Living Document

Team Info

Team Members:

- Douglas Sandford - Developer
- Kaden Allen - Developer/QA Engineer
- Joseph Liefeld - Designer / Developer
- Jonathan Hotchkiss - Developer, Tester
- Graham Glazner - QA Engineer, Designer
- Wyatt Fujikawa - Developer / Designer
- Owen Wickman - Developer/QA Engineer

Github

https://github.com/LitchDoctor/winter2025-group2-rpg.git

Trello

https://trello.com/invite/b/6781aa06f4132bde26a088ae/ATTI281728df6d053a134dd1f0f5fcf09ab6E74188A9/pt-2-group-2

Communication

Discord and Text Group chat

Rules

- Check Discord regularly
- Respond within 8 hours if pinged on discord
- Text Group chat for high priority messages

Project Description:

Protect Humanity RPG

Abstract

Our project is a role-playing game where the player plays a robot who travels through a dangerous, apocalyptic world where humans have mutated into horrific creatures, and your job is to escort the last living human to safety. The map includes enemy encounters that enter the player into a turn-based battle. For each turn, the player must choose an action to keep the human alive. The human will also act, though they may not always be helpful.

Goal

We are trying to provide the large video gaming community with a novel and entertaining game.

Current Practice

Today's RPGs are typically developed with game engines such as Unity or Unreal Engine. These games often feature vast open worlds and real-time combat. Many RPGs also integrate online multiplayer and dungeon raids with a group of players.

Novelty

Unlike previously made RPG's, our game will have the player be protecting another entity, which differs from other games where the focus is just on the player's survival.

Effects

In our day and age, the video gaming industry is at an all-time high demonstrating how highly our society values entertainment. The game we propose will inject itself into the said industry and provide the community with a new and valuable source of entertainment.

Responsibility

- Combat
    - Enemy/Human Decision making(RNG, Decision tree, whatever we decide)
    - Damage and HP
    - Battle UI
        - Player Decision UI
        - Player Display
        - Enemy Display

        - Human Display
        - Descriptive Text (e.g. "*The enemy uses laser attack!*","*The human trips on a rock!*", etc.)
        - Battle Dynamic Graphics
    - Game States(Start, Player's Turn, Won, etc.)

- Top Down Exploration:
    - Player Movement
    - Player Movement Animation
    - World Design
    - Hazards/Blocks
    - In Game Menu

- Opening Menu
    - Start Game
    - Quit

- Player Progression(leveling up)
- Sprite Designs

Stretch Goal Responsibilities:

- Items/Player Inventory
- Save files
- NPC dialogue encounters -> overarching story
- World interactables (use axe to cut down tree)
- Quests

Other Responsibilities:

- Manage/Update Trello Project
- Keep TA Updated

Use Cases

Use Case: As a player, I want to enter an enemy battle encounter so that I can progress the game.*Author: Graham Glazner*

1. Actors
    a. The Player
2. Triggers
    a. The Player moves their Character into the same square as an enemy.
3. Preconditions
    a. The enemy is visible on the screen. The character is not currently in a battle. The character has a clear path to the enemy.
4. Postconditions (success scenario)

    a. The character is in a battle with the enemy they ran into. The enemy, the character and the human npc are displayed. The character's battle options are shown.
5. List of steps (success scenario)
    a. The player moves their character into the enemy's square.
6. Extensions/variations of the success scenario
    a. The transition to battle animation plays.
7. Exceptions: failure conditions and scenarios
    a. The Character goes through the enemy sprite without entering a battle.

Use Case: As a user, I want to heal the "pet" during an encounter so that they do not die
Author: Jonathan Hotchkiss

1. Actors
    a. User
2. Triggers
    a. User selects to heal "pet"
3. Preconditions
    a. "Pet" is alive (hp > 0) and not at max hp
    b. Heal cooldown has expired
4. Postconditions (success scenario)
    a. Heal cooldown increased
    b. "Pet" hp has increased
5. List of steps (success scenario)
    a. User enters into an encounter with an enemy
    b. At least one round of fighting takes place to reduce "pet" health
    c. User selects heal when battle prompts are shown
6. Extensions/variations of the success scenario
    a. The user chooses to defend or attack and wait another round before healing
7. Exceptions: failure conditions and scenarios
    a. The user does not heal the "pet" and it dies

