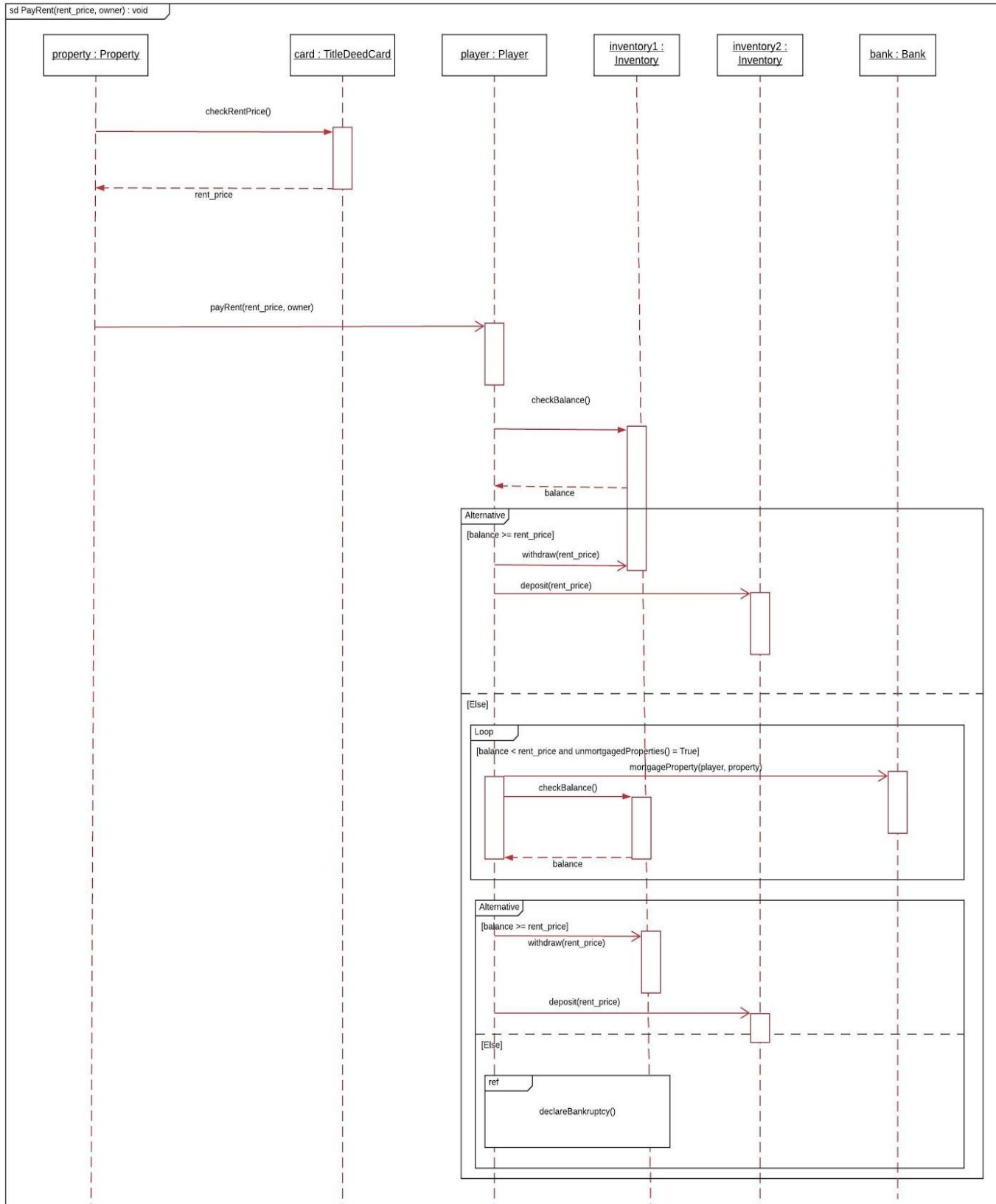
## Sequence Diagrams





