The Problem and the Early Solution

Group 45

Authored by: Adam Logan

XXXXX

XXXXX

XXXXX

XXXXX

Queen's University Belfast

Technopoly

Contents

Peer Assessment 1 Form

Evaluation

Evaluation	Group Number: 43			
Name	Contribution to team-working and motivation[1]	Contribution to PDF Report 1 [1,2]	Contribution to Interim Demo[1,2]	**Peer Score** (Range 85 – 115)
Adam Logan	5	5	5	115
XXXXX	5	4	5	105
XXXXX	1	2	2	95
XXXXX	5	4	2	95
XXXXX	4	4	2	100

Values for contribution: 1 = Minimal Contribution; 2 = Reasonable Contribution; 3 = Good Contribution; 4 = Very Good Contribution; 5 = Excellent Contribution

Declaration

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!)*
Adam Logan	01/12/2021	I agree to the terms of the declaration
XXXXX	01/12/2021	I agree to the terms of the declaration
XXXXX	01/12/2021	I agree to the terms of the declaration
XXXXX	01/12/2021	I agree to the terms of the declaration
XXXXX	01/12/2021	I agree to the terms of the declaration

Personal Statements

Personal statement of XXXXX:	
The following were my most significant contributions to the Semester 1 Deliverable (100 words or less):	
The use case for winning and player eliminated.	

Personal statement of XXXXX:	
The following were my most significant contributions to the Semester 1 Deliverable (100 words or less):	
Use cases for Income Event, Request to buy warehouse, and system builds warehouse. Class diagram. Gantt chart. Code for product class and square class. Use case realisation for player options. Explanation of the game in the report. Weekly minutes report 15/11/21.	

Personal statement of Adam Logan:	
The following were my most significant contributions to the Semester 1 Deliverable (100 words or less):	
All use case realisations/sequence diagrams, gantt chart, class diagram, the Game class for demo, weekly minutes for weeks 25/10/2021, 01/11/2021, 08/11/2021 and the use case descriptions for Choice to Play Game, Stock Exchange Game, Lost Stock Exchange, Money Out, Anti-Trust, Money In and Charged Product.	

Personal statement of XXXXX:	
The following were my most significant contributions to the Semester 1 Deliverable (100 words or less):	
Use Cases and descriptions: Start of quarter, dice rolling, fields. Weekly minutes report for the week of 29/11/2021. Game board draft layout, table of prices when paying commission for products, Gantt chart, code for dice method and class diagram.	

Personal statement of XXXXX:	
The following were my most significant contributions to the Semester 1 Deliverable (100 words or less):	
Use cases for Warehouses and Product Bidding (including descriptions). Weekly minutes report for 22/11/2021. Also completed moneyOut use case realisation. Completed the class diagram, gantt chart, code for Player class.	

Game Description

The game begins with all players on the red 'Start of Quarter' square on the bottom right of the board, the same place as 'GO' in monopoly. Players take turns to roll the dice and move accordingly with the number that they roll. When a player lands on a product square which hasn't been purchased already, they will be given an option to buy this product.

There is a 'Product Bidding' square, and when a player lands on this their products are put up for auction, meaning that all players on the board have the opportunity to bid on a product of the player who landed on the square. Whichever player bids the highest will receive ownership of the product.

When a player lands on a square which is owned by another player, they are charged for the use of the product, with the amount charged being dependent on what the product is. Some products will not charge players too much to use, while some will be more expensive. Players should aim to own all squares of a certain company; this is because once they own all products of a company, they get the opportunity to build warehouses on their products. When a player wants to build a warehouse, provided they are able to, they are charged for each warehouse they build. When a player lands on a square with a warehouse on it, they are charged an increased rate on what the standard cost to use the product would be, this gives an incentive for players to build warehouses. The reason why there is an increased rate is due to the product being improved by them being worked on in the warehouse.

When a player has built warehouses on all squares of a field, they are then given the option to select a region. Selecting a region comes at a cost, but it also comes with a reward as it allows a player to expand their business to the whole region, meaning that when a player lands on their square, they will have to pay an even higher rate than before. There are different regions that are available to select, and each has its own pros and cons, for example some regions will cost more to build on but will charge more money for players who land on their squares.

Each time a player passes the 'Start of Quarter' square they receive a bonus. When a player passes the 'Start of Quarter' square four times, marking that a full year has passed, the income event will trigger. This event checks the player's income for the year, and if it is below a certain amount then the player is eliminated from the game. The amount of income required will depend on how many players are left in the game, and when a player is eliminated, the amount will go up as the remaining players will have had the opportunity to purchase the products which were owned by the eliminated player.

There are 'Stock Exchange' squares around the board, and when a player lands on them they can decide to take a gamble, picking a number between a certain range; if the player guesses the same number as the computer then they will double their money, but if they're wrong then they will lose money, the amount they lose depends on how far away their guess was from the value produced by the computer.

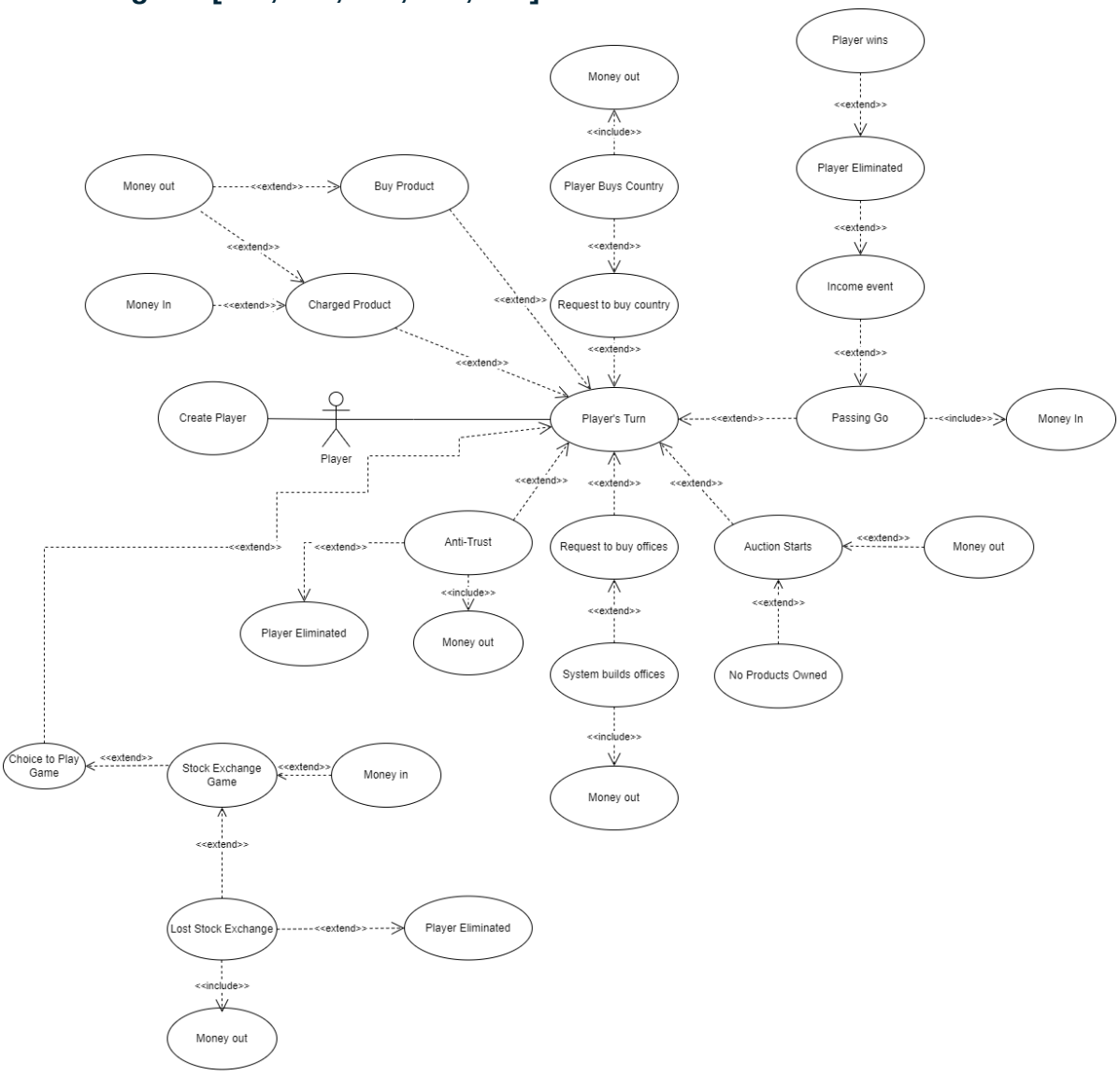
There is a 'Tax' square which when landed on, charges the player 10% of their current value. There is also a 'Bank Holiday' square which does nothing, this is not very useful in the beginning of the game, but towards the endgame it means the player gets a free pass on their turn instead of landing on a product owned by someone else and having to pay them.