Use Case: Player movement
Author: Owen Wickman

1. Actors
    a. The player
2. Triggers
    a. The player presses any of the movement keys W, A, S, or D
3. Preconditions
    a. The player is in the main world (not in a menu or encounter)
    b. The players movement is not restricted
4. Postconditions (success scenario)

    a. The player moves in a direction depending on the key pressed
        i. W -> Up
        ii. A -> Left
        iii. S -> Down
        iv. D -> Right
    b. If two keys are pressed, the player will move in the average direction
        i. W and D for example will cause the player to move up and to the right
5. List of steps (success scenario)
    a. The user presses any movement keys
        i. The player moves in the correct direction ex: W -> Up
6. Extensions/variations of the success scenario
    a. The user presses multiple movement keys simultaneously
        i. The player moves in the average direction ex: S and D -> Down and Right
7. Exceptions: failure conditions and scenarios
    a. The player fails to move when movement keys are pressed or the player moves in the wrong direction

Use Case: Defeating an Enemy (Author: Joseph)
Actor: User playing the game

Trigger(s): Using an attack that reduces the enemy's health points to less than or equal to zero

Precondition(s):
- In battle state
- Enemy has remaining health (hp > 0)

Postcondition(s):
- Sprite disappears from screen
- "Enemy defeated" text box is displayed

List of Steps:
- User enters a fight
- User uses abilities that deal damage to enemy

Extension/Variation:
- User dies before defeating enemy

Exceptions (failure conditions + scenarios):
- Enemy's HP is reduced to <= 0, yet the enemy remains one screen and continues taking turns

Use Case: As a user, I want my character to level up as a reward for killing an enemy so that I have a reason to kill enemies and get stronger.
Author Wyatt
Actor: player

Trigger(s): passing or meeting the experience threshold required to level up

Precondition(s):

- Player must defeat an enemy or gain enough experience to pass the amount of experience required to level up

Postcondition(s):

- Pop up message saying “You have leveled up!”
- Experience data is saved and added to player class in case player doesn’t level up
- Music queue indicating level up?
- Random increase in all player stats (1-5)
    - Not sure which stats we want a player to have
        - HP, MP, strength, resilience, speed, etc. (open to ideas)
- More experience is required to level up again
- Experience is “reset” after every level up

List of steps (success scenario)

1. Player defeats entity or gains experience another way
2. Meets the experience requirements
3. Player levels up and a pop up message is generated
4. Another pop up is generated showing statistical increase
    a. Example: 1 → 4
5. Pop up closes and user is returned to the main scenario

Extensions/variations of the success scenario

- Learning new abilities?
- Skipping levels if enough experience is gained leading to greater stats increase, new abilities, etc.
- Can both the player and the robot level up?

Exceptions: failure conditions and scenarios

- Experience isn’t properly saved to the player class after an enemy is defeated
- Messages don’t pop up when player levels up
- Player stats aren’t properly updated or saved
- No new abilities are learned even if they’re supposed to be (if we add new abilities)

Use case: Game over conditions
Author Kaden
Actor: Player

Triggers: Player character or AI companion reaches 0 hp

Preconditions: Player is in combat and an allied character reaches 0 hp.

Postconditions: Depending on which entity dies, either combat continues without user input(player death) or the game over screen displays(AI death)

List of steps (success scenario)

1. Player enters combat with an enemy
2. Player is damaged by enemy
3. Player's Hp falls to 0
4. Death message displays
5. Combat continues without player input. (Until AI dies or battle is won)

Extensions/variations of the success scenario

- If the AI companion dies instead of the player character, combat ends with a game over screen.
- If the player character is revived mid combat, they should be able to act again.

