Game Proposal:

Farmer Defense - The Last Days

CPSC 427 – Video Game Programming

Team6: Farmer Nation

Kung Lim	27323864
Daniel Lee	48465041
Anna Tao	76542653
Ziqing Wang	22270649
Haonan Zhang	98313232

Story:

Briefly describe the overall game structure with a possible background story or motivation. Focus on the gameplay elements of the game over the background story.

You are a hardworking farmer running a small farm far away from the city. Unbeknownst to you, an apocalypse began in the middle of the night when a meteor carrying an unknown virus struck the city. Half asleep, you wander over to your window to investigate the loud sound that woke you up; seeing the peaceful rows of crops, nothing looked suspicious. You went back to sleep, unknowingly avoiding the initial virus outbreak. The next day, you wake up and as part of your daily farmer tasks, start planting seeds as usual. Surprisingly, the seeds start growing at a supernatural speed. Some of them develop limbs, and emerge from the ground, able to walk around, and some of them bear fruits that are actually weapons you can use. Before you have time to freak out, zombies begin approaching your farm from all sides. Armed with only your pitchfork and your crop allies, you must fight to protect your land. Each day brings new challenges: tougher zombie variants, dwindling resources, and the constant need to expand your farm to grow more seeds. As you fend off the undead, you grow stronger, learning new skills and acquiring new weapons and seeds to improve your chances of survival. But your tools have limited durability, so you'll need to use them wisely.

The question is no longer about being a farmer. It's about how long you can survive in this new world where your crops are your lifeline, and your farm is your fortress.

Gameplay:

The goal of Farmer Defense is to protect yourself from zombies for as long as possible. As time progresses and you attempt to defeat the zombies, you are forced to make use of the given farmland and grow plants that can assist you in this cause. High scores in the game are determined by the number of days that you have survived, with the time spent playing the game in total used for any potential tiebreakers.

There is only one map/environment in our game. In this map, there are various plots of farmland. Seeds can only be planted at these given positions of farmland. If the player runs out of farmland, a hoe can be used to expand the area in which seeds can be planted, but the hoe itself cannot be used indefinitely. In addition, the hoe can only be obtained by growing the plant that spawns the hoe upon harvest. The purpose of restricting the farmland in such a measure is to force the player to cover ground and leave some farmland vulnerable to attack by zombies.

At the outset of the game, the player is provided with a set number of seeds and a set amount of farmland. The only way to attain more seeds is to level up, and the only way to attain more farmland is to plant the seeds that grow into the plant that spawns the hoe and then use it to create more farmland.

The player can move in the X and Y coordinates using the WASD keys. There are no jumping or climbing mechanisms, and as such the entire environment is a flat surface. The camera perspective is from a top down angle.

Zombies will spawn from all four corners of the game map. The zombies will walk primarily towards the player and will attack the player when they reach them. The zombies can also attack saplings and plants, but they must be within a certain proximity to a plot of farmland in order to do so; otherwise they will go straight to the player. Zombies can have different attributes such as move speed, damage inflicted on the player, and health. When the zombies attack the player, the player's health bar will drain. The game ends when the player's health bar reaches zero.

The player can attack the zombies with a weak pitchfork from the very beginning. Although this is the only weapon that can be used indefinitely, it will take multiple uses for the player to successfully kill a zombie, and it would be very inefficient to do so. This incentivizes the player

to plant seeds that will kill the zombies better, because if they solely rely on the given weapon, they will lose very quickly.

The player plants seeds on various plots of farmland by using the F key. These seeds can only be planted with respect to a visible grid system. If one plant is placed within a given square, no other plants can be planted in that square.

After a seed is planted, it will take time for it to grow. For instance, it will take time for a sapling to form, and then it would take time for it to grow tall. During this time, the zombies can go towards the saplings and then destroy the plants. If the zombies successfully reach a plant, they can destroy it, and the plant will die without spawning anything. Therefore, the player must go and kill the zombies before they are able to destroy any plants.

After a set period of time, plants can be harvested, also by means of using the F key. This involves different types of plants. For instance, one class of plants involves those that can grow into solid defenses that can attack zombies from a stationary position. One example of a solid defense can shoot at zombies from afar, and another example can attack zombies when they get super close. These solitary defenses can be killed by the zombies. Another class of plants involves those that become non-player characters that can move towards the player's current location and attack and kill any zombies that get in their way. These NPCs can take damage from zombies and eventually be killed. Another class of plants involves those that spawn weapons when harvested. These weapons can include a single-use grenade as well as a bow-and-arrow combination. However, the player cannot use these items indefinitely, as they will eventually wear down. Finally, another class of plants involve those that spawn items that increase the value of the player's health bar when used. All of these harvests can be stored in a toolbar for safekeeping.