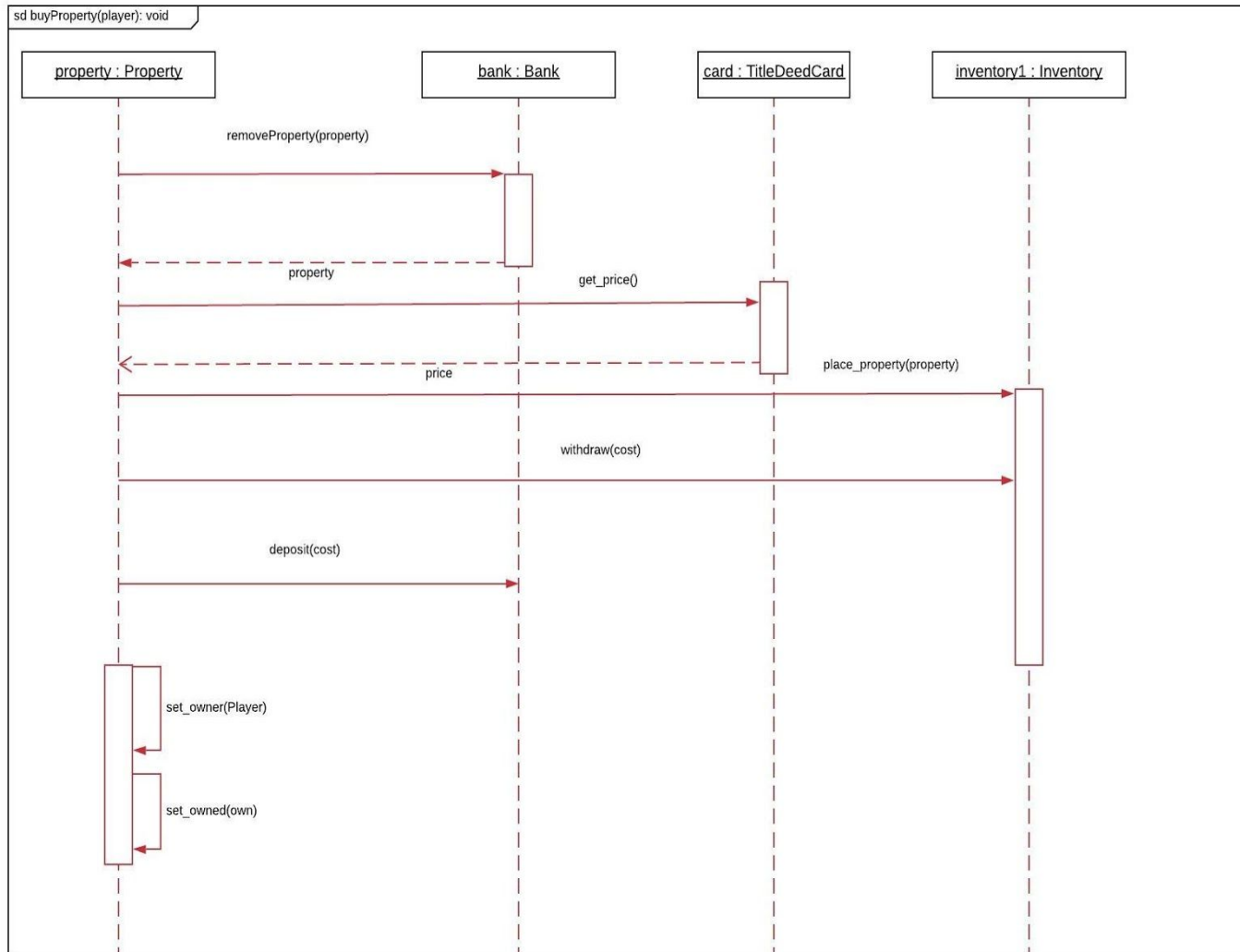
# Design and Analysis of a Game of Monopoly