Exceptions: failure conditions and scenarios

- Combat continues after the AI companion dies
- Player remains able to act after reaching 0 hp
- Game over screen fails to display
- Combat is treated as a victory even if the player loses

Use case: Talk to an NPC
Author Douglas
Actor: Player

Triggers: The player character is near an NPC with an icon above their head and starts an interaction with them

Preconditions: The player is able to walk around the world and not in battle

Postconditions: The NPC dialog is starting allowing the player to read what the NPC is saying

List of steps (success scenario)

1. The player moves to the NPC
2. The player can progress the dialog with a click or a button
3. The dialog starts

Extensions/variations of the success scenario

- the player receives the next quest or item

Exceptions: failure conditions and scenarios

- The NPC is not able to be interacted with
- Player cant reach NPC due to error

Non-Functional Requirements

- Each time the player lands on a battle, the battle animation/screen should load up within 3 seconds.
- Player movements must occur within .5 seconds of the user providing input
- The game should maintain a consistent frame rate of at least 30 FPS
- The game should offer colorblind options
- The system should be highly modular making it easy to implement new content

External Requirements

- The program will be able to withstand reasonable user input errors and will respond accordingly.
- The game must be downloadable and playable for users on their own devices.
- Software used for the entire program should be accessible to others so that they can develop their own versions. As a result, the program should be well-documented and understandable by others.

Team Process Description

- Specify and **justify** the software toolset you will use.
    - Unity includes a 2D preset that we can build off of and is designed for game creation. Our art requirements are not large, so we should be able to make everything that we need in Piskel.

- Define and **justify** each team member's role: why does your team need this role filled, and why is a specific team member suited for this role?
    - Kaden: Developer/ QA Engineer
        - Our team needs many developers since there is a lot of different types of development work to be done. I have played many different video games so I should be able to identify quality problems before they become ingrained in our systems.
    - Douglas: Developer/ Combat Engineer
        - Our team needs people to design each part of the game. I have experience in playing multiple different kinds of turn-based games. I should be able to efficiently make changes based on testing.
- Provide a schedule for each member (or sub-group) with a *measurable*, concrete milestone for each week. "Feature 90% done" is not measurable, but "use case 1 works, without any error handling" is.
    - Combat Team (Doug/Kaden)
        - Week 1: Explore different ideas for the combat system
        - Week 2: Formalize combat requirements
        - Week 3: make characters classes
        - Week 4: Create an ability structure
        - Week 5: Simple combat system
        - Week 6: Add in-game over system
        - Week 7: Create an Item system
        - Week 8: add the ability to use the items
        - Week 9: Bug hunting/polish game
        - Week 10: Ensure successful product launch

- Specify and explain at least three major risks that could prevent you from completing your project.
    - None of our group members have worked on a project like this before, so if there is not a relevant tutorial that we can reference, we may not be able to create the features that we want.
    - Creating art may take too much time so we may have to cut corners on looks.
    - Different teammate schedules may keep us from meeting consistently.

- Describe at what point in your process external feedback (i.e., feedback from outside your project team, including the project manager) will be most useful and how you will get that feedback.
    - External feedback will be most helpful once we get the majority of our systems online towards the beginning of beta testing. This feedback can be given by the TA or anyone not involved in the project. We will mostly be interested in game

feel and player experience. This info should allow us to tweak our final product to be more suitable for a wider audience.

Technical Approach

We will create the game using the Unity game engine and C #. The sprite assets will be created using pixel art and the online sprite editor Piskel.

Risks

Our team has limited experience with Unity, and when it comes to RPG's it is easy to overestimate what can be realistically completed in the timeframe. We will combat these risks by focusing on learning Unity early on as well as building the most integral features.

Features

Major

- Top Down Exploration
- Aesthetic Visuals
- Satisfying Encounter-Based Combat System
- Exciting Player Progression
- Quest Oriented Progression
- Item System

Stretch Goals

- Save/Load Game
- Items & Inventory
- Multiplayer
- Story

Core Elements

- Combat
    - Player Decision Menu
    - Enemy/Human Decision making(RNG, Decision tree, whatever we decide)
    - Battle Graphics
    - Descriptive Text (e.g. "*The enemy uses laser attack!*","*The human trips on a rock!*", etc.)

- Top Down Exploration
    - Player Movement

    - Player Movement Animation
    - World Design

- Player Progression(leveling up)
- Sprite Designs

Secondary Elements

- Items/Player Inventory
- Save files
- NPC dialogue encounters -> overarching story
- World interactables(use axe to cut down tree)
- Quests