CSC7053 Software Engineering

Artemis Lite Project

# Group 19

**Jordan Davis [JD]**

**David Finlay [DF]**

**Joseph Mawhinney [JM]**

**Andrew Pickard [AP]**

# Project Purpose and Aim

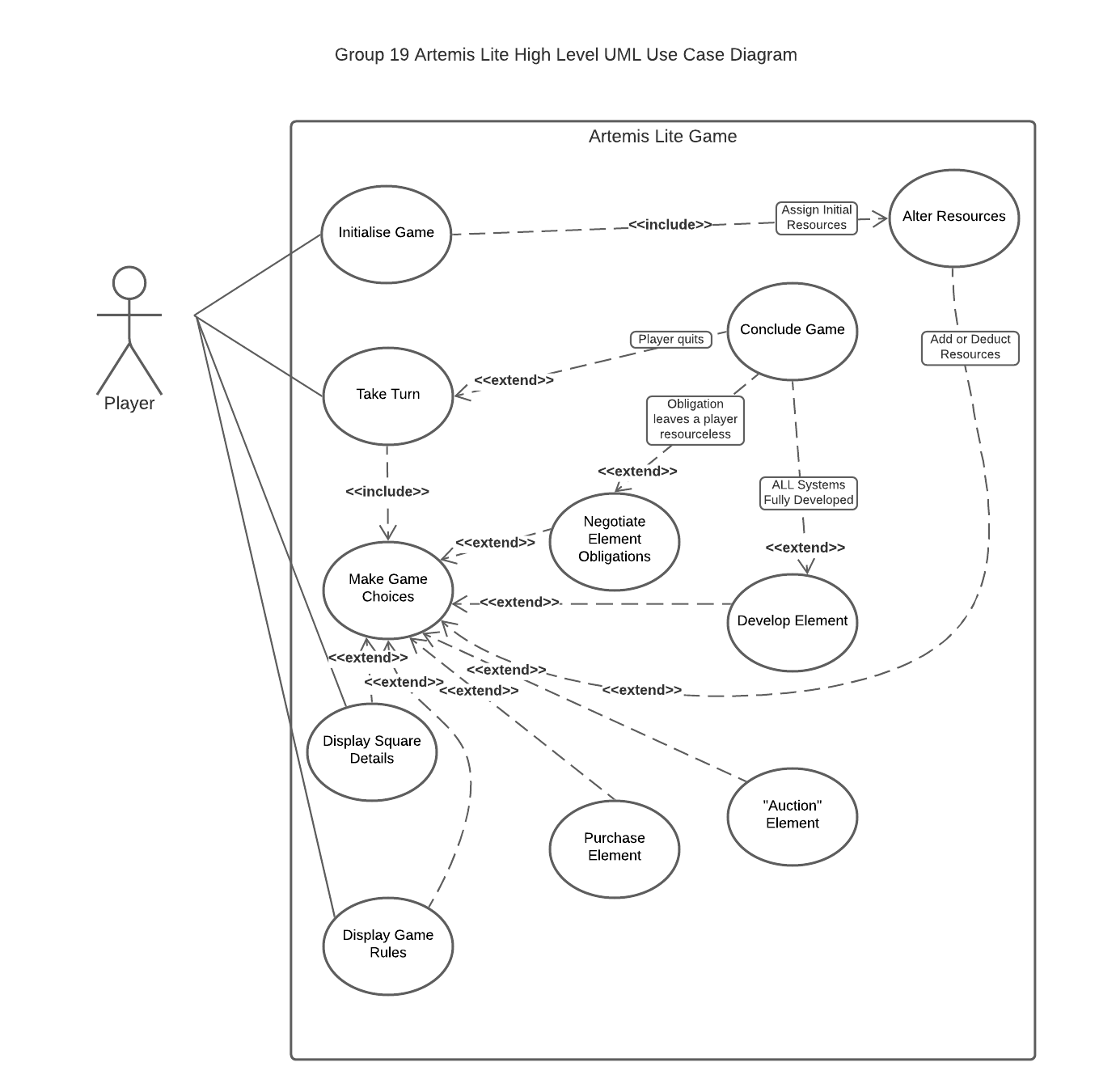
The purpose of this project was to demonstrate understanding and use of best practices in object-oriented software engineering within a software engineering team. The team was tasked with deploying a use-case driven methodology with the aim of developing a software board game. That board game was to be played within the console of an Integrated Development Environment such as Eclipse. The game was based upon a manned space mission in development (current of 2021) at NASA, named Artemis. Simplified in comparison to the real-world mission, the software board game was named Artemis Lite.