Each product has its own unique value, and this value changes with each upgrade that is purchased. The last player left on the board will be the winner after all other players have been eliminated.

Use Case Requirements Specification and Planning

Use Cases

Use Case Diagram [A.L, X.X, X.X, X.X, X.X]



Use Case Descriptions [A.L, X.X, X.X, X.X, X.X]

Choice to Play Game [A.L]	
Objective	To give the player a choice to play the Stock Exchange game
Precondition	A player lands on the square
Main Flow	1. A message is displayed to the player giving them a choice to either play the game or not 2. The player selects one of the options

Alternative Flows	• At 2 the player:
	  ○ Chooses to play the game then the 'Stock Exchange Game' use case is triggered
	  ○ Chooses to not play the game and a quit message is displayed

Post-condition	The player either plays the game or not

Stock Exchange Game [A.L]	
Objective	The player will guess a number between
Precondition	A player has chosen to play the Stock Exchange game
Main Flow	1. A random number is generated between 1 and 10 2. A message is displayed to the player prompting them to select a number between 1 and 10 3. The player enters a number

Alternative Flows	• At 3 the player:
	  ○ Selects the correct number and the 'Money In' use case is triggered
	  ○ Selects an incorrect number and the 'Lost Stock Exchange' use case is triggered

Post-condition	The player will either have earned money or have lost money

Lost Stock Exchange [A.L]	
Objective	The player will have lost money
Precondition	A player has chosen the incorrect number in the Stock Exchange Game
Main Flow	1. The number the player has entered is checked to see how far away the player was to the incorrect answer 2. The player loses money

Alternative Flows	• At 2 the player loses:
	  ○ A small amount of money if they are 2 off the correct answer
	  ○ A large amount of money if they are more then 2 off the correct answer

Post-condition	The player has gained money

Money Out [A.L]	
Objective	The money is deducted from the player's balance
Precondition	N/A
Main Flow	1. An amount of money is supplied 2. This amount of money is subtracted from the player's balance
Alternative Flows	N/A
Post-condition	The player has lost money

Anti-Trust [A.L]	
Objective	The player loses a certain percentage/predetermined amount of money
Precondition	A player lands on the square
Main Flow	1. A message is displayed to the player stating that they have lost money 2. Money is deducted from the account of the player
Alternative Flows	N/A
Post-condition	The player has lost money

Money In [A.L]	
Objective	The money is increased from the player's balance
Precondition	N/A
Main Flow	1. An amount of money is supplied 2. This amount of money is added from the player's balance
Alternative Flows	N/A
Post-condition	The player has gained money

Charged Product [A.L]	
Objective	A player is charged for using a product in which they have landed on
Precondition	A player has landed on a product area which another player already owns
Main Flow	1. A certain amount of money is deducted from the player's balance who landed on the product 2. The same amount of money is added to the player's balance who own the product
Alternative Flows	N/A
Post-condition	One player has lost money and another has gained money

Player's Turn [X.X]	
Objective	The player will be given options and then they will roll the die
Precondition	It is the player's turn
Main Flow	1. The player will be given the option to purchase an warehouse or country or to do nothing 2. The system will 'roll the dice' by randomly generating two random numbers between 1 and 6 3. Player will be moved to the area counting the total from their current position

Alternative Flow	• At 1 the player chooses to purchase:
	  ○ An warehouse and the 'Request to buy warehouses' use case is triggered
	  ○ A country and the 'Request to buy warehouses' use case is triggered
	• At 2:
	  ○ If the system generates two of the same number this step repeats itself
	  ○ If the system generates two of the same number three times in a row then the player is moved to the auction square
	• At 3 if the player lands on the:
	  ○ Product area the 'Lands on product' use case is triggered'

Post-Condition	Player rolls dice and two numbers are randomly generated

Passing Go [X.X]	
Objective	When the player passes by the starting spot, they will receive 20% bonus of their total balance
Precondition	Player must first pass by the area where the game started
Main Flow	1. The system will store the number of times a player has passed by the start 2. Player receives 20% of their total balance
Alternative Flow	• If the player has passed the starting spot four times the 'Income event' use case is triggered
Post-Condition	Player rolls the die number that allows them to land on or pass by the starting square, start of quarter will trigger and if passed four times income event will too

Buy Product [X.X]	
Objective	System gives the product to the player and takes the proper funds from their balance
Precondition	Player must purchase an available field from the system
Main Flow	1. Player purchases an available field from system 2. The system will give the player the product 3. The 'money out' use case will be triggered
Alternative Flow	• At 1: ○ The player does not have enough funds to purchase the product and therefore the system does not give the player the product but displays an appropriate message instead
Post-Condition	The player purchases a field, the system will give it to them, while also removing the amount of funds needed from their balance

Winning [X.X]	
Objective	This use case occurs when only one player is left and wins the game
Precondition	Only one player is left in the game
Main Flow	1. Display congratulations message 2. Close the application
Alternative Flows	N/A
Post-condition	The game will no longer continue and the application will be closed

Player Eliminated [X.X]	
Objective	This use case occurs when one player loses the game
Precondition	The player must have 0 cash
Main Flow	1. Display message 'player's name is out of the game!' 2. Remove the player out of the game 3. Return all companies owned by the player back to the system
Alternative Flows	N/A
Post-condition	The player who lost will no longer be able to play in the game. The companies previously owned by the player will go back to the system and be able to be purchased by the other remaining players for the duration of the game.

Create Player [X.X]	
Objective	This displays the steps in which the player must take to create their player for the game
Precondition	The game has started
Main Flow	1. Display the message 'What is your name?' 2. Player will enter the name they wish to display during the game
Alternative flows	• At 2: ○ If the player enters a non-letter value into their name, the system will display the message 'Error. Please enter a name with only letters'
Post-condition	All players will be put into the game and the first players turn will begin.

Request to buy country [X.X]	
Objective	Player will be able to purchase warehouses in a selection of 4 different countries (each country has different rent).
Precondition	Users must have all properties of the same field owned, must have 3 warehouses on each field bought before they relocate to a country.
Main Flow	1. Player choices product which they want to build an warehouse for 2. Player choices which country that they wish to purchase 3. System checks if they own the required products and have enough money. 4. System checks if the player owns 3 warehouses on all of the same-coloured fields. 5. The cost of landing on the product is increased to the appropriate amount 6. The 'System build country' use case is triggered

Alternative flows	• At 3:
	  ○ The player does not own the selected product and therefore an appropriate message is displayed and the 'System build warehouse' use case is not triggered
	  ○ The player does not have enough money within their balance and therefore an appropriate message is displayed and the 'System build warehouse' use case is not triggered
	• At 4:
	  ○ The player does not own the required warehouses on the selected product and therefore an appropriate message is displayed and the 'System build warehouse' use case is not triggered

Post-condition	Player has purchased a country

Player Buys Country [X.X]	
Objective	The player will own a country
Precondition	Players must meet the criteria to own a country.
Main Flow	1. System gives ownership of country on desired product 2. The cost of landing on the product is increased to the appropriate amount 3. The 'money out'' use case is triggered
Alternative flows	N/A
Post-condition	The player owns a country

Auction Starts [X.X]	
Objective	One of the player's products goes up for auction for other users to bid on.
Precondition	Player lands on specific square on board
Main Flow	1. System puts one of the player's product up for auction 2. System asks other players which product they would like to bid on from the player. 3. System takes turns to ask users how much they would like to bid. 4. At the end of the auction (once 1 player is left, or time is up), System then determines who had the highest bid. 5. System transfers ownership of the product to the winning player and charges the winning player the amount of the bid
Alternative flows	• At 2: ○ If the player doesn't have a company to auction up, then the 'No Products Owned' use case is triggered
Post-condition	Company will be auctioned, and winning player will get the company

No Products Owned [X.X]	
Objective	To move the player to the correct area
Precondition	If player has no products and lands on the auction area
Main Flow	1. Displays an appropriate message to user 2. Moves player to most expensive field and, if owned, has to pay player money
Alternative flows	N/A
Post-Condition	The player has moved to the correct area

