

**Designed by Freepik**

**Witch’s Dystopia**

**Project Phase 3**

**COSC419l Game Engineering**

**Student Name Student Number**

Ryan McQueen 54198149

Ryan Kramer 24604143

Kevin Van Kessel 37845138

# 

# 1. Introduction

In this report, there are a number of items introduced, and discussed. Initially, the concept of Witch’s Dystopia is explored, detailing the key characteristics of the game, and the genre it follows. Secondly, the engineering methodology is outlined to indicate the methods used in the development of the game. Following the methodology, the engineering artifacts are explored, paying exceptional detail to user stories around functional and nonfunctional tasks related to the development of Witch’s Dystopia and the acceptance criteria associated with these user stories, a few architectural diagrams such as the component and class diagrams. Included in the engineering artifacts section, the outline of how WItch’s Dystopia was tested is demonstrated. Finally, a brief biography about the team is mentioned, followed by a sample of Witch’s Dystopia.

# 2. Game Concept

The game proposed is a lightweight mix of survival and roguelike genres. The game is about gathering supplies to survive and craft better equipment so that the player can travel to more dangerous areas that have better or more supplies as they work to gather specific items to beat the game. The game alternates between two areas, an overworld where the player manages one or more bases, and travels between areas. The other portion of the game is a 3D/isometric level where the user controls the character and searches the environment of the level for supplies while dealing with opposition from enemies and the environment.

* Player has to explore areas to get supplies to survive and build better gear
* Player manages one or more bases, which have automated workers for gathering basic supplies and materials the player dictates
* Player moves across world on 2D overworld map, moving between areas that are linked to other areas, where all but the starting areas are locked until the area before it is explored
* Player explores area’s in 3D or isometric view and go around level looking for loot and supplies while dealing with enemies and environmental dangers
* Player has to maintain health, food,water and sleep during the game, which can be adversely affected by the environment and enemies, which in turn can be dealt with by the player if they have the supplies (ex: medicine cures poison, which caused the players food meter to drop rapidly)
* Player uses supplies and materials to craft new and upgraded gear, which is used to explore new areas, as areas farther from the center of the map are more dangerous
* Player wins when they gather all the key parts scattered on the outer edge of the map, which is the most dangerous areas in the game

# 

# 3. Game Engineering Methodology

**3.1) Requirements Engineering**

|  |  |  |
| --- | --- | --- |
| Methodology Element | Description | Citation |
| Standards Adopted | UML | [14] |
| Techniques | User Stories | [14] |
| Notation | UML | [14] |
| Languages | --- |  |
| Templates | --- |  |
| Tools | StarUML | [13] |

**3.2) Design**

|  |  |  |
| --- | --- | --- |
| Methodology Element | Description | Citation |
| Standards Adopted | UML | [14] |
| Techniques | Component Design:   1. Class Diagrams 2. Sequence Diagrams   Architecture:   1. Package/Component Diagrams 2. Activity Diagrams | [16]  [16] |
| Notation | UML | [14] |
| Languages | -- |  |
| Templates | -- |  |
| Tools | StarUML | [13] |

**3.3) Implementation**

|  |  |  |
| --- | --- | --- |
| Methodology Element | Description | Citation |
| Standards Adopted | C# Language Specification, version 5  Coding Standard Specification, version | [1]  [2] |
| Techniques | Version control | [7] |
| Notation | Google Java Style Guide | [15] |
| Languages | C#, version 5.0 | [8] |
| Templates | --- | --- |
| Tools | MonoDevelop IDE, version 6.1  GIT version control system, version 2.11.1  GitHub version control repository  VirtualBox virtual machine, version 5.1  Unity version 5.5.1f1  Unity Asset Store  Unity Collaborate | [3]  [6]  [7]  [5]  [10]  [11]  [12] |