Aifric Nolan, Aine McKeon, Comfort Dopamu, Mahjabeen Soomro, Rachel white



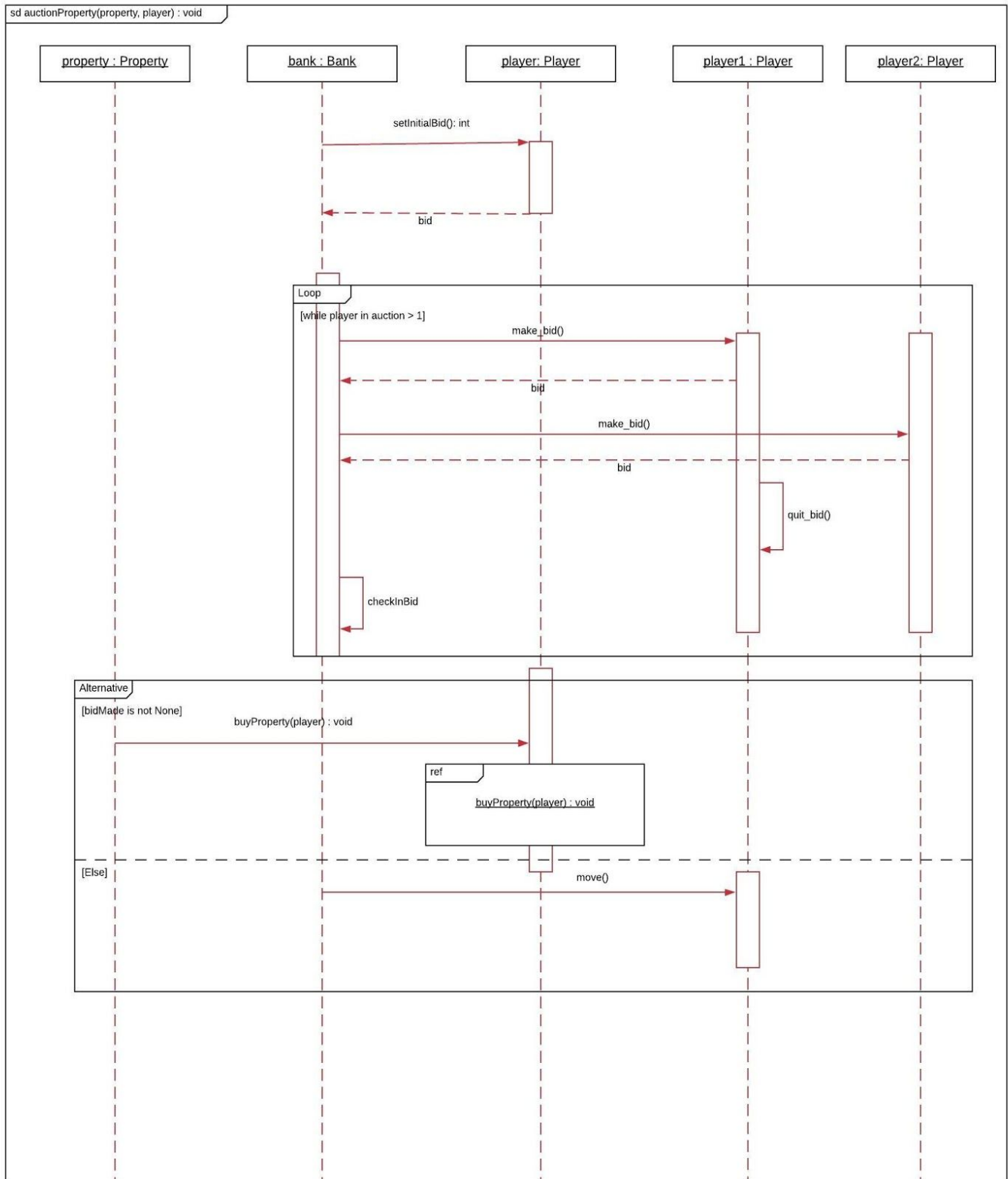
# Design and Analysis of a Game of Monopoly

