CSC7053 Software Engineering

**Software Engineering Project:**

**Technopoly**

**Group 3:**

Barbara Murtland

Colette Casey

Dave Kennedy

Ismael Floritt

Time Lewis

**Table of Contents**

1. Introduction…………..…………………………………………………………………………………………………………..1
2. Requirements Analysis ..…………………………………………………………………………………………………….1
   * 1. Game Guide & Graphical Representation of Board……………………………………1
     2. Use Case Descriptions……………………………………………………………………………….
     3. Use Case Diagram………………………………………………………………………………………
3. Realisation………………….………………………………………………………………………………………………………
   * 1. Sequence Diagram……………………………………………………………………………………
4. Design……………………………………………..………………………………………………………………………………..
   * 1. UML Class Diagram…………….. ……………………………………………………………………
5. Conclusion……………………..…………………………………………………………………………………………………

**Appendix 1**

1. Photos of use case and sequence diagrams
2. Full user guide and graphical representation of board
3. Test Plan
4. Test Plan – Junit tests conducted
5. Glossary of Key Terms

**Appendix 2**

1. Weekly Team Meeting Minutes

Introduction

The aim of this report is to document the planning, design, development and implementation of a Monopoly style interactive game known as Technopoly. The report has focused on the requirements analysis, system design, software implementation and system testing.

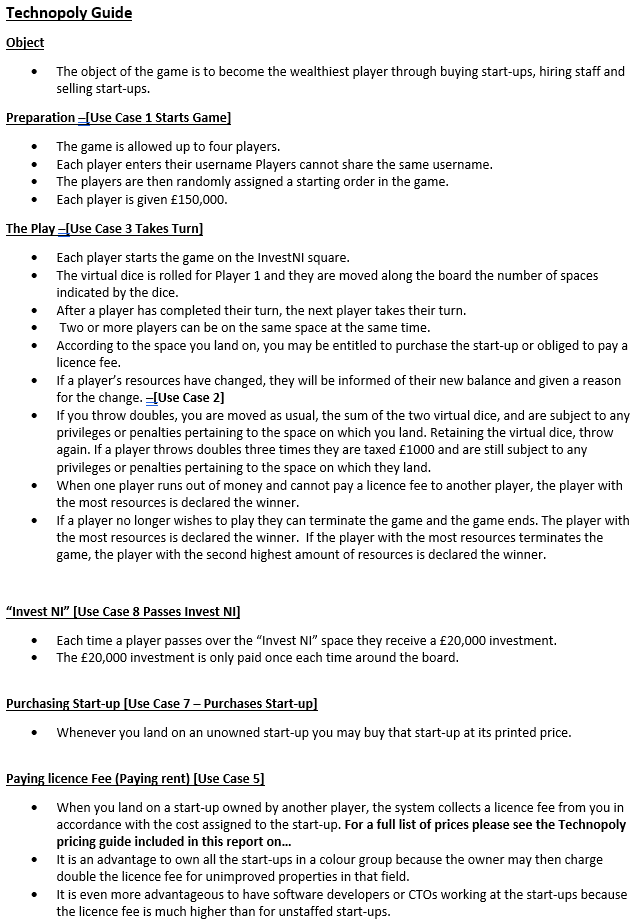
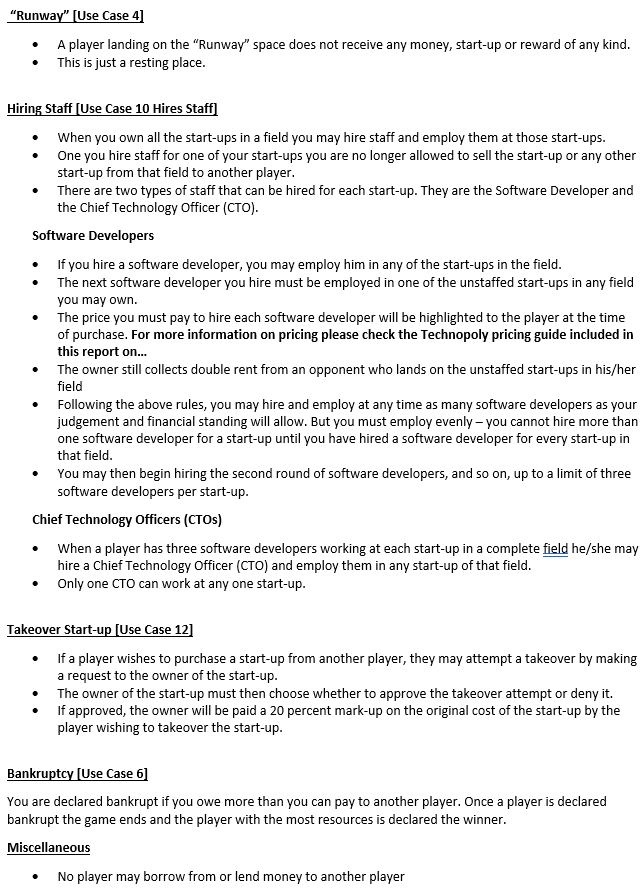
Technopoly has been developed in the Eclipse IDE in Java, using the console to create a textual interface. A use case driven development process was adopted by the team in order to effectively manage and meet the requirements laid down by the ‘customer’.