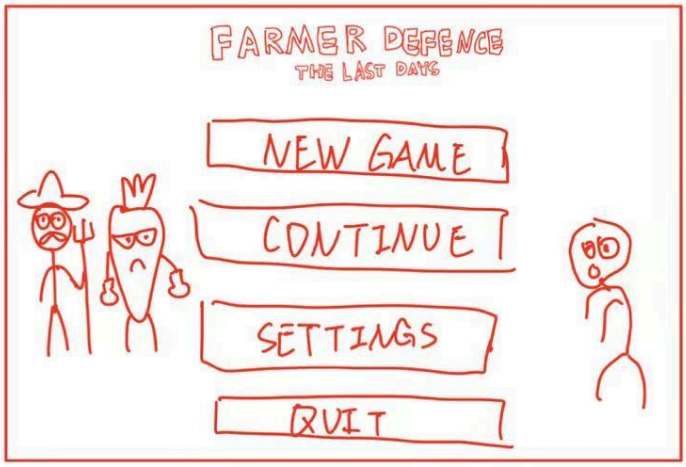
Progress in the game is split using a day system. After a set period of time in which the player survives zombie attacks and defends their plants, a quick day/night transition occurs, and the player gains a new level with respect to experience. At this point, they will be given the option to choose from one of three packs of seeds. With each passing day, more types of zombies will be introduced, such as those with different move speeds and higher levels of resistance or attack power.

In addition to the day system, the player can also gain experience by killing zombies. Killing a single zombie results in a rise in the experience level. After killing a certain number of zombies, the player will reach a new level with respect to experience and can make a selection between one of three packs of plant seeds. But as the player becomes stronger, the quantity of zombies will increase, and their attributes will start to vary to form new types of zombies, presenting an ever greater challenge. In addition, a day cannot end unless the player kills all of the zombies that are present on the given map.

Scenes:

Produce basic, yet descriptive, sketches of the major game states (screens or scenes). These should be consistent with the game design elements, and help you assess the amount of work to be done. These should clearly show how players will interact with the game and what the outcomes of their interactions will be. For example, jumping onto platforms, shooting projectiles, enemy pathfinding or 'seeing' the player. This section is meant to demonstrate how the game will play and feeds into the technical and advanced technical element sections below. If taking inspiration from other games, you can include annotated screenshots that capture the gameplay elements you are planning to copy.

Game Start:



We are planning to have 2 screens. The first one is showing the splash screen with the game title. After you click your mouse, it will lead you to the main screen with options to start a new game, continue your saved game, modify your settings, or quit the game.

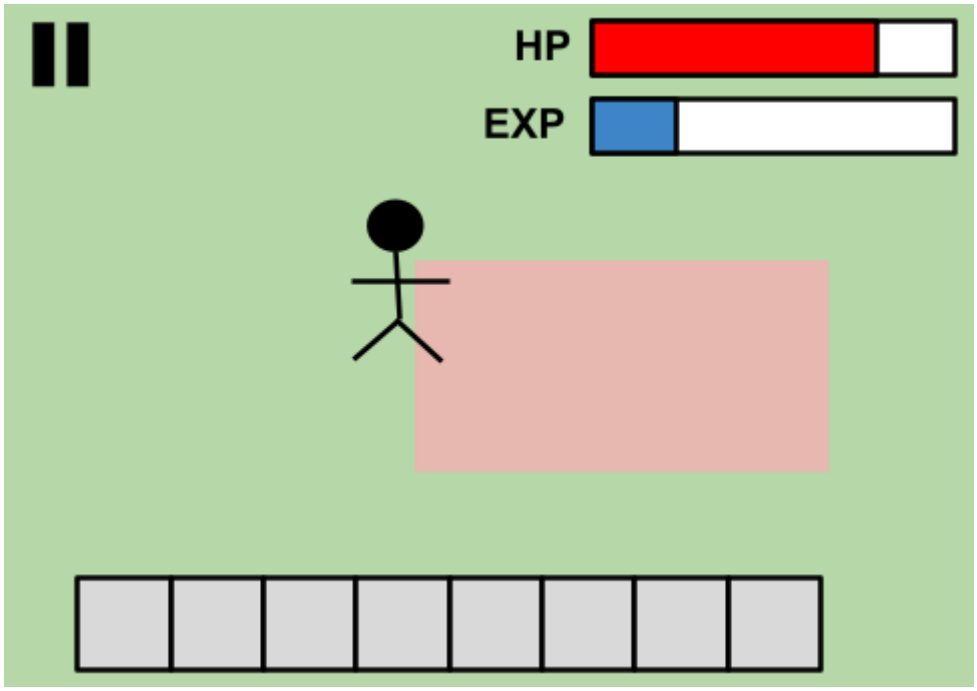
Game Over:

We plan on doing something similar to vampire survivors, where the map will fade into the background and a big “GAME OVER” will appear on the screen:

We will also display the option to “QUIT” or “RETRY” in a layout similar to the screenshot above, however the colour of the buttons have yet to be determined.