**3.4) Testing**

|  |  |  |
| --- | --- | --- |
| Methodology Element | Description | Citation |
| Standards Adopted | Playtesting  User Testing | [14] |
| Techniques | Acceptance Testing | [17] |
| Notation | English | -- |
| Languages | --- | -- |
| Templates | Self Created | NA |
| Tools | Unity version 5.5.1f1 | [10] |

# 4. Engineering Artifacts

### 4.1) Requirements

**Non Functional:**

|  |  |  |  |
| --- | --- | --- | --- |
| User Story | Category | Acceptance Criteria | Priority |
| As a user, I would like to have my attributes update when a new item is equipped | Attributes/Gameplay | The attributes (HP, defence, attack) on the equipment screen must update instantly based on the item’s being equipped/unequipped | Low |
| As a user, I would like there to be safety measures in place to ensure I cannot fall off the map | Gameplay | Boundaries must be established around the perimeter of the map which prevent the user from walking through them, and must be tall enough to prevent the player from jumping over them | High |
| As a user, I would like to be unable to walk through trees in the environment | Gameplay | Trees must have boundaries surrounding their shape which prevent movement through them, and stop the user if they try to walk through them | High |
| As a user, I would like to have crafted items automatically appear in my inventory | Crafting | When an item has been crafted, the user will not be required to drag it into their inventory, it should automatically appear in a free spot within their inventory | Medium |
| As a user, I would like to have the delay between pressing the craft but be no more than 1 second | Crafting | The crafting button must perform the crafting operation within | Medium |
| As a user, I would like to have the crafting button disable if I do not have sufficient materials | Crafting | The crafting button will become disabled, and red when a user does not have the sufficient materials | High |
| As a user, I would like to have visual indications as to what the item looks like in the crafting window | Crafting | The crafting buttons must contain an image which is relevant to the item which they have clicked on to craft | High |
| As a user, I would like to be able to see where I am currently placed on the map in regards to my current location | Map | The map must indicate the user’s accurate current position in the world of the game | Low |
| As a user, I would like to have an indication as to what effects an item has on me | Consumables | When a user is to hover over an item, display information about what the item does, and how it affects the character | Medium |
| As a user, I would like to have the worker store an item every 10 minutes into my storage inventory | Workers | When a user is interacting with the worker NPC, it must deposit an item every 10 minutes from what they have chosen | Low |