Requirements Analysis

This project has been designed around a use case driven development process. The team met to fully discuss the uses cases which were needed, running through a mock up the game. The use cases were then split amongst the team before being collated.

i. Game Guide and Graphical representation

//Diagram of board here – image – ref both in appendix



ii. Use Case Descriptions

Please refer to the Use Case Diagram. In order to assist the reading of the below descriptions all use cases have been listed numerically. All extension and inclusion points have been assigned a letter, red for Extension Points, blue for inclusions. Actors have been italicised.

|  |  |
| --- | --- |
| **Use Case 1** | **Starts Game** |
| **Summary** | Establish the number of players(2-4) and their names. Randomly generate the order of play. Game starts. |
| **Actors** | *Player1, Player2,* (*Player3, Player4* are optional) |
| **Triggers** | 2-4 players wish to set-up Technopoly |
| **Pre Conditions** | * There are a minimum of 2 and a maximum of 4 players |
| **Post Conditions** | * The valid number of players selected is displayed. * A random order is generated. * The players can either view the rules [Use case 2 Views Rules] or press start to begin the game. |
| **Flow** | 1. The *player(s)* are prompted to enter the number of valid players(between 2 and 4 players) 2. The *players* will be prompted in turn to enter a valid name (2-15 characters) 3. Once all player names are added to the game a random order is assigned to the players and the game begins. |
| **Alternative Flow** | * At flow point 1:- If the number of players is invalid, the *player(s)* will be prompted again and reminded that the valid number of players is between 2 and 4 * At flow point 2:- If the name of the player is invalid or already exists in the game, the *player* will be prompted again and reminded that a valid name is between 2 and 15 characters(including – and ‘) excluding whitespace. |
| **Extension Points** | None |
| **Inclusions Points** | None |