General User Interface:

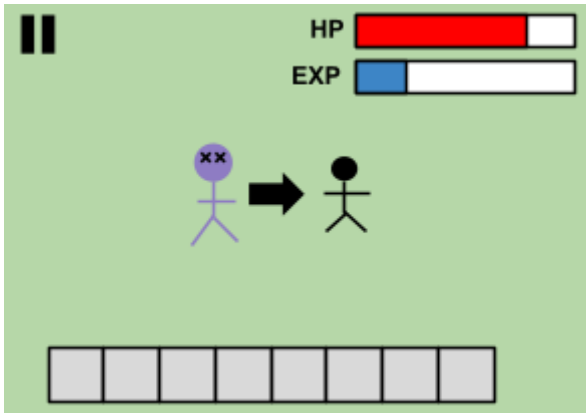
We plan on having our general game User Interface look similar to this:



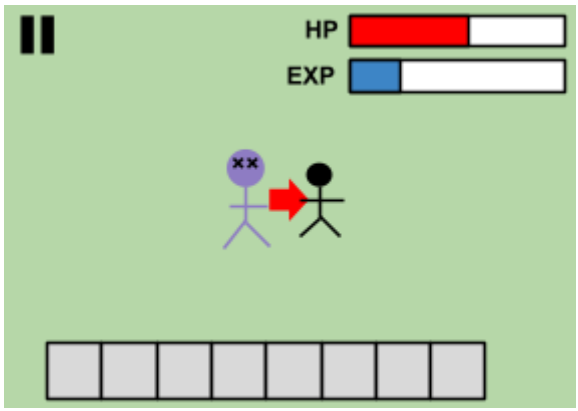
As you can see, there is a pause button on the upper left, the HP and experience bars on the upper right, and the toolbar on the bottom. In addition, we plan to add a "days survived" counter below the experience bar (it is not shown to be consistent with later examples). Finally, the beige plot of land represents the farmland that you use to plant plants, with the green color representing non-farmland.

Enemies:

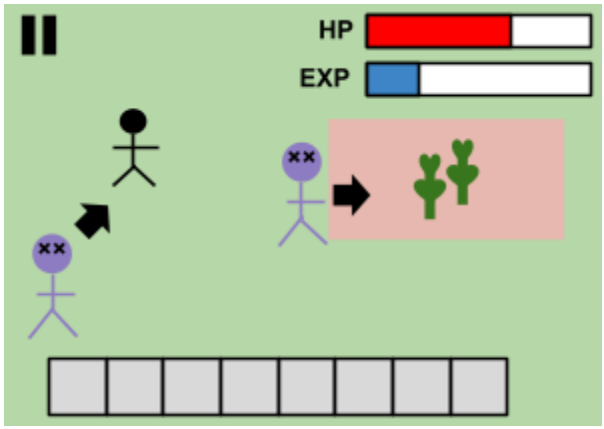
Zombies will always target the player:



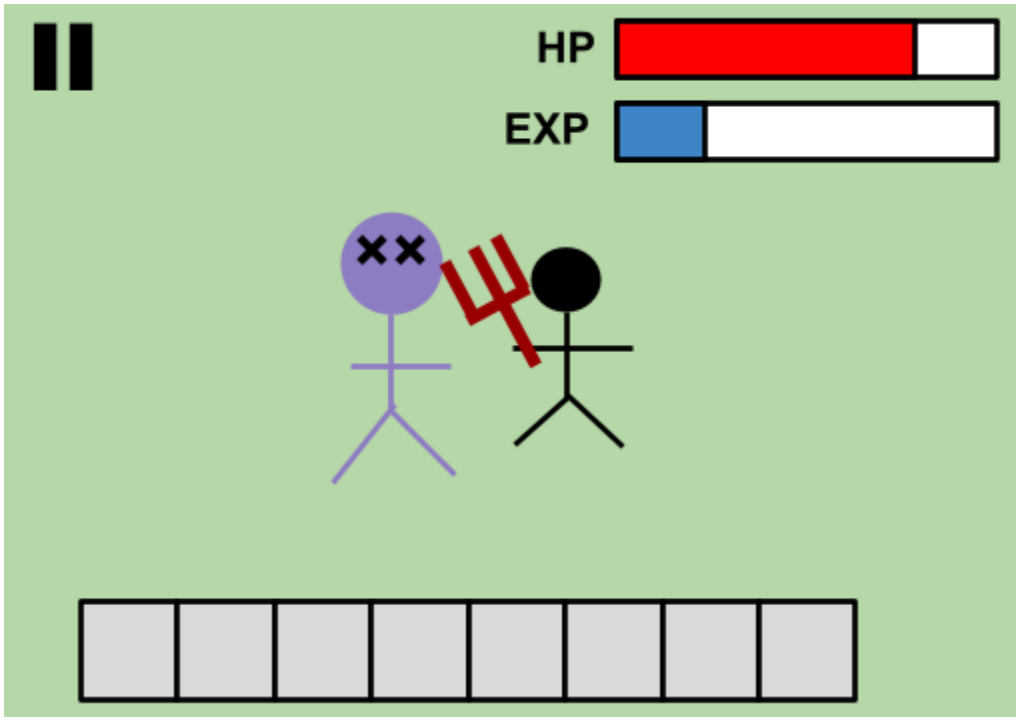
When they reach the player, zombies will deal damage to the player and reduce their health bar:



Zombies will only approach and attack farmland if they are in close proximity to the farmland, otherwise they will target the player. Once they reach the farmland, they can destroy the plants:

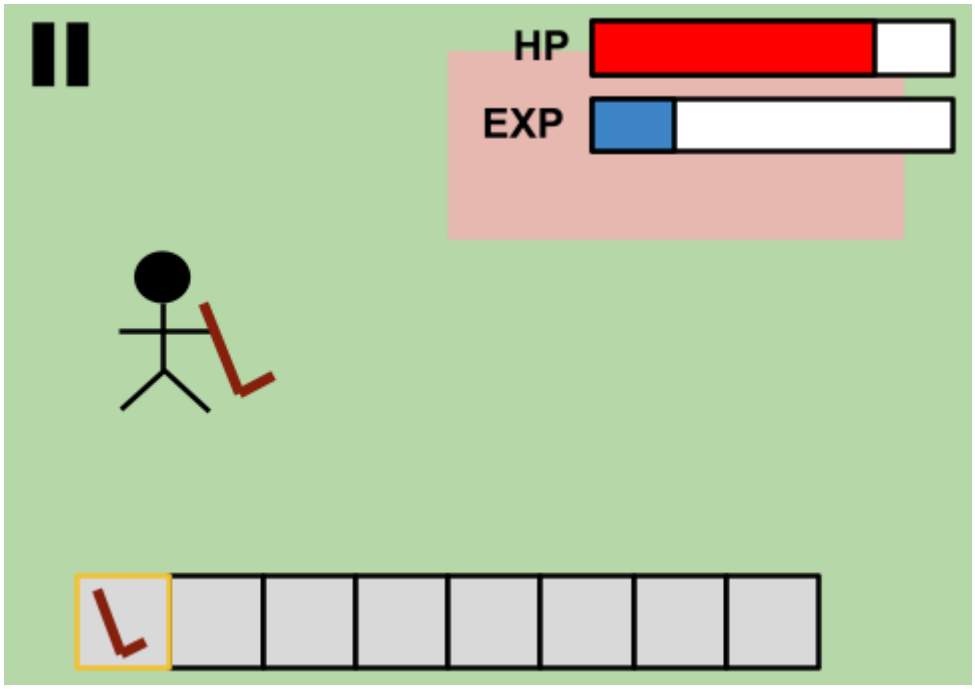

HP
EXP

Attacking:

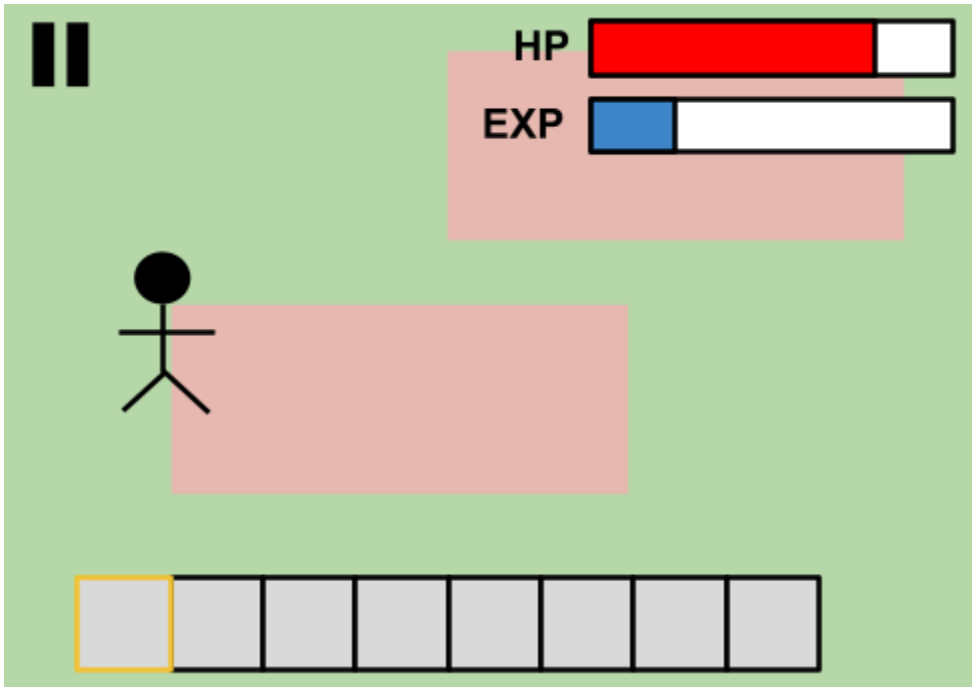


The player always has a basic weapon. Unlike all other weapons, the basic weapon can be used indefinitely. However, the damage done is very small and cannot be used to achieve success in the game.

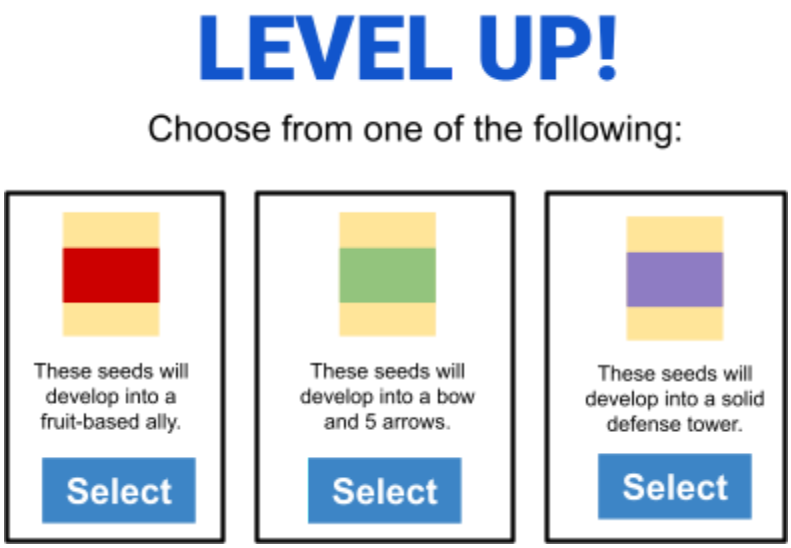
Using the Hoe:



After usage, the hoe disappears, and more farmland is created:

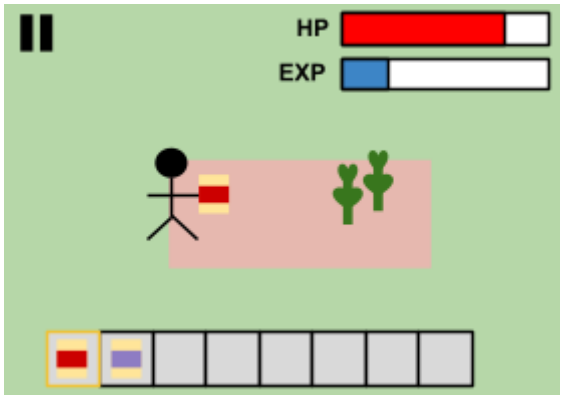


Level Up:

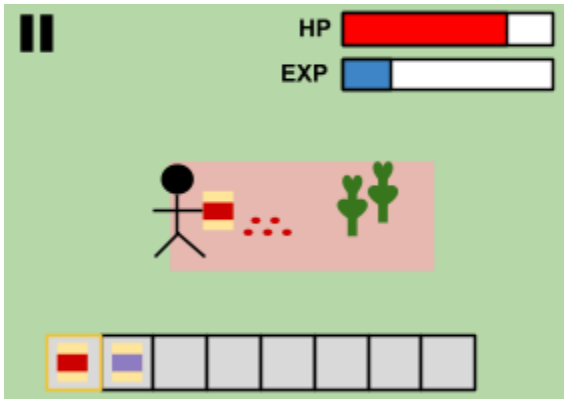


This is what the player sees once they level up.