This report outlines the key stages within this development process.

# Requirements Analysis

From the initial, supplied requirements, a procedural requirements document was created. This document was iterated over during the initial team meetings [Appendix 2] to determine the main sets of sequences of actions that would eventually inform the following final UML Use Case Diagram.

## UML Use Case Diagram



*Figure 1. UML Use Case Diagram*

It became apparent from both the initial requirements document and the Use Case Diagram [Figure 1] that not only was there a potential for several alternative flows, but also that effort would not be evenly distributed through the use cases, and some would be more heavily weighted. This is reflected in the use case descriptions.

## Use Case Descriptions

|  |  |
| --- | --- |
| **Use Case 1** | **Initialise Game** |
| **Summary** | Players added and their names are stored.  Turn order decided by dice roll.  Game begins. |
| **Actors** | *All players (2 min, 4 max)* |
| **Triggers** | A minimum of 2 players begin a game. |
| **Pre Conditions** | Between 2 and 4 players. |
| **Post Conditions** | * Players names are displayed. * Player order is generated. |
| **Flow** | 1. Players are prompted to add player. 2. Players are prompted to enter their name. 3. 1 and 2 repeated until all players are added. 4. A die is rolled by each player to determine turn order. 5. Game begins. |
| **Alternative Flow** | * At 1, players are prompted if number of players is invalid. * At 2, players may choose to modify their name. |
| **Extension Points** | None |
| **Inclusions Points** | Alter resources. |

|  |  |
| --- | --- |
| **Use Case 2** | **Take Turn** |
| **Summary** | Processing the active player as they initiate and conclude a game move. |
| **Actors** | Active Player |
| **Triggers** | Active player prompted to roll dice. |
| **Pre Conditions** | Use Case 1: Initialise Game  Valid number of players (2-4)  Player order predetermined on dice roll. |
| **Post Conditions** | Active player has made a game choice and the game has progressed by one move.  Active player chooses to End Turn or End Game. |
| **Flow** | 1. Active player prompted to roll dice. 2. Display dice total, active player’s new square position post-move and element details for this current square. 3. Active player is prompted to continue. 4. On continuing, game state, and available game choices are displayed. This includes options to conclude the turn and to End Game. 5. Wait for active player’s choice. 6. Game flow proceeds depending on active player’s choice. 7. Turn concludes on selection either End Turn or End Game options. |
| **Alternative Flow** | 1. Invalid option is chosen for game choice. 2. Active player informed of invalid choice and redirected back to game choice menu. |
| **Extension Points** | Conclude Game |
| **Inclusions Points** | Use Case 3: Make Game Choices |

|  |  |
| --- | --- |
| **Use Case 3** | **Make Game Choices** |
| **Summary** | Player lands on a square and is prompted to perform an action from a menu. |
| **Actors** | Active player |
| **Triggers** | Player moves onto a square. |
| **Pre Conditions** | Player lands on a square after moving. |
| **Post Conditions** | * Player carries out chosen action and chooses to end turn. |
| **Flow** | 1. Player lands on a square after carrying out their move. 2. Player is prompted to choose an option from a menu. 3. When player chooses to end turn, play is passed to the next player. |
| **Alternative Flow** | None (Note: check this) |
| **Extension Points** | * Negotiate element obligations. * Develop element. * Auction element. * Purchase element * Display game rules * Display square details. |
| **Inclusions Points** | * Take turn |