|  |  |
| --- | --- |
| **Use Case 2** | **Views Rules** |
| **Summary** | The *player* views the rules of the game before selecting ‘Start Game’. |
| **Actors** | Player(s) |
| **Triggers** | [USE CASE 1 Start Games] After selecting entering a valid number of players and entering valid name the players can view the rules. |
| **Pre Conditions** | * The game has not yet started |
| **Post Conditions** | * The rules of the game are displayed. |
| **Flow** | 1. The *player* selects the “View Rules” option. 2. The system displays the rules of the game to the player(s). |
| **Alternative Flow** | The players do not select ‘View Rules’ in which case the only other option available to them is to ‘Start Game’. |
| **Extension Points** | None |
| **Inclusions Points** | None |
| **Use Case 3** | **Takes Turn** |
| **Summary** | *Player* rolls dice and moves to the corresponding space. Three options are then posed to the player |
| **Actors** | Player(s) |
| **Triggers** | The game starts OR the player before has ended their turn |
| **Pre Conditions** | * [USE CASE 1 Starts Game] :- The game is set up * It’s the *player* in question’s turn |
| **Post Conditions** | * [USE CASE 11 Ends Turn] :- The *player* must select ‘end turn’ once they have completed their turn. The next player will repeat this operation. |
| **Flow** | 1. The player is shown: Player Number, Name, Balance, Properties Owned including Staff, the dice score thrown and the space they have landed on. Eg, Player 1 | Joe Bloggs | £xx.xx | Owns: [AProperty, BProperty] | Rolled [2,6] | Landed on X. 2. The *player* is ‘moved’ to the corresponding space. Depending on the space landed on the *player* can: 3. Purchase Start-up [Extension D] 4. Pay Licence fee [Extension B] 5. The *player* can view menu [Inclusion A] |
| **Alternative Flow** | 1. At flow point 1 and 2:- If the *player* rolls a double (same value on both dice) the *player* is moved to the corresponding space. They finish that turn and take another go immediately. 2. At flow point 1 and 2:- If doubles is rolled three times in a row; on the third turn the *player* suffers a penalty (tax or jail depending on the game iteration) 3. At flow point 4:- If the *player* has insufficient funds to pay a Licence fee, then that *player* is declared bankrupt and the game is over. The player with the highest net worth is deemed the winner. [Extension C] |
| **Extension Points** | D. Purchase Start-Up – [Use Case 7]  B. Pay Licence Fee –[ Use Case 5]  C. Declares Bankruptcy – [Use Case 6]  A. Lands on Runway – [Use Case 4]  E. Passes Invest NI – [Use Case 8] |
| **Inclusions Points** | 1. Views Menu – [Use Case 9] |
| **Use Case 4** | **Lands on Runway** |
| **Summary** | Player lands on the ‘Runway’ field |
| **Actors** | *Players* |
| **Triggers** | The Player has moved onto the “Runway” field |
| **Pre Conditions** | * It’s the *player* in question’s turn [Use Case 3 Takes Turn] |
| **Post Conditions** | * The *player’s* turn continues |
| **Flow** | 1. The *player* lands on “Runway”. This space cannot be owned. No licence fee is applicable. |
| **Alternative Flow** | None |
| **Extension Points** | None |
| **Inclusions Points** | None |
| **Use Case 5** | **Pays Licence Fee** |
| **Summary** | The p*layer* lands on a space with a start-up owned by another *player*. The *player* must pay the appropriate licence fee to the other *player*. |
| **Actors** | *Players* |
| **Triggers** | A *Player* lands on a square owned by another *player*. |
| **Pre Conditions** | * It’s the *player* in question’s turn [Use Case 3 Takes Turn] * The start-up on the space is owned by another *player* * The *player* has sufficient funds to pay the licence fee |
| **Post Conditions** | * The *player’s* balance is updated to reflect paying the licence fee * The new balance is displayed by the system * The *player* has sufficient funds to continue to play (Balance >=0) |
| **Main Flow** | 1. The *player* lands on a square for a start-up owned by another *player* 2. The system displays the licence fee and the other *players* name/number 3. The *player* must have the funds to pay the appropriate licence fee 4. *Player’s* balance is reduced 5. *Other players* balance is increased 6. The updated balances for each player are displayed by the system |
| **Alternative Flow** | 1. At main flow point 3:- the *player* does not have sufficient funds to pay the licence fee [Extension C]. |
| **Extension Points** | C. Declares Bankruptcy – [Use Case 6] |
| **Inclusions Points** | None |