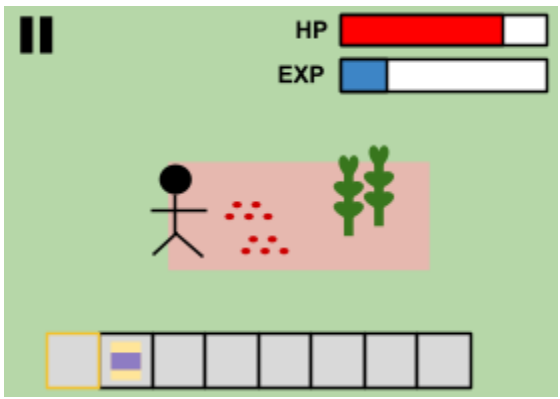
Planting Seeds:



The plants that are initially depicted are saplings and are not ready to be harvested.



Eventually, the seeds will run out and the saplings will grow taller, as demonstrated below:



Plant Towers, NPCs, and Weapons:

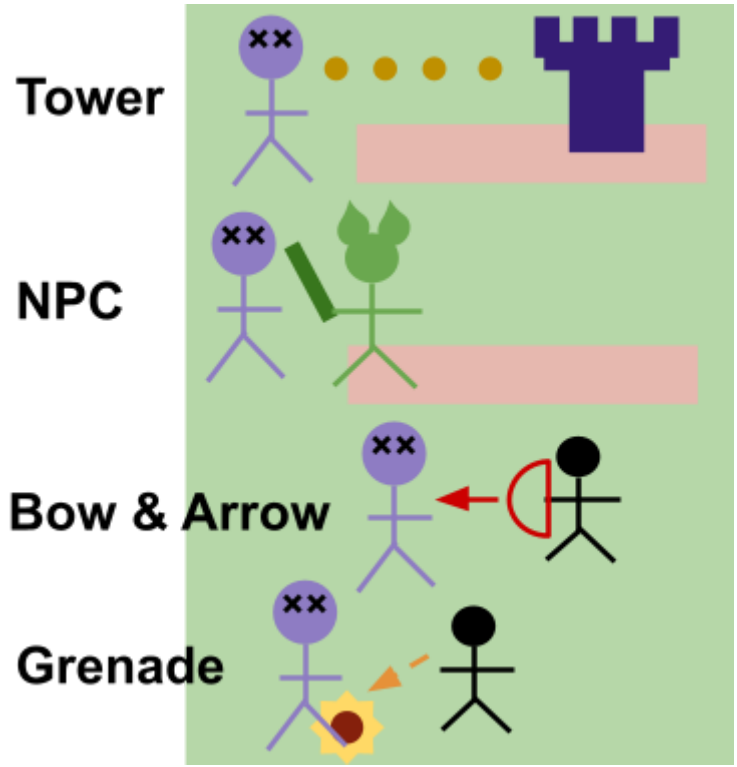
In this diagram, the UI is not illustrated for simplicity purposes.

In the tower, the given defense is shooting projectiles from a distance to attack the zombie.

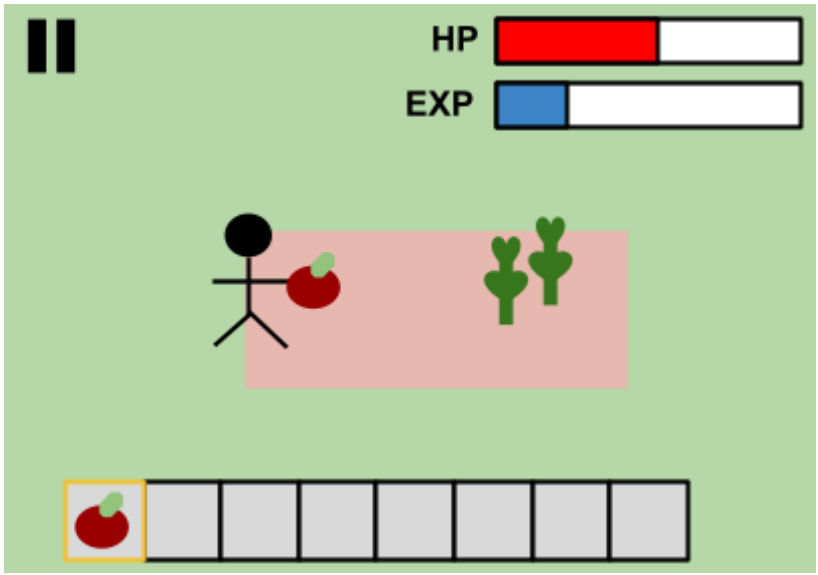
In the NPC, the plant actually walks to the zombie to attack it.

There is a set number of arrows at each time, and the grenade is single use.