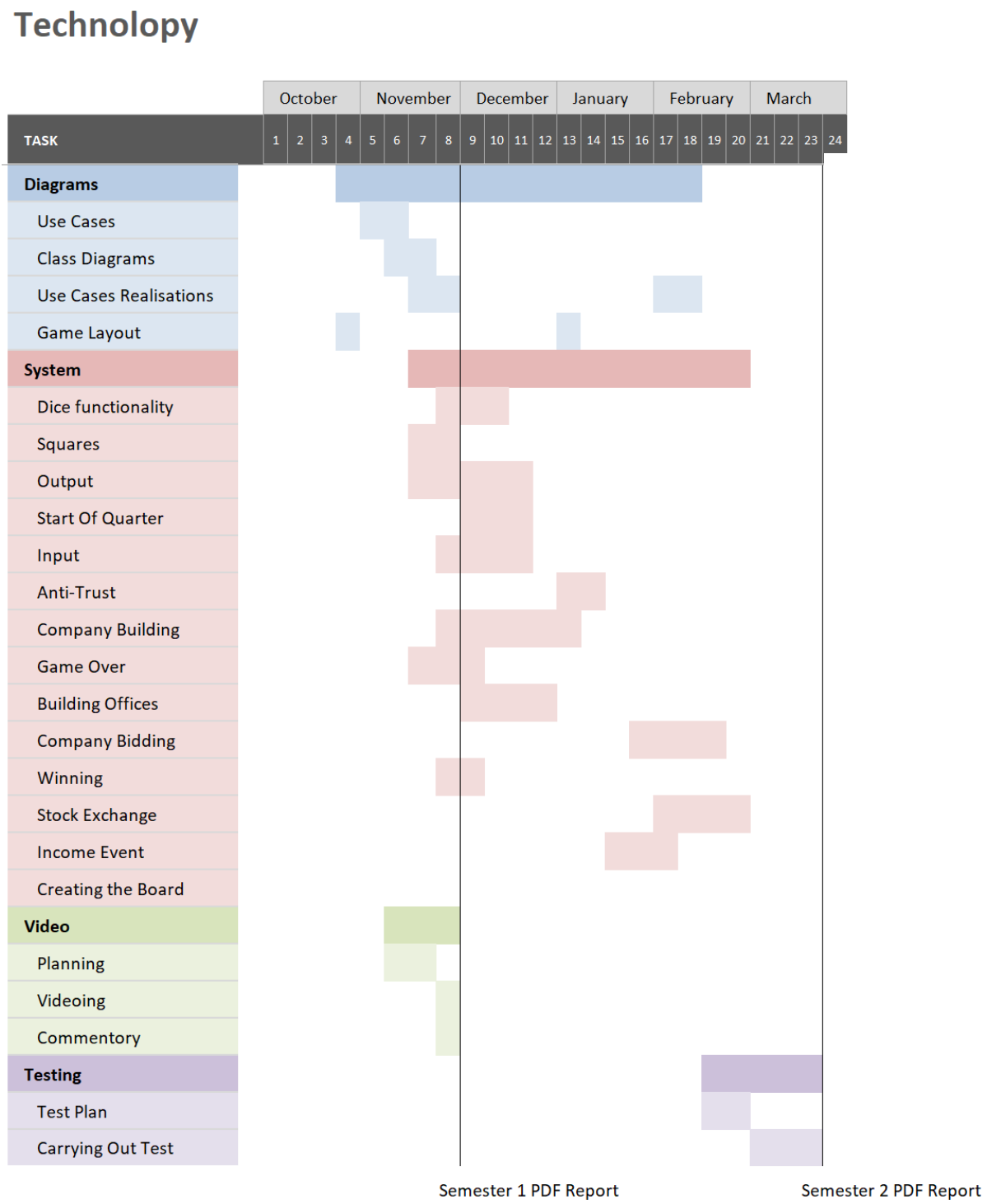
Request to buy warehouse [X.X]	
Objective	Player requests to buy an warehouse.
Precondition	It must be the specific player's turn.
Main flow	1. Player choices product which they want to build an warehouse for 2. Player requests to buy an warehouse. 3. System checks if they own the required fields and have enough money. 4. The 'System build warehouse' use case is triggered
Alternative Flows	• At 2: ○ The player does not own the selected product and therefore an appropriate message is displayed and the 'System build warehouse' use case is not triggered ○ The player does not have enough money within their balance and therefore an appropriate message is displayed and the 'System build warehouse' use case is not triggered
Post-Condition	Player requests to buy warehouses.

System builds warehouse [X.X]	
Objective	Warehouses are built for player.
Precondition	Player must meet the criteria to build an warehouse.
Main Flow	4. System builds warehouses on desired product 5. The cost of landing on the product is increased to the appropriate amount 6. The 'money out'' use case is triggered
Alternative Flows	N/A
Post-Condition	Warehouses are built.

Income Event [X.X]	
Objective	Player receives money after they pass 'go' every four times
Precondition	The player has passed go 4 times
Main Flow	1. System checks if the player has enough money to pass the threshold
Alternative Flows	• At 1: ○ If the player does not have enough money to pass the threshold the 'Player is eliminated' use case is triggered
Post-Condition	N/A

Gantt Chart [A.L, X.X, X.X, X.X, X.X]

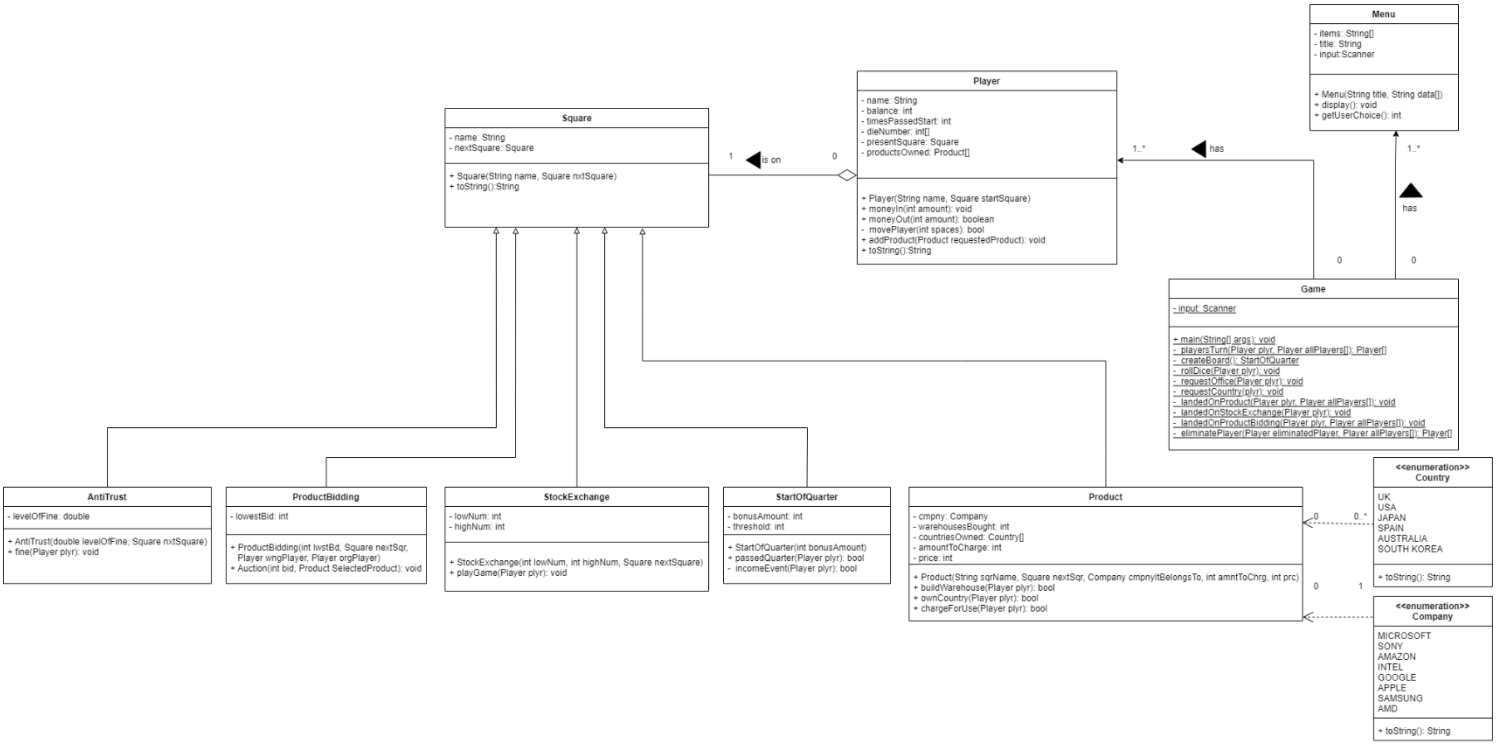
This Gantt chart shows the progress of the project in terms of weeks.