| **Use Case 6** | **Declares Bankruptcy** |
| **Summary** | *Player* is declared bankrupt by the system. |
| **Actors** | *Player(s)* |
| **Triggers** | The *player* lands on an start-up owned by another *player* without sufficient funds to pay the licence fee. |
| **Pre Conditions** | * It’s the *player* in question’s turn [Use Case 3 Takes Turn] * The *player* has landed on any owned start-up that is not their own [Use Case 5 Pay Licence Fee] * The *player* has less funds than necessary to pay a Licence Fee. |
| **Post Conditions** | * The player is declared bankrupt. The game ends and a winner is declared. |
| **Flow** | 1. The *player* lands on an owned start-up that is not his own. 2. When unable to pay the fee, the system declares the *player* bankrupt. 3. The system ends the game and declares the winner (player with most resources). |
| **Alternative Flow** | None |
| **Extension Points** | None |
| **Inclusions Points** | None |
| **Use Case 7** | **Purchase Start-up** |
| **Summary** | The *player* lands on a space with an available start-up. If they decide to purchase the start-up, they pay the listed price. If they buy all the start-ups in the same colour group(field), they have a tech monopoly (technopoly) and can charge double the listed license fee. |
| **Actors** | *Player*(s) |
| **Triggers** | *Player* lands on a new space with an available start-up |
| **Pre Conditions** | * It’s the *player* in question’s turn [Use Case 3 Takes Turn] * The start-up on the space is available to purchase * The player has sufficient funds to buy the start-up |
| **Post Conditions** | * The *player’s* balance is updated to reflect the purchase * The *player’s* properties are updated to list the company * The *player’s* properties are updated if they own a complete field |
| **Main Flow** | 1. The *player* lands on a space with an available start-up 2. *Player* decides to purchase start-up by selecting “Purchase” from the menu [Inclusion A] 3. The system checks the *player* has sufficient funds, then:    1. Reduces *player’s* balance by the listed price    2. Adds the start-up to the *player’s* list of companies    3. Displays a message to confirm the purchase and the new balance   4. The *player* can either return to the menu/end their turn [Extension H] |
| **Alternative Flow** | 1. At main flow point 1:- The *player* decides not to purchase the start-up, Views Menu and ends turn [Inclusion A and Extends H] 2. At main flow point 3:III:- If the purchase completes the *player’s* ownership of a colour field, the system will display a message to announce this. The player is brought to the menu [Inclusion A] 3. At main flow point 3:- The *player* does not have sufficient funds. The System displays a warning message. The player cannot purchase the space. Player must end their turn. [Extension H] |
| **Extension Points** | H. Ends Turn [Use Case 11] |
| **Inclusions Points** | 1. Views Menu - [Use Case 9] |
| **Use Case 8** | **Passes Invest NI** |
| **Summary** | The *player* receives a £20,000 investment each time they are on or pass the Invest NI space.  Their balance is updated to reflect the changes |
| **Actors** | *Player* |
| **Triggers** | *Player* is on or passes the Invest NI space |
| **Pre Conditions** | * The game has been started [Use Case 1 Starts Game] * It is the *player’s* turn [Use Case 2 Takes Turn] |
| **Post Conditions** | * The *player’s* balance updates to reflect an investment of £20,000 |
| **Main Flow** | 1. The *player* is on or passes the Invest NI space 2. The player receives an investment of £20,000 |
| **Alternative Flow** | None |
| **Extension Points** | None |
| **Inclusions Points** | **None** |
| **Use Case 9** | **Views Menu** |
| **Summary** | Player views menu, which describes the options available to take during their turn. |
| **Actors** | *Player*(s) |
| **Triggers** | * It is the *player’s* turn [Use Case 2 Takes Turn] * Player has finished any actions other than “Quits Game”. |
| **Pre Conditions** | * It is the *player’s* turn [Use Case 2 Takes Turn] * The player has ‘rolled’ and if necessary paid any licence fees applicable |
| **Post Conditions** | * Player will either perform the selected action or quit the game. |
| **Flow** | 1. The *player* views the menu. 2. The player may choose any option on the menu: 3. Purchases Start-up – [Extension D]. 4. Takes Over Start-up – [Extension F]. 5. Hires Staff – [Extension I]. 6. Ends Turn – [Extension Point H]. 7. Terminates Game – [Extension G]. |
| **Alternative Flow** | None |