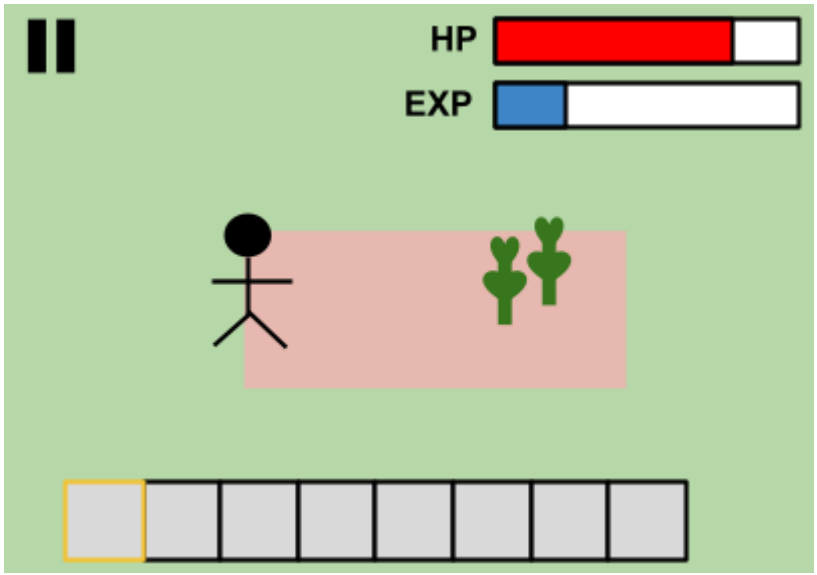
This is just for illustrative purposes only, and there can be other types of plants not depicted.



Plant Items:



This example involves the item that increases your health. After you consume it, it disappears and your health goes up:



There is also an item to increase move speed, but that is harder to illustrate.

Technical Elements:

Identify how the game satisfies the core technical requirements: rendering; geometric/sprite/other assets; 2D geometry manipulation (transformation, collisions, etc.); gameplay logic/AI, physics.

Graphics:

Simple rendering effects

- Animations for the player (idle/walk/attack)
- Visual indicators when taking damage
- Death animations for the player, plants, and zombies

Physics & Simulation:

Basic physics

- Hitbox-based collision system for player, plants, and zombies
- Range-based detection system for plants and zombies
- Plant growth area restrictions

AI Features:

Simple pathfinding

- Enemies move and target the player
- Zombie navigation to avoid obstacles (optional)
- Tower plants stay in place and attack the nearest enemy
- NPC Plants follow the player and attack the nearest enemy

Enemy group behavior

- Zombie pack formation
- Different zombie types with varied behaviors (optional)

UI & Input/Output:

Camera controls

- Make the camera follow the player

Quality & user experience (UX)

- Original assets for the player, plants, and enemies
- Game Balance: The game tries to ensure a linear increase in difficulty (quantity and type of zombies)
- Story elements: A basic text to introduce the background

Audio feedback

- Plant growth sounds
- Combat effects and music
- Ambient farm/zombie sounds

Basic integrated assets

- Custom plant sprites with growth stages
- Unique zombie variants
- Farm environment tiles

Reloadability (optional)

- Players can save and load the game, including positions of enemies, positions of towers, items, day (level), HP

Gameplay Logic

- Player Movement: The player can move up/down/left/right
- Player Attack: The player can melee attack adjacently
- Player Interact: The player can interact with farmland to plant seeds

Advanced Technical Elements:

List the more advanced and additional technical elements you intend to include in the game prioritized on likelihood of inclusion. Describe the impact on the gameplay in the event of skipping each of the features and propose an alternative.

Particle systems:

- Zombie death particles
- Attack impact effects
- Plant growth sparkle effects

  Impact: Reduced visual feedback for key game events
  Alternative: Improving animations for deaths/impacts

2D dynamic shadows:

- Shadow casting from plants and characters
- Time-of-day system affecting gameplay

  Impact: Lack of immersion for players
  Alternative: Using static shadow

Basic physics:

- Movement speed variations based on terrain

  Impact: May make players feel that the game lacks variety
  Alternative: Putting effort into other areas to divert players' attention

Camera controls:

- Zoom the camera based on mouse

  Impact: Restricting the player's view
  Alternative: Make the game a fixed view

Devices:

Explain which input devices you plan on supporting and how they map to in-game controls.