|  |  |
| --- | --- |
| **Use Case 4** | **Display Square Details** |
| **Summary** | Display to console all pertinent details of the current game square. |
| **Actors** | Active Player |
| **Triggers** | Active player rolls dice and moves to new square.  Active player passes the Resources square.  Active player chooses to view their elements. |
| **Pre Conditions** | Active player moves to new square.  Active player chooses to view elements. |
| **Post Conditions** | Element details for the square are displayed to the console. |
| **Flow** | 1. Active player rolls dice and moves to square -OR- active player chooses the option to view elements or to get current square details. 2. If the square contains an element that is owned by another player, display the owner, along with other development details. 3. If the square contains an un-owned element, display the purchase cost along with other development details.   -OR-   1. Active player passes the resource square. 2. Display resources acquired from the square. |
| **Alternative Flow** | None |
| **Extension Points** | None |
| **Inclusions Points** | None |

|  |  |
| --- | --- |
| **Use Case 5** | **Alter Resources** |
| **Summary** | Players’ resources are altered as a result of game actions. |
| **Actors** | All players |
| **Triggers** | Player carries out a transaction or passes the start square. |
| **Pre Conditions** | Player carries out an action that results in their resource total changing. |
| **Post Conditions** | Players’ resources altered. |
| **Flow** | 1. Player carries out action. 2. Active player’s resources are altered accordingly. 3. If another player is involved (e.g. paying rent) their resources are also altered. |
| **Alternative Flow** | * At 1, player will be prompted if action will lead to bankruptcy. * At 2 or 3, if player is bankrupt game ends. |
| **Extension Points** | None |
| **Inclusions Points** | None |

|  |  |
| --- | --- |
| **Use Case 6** | **Auction Element** |
| **Summary** | The active player offers an element that they either do not want or cannot afford to the other players to purchase. |
| **Actors** | *All players* |
| **Triggers** | * Active player lands on an unowned element, that they do not wish to purchase. * Active player lands on an unowned element, that they cannot afford to purchase. |
| **Pre Conditions** | * The active player has landed on an element that is not owned. * The active player has indicated that they do not want / cannot afford the element. * At least 1 other player can afford the element in question. |
| **Post Conditions** | * The active player continues their turn. * The element MAY be owned by a new player if the auction was successful. * The auction winner’s resources have decreased by the purchase cost of the element. |
| **Flow** | 1. Each player (excluding the active player) that can afford the element is offered (in turn order) to purchase the element and shown their current resources and cost of the element. 2. Eligible players are prompted for a Yes / No response. 3. Players who indicated (Yes) that they wish to purchase the element roll the dice to determine who gets to purchase the element. 4. The highest rolling player is now set as the owner of the element in question. 5. The highest rolling player has the price of the element deducted from their balance of resources. 6. The players are notified of the changes to element ownership and resources. 7. The active player is told that their turn now continues following the conclusion of the auction. |
| **Alternative Flow** | * At flow point 1:- if no players are eligible (ie. cannot afford the element), it is stated that there are no eligible players, the property remains unowned and the active player continues their turn. * At flow point 3:- if only one player indicated they wish to purchase the element, there is no roll. * At flow point 4/5:- if there is no highest roll (ie. a draw) all players who rolled lower are removed from contention, and the players who rolled the draw are prompted to roll again, until a single highest roller remains. |
| **Extension Points** | None |
| **Inclusions Points** | None |

|  |  |
| --- | --- |
| **Use Case 7** | **Purchase Element** |
| **Summary** | The active player purchases an unowned element they landed on. |
| **Actors** | Active player |
| **Triggers** | * Active player lands on an unowned element |
| **Pre Conditions** | * The element is not currently owned. * The active player has sufficient resources to purchase the element. |
| **Post Conditions** | * The active player becomes the owner of the element. * The active players resources have decreased by the purchase cost of the element. |
| **Flow** | 1. The purchase cost of the element is displayed. [Extension A] 2. The active player is told their current balance of resources and asked if they wish to purchase the element. 3. The active player is prompted for a Yes / No response. [Extension A] 4. Following a Yes response, the active player has their resources reduced by an amount equal to the purchase cost of the element. 5. Ownership of the element is now transferred to the active player. 6. Players are notified of the change to the active players resources and the ownership status of the element in question. 7. The active player resumes their turn. |
| **Alternative Flow** | * At flow point 1:- if the active player cannot afford the element, it will instead go to auction. [Extension A] * At flow point 3:- if the active player indicates that they do not wish to purchase the element, it will instead go to auction. [Extension A] |
| **Extension Points** | 1. Auction Element – Use Case 6 |
| **Inclusions Points** | None |