**Functional:**

|  |  |  |  |
| --- | --- | --- | --- |
| User Story | Category | Acceptance Criteria | Priority |
| As a user, I would like to be able to control my character, and move forward, backward, left, and right | Movement | When pressing the W key, the character must move in a forward direction based on where they are  When pressing the S key, the character must move in a backward direction based on where they are  When pressing the A key, the character must move in a leftward direction based on where they are  When pressing the D key, the character must move in a rightward direction based on where they are | High |
| As a user, I would like to be able to rotate my character 360 degrees, or more in order to turn around | Movement | When the player presses Q or E, the character model rotates in its current location until the keys are released, and does not affect the character’s current motion direction | High |
| As a user, I would like to be able to view my character from a top-down view with a minor angle in order to see my character | View/Camera | The camera facing the character must show the character from a higher approach than the character  Must be on 25 degree angle to not limit their vision, but keep the camera angled | High |
| As a user, I would like to be able to obtain items to improve my character | Storage/Inventory | There must exist objects in the world in which a character can interact with in order to receive an item that fits within their equipment slots (Weapon, Helmet, Tunic, Pants, Gloves, Wrist-guards, and Boots) | High |
| As a user, I would like to be able to store items to use at any time | Storage/Inventory | There must be objects such as crates, or chests that a character can interact with, and place items from their inventory into | High |
| As a user, I would like to be able to view stored items | Storage/Inventory | The objects storing a character’s items must visually show which items of the character’s that they currently contain | Medium |
| As a user, I would like there to be objects scattered across the environment to interact with | Interaction | Crates that the character can open/close  Enemies that the character can attack, or interact with  Trees that the user can collide with | Low |
| As a user, I would like to be able to interact with objects in order to obtain new items | Interaction | Crates/chests that the character can open/close and are unique (Ex. golden swords) | High |
| As a user, I would like to be able to visually see what items are present in an object | GUI/Interaction | Crates/chests that when opened, display the item icon for the user to visually see | High |
| As a user, I would like to be able to equip items so that I can use them for combat | Equipment | Items that fit within the character’s equipment slots (Weapons, and armours) | Medium |
| As a user, I would like to be able to see what items I have equipped | Equipment | A window that the user can open/close by pressing C  Slots that hold helmets, tunics, gloves, wrist-guards, pants, and boots | Medium |
| As a user, I would like to have a way to view what my character attribute points are | Attributes | On the equipment window, must have a heading called “Attributes” and have HP, Defence, Attack, and Food levels. | Medium |
| As a user, I would like to be able to view the health of my character | Attributes | Health points (HP) that the character currently has from their food level |  |
| As a user, I would like to be able to view the attack points of my character | Attributes | Attack points that the character currently has from their weapon(s); this is a numeric value | High |
| As a user, I would like to be able to view the defence rating of my character | Attributes | Defence points that the character currently has from their armour; this is a numeric value | High |
| As a user, I would like to be able to view the food level of my character | Attributes | Food level that the character currently has from their consumption of food items. It is on a numeric scale from 0-100 (starting at 100) This must decrease when prolonged durations of hunger persist  24 hours lowers it by 1 per additional day until 48 hours is reached  48 hours lowers it by 3 per additional day until 96 hours is reached  96 hours lowers it by 5 per additional day until 192 hours is reached  192 hours results in the HP of the player being set to 0, and dying | Medium |
| As a user I would like to be able to combine items and create new ones from them | Crafting | A user must be able to drag items into the crafting inventory, and place them in the free slots  The new items must not be the same as their original inputted items  Crafted items must deduct required item quantities upon the new items creation | Medium |
| As a user, I would like to have a map showing the game world, and where I can travel to | Map | The map must display the layout of the game world  The map must be informative as to where the other areas of the world are in terms of the location of where the user currently is | Low |
| As a user, I would like to have workers to interact with | Workers | The workers can combine | Medium |
| As a user, I would like to have workers perform item storage for me | Workers | The worker must perform a task of storing an item into the user’s storage inventory | Low |
| As a user, I would like to be able to travel between areas | Travel | Need to transport the user from one zone to the next based on their input  Update the user’s current location to the new area | Medium |

### 

### 

### 

### 

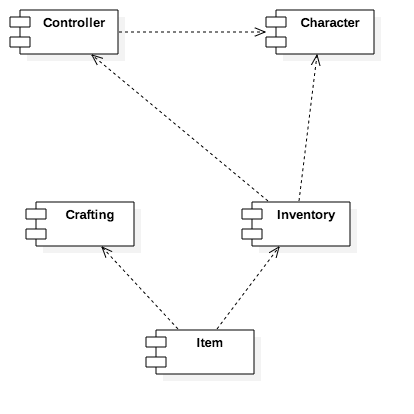
### 

### 4.2) Design

**Software Architecture:**

The architecture pattern, or style, being used is object-oriented programming, and does not follow any other typical design patterns that would be used in this type of architecture.

**Figure 1.1**: Component Diagram of Witch’s Dystopia



**Controller:**

Handles interactions between the Character and the environment of the game. This entails: Item interactions, Character menus (inventory, attributes).

Files Included:

* MapBorder.cs
* initScene.cs

**Character:**

Consists of the Enemy and Player object. This also includes the interactions based on a player such as the attribute bars, the controls of the player, and the camera view of the player.

Files Included:

* Enemy.cs
* Player.cs
* statsBars.cs
* PlayerControl.cs
* LookAtCam.cs
* CharacterAttackHitbox.cs

**Crafting:**

Incorporates individual items, and combinations of items to create different types of items.

