Introduction To Programming Final Project

# Objective:

The player has been chosen for an important mission, involving two scientists that were conducting research at a facility and need to be rescued. The player is dropped off on the map to complete the objective. Prior to rescuing the two scientists, you must collect nine nuclear cores located throughout the map. However several wild animals are also wandering throughout the map. Once all the cores have been collected and given to the scientists, they are able to hide in a safe place before reinforcements arrive. If the player loses a battle before they are able to collect and give all 9 cores to the scientists, the player loses the game.

# Gameplay Instructions:

## Controls

The basic controls for the game:

* **(1) Use item:** This allows the player to use the items that the player chose to start with.
* **(2) Move:** This allows the player to move around the map.
* **(3) View Map:** This tells the player what area they are in, while also showing a picture of map and the player’s location.
* **(4) Search Surroundings:** This allows the player to search for items that might be scattered around the map
* **(5) Rest:** Lets the player rest, giving them a chance to recover some health
* **(6) Check Character:** Shows the player their characters stats, such as their ammo, health and the items in their backpack

# Allocation of Duties for Each Team Member

| **Aditya** | **Dutt** | **Angus** |
| --- | --- | --- |
| * Main programmer | * Assistant programmer * Bug reporting * Testing * Idea generation * Planning Document * Organiser | * Assistant programmer * Bug reporting * Testing * Idea generation * JavaDoc implementation |

# 

# Bug Report

1. **Player inputs were causing errors and not behind handled properly**

* Discovered by Aditya
* Fixed by replacing **sc.next()** with **sc.nextLine()** to properly capture input

1. **The displayed location name was incorrect whenever when the player moved**

* Discovered by Angus
* Fixed by updating the location name before and after the player changes locations to correctly display location names

1. **The player couldn't rest at new location which they had never rested before**

* Discovered by Aditya
* Fixed by switching from using the visited locations to its own variable to keep track of resting

1. **Using items in different orders caused errors due to items shifting around in the inventory**

* Discovered by Dutt
* Fixed by implementing a more reliable method for handling item

1. **Items found during a search weren't added to the backpack, making them unusable.**

* Discovered by Dutt
* Fixed by ensuring searched items are added to the backpack after they are searched

1. **Player damage output was doubled due to damage being added twice during combat**

* Discovered by Aditya
* Fixed by correcting the combat logic to apply damage only once per attack

1. **Ammo count did not decrease after using it in combat, leading to infinite ammo**

* Discovered by Angus
* Fixed by decreasing the ammo count after a weapon is used

1. **Flee chance increased each time the player tried to flee, making fleeing easier**

* Discovered by Aditya
* Fixed by adjusting the flee mechanic to decrease the chance of fleeing on repeated attempts

1. **Replaying the game after it ended didn't reset game state, causing previous items and location data to remain attached to the player**

* Discovered by Angus
* Fixed by creating a new player instance on each game restart to fully reset the game state

1. **After defeating an enemy, players couldn't encounter any enemies in other locations**

* Discovered by Dutt
* Fixed by resetting the enemy encounter state after each battle

1. **Story items were unable to be 'used' when outside of combat**

* Discovered by Aditya
* Fixed by using a **boolean** on the the player instead using one that was used for combat

1. **The player was still able to play the game after being defeated by an enemy**

* Discovered by Angus
* Fixed by using a global variable in the **Combat** class instead of instantiating one every time the fight began

1. **Items selected at the start of the game were not being added to the backpack**

* Discovered by Aditya
* Fixed by the **switch case** statement as it was checking for **chars** instead of **ints**

# Game Planning:

## Inventory System:

Throughout the game, the player will be able to utilise items that give the player health and ammo. Healing and ammo items can also be selected at the beginning of the game but some other items are able to be picked up across the map. Each item is different in terms of its weight and how much they provide to the player.

| **Weapon** | **Damage** | **Ammo** | **Range** | **Weight** |
| --- | --- | --- | --- | --- |
| **Pistol** | 6 | 16 | 1 | 1 |
| **Rifle** | 10 | 12 | 3 | 3 |
| **Sniper** | 14 | 8 | 5 | 5 |
| **Item** | **Action** | | |  |
| **Food Pack** | Healing - 10 | | | 1 |
| **Ammo Box** | Ammo - 18 | | | 2 |
| **First Aid Kit** | Healing - 25 | | | 3 |
| **Bundle of Rotten Fruit** | Healing - 4 | | | 1 |
| **Pack of Pain Killers** | Healing - 8 | | | 1 |
| **Handful of Loose Ammo** | Ammo - 6 | | | 1 |
| **Used Magazine** | Ammo - 12 | | | 1 |

## 

## Player Details:

The player chooses their name at the start of the game and begins with 100 health. To regain health they can heal with items or through resting which has a change to regenerate some health. They also have an inventory system in the form of a backpack to hold their items which they can use. They can search for items on the map which has a chance to give them a story related item or another item they can use.

## 

## Combat System:

Battles are turn-based and occur randomly as the player explores the map. On each turn of the battle, the player can choose to fight the enemy with or without items, or flee from the encounter. Multiple enemies can appear in each encounter, each with unique attacks and health. The rate at which the player encounters enemies is based on their weight, heavier loadouts result in a higher weight causing more encounters. Their attacks are also dependent on their weapons range – weapons with a shorter range are more likely to miss the enemies. The player can also flee these encounters, however, repeatedly fleeing results in lower chances to flee other possible encounters.

## Navigation and Map:

Players can explore the game's map freely across a 5x5 grid, where each tile represents a unique location and different encounters and items. Movement is handled through input from the user, picking the direction where they want to go, with a visual map that shows their current location and what’s nearby.



Each location contains data to determine whether the location has:

* Combat, and if so how many encounters are possible
* Searchable items (normals items, quest items, or NPCs)
* Restable locations

This data is changed after the player performs an action in that location, making them unable to perform that action again, for example, the player cannot rest multiple times in the same location.