|  |  |
| --- | --- |
| **Use Case 8** | **Manage Element Obligations** |
| **Summary** | The active player pays the element owner for landing on their element. |
| **Actors** | Active player & Element owner (player) |
| **Triggers** | * The active player lands on an element that is already owned by another player. |
| **Pre Conditions** | * Active player has landed on an element. * The element is owned by another player. |
| **Post Conditions** | * The active player MAY have the cost of rent deducted from their balance of resources. * The element owner MAY have the cost of rent added to their balance of resources. * Conclude Game – Use Case 10 |
| **Flow** | 1. The active player is told that they have landed on an element owned by another player, and the name of that player. 2. The active player is told the cost of rent on this element and shown their current balance of resources. 3. The element owner is asked if they would like to charge the active player rent. 4. The element owner is prompted for a Yes / No response. 5. Following a Yes response, the active player has their resources reduced by an amount equal to that of the rent on the element in question. 6. The element owner has their resources increased by the same amount. 7. The players are informed of the changes to their balance of resources. 8. The active player is told that their turn now continues following fulfilment of their obligations on this element. |
| **Alternative Flow** | * At flow point 3:- if the active player does not have enough resources to pay the full rent, the element owner is notified that if they choose to charge rent they will make the active player bankrupt and it will end the game.  1. Following a Yes response, the rent value is transferred from the active player to the element owner. 2. The active player is now bankrupt, and the game ends. 3. The game end sequence occurs. [Extension A]  * At flow point 5:- if the element owner decides not to charge the active player rent, then no transfer of resources occurs and the active player may resume their turn. |
| **Extension Points** | 1. Conclude Game – Use Case 10 |
| **Inclusions Points** | None |

