Modelling Document

Software Engineering –

Alex Sinclair

Abdalla Shita

Dion Upton

Jaimie Withers

Jamie Stening

# High Level Models

## Domain diagram

Our software will comprise of a user interface, UI, an artificial intelligence, AI, and the game program.

The interface will be aesthetically pleasing and react to any user inputs accordingly.

The game program will outline the game and ensure the requirements set out for Property Tycoon are all met in terms of game play. This is the core of the game, it will handle any game responsibility not related to user interaction such as selecting a card, keeping player assets in line with game development and the general game concept.

The AI will be available to provide additional players, the AI will react to the game in an intelligent yet unbiased way.

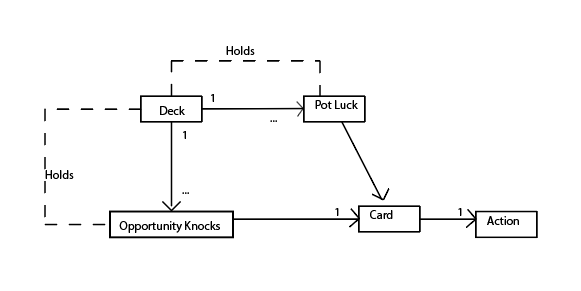
## User Case Diagrams

### User Diagram

### 

The user will play the game by interacting with the UI, for every possible action the player will select a button displayed on the custom-built UI corresponding to their game choice. Almost every action will initially require the ‘Roll Dice’ button to begin that player’s turn.

### Board diagram

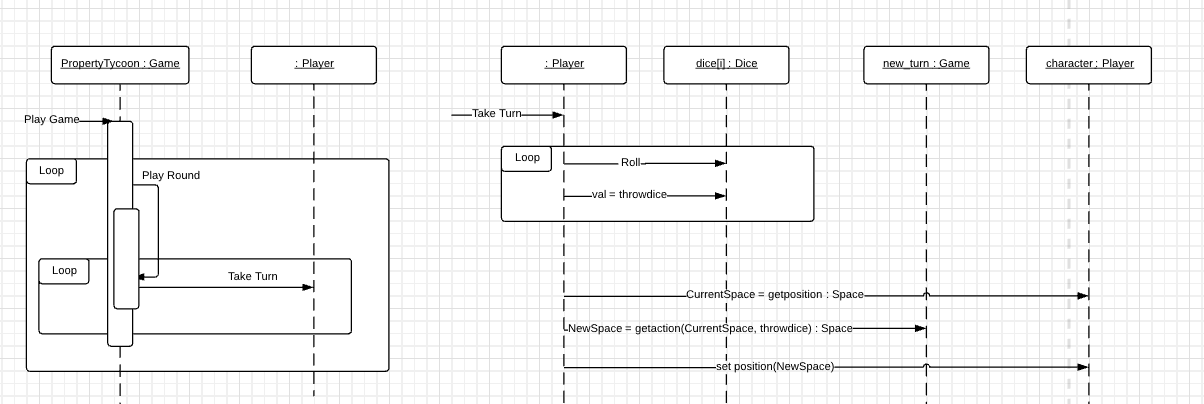
This diagram demonstrates the way in which the game will hold the two types of cards related to Property Tycoon. A deck will hold an unknown number of ‘Pot Luck’ or ‘Opportunity knocks’ cards that can then be selected one at a time, as appropriate, the action of this card can then take place.

# 

This is a domain diagram of the board game itself. It demonstrates an outline of the game that we are generating.

# Low Level Models

Sequence diagram

This sequence diagram focuses on the way a player would take a turn during the game, it shows the interaction within the game as well as the time ordering of the implementation. The first diagram shows the steps that take place as an overall view, the game is started and depending on the number of players will decide the length of the first round. Each player turn is then also seen as one depending on the player’s actions as to how long this will go on.

checkSpace =

The second diagram shows the inner workings of a player’s turn. The ‘Roll dice’ action will be initialized by the user, at this time the position of the player’s current position will be found, whilst the new space will be found based on the instructions given by the ‘Roll Dice’ action. The player will then be allocated to it’s new position where any further action can be implemented.

## Class Diagram(s)

### Initial

Progressed

Final