| **Extension Points** | D. Purchases Start-up – [Use Case 7]  F. Takes Over Start-up – [Use Case 12]  I. Hires Staff – [Use Case 10]  H. Ends Turn – [Use Case 11]  G. Terminates Game – [Use Case 13] |
| **Inclusions Points** | None |
| **Use Case 10** | **Hires Staff** |
| **Summary** | *Player* hires a new member of staff for their Start-up space. The player must own all spaces in that field. They can then hire a software developer OR if three developers are present on the space, then they can purchase one CTO to place on that space. Each space can only have a maximum of three developers OR one CTO present. |
| **Actors** | *Player* |
| **Triggers** | *Player* selects ‘Hire Staff’ from the menu[Use Case 9 Views Menu] |
| **Pre Conditions** | * It is the *player’s* turn [Use Case 2 Takes Turn]. * The player is on any space. * The player owns all corresponding spaces relating to that field. * The player has sufficient funds to employ the corresponding Developer/CTO. * The number of staff on the relevant field will not exceed the limit of staff once this hiring process is complete. |
| **Post Conditions** | * Player will have relevant amount of money debited from their account. * Player will have a new member of staff on the desired field. * Player will be shown new account balance. * Player will be told they have hired the relevant staff member at the relevant start-up. |
| **Flow** | 1. The *player(s)* selects the ‘Hires Staff’ menu option. 2. The player selects the area they wish to develop. 3. The player confirms they wish to proceed with the hiring of staff [Inclusion D] 4. The relevant amount of money is deducted from the players account. |
| **Alternative Flow** | 1. At flow point 1:- If the player has insufficient funds to hire a member of staff at this start up a message is displayed stating so and player is returned to view menu [Use Case 9 Views Menu]. 2. At flow point 2:- If the player already has too many staff hired at the startup in question; a message is displayed stating so and the player is returned to the Views Menu [Use Case 9 Views Menu]. |
| **Extension Points** | None |
| **Inclusions Points** | D. Verifies Choice [Use Case 14] |
| **Use Case 11** | **Ends Turn** |
| **Summary** | A player ends their turn in the game. The game continues. |
| **Actors** | Player(s) |
| **Triggers** | The *player* selects the “end turn” option from the menu. |
| **Pre Conditions** | * It is the *player’s* turn [Use Case 2 Takes Turn]. * The *player* has already ‘rolled’ the dice and landed on a new space. * If required, the player has paid the necessary licence fee to another *player*.[Use Case 7 Pays Licence Fee] * The *player* has selected the “view menu” option [Use Case 9 Views Menu]. |
| **Post Conditions** | * The player’s turn ends * The system produces a message declaring that it is the next player’s turn. |
| **Flow** | 1. The *player* chooses the “view menu” option. [Inclusion A] 2. The *player* chooses the “end turn” option.[Extension H] 3. The system asks the *player* if they are sure they want to end their turn. [Inclusion C] 4. The *player* chooses the “yes” option. 5. The *player’s* turn ends. |
| **Alternative Flow** | 1. At flow point 4:- The player chooses the “No” option and they are returned to the menu. Their turn has not ended. |
| **Extension Points** | H. Ends Turn [Use Case 11] |
| **Inclusion Points** | C. Verifies Choice [Use Case 14]  A. Views Menu [Use Case 9] |
| **Use Case 12** | **Takes over Startup** |
| **Summary** | Player stages a Takeover, offering the initial price of the start-up(they wish to Take Over) plus 20% on top. Takeovers need to be agreed by **both** parties. |
| **Actors** | *Players (Always 2 players)* |
| **Triggers** | The *Player(whose turn it is)* has selected the “Trades Start-up” option on the Menu. |
| **Pre Conditions** | * It is the *player* who wishes to perform the takeover’s turn – [Use Case 3]. * The player selects “Takeover Start-up” option from the play menu. |
| **Post Conditions** | * The current player has the required money for the transaction removed from their inventory. * The relevant player, who sold their start-up with the current player, has the start-up in question removed from their inventory.(If they agree to the Take Over). * The current player gains the Start-up from the other player and it is added to the current players inventory. * The relevant player, who sold their start-up, gains the relevant amount of money and it is added to their inventory. |