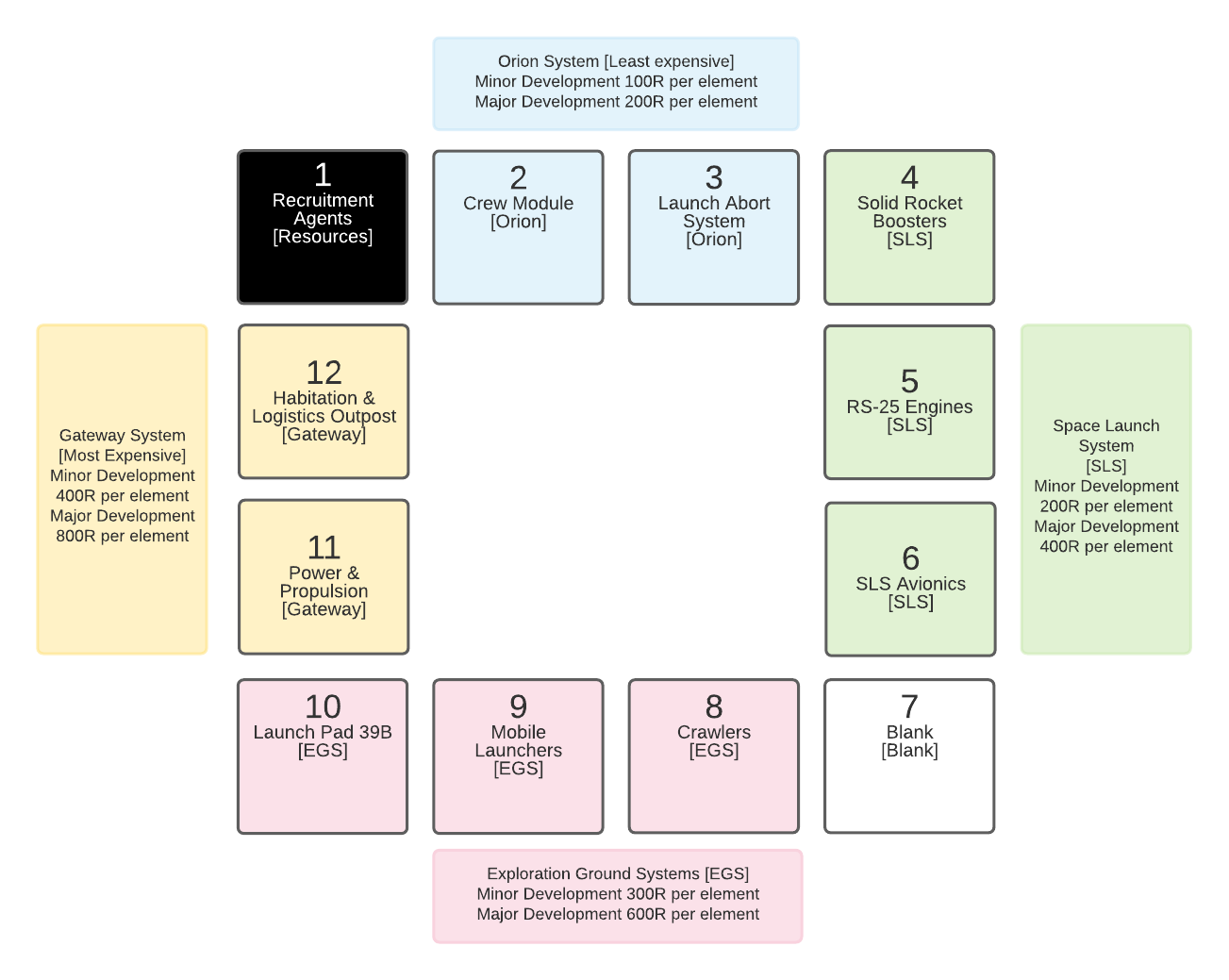
|  |  |
| --- | --- |
| Use Case 9 | **Develop Element** |
| **Summary** | The active player can increase both the minor and major development level for an element once they are in control of all other elements of the same system. |
| **Actors** | Active Player |
| **Triggers** | Active player selects develop element. |
| **Pre Conditions** | Player must own all elements in a system prior to starting a development.  Player must reach max minor development level before they can increase major development level. |
| **Post Conditions** | Development level of element increased.  Rent value of property increased. |
| **Flow** |  |
| **Alternative Flow** |  |
| **Extension Points** |  |
| **Inclusions Points** |  |

|  |  |
| --- | --- |
| Use Case 10 | **Conclude Game** |
| **Summary** | This should be split to Game win, goes bankrupt and Quit game? |
| **Actors** |  |
| **Triggers** |  |
| **Pre Conditions** |  |
| **Post Conditions** |  |
| **Flow** |  |
| **Alternative Flow** |  |
| **Extension Points** |  |
| **Inclusions Points** |  |

|  |  |
| --- | --- |
| Use Case 11 | **Display Game Rules** |
| **Summary** |  |
| **Actors** |  |
| **Triggers** |  |
| **Pre Conditions** |  |
| **Post Conditions** |  |
| **Flow** |  |
| **Alternative Flow** |  |
| **Extension Points** |  |
| **Inclusions Points** |  |

In concert, the initial requirements document, Use Case Diagram and Use Case Descriptions allowed for the creation of a virtual board. While this board is not implemented in any graphical form within the project, it was produced as a visual cue for both the project build and test stages.