System Analysis

Class Diagram [A.L, X.X, X.X, X.X, X.X]

The class diagram itself [A.L, X.X, X.X, X.X, X.X]



Description of class diagram [A.L]

The class diagram above contains the candidate classes and their relationships for the system. **The getters and setters are not shown in the above diagram, but all the classes have both getters and setters for all their attributes.**

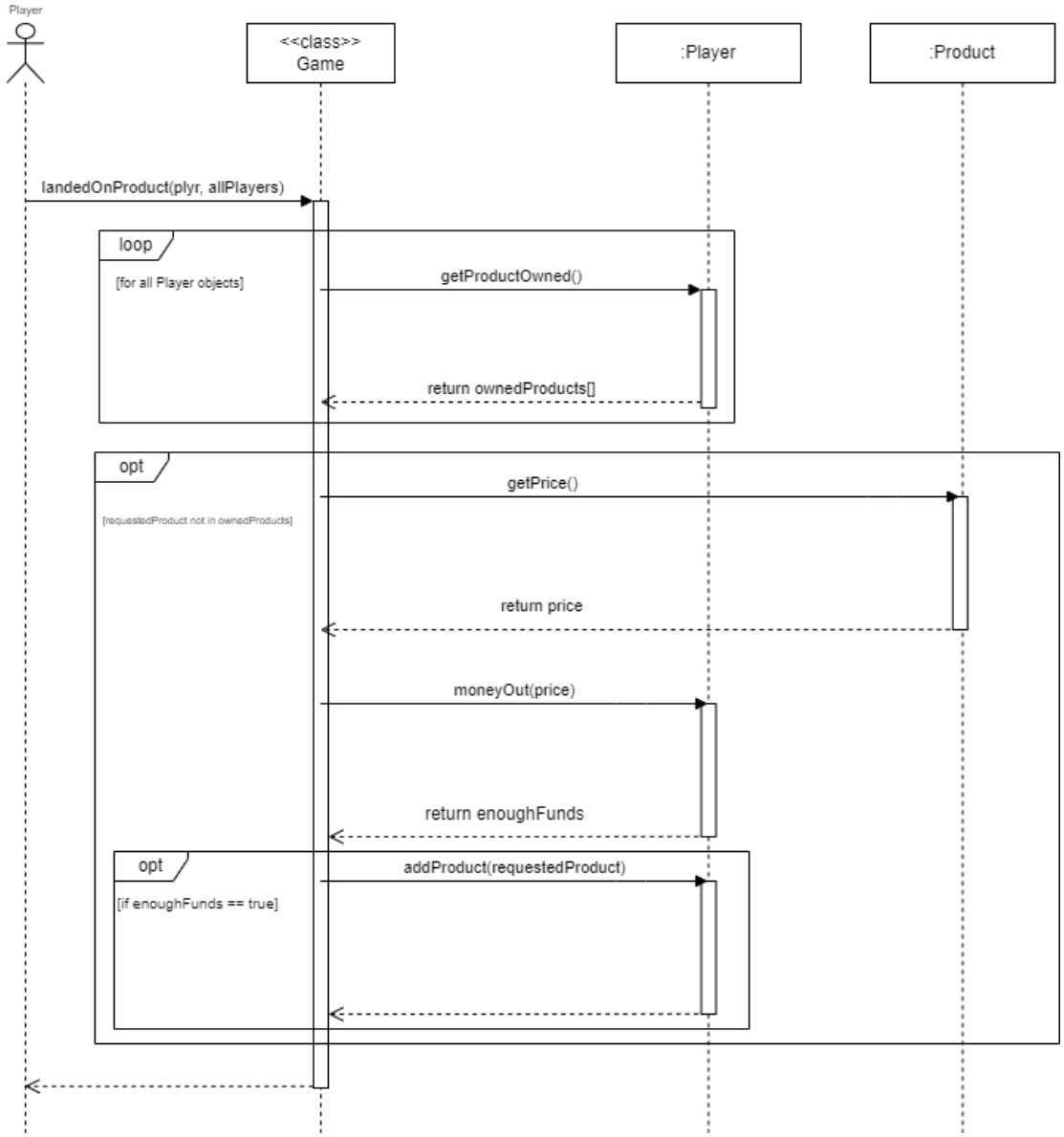
The Game class is the class which contains the 'main' method and is how the user/player interacts with the system. The Menu class is simply an easy way to create a menu for the user/player with validation. The Menu class was taken from a module in level 1, which all members of the team had enrolled. The relationship between Game and Menu is a simple association as the Menu objects are created within methods of the Game class and not used as attributes. The Menu class does not use an Object of the Game class, but the Game class creates many objects of the Menu class.

The relationship between Game and Player is a simple association as the Player objects are created within methods of the Game class and not used as attributes. The Player class does not use an Object of the Game class but the Game class creates many objects of the Player class. The relationship between the Square class and the Player class is Aggregation as if the Player object is deleted the Square class continues to exist. AntiTrust, ProductBidding, StockExchange, StartOfQuarter and Product are all special cases of Square and therefore inherit from the Square class. Both Company and Country are enumerations and are attributes of the Product class.

Use Case Realisations [A.L]

Buy Product [A.L]

Sequence Diagram

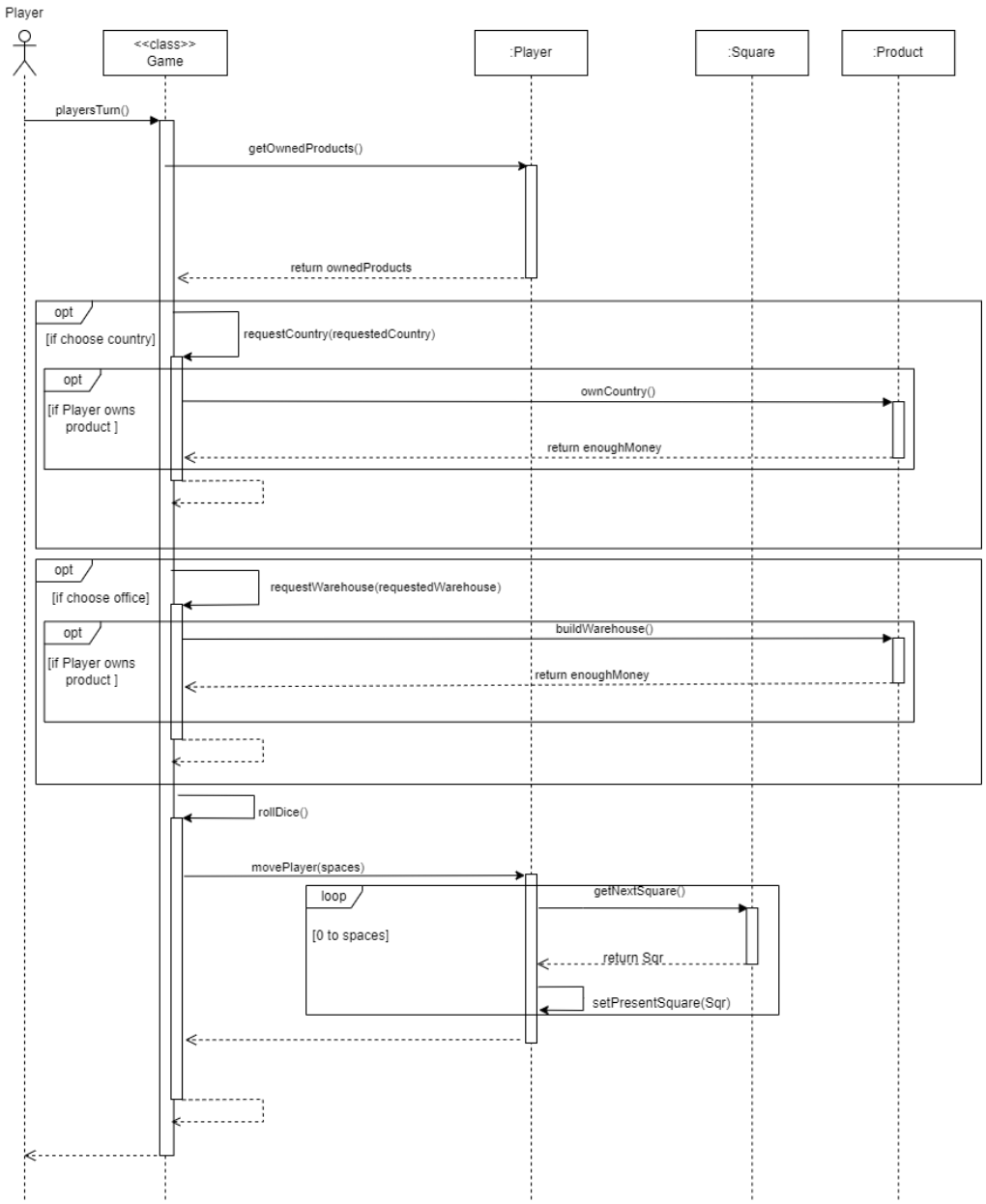


Description

This is the sequence diagram for the use cases 'Buy Product' which includes the 'Money Out' use case. First the system will loop through all the players in the game, to check if they already own the product which the user wishes to purchase. The system does this by getting what products each player owns and checks this against the requested product. If none of the players own the product then the system will get the price of the product, deduct that amount from the player and then update the player object with the product. The 'return eliminated' for the 'moneyOut(price)' method is a boolean variable which is true when the player does not have the balance required to purchase the product.