Files Included:

* craftingBtn.cs

**Inventory:**

Holds the items of a Character, and the items of a Crate

Files Included:

* BaseController.cs
* timeTracker.cs
* craftingBtn.cs
* Create.cs
* Slot.cs

**Item:**

The types of objects which are storable into an inventory slot. These include weapons, armors, and consumable items.

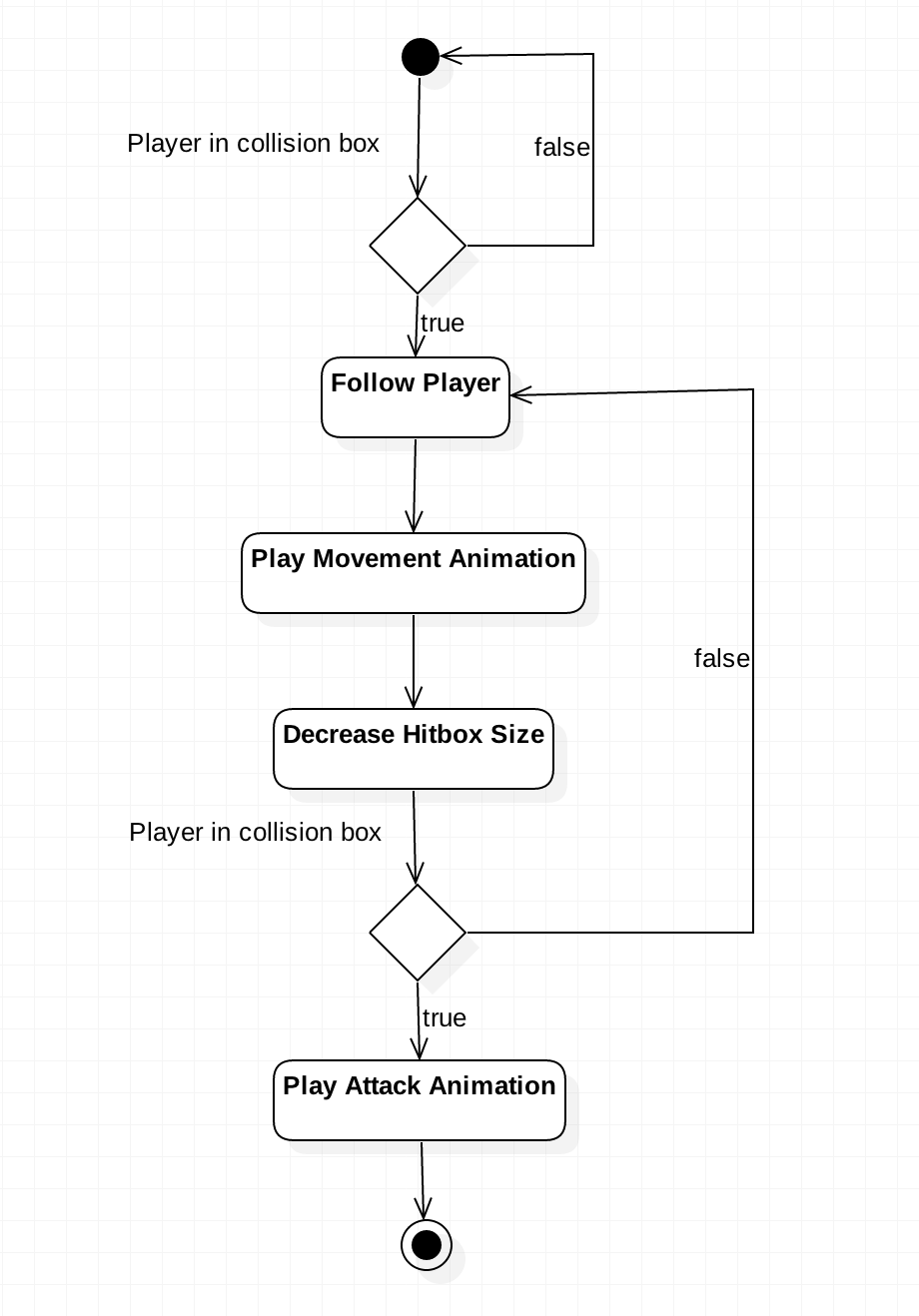
Files Included:

* ConsumeableItem.cs
* MovingItem.cs
* Armor.cs
* Weapon.cs

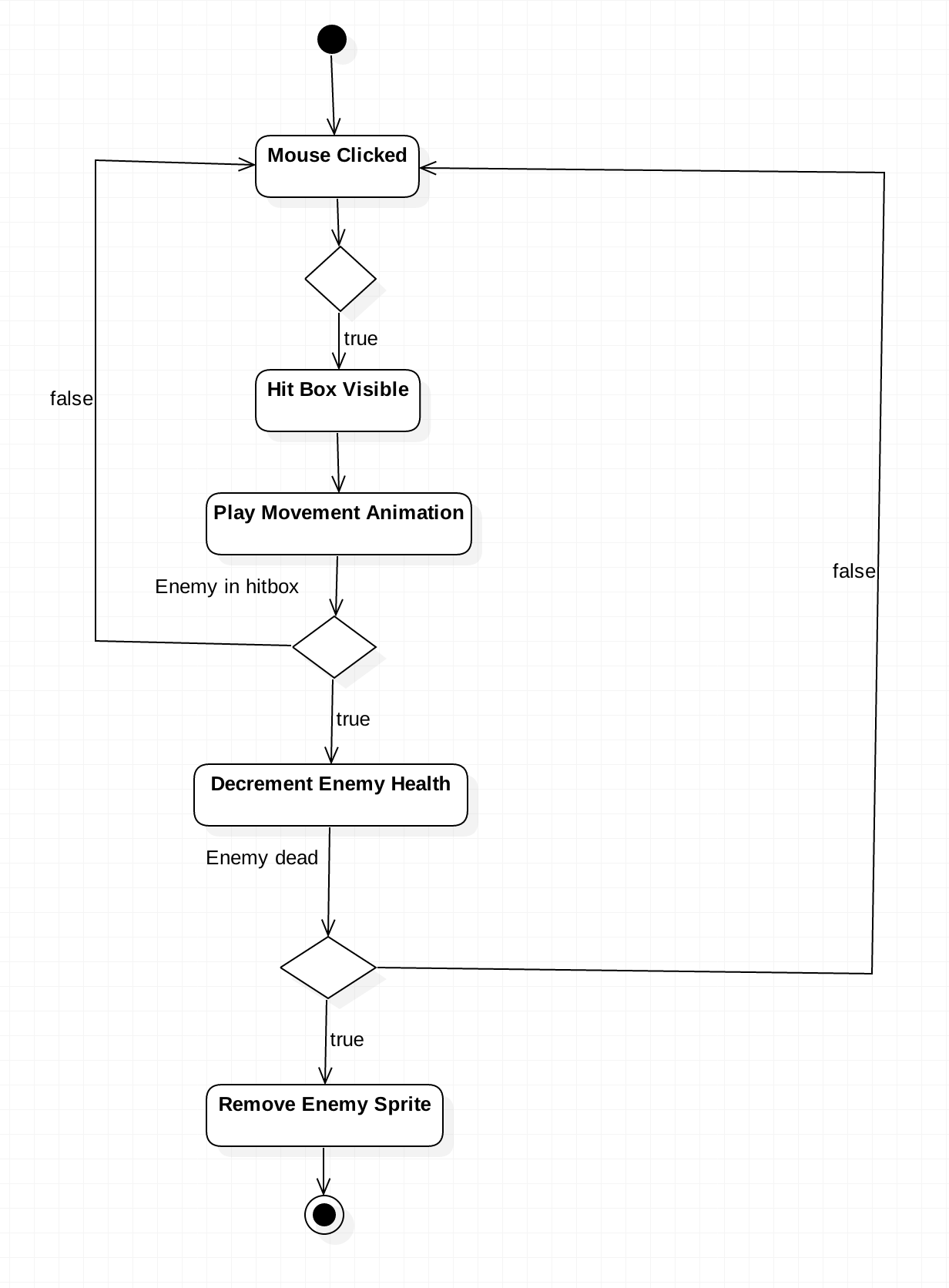
**Figure 1.2:** Class Diagram of Witch’s Dystopia