## Virtual Board



*Figure 2. Virtual Board*

# Realisation

comprising a number of UML Sequence Diagrams with a brief written commentary. Your sequence diagrams should show how your software components make method calls to each other, and interact with the players, in order to realise the behaviour of a selection of important use cases described in the previous section: clearly identify the use cases for which you have chosen to show realisations. Remember: any class that you show at the top of lifeline in a sequence diagram should also appear somewhere in your class diagram – see below. (Realisation: 20 marks)

# Design

comprising a UML Class Diagram, that describes the system components. The class diagram will correspond closely to the coded implementation of the game; it should show classes and methods that support the sequences of method calls described in the previous section. Again, provide a brief written commentary on your design, pointing out any instances of good design where you have considered questions of maintainability and extensibility. (Design: 20 marks)

# Test

comprising a sample from your Test Plan with a brief description of your approach. (Your Test Plan may take the form of an Acceptance Plan, like the one shown in Chapter 12 of your lecture notes. Your full test plan and test strategy should be documented in Appendix I – see below.) (Test: 5 marks)

# Appendix 1 Full Test Plan for the implemented system

along with your acceptance tests you may provide evidence of unit testing (for example, screen dumps showing that unit tests ran successfully).

# Appendix 2 Weekly Team Minutes

# Minutes for Group: 19 Week commencing: 01/02 Date of this minute: 02/02

# Meeting No: 01 Meeting duration: 45 min

The following team members were present

|  |
| --- |
| **NAME** |
| **DAVID FINLAY** |
| **JORDAN DAVIS** |
| **JOSEPH MAWHINNEY** |
| **ANDREW PICKARD** |

**Task Reporting**

**Jordan:**

* Produced a draft UML use case diagram

**During session:**

Explored our understanding of how the game should work, to make sure we’re all ‘on the same page’ with what the finished product should look / feel like.

We discussed as a group and added to Jordan's preliminary UML, and split the diagram into 4 individual flow / use cases that each group member could create use case descriptions for by the next meeting.

**Actions Planned**

**Everyone:**

* Create use case descriptions from the assigned elements within the use case diagram.

# Minutes for Group: 19 Week commencing: 08/02 Date of this minute: 09/02

# Meeting No: 02 Meeting duration: 1hr 30min

The following team members were present

|  |
| --- |
| **NAME** |
| **DAVID FINLAY** |
| **JORDAN DAVIS** |
| **JOSEPH MAWHINNEY** |
| **ANDREW PICKARD** |

**Task Reporting**

**David:**

* Use case description written for:  
  - Display game state  
  - Move player  
  - Display new square

**Jordan:**

* Use case description written for:  
  - Auction square  
  - Buy square  
  - Display square details  
  - Pay debt / cancel debt

**Joseph:**

* Use case description written for:  
  - Launch game  
  - Register Player  
  - Verify valid name

**Andrew:**

* Use case description written for:  
  - Take turn  
  - Display current square stats  
  - Display options

**During session:**

We set up some shared resources, so that we could all work on the same documents remotely, including; shared Lucid Chart sheets to make Use Case diagrams and UML Class Diagrams, when we get on to them; shared word files to come up with a more succinct requirements (derived from the customer core requirements provided).

Discussed some core game features in more detail, and tried to come to a consensus on how these features would be implemented within our system (eg. determine player turn order, resolve an auction where 2-or-more players want the property etc.).

**Actions Planned**

**Everyone:**

* Use the shared resources to put forward ideas for what certain parts of the game will be named (eg. the elements) .
* Create a normal flow of how the game would play out from start to finish.
* Create initial draft of UML Class Diagrams.

# Minutes for Group: 19 Week commencing: 15/02 Date of this minute: 16/02

# Meeting No: 03 Meeting duration: 1hr 45 min

The following team members were present

|  |
| --- |
| **NAME** |
| **DAVID FINLAY** |
| **JORDAN DAVIS** |
| **JOSEPH MAWHINNEY** |
| **ANDREW PICKARD** |

**Task Reporting**

**Everyone:**

* Everyone created UML class diagrams to compare and collate in to a single diagram during the meeting.

**David:**

* Created a more up-to-date use case diagram, incorporating the features we had discussed in previous meetings.