Player's Turn [A.L]

Sequence Diagram

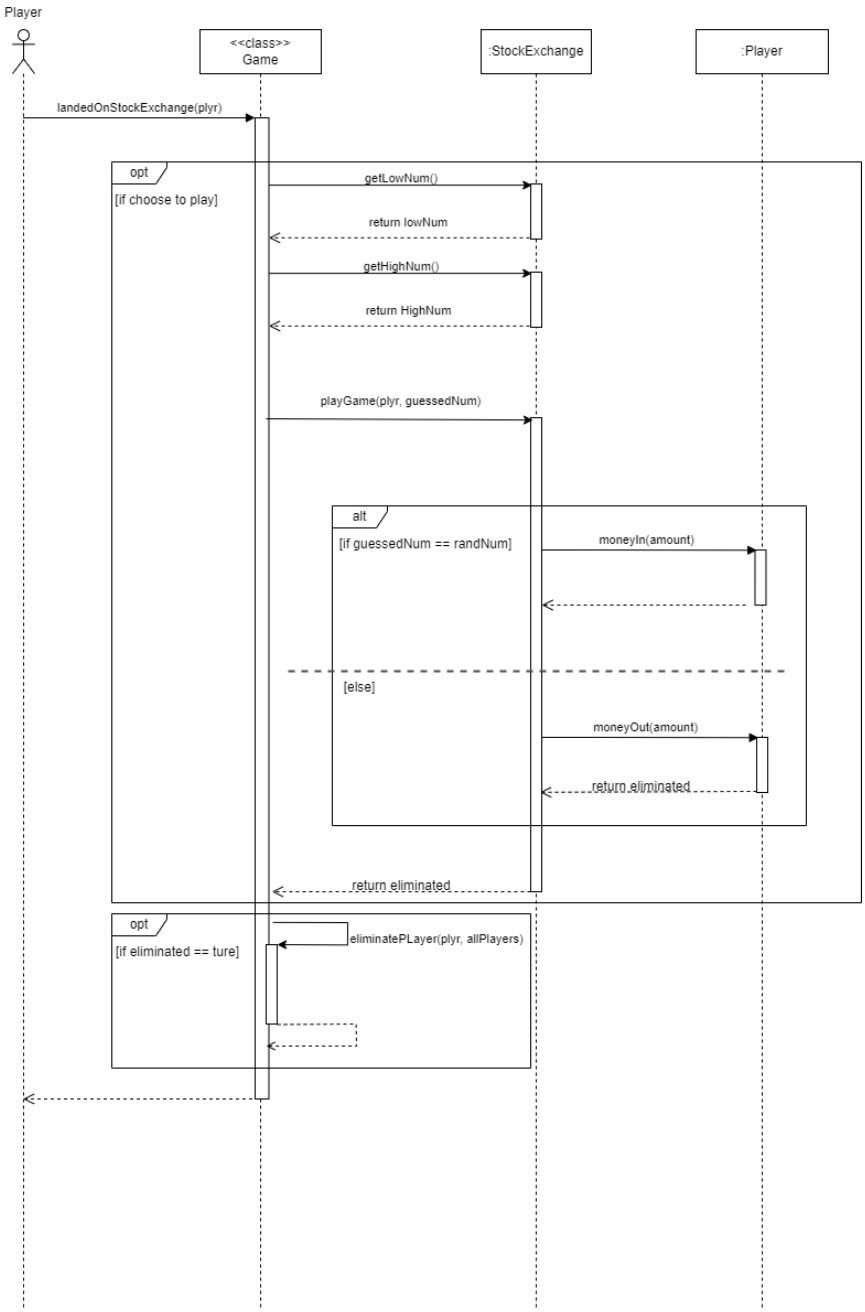


Description

This is the sequence diagram for the use cases 'Player's Turn', 'Request to buy country', 'Player Buys Country', 'Request to buy warehouse' and 'System builds warehouse'. This sequence diagram has omitted the 'Money Out' use case which always occurs within the use cases 'Player Buys Country' and 'System builds warehouse' to make the sequence diagram more readable. The system gets the products owned by the player and gives them the option to buy a country or a warehouse and when this is selected they will be displayed a list of their owned products. A check will be carried if they can purchase this on their product and if it passes then the system builds the office. After this has occurred the system moves the player by setting the 'presentSquare' attribute to the correct square.

Stock Exchange [A.L]

Sequence Diagram

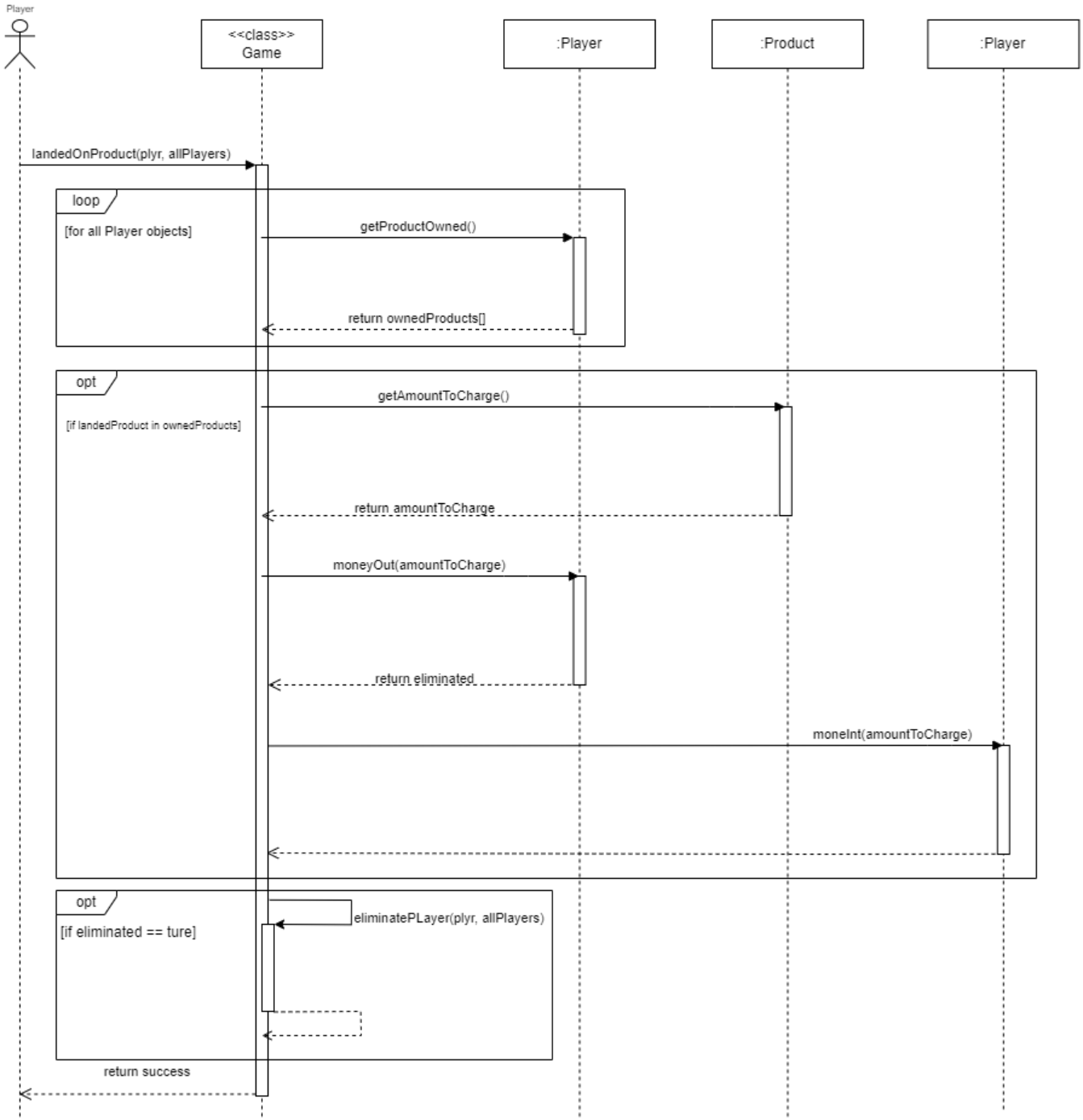


Description

This is the sequence diagram for the use cases 'Choice to Play Game', 'Stock Exchange Game', 'Money In' 'Lost Stock Exchange', 'Money Out' and 'Player Eliminated'. First the system will give the player the option to play the stock exchange game and if they choose yes, they will have to guess a number between a certain range. If the correct number is guessed then a certain amount of money is added to their balance but if they guess a wrong number then money id deducted from their bank account. Depending on how far their guess was to the correct number the amount of money deducted from their balance will change. If the user is does not have sufficient funds the 'moneyOut()' method will return true and the player will be eliminated form the game.