Aifric Nolan, Aine McKeon, Comfort Dopamu, Mahjabeen Soomro, Rachel white



# Design and Analysis of a Game of Monopoly

Aifric Nolan, Aine McKeon, Comfort Dopamu, Mahjabeen Soomro, Rachel white

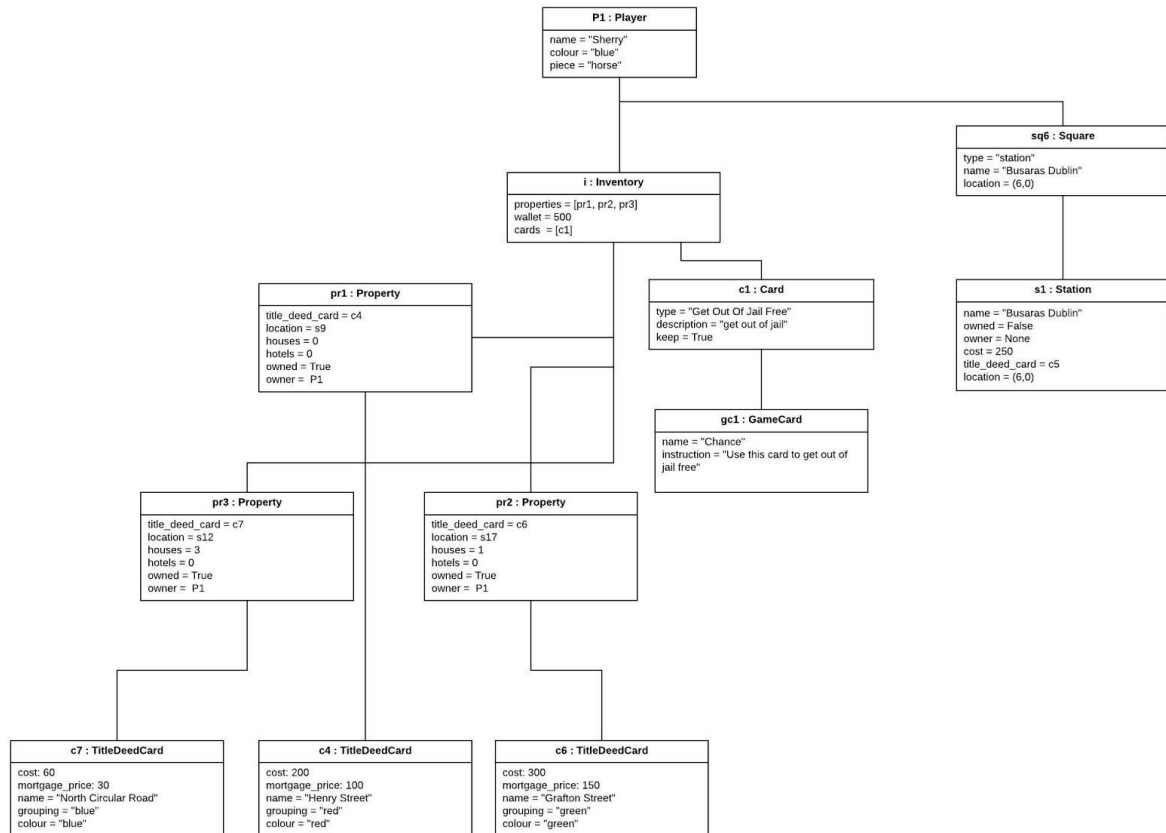


# Design and Analysis of a Game of Monopoly

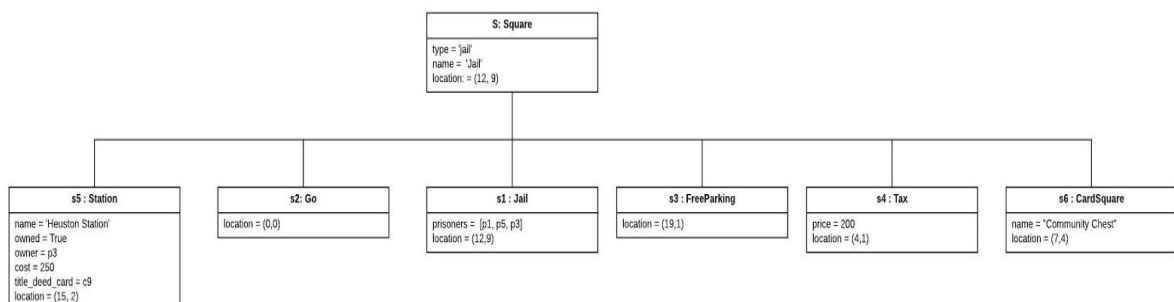
Aifric Nolan, Aine McKeon, Comfort Dopamu, Mahjabeen Soomro, Rachel white