**\*\*Attached as a separate document labeled “class\_diagram.png” to improve readability\*\***

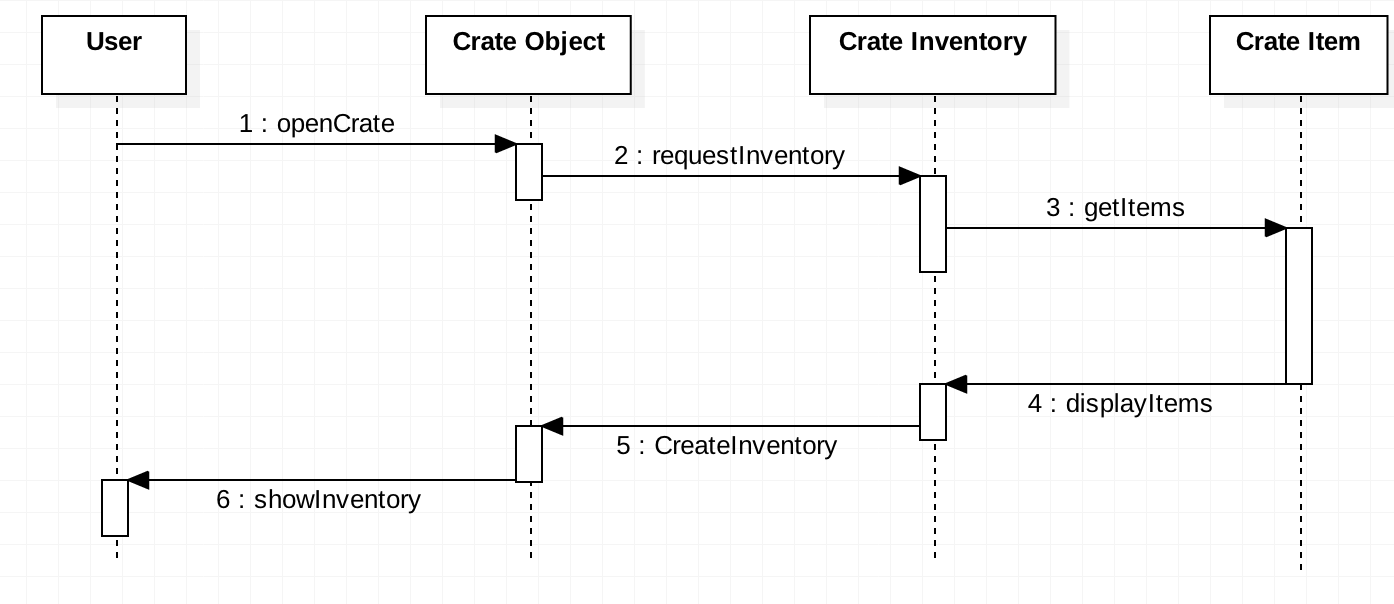
**Figure 1.3:** Goblin Attack Activity Diagram



**Figure 1.4:** Player Attack Activity Diagram



**Figure 1.5:** Crate Looting Sequence Diagram



**\*\*Our GDD can be found in a separate document titled “Witch’s Dystopia GDD.pdf”\*\***

### 4.3) Code Repository

Note: We primarily use Unity Collaborate**[12]** now as it is exceptionally easy to use as we do not need to worry about the hassles of developing on separate branches and the merge conflicts associated with this. We still post our project on GitHub as a backup however.