Charged Product [A.L]

Sequence Diagram

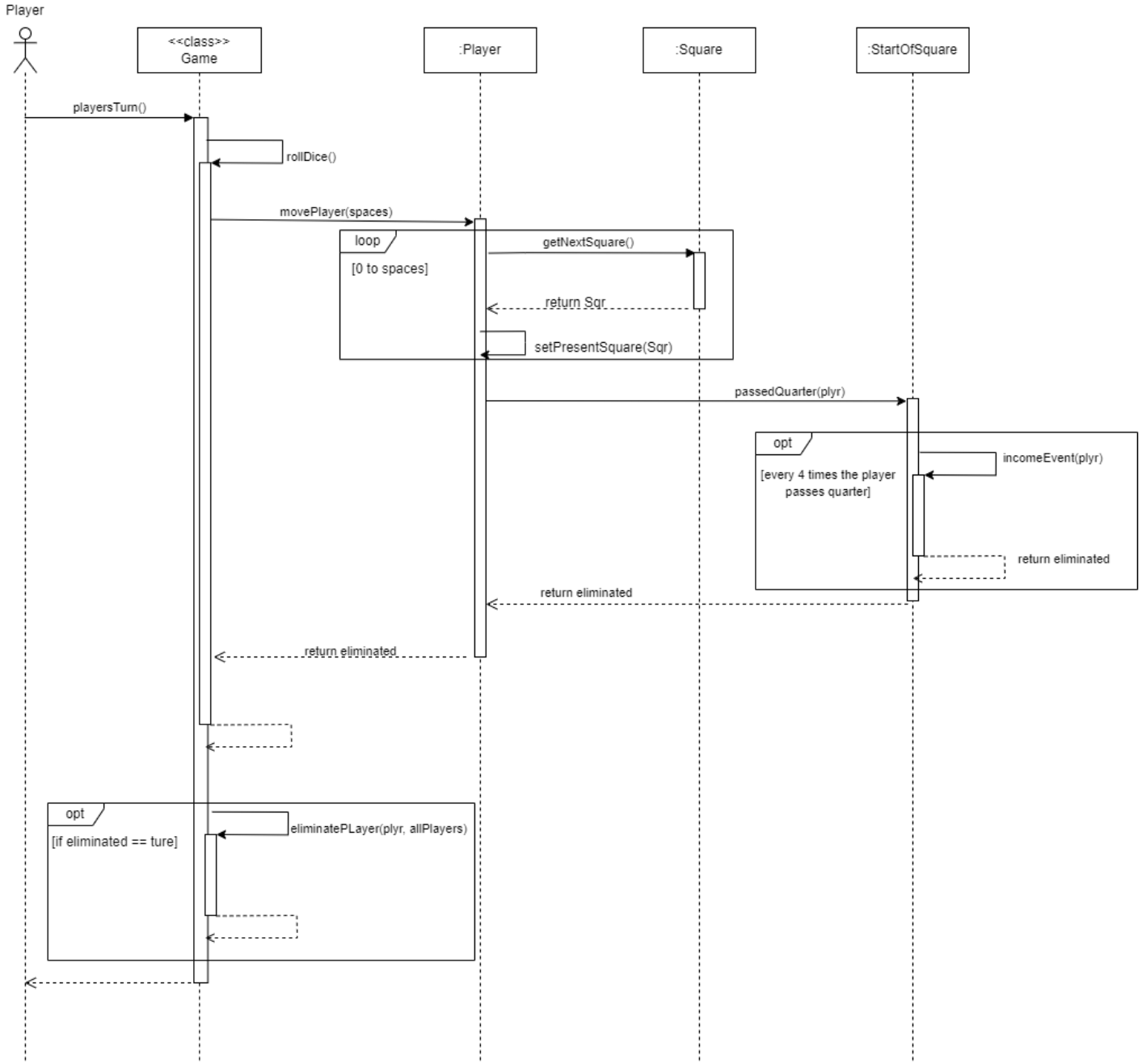


Description

This is the sequence diagram for the use cases 'Charged Product', 'Money In', 'Money Out' and 'Player Eliminated'. The for loop iterates through an array of Player objects and checks if the current player has landed on a product which another player owns. If this is the case the amount of money that needs to be charged to the player will be deducted and the player which owns the product will receive that same amount. If the charged player does not have sufficient funds the 'moneyOut()' method will return true and the player will be eliminated form the game.

Income Event [A.L]

Sequence Diagram

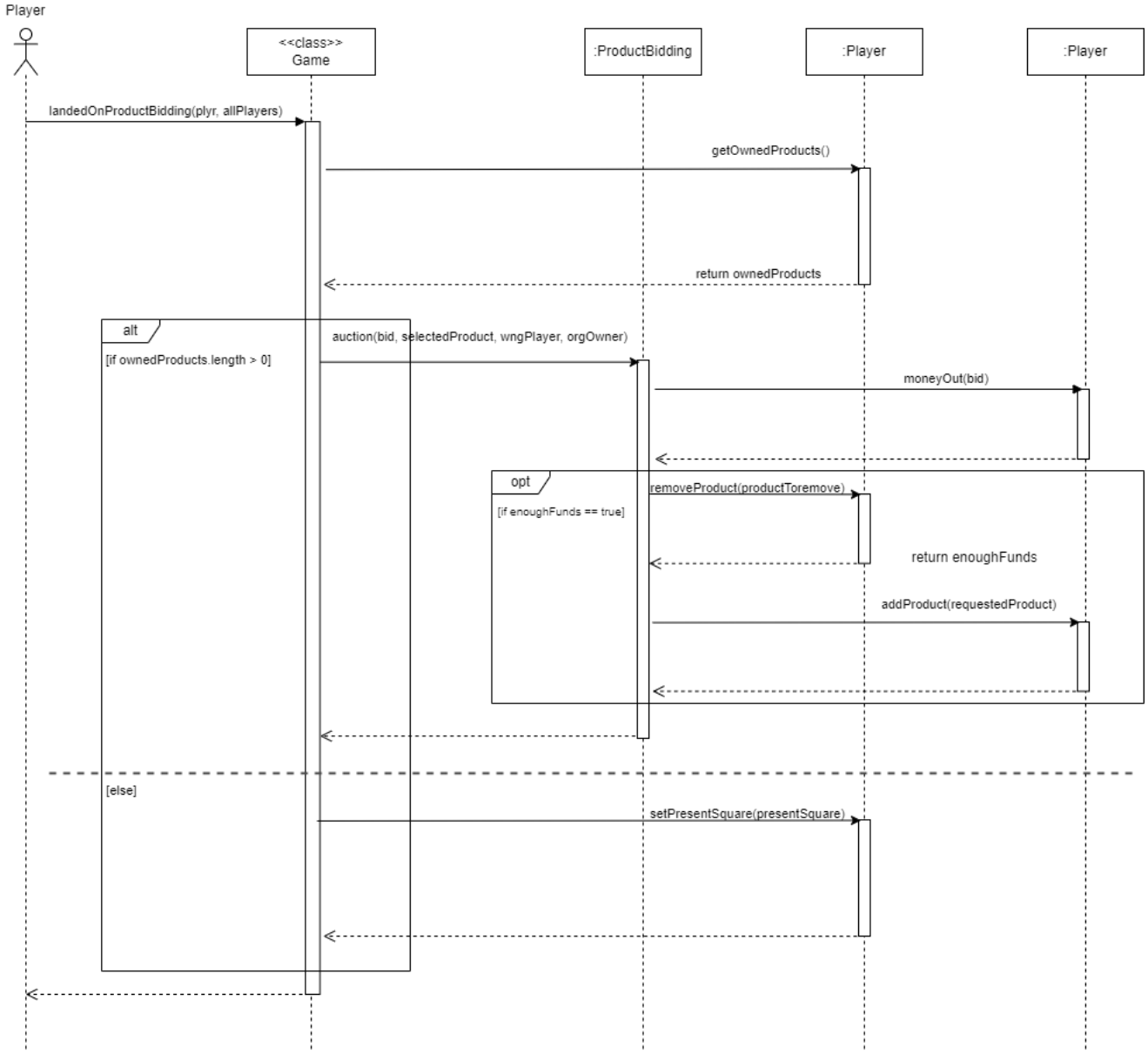


Description

This is the sequence diagram for the use cases 'Passing Go', 'Income Event' and 'Player Eliminated'. The system will trigger the income event every time the player has passed the 'start of quarter' 4 times and check if their income is above the threshold that is required. If their income is below the threshold then 'eliminated' will be true and the user will then be eliminated from the game.