Keyboard:

- support the movement of the player:
    - W: move up
    - A: move left
    - S: move down
    - D: move right
    - F: interact
        - plant seeds
        - pick up items
- shortcut to interact with different UI components:
    - TAB: open the inventory
    - ESC: pause the game and show the menu
    - number keys(1-9): quickly swap between tools, order is the same as the order in the toolbar

Mouse:

- wheel: zoom-in and zoom-out the visible field
- left button
    - make attacks to the direction the mouse is pointed to
    - hold to charge for power attacks
- right button
    - defend, if you have defensive equipment

Tools:

Specify and motivate the libraries and tools that you plan on using except for C/C++ and OpenGL.

Drawing 2D assets

- Photoshop
- https://www.youtube.com/watch?v=KyF2GMBZKX8 <-inspiration
- free 2d assets website

Load map

- Text file to create the map. Have a function to read the text file and render the map onto the screen
- https://www.youtube.com/watch?v=GcNOlhGGhRI <-inspiration (check 2:05)

Sound

- sound effects / BGM
- https://freesound.org/
- use microphone to record our own sound effects, and Audacity to edit the sound
- https://www.youtube.com/watch?v=Kux_LvRl57U <-inspiration

Team management:

Identify how you will assign and track tasks and describe the internal deadlines and policies you will use to meet the goals of each milestone.

The task backlog is a comprehensive multi-sheet document that organizes all tasks. While the main tasks involve coding, there are other tasks to account for UI design and gathering art/audio assets. Tasks are assigned a priority (core/extra) and milestone (1/2/3/4) based on their relevance to the project. There are also columns to track the time spent on each task and the person responsible for it. Furthermore, there are sheets to record the data for each enemy and plant.

Tasks are assigned based on the role of each member. This occurs during the meeting at the beginning of each milestone with priority given to core features. Please note that the role of each member is flexible depending on the requirements for each milestone. In general, we will try to assign a fair workload based on results over time. Also, while everyone will have the opportunity to code, there are some non-programming tasks such as gathering assets which will need to be assigned.

Each milestone occurs over a period of two weeks with the deadline on Sunday. Tasks are allocated at the beginning of each milestone after cross-play. This will give people the opportunity to begin work on the project while taking into account player feedback. Ideally, development should be completed a few days before the deadline to account for testing. The last day will be used to review the final prototype before submission. Also, we will have bi-weekly meetings outside of class to manage problems.

Sun	**Mon**	**Tue**	**Wed**	**Thu**	**Fri**	**Sat**
	Crossplay		Allocate			
			Merge		Testing	
Submit						

The policy for GitHub is that all code must be developed individually in separate branches, named after the contributor and corresponding milestone. Merging is only permitted after a specific day.

As per the teamwork agreement, any concerns with the ability to complete your task should be addressed early. If you are unable to complete your task, then you must inform the group before 48 hours of the deadline. You must provide an explanation and the task may be reassigned to someone else. The quality of your work must be satisfactory as determined by the group.

Development Plan:

Provide a list of tasks that your team will work on for each of the weekly deadlines. Account for some testing time and potential delays, as well as describing alternative options (plan B). Include all the major features you plan on implementing (no code).

Task Backlog (9 Pages):

https://drive.google.com/file/d/1EoQ6-z6onmrQr7FYaItt4WHGi1ie0M61/view?usp=drive_link

To account for delays, the task backlog is divided between core and extra features. Core features are mandatory and should be implemented first. The design is flexible to account for changing requirements.

Please note that the development plan is not fixed and is expected to change over the course of the project. This is to account for changing requirements, technical knowledge, and the actual work completed each milestone.

Each person is responsible for testing their own code, but manual testing will occur once before the end of each milestone. This will cover design, functionality, and performance.

Milestone 1: Skeletal Game

In this milestone, the basic combat system is implemented. Within the map, the player only has the ability to move and attack. Zombies can trace the character and attack. There are a few zombies, but no system to spawn waves over them. By keeping the requirements minimal, this will allow time to set up the ECS and gather assets (art/audio) for the next milestone. There are no menus, no plants, and only grayboxing for assets.

Milestone 2: Minimal Playability

In this milestone, the menus and plants are added, including the Game Start and Game Over screens. Art replaces grayboxing and a full UI is included. The mechanic for growing plants from seeds is introduced, allowing them to develop into either towers or items. The enemy spawn system is partially integrated for testing. Plant NPCs are included in the next milestone.

Milestone 3: Playability

In this milestone, the game is created. The enemy spawn system is fully implemented and a few days (levels) are added. This enables a basic progression where the player must use plants to survive increasingly difficult waves. Additionally, the mechanic for plant NPCs is introduced.

Milestone 4: Final Game

In this milestone, the game is expanded. While game balance is a concern, the focus is on utilizing mechanics in the previous milestones to make a complete experience. New plants/enemies are introduced and multiple days (levels) are developed. Non-essential features such as the pause menu and save/reload functionality are also implemented during this stage.