**Repository:** <https://github.com/rcMcQueen/COSC419l>

You can find all of the code in: Assets/Scripts/

### 4.4) Testing

**Functional Test Cases:**

* As shown in our 4.1), we have our functional and nonfunctional requirements clearly indicated, with the appropriate acceptance criteria which is what we used to gauge whether or not we had satisfied the user story, and in turn, completed the task.
* Although no list detailing the breakdown of each task even further during the testing phase, all functional requirements were deemed complete as they were validated by their respective acceptance criteria

**Playtesting Material:**

* This can be found in the document labelled “Playtest\_feedback.pdf”
* Although the non-functional requirements are not explicitly stated within the Playtest script, they expand further into the main categories as listed in the non-functional requirements table in section 4.1.

We only had one person to playtest the game which was not ideal; however, due to other commitments with school this is all we were able to do.

# 4. Team Bio

**Ryan Kramer**: 4th year Comp Sci Major, lots of Unity experience and worked on multiple games.

**Ryan McQueen:** 3rd year Computer Science major, enjoy writing unreadable Python lambda expressions in my spare time, and reading tech related books

**Kevin Van Kessel:** 3rd year Computer Science Major. Experience with game engines

# 5. References

[1] *C# Language Specification*, 5th ed. Microsoft USA: Microsoft, 2015.

[2] Microsoft, "C# coding conventions (C# programming guide)," 2015. [Online]. Available: https://msdn.microsoft.com/en-us/library/ff926074.aspx. Accessed: Feb. 11, 2017.

[3] Mono Project, "MonoDevelop," 2017. [Online]. Available: http://www.monodevelop.com/download/. Accessed: Feb. 11, 2017.

[4] StyleCop, "StyleCop/StyleCop," GitHub, 2016. [Online]. Available: https://github.com/StyleCop/StyleCop. Accessed: Feb. 11, 2017

[5] "Oracle VM VirtualBox,". [Online]. Available: https://www.virtualbox.org/. Accessed: Feb. 11, 2017.

[6] "Git,". [Online]. Available: https://git-scm.com/. Accessed: Feb. 11, 2017.

[7] GitHub, "Build software better, together," GitHub, 2017. [Online]. Available: https://github.com/. Accessed: Feb. 11, 2017.

[8] "C# Programming Guide", *Msdn.microsoft.com*, 2017. [Online]. Available: https://msdn.microsoft.com/en-us/library/67ef8sbd.aspx. [Accessed: 14- Apr- 2017].

[9] "Travis CI - test and deploy your code with confidence,". [Online]. Available: https://travis-ci.org/. Accessed: Feb. 11, 2017.

[10] U. Technologies, "Unity - game engine," in Unity, Unity, 2017. [Online]. Available: https://unity3d.com/. Accessed: Feb. 13, 2017.

[11] "Asset store," in Unity Asset Store. [Online]. Available: https://www.assetstore.unity3d.com/en/#!/. Accessed: Feb. 13, 2017.

[12] “Unity Services - Collaborate,” Unity. [Online]. Available: https://unity3d.com/services/collaborate. [Accessed: 21-Mar-2017].

[13] "StarUML", *Staruml.io*, 2017. [Online]. Available: http://staruml.io/. [Accessed: 21- Mar- 2017].

[14] Fullerton, Tracy. Game Design Workshop. 2nd ed. Boca Raton, FL: CRC Press, 2014.

[15] "Google Java Style Guide", *Google.github.io*, 2017. [Online]. Available: https://google.github.io/styleguide/javaguide.html. [Accessed: 21- Mar- 2017].

[16] "UML Component Diagrams: Guidelines", *Msdn.microsoft.com*, 2017. [Online]. Available: https://msdn.microsoft.com/en-us/library/dd409393.aspx. [Accessed: 23- Mar- 2017].

[17] "Practices Acceptance Test Engineering Guidance", *CodePlex*, 2017. [Online]. Available: http://testingguidance.codeplex.com/. [Accessed: 14- Apr- 2017].

# 6. Source Code

Attached as a separate folder titled “source\_code”

# 7. Video file of current game

\*\***Located on YouTube due to size limitations on Connect\*\***

<https://youtu.be/j_E9t6F3De8>