Auction Starts [A.L]

Sequence Diagram

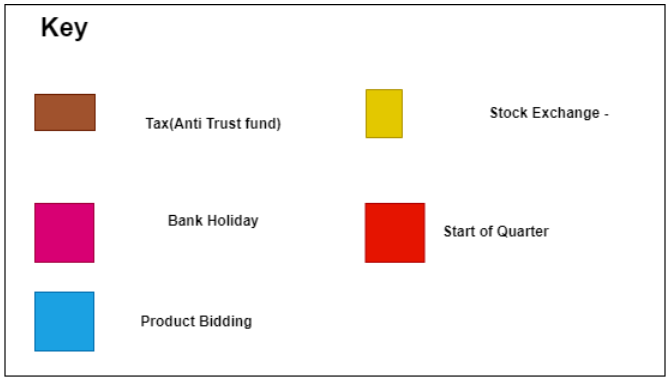


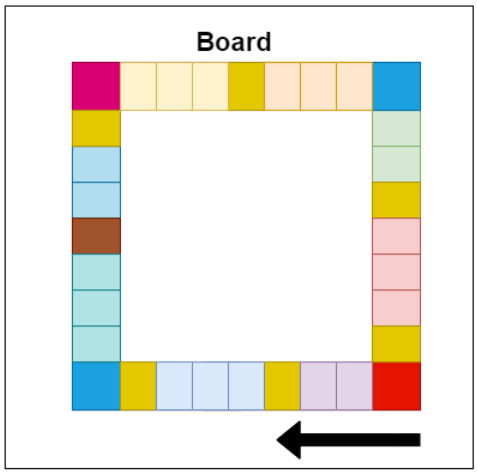
Description

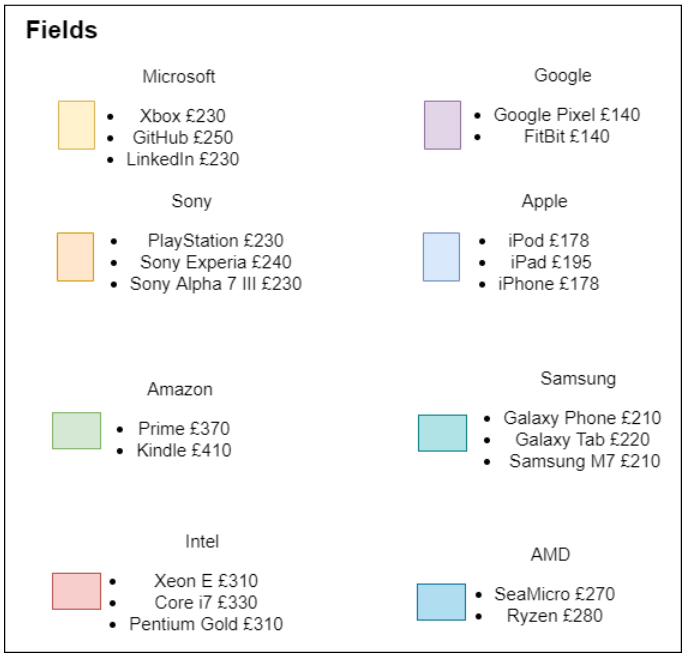
This is the sequence diagram for the use cases 'Auction Starts', 'Money Out' and 'No Products Owned'. First the system will get the products which the player who landed on the square and, if the player owns any products, the system will allow the other players to select a product to bid on. Once this has taken place the players will bid on the product and the winning player will be given the product. This works by first checking if the winning player has the funds to place the bid, removing the product from the original player and then giving the winning player the product. If the player who landed on the square has no products the system finds the most expensive square and moves the player there.

Draft Game Layout [X.X]

The Board [X.X]







Prices For Products [X.X]

Name	Basic	1 Warehouses	2 Warehouses	3 Warehouses	4 Warehouses	Country
Google Pixel	5	25	75	225	400	650
FitBit	5	25	75	225	400	650
iPod	7	35	105	315	460	780
iPad	9	45	135	405	595	890
iPhone	7	35	105	315	460	780
Galaxy Phone	10	50	150	450	600	895
Galaxy Tab	12	60	180	540	720	990
Samsung M7	10	50	150	450	600	895
SeaMicro	13	65	195	520	740	910
Ryzen	14	70	210	560	800	980
Xbox	15	75	225	575	825	1050
GitHub	18	90	270	660	990	1270
LinkedIn	15	75	225	575	825	1050
PlayStation	19	95	285	685	1000	1210
Sony Experia	21	105	315	755	1090	1440
Sony Alpha 7 III	19	95	285	685	1000	1210
Prime	35	185	525	980	1400	1900
Kindle	40	200	600	1120	1600	2300
Lenovo	26	130	390	805	1170	1560
HP	29	155	435	900	1305	1750
Dell	26	130	390	805	1170	1560

Appendix

Weekly Team Minutes [A.L, X.X, X.X, X.X]

25/10/2021 [A.L]

Minutes for CSC2058 Project 45 Week commencing 25/10/2021 Date of this minute - 26/10/2021

The following team members were present on Teams when minutes were discussed:

Name (printed/typed)	Signature (agreed bitmap or initials)
Adam Logan	A.L
XXXXX	X.X
XXXXX	X.X
XXXXX	X.X

Task Reporting

N/A as this was first meeting

Actions Taken During Meeting

During this meeting the team completed the gantt chart and the board layout together

Actions Planned

Adam Logan:

- To Complete the following Use Cases:
  - Stock Exchange
  - Anti-Trust

XXXXX:

- To Complete the following Use Cases:
  - Start of Quarter
  - Dice Rolling
  - Fields

XXXXX:

- To Complete the following Use Cases:
  - Game Over
  - Winning

XXXXX:

- To Complete the following Use Cases:
  - Countries
  - Company Bidding

XXXXX:

- To Complete the following Use Cases:
  - User
  - Income Event
  - Warehouse

01/11/2021 [A.L]

Minutes for CSC2058 Project 45 Week commencing 1/11/2021 Date of this minute – 3/11/2021

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)
Adam Logan	A.L
XXXXX	X.X
XXXXX	X.X
XXXXX	X.X
XXXXX	X.X

Task Reporting

Adam Logan:

- Completed the following Use Cases:
  - Stock Exchange
  - Anti-Trust

XXXXX:

- Completed the following Use Cases:
  - Start of Quarter
  - Dice Rolling
  - Fields

XXXXX:

- Completed the following Use Cases:
  - Game Over
  - Winning

XXXXX:

- Completed the following Use Cases:
  - Countries
  - Company Bidding

XXXXX:

- Completed the following Use Cases:
  - User
  - Income Event
  - Warehouses

Actions Taken During Meeting

During this meeting the team discussed and modified the individual use cases and merged them together into a singular use case diagram

Actions Planned

Adam Logan:

- To Complete the following Class Diagrams:
  - Stock Exchange
  - AntiTrust
  - Square
  - Fields
  - Products

XXXXX:

- To Complete the following Class Diagrams:
  - IncomeEvent
  - Dice
  - StartOfQuarter

XXXXX:

- To Complete the following Class Diagrams:
  - GameOver
  - Winning

XXXXX:

- To Complete the following Class Diagrams:
  - CompanyBidding
  - Country

XXXXX:

- To Complete the following Class Diagrams:
  - Player
  - Warehouse

08/11/2021 [A.L]

Minutes for CSC2058 Project 45 Week commencing 8/11/2021 Date of this minute 12/11/2021

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)
Adam Logan	A.L
XXXXX	X.X
XXXXX	X.X
XXXXX	X.X
XXXXX	X.X

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

Adam Logan:

- Completed the following Class Diagrams:
  - Stock Exchange
  - AntiTrust
  - Square
  - Fields
  - Products

XXXXX:

- Completed the following Class Diagrams:
  - IncomeEvent
  - Dice
  - StartOfQuarter

XXXXX:

- Completed the following Class Diagrams:
  - GameOver
  - Winning

XXXXX:

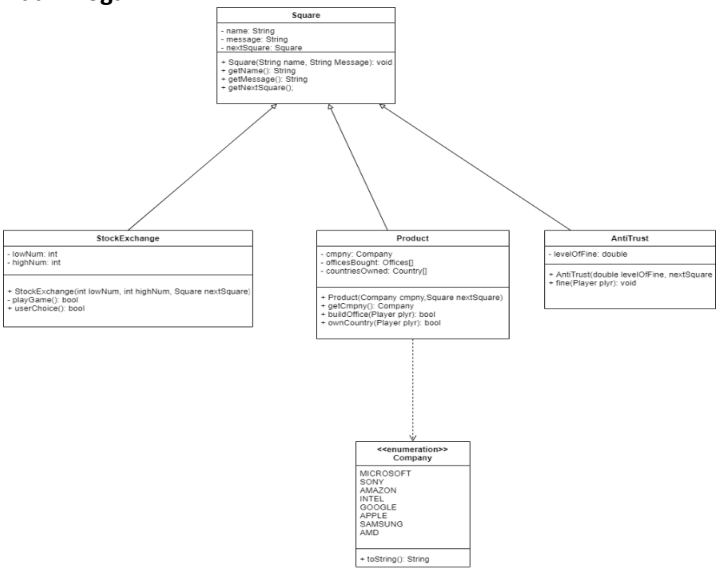
- Completed the following Class Diagrams:
  - CompanyBidding
  - Country

XXXXX:

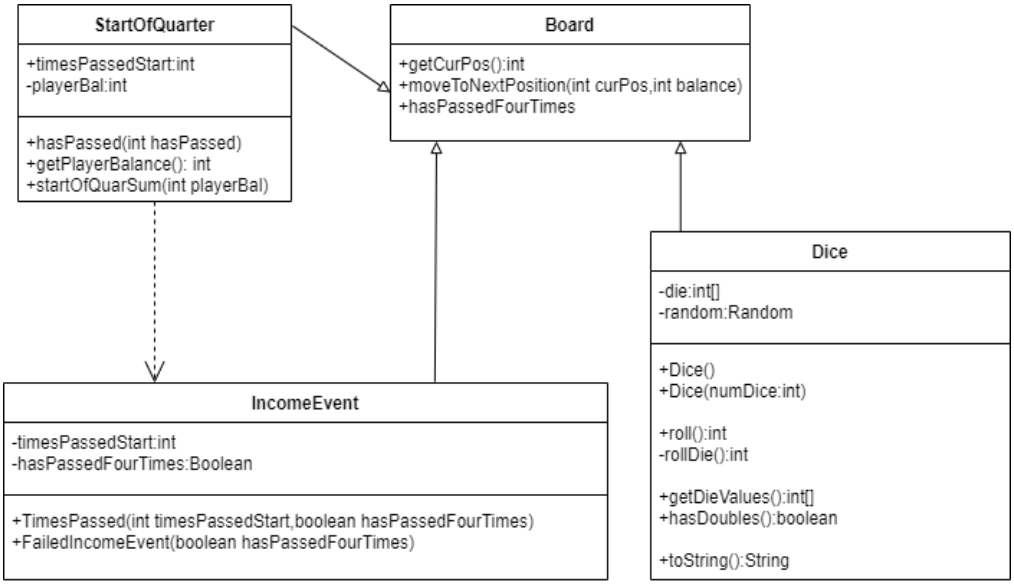
- Completed the following Class Diagrams:
  - Player
  - Warehouse

Interim Deliverables:

Adam Logan:



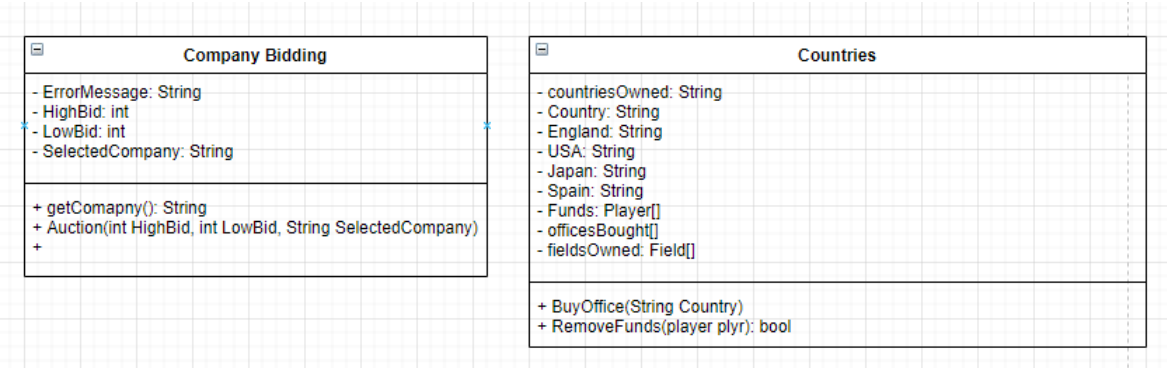
XXXXX:



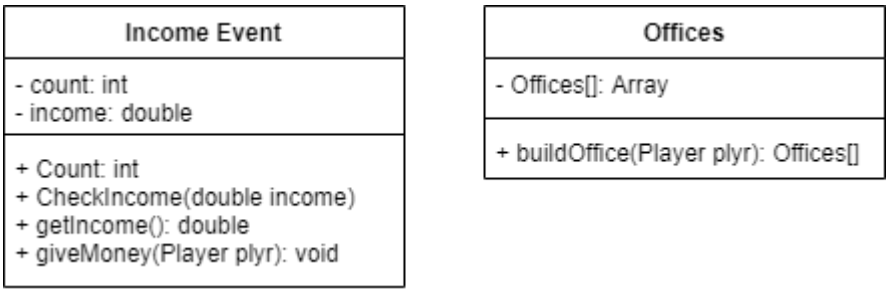
XXXXX:

(The team decided that the classes assigned to Marius should no longer be classes)

XXXXX:



XXXXX:



Actions Taken During Meeting

During this meeting the team discussed and modified the individual class diagrams and merged them together into a singular class diagram

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

Team members will decide on which use cases should be realised and will work together to create the sequence diagrams for these use cases.

15/11/2021 [X.X]

Minutes for CSC2058 Project 45 Week commencing 15/11/2021 Date of this minute - 17/11/2021

The following team members were present on Teams when minutes were discussed:

Name (printed/typed)	Signature (agreed bitmap or initials)
Adam Logan	A.L
XXXXX	X.X
XXXXX	X.X
XXXXX	X.X
XXXXX	X.X

Task Reporting

The team looked over everybodys' use case diagrams and agreed that they were all correct after making appropriate changes where needed.

Actions Taken During Meeting

During this meeting the team combined our use case diagrams and decided on use case realisation diagrams for each person to do.

Actions Planned

Adam Logan:

- To Complete the following Use Case Realisation:
  - Charged Product
  - Stock Exchange
  - Roll Dice
  - Player Turn
  - Pass Go

XXXXX:

- To Complete the following Use Case Realisation:
  - Country

XXXXX:

- To Complete the following Use Case Realisation:
  - Money In
  - Buy Product

XXXXX:

- To Complete the following Use Case Realisation:
  - Player Options

22/11/2021 [X.X]

Minutes for CSC2058 Project 45 Week commencing 22/11/2021 Date of this minute - 24/11/2021

The following team members were present on Teams when minutes were discussed:

Name (printed/typed)	Signature (agreed bitmap or initials)
Adam Logan	A.L
XXXXX	X.X
XXXXX	X.X
XXXXX	X.X

Task Reporting

N/A

Actions Taken During Meeting

During this meeting, the team completed code for some of the classes, reorganised the class diagram and finished the use case realisations

Actions Planned

Adam Logan:

- To Complete the following classes / Use Case Realisations:
  - Square class
  - playerTurn use-case realisation

XXXXX:

- To Complete the following classes / Use Case Realisation:
  - Dice class
  - Roll dice use-case realisation

XXXXX:

- To Complete the following classes / Use Case Realisation:
  -
  -

XXXXX:

- To Complete the following classes / Use Case Realisation:
  - Player Class
  - Buy product use-case realisation

XXXXX:

- To Complete the following classes / Use Case Realisation:
  - Product class
  - Country use-case realisation

29/11/2021 [X.X]

Minutes for CSC2058 Project 45 Week commencing 29/11/2021 Date of this minute - 30/10/2021

The following team members were present on Teams when minutes were discussed:

Name (printed/typed)	Signature (agreed bitmap or initials)
Adam Logan	A.L
XXXXX	X.X
XXXXX	X.X
XXXXX	X.X

Actions Taken During Meeting

During the meeting, the team had finished the last sequence diagrams, fixed board layout, started adding to the report and planned the commentary video.

Task Reporting

Adam Logan:
Completed following Use Case realisations:

- playerTurn
- buyProduct

Also fixed:

- Class Diagram
- Use Case diagram

XXXXX:
Completed the following:

- Board draft layout
- Table of prices
- Dice code class

XXXXX:

- N/A

XXXXX:
Completed the following:

- Player code class

XXXXX:

- Completed explanation of game

Actions Planned

Group:

- To complete the peer assessments
- To finish interim video
- Finish report