| **Flow** | 1. The current *player* selects the other player they wish to purchase a start-up from in a sub menu (each player is numbered, current player is not shown in sub menu). 2. The current player selects the start-up they wish to acquire from the previously selected player from a sub menu (each owned start-up is numbered). 3. The current player is prompted with a message displaying the cost of the property they are about to purchase and asked if they wish to proceed.[Inclusion F] 4. The other player involved in the transaction is asked to confirm the sale of said startup. [Inclusion F] 5. The trade is completed. 6. The current player is shown the Menu – [Use Case 9] |
| **Alternative Flow** | 1. At flow point 1:- The current player may choose to cancel the takeover from the provided sub menu.    * Current player Views Menu [Use Case 9]. 2. At flow point 2:- The current player may choose to cancel the takeover from the provided sub menu.    * Current player Views Menu [Use Case 9]. 3. At flow point 3:- The current player may choose to cancel the trade from the provided verification message.    * Current player Views Menu [Use Case 9]. 4. At flow point 4:- The other player involved may choose No when prompted if they wish to proceed with the takeover.    * Current player Views Menu [Use Case 9]. |
| **Extension Points** | None |
| **Inclusions Points** | F. Views ‘Verifies Choice’ [Use Cases 14]. |
| **Use Case 13** | **Terminates Game** |
| **Summary** | A player terminates the game and the player with the most resources is declared the winner. |
| **Actors** | Player(s) |
| **Triggers** | The *player* selects the “terminate game” option from the menu. |
| **Pre Conditions** | * It is the *player’s* turn [Use Case 2 Takes Turn]. * The *player* has already ‘rolled’ the dice and landed on a new space. * If required, the *player* has paid the necessary licence fee to another player.[Use Case 7 Pays Licence Fee] * The *player* has selected the “view menu” option [Use Case 9 Views Menu]. |
| **Post Conditions** | * The system produces a message declaring that a *player* has terminated the game and the game is over. * The *player* with the most resources is declared the winner. |
| **Flow** | 1. The *player* chooses the “View Menu” option.[Inclusion A] 2. The *player* chooses the “Terminate Game” option. 3. The system asks the *player* if they are sure they want to terminate the game. [Inclusion E] 4. The *player* chooses the “yes” option. 5. The game is terminated, and the winning *player* is displayed |
| **Alternative Flow** | * At flow point 3:- The player chooses the “No” option and is returned to the menu. |
| **Extension Points** | None |
| **Inclusions Points** | A. Views Menu [Use Case 9]  E. Verifies Choice [Use Case 14] |
| **Use Case 14** | **Verifies Choice** |
| **Summary** | The *player* verifies a selected action from the menu presented to them. They can only select yes or no. No returns them to the menu. Yes carries out the required action. |
| **Actors** | *Player(s)* |
| **Triggers** | * The *player* attempts to use the “Purchase Start-up” option.[Use Case 7] * The *player* attempts to use the “Takes Over Start-up” option.[Use Case 12] * The *player* attempts to use the “Terminate Game” option.[Use Case 13] * The *player* attempts to use the “Ends Turn” option. [Use Case 11] * The *player* attempts to use the “Hires Staff” option. [Use Case 10] |
| **Pre Conditions** | * The *player* attempts to use the “Purchase Start-up” option.[Use Case 7] * The *player* attempts to use the “Takes Over Start-up” option.[Use Case 12] * The *player* attempts to use the “Terminate Game” option.[Use Case 13] * The *player* attempts to use the “Ends Turn” option. [Use Case 11] * The *player* attempts to use the “Hires Staff” option. [Use Case 10] |
| **Post Conditions** | The selected option is confirmed and the system notifies the *player* with the corresponding confirmation message. |
| **Flow** | 1. The player attempts to use one of the following options in the menu: 2. “Purchase Start-up”[Use Case 7] 3. “Takes Over Start-up” Use Case 12] 4. “Terminate Game” Use Case 13] 5. “Ends Turn” [Use Case 11] 6. “Hires Staff” [Use Case 10]   2. The system prompts the user to verify the selected choice.  3. The *player* confirms the choice.  4. The system displays a confirmation message to the *player.* |
| **Alternative Flow** | 1. At flow point 3:- The *player* may choose to cancel the selected choice.   * + Current player Views Menu [Use Case 9]. |
| **Extension Points** | None |
| **Inclusions Points** | None |

iii. Use Case Diagram