**During session:**

Discussed the class diagrams, worked together in a shared Lucid Chart sheet while on voice chat to put together an initial class diagram that we could work off to begin coding the Artemis Lite application.

**Actions Planned**

**Everyone:**

* Set up / familiarise with GitLab
* Start some initial coding, using the UML class diagrams as a basis.

NOTE: Plan of action was purposefully sparse for the near future, as we had an exam for programming module + hand in for WebDev.

# Minutes for Group: 19 Week commencing: 01/03 Date of this minute: 02/03

# Meeting No: 04 Meeting duration: 15 min

The following team members were present

|  |
| --- |
| **NAME** |
| **DAVID FINLAY** |
| **JORDAN DAVIS** |
| **JOSEPH MAWHINNEY** |
| **ANDREW PICKARD** |

**Task Reporting**

**Everyone:**

* Ongoing work on coding the application.

**During session:**

We discussed the coding plan going forward, and went over some of the functions we had implemented so far.

**Actions Planned**

**Everyone:**

* Continue to work on coding.

# Minutes for Group: 19 Week commencing: 08/03 Date of this minute: 09/03

# Meeting No: 05 Meeting duration: 1hr 25 min

The following team members were present

|  |
| --- |
| **NAME** |
| **DAVID FINLAY** |
| **JORDAN DAVIS** |
| **JOSEPH MAWHINNEY** |
| **ANDREW PICKARD** |

**Task Reporting**

**Everyone:**

* Continued work on coding.

**Jordan:**

* Put together an initial Sprint log.

**During session:**

Having decided between this and the previous meeting that we would have a 2 week sprint, we spent this session outlining the user stories, and writing out story descriptions to populate the Product Backlog.

We then decided on items to move from the Product Backlog on to the Sprint Backlog to work on over the coming 2 weeks.

**Actions Planned**

**Everyone:**

* Work from the Sprint log to complete the items on the Sprint Backlog – marking items that you are currently developing with your initials.

# Minutes for Group: 19 Week commencing: 08/03 Date of this minute: 11/03

# Meeting No: 06 Meeting duration: 15 min

The following team members were present

|  |
| --- |
| **NAME** |
| **DAVID FINLAY** |
| **JORDAN DAVIS** |
| **JOSEPH MAWHINNEY** |
| **ANDREW PICKARD** |

**Task Reporting**

**Everyone:**

* Continued work on coding.
* Lots of communication via Teams chat while working on the coding, including short voice chat to talk through issues that arise.

**During session:**

Short ‘Stand up’ like meeting to go over some of the features each of us plan to work on.

**Actions Planned**

**Everyone:**

* Continue work on coding, and keep in close contact to help with code integration / mergers.

# Minutes for Group: 19 Week commencing: 15/03 Date of this minute: 16/03

# Meeting No: 07 Meeting duration: 50 min

The following team members were present

|  |
| --- |
| **NAME** |
| **DAVID FINLAY** |
| **JORDAN DAVIS** |
| **JOSEPH MAWHINNEY** |
| **ANDREW PICKARD** |

**Task Reporting**

**Everyone:**

* Continued work on coding.

**During session:**

We realised that we had more time than expected to work on the coding, and consequently progressed through the items on the Sprint Backlog at a faster rate than anticipated.

We subsequently decided to move the remaining items from the Product Backlog onto the Sprint Backlog, after having talked through the corresponding user stories to understand them better.

**Actions Planned**

**Everyone:**

* Continue coding.

# Appendix 3 Use of GitLab

place evidence of the team’s use of GitLab (e.g. a representative sample of screen dumps from GitLab’s Activity record) in Appendix III.

# Appendix 4 Other Evidence of Day-to-Day Project Management

(e.g. brief descriptions of backlog items and tasks, with estimates; screen dumps of sprint backlog tables and team/teammember burndown charts, project timelines, etc.) should be placed in Appendix IV (Process: 5 marks)

The main body of the report (excluding appendices and the peer assessment) should not exceed 20 pages. Individual team members should place their initials (e.g. [A.B.; C.D.]) next to the sections for which they were the principal authors.