## Object Diagrams

### Player



### Square



# Design and Analysis of a Game of Monopoly

Aifric Nolan, Aine McKeon, Comfort Dopamu, Mahjabeen Soomro, Rachel white

## Communication Diagrams

Blah blah blah

## Revised Object Diagrams

Blah blah blah

## More Refined Class Diagrams

Blah blah blah

## Class Skeletons

We have constructed class skeletons based on the refined class diagram. Each class skeleton is shown below,

### Die Class

```
import random

class Die:

    def __init__(self):
        self.sides = [1,2,3,4,5,6]

    def roll(self):
        """Roll the dice and return the sum of the 2 die"""
        index = random.randint(0,5)
        index2 = random.randint(0,5)
        return self.sides[index] + self.sides[index2]
```

# Design and Analysis of a Game of Monopoly

Aifric Nolan, Aine McKeon, Comfort Dopamu, Mahjabeen Soomro, Rachel white

## Class Square

```
class Square:

    def __init__(self, type, name, location):
        """Initialises the square"""
        self.type = type #Is a string that represents the type of square
        self.name = name #Is a string that is the name of the square
        self.location = location #Is a tuple that represents the location on
the board

    def get_type(self):
        """Returns the type of square"""
        return self.type

    def get_location(self):
        """Returns the location of square"""
        return self.location

    def get_name(self):
        """Returns the name of the square"""
        return self.name
```

## Board Class

```
class Board:

    def __init__(self, squares):
        """Initialises the board"""
        self.dimensions = (11,11) #The dimensions of the board
        self.num_squares = 40 #Number of squares on the board
        self.squares = squares #A list of squares on the board
        self.posx = posz #X Position of the board on the screen
        self.posy = posy #Y Position of the board on the screen
```

# Design and Analysis of a Game of Monopoly

Aifric Nolan, Aine McKeon, Comfort Dopamu, Mahjabeen Soomro, Rachel white

## Appendix

### Team Meetings

#### Meeting 1

**31st October**

**Attendees:** Aifric, Comfort, Rachel, Aine

**Topic:** Sequence diagrams

**Minute Taker:** Rachel

**Leader:** Aine

**Deputy:** Aifric

- For today's meeting, we wanted to get all the hard bits done first such as the object diagram, sequence and its sub-diagram and the state diagram.
- We looked and discussed the object diagrams that Afric had made and then decided it was very good, we discussed how the sequence diagram was going to be implemented using the object diagrams.
- We revised our previous use case and discussed on how the design should look like, at first we were all confused on how it should be designed but after looking at various examples and the notes on loop we finally decide on what use case should be used for the sequence diagram and the sub-diagram.
- We drew out the sequence diagram on paper to understand the flow better and make functions that can connect classes accurately.

#### Meeting 2

**8th November**

**Attendees:** Aifric, Comfort, Rachel, Aine, Mahjabeen

**Topic:** UI Mockups

**Minute Taker:** Aifric

**Leader:** Rachel

**Deputy:** Aine

- We focused on the UI mockups for this meeting. We designed every screen on paper first and then started to translate it to the screen.
- It was difficult to design because we decided to use pygame which imposes a lot of limitations on the UI and user interactions. We couldn't include any buttons which was difficult.

# Design and Analysis of a Game of Monopoly

Aifric Nolan, Aine McKeon, Comfort Dopamu, Mahjabeen Soomro, Rachel white

- Jabeen emailed Renaat to enquire if we should include all our mockups in the assignment or just a few and we were told 3 would do.
- We created the start screen and the general game screen with Lucid Chart.
- We decided Aifric will design the end screen and perhaps give a few 'You Win' endings so the team can choose the best one.
- We are going to decide the last mockup and start the communication diagrams on Tuesday.

## Sprint Burndown Charts

### Legend

Colour	Meaning
Red	Expected Burndown
Blue	Actual Burndown
Light Blue	Trending Burndown



# Design and Analysis of a Game of Monopoly

Aifric Nolan, Aine McKeon, Comfort Dopamu, Mahjabeen Soomro, Rachel white

## Sprint B

### Burndown Chart - Sprint B

