# Technical Design Document: Farming

## Core Loop

*Define the main activities that the player will perform in the prototype.*

Action(s): Grow Crops, Sell Crops, Kill Enemies

Reward(s): Currency, Combat Items

Expansion(s): Farm Growth

### Core Loop Diagram



This is the games core loop, The player can choose to ignore the combat elements of the game and simply focus on growing and selling crops living a peaceful farming life if they so wish, this section of the game is self-sufficient. However, this would most likely become boring on its own. So the player can use their money to buy equipment from the shop to venture out into the wilds to kill enemies. Killing the world bosses will reward the player with new types of seeds to plant supplementing the farming loop.

### Actions

*Define the main actions the character will perform to play the game.*

|  |  |  |
| --- | --- | --- |
| **Action** | **Description**  | **How does the player perform and complete this action** |
| Grow Crops | The player can grow crops | The player can plant crops on Plots placed throughout the farm, these take time to grow but can then be harvested to later be sold. |
| Sell Crops | The player can sell grown crops | The player can take the crops they’ve grown and sell them at a shop to gain money which can later be used to buy equipment. |
| Kill Enemies | The player can kill enemies | The player can kill enemies in the wilds to gain money to improve their equipment or to expand their farm. |

### Rewards

Define the rewards that the player will be received when they successfully complete the actions.

|  |  |  |
| --- | --- | --- |
| **Reward** | **Description**  | **How and when does the player earn this reward** |
| **Currency** | Gold for use in the shop | The player gets gold by selling items such as crops or unwanted armour, by destroying certain world elements such as crates and by killing enemies. |
| **Combat Items** | Items to improve combat | The player can buy combat items from the shop or occasionally earn them by destroying world elements or killing enemies. |

### Expansions

Define the expansions that the player will obtain to continue playing the game.

|  |  |  |
| --- | --- | --- |
| **Reward** | **Description**  | **How and when does the player use this expansion** |
| Farm Growth | The farm can be expanded | The player can use gold to unlock additional areas of the farm giving access to more Plots for growing crops. |

## Game Rules

Define the main game rules that the player will perform in the prototype.

Include flowcharts and diagrams to support this.

### Win

The player wins by killing all 3 world bosses, Then [paying off the farms debts/buying a boat and sailing off into the sunset]

### Death

The player can die during combat, when this happens they will lose a set amount of money as a respawning fee

### Respawn

The player respawns at the farm after dying.

### Timers

The crops will be grown using real world time, more valuable crops will often take longer to grow. During growth time the player has opportunity to go out killin’

### Money

Money is the main scoring element of the game as well as being a valuable resource used for progression.

## Camera

The game will have a first-person camera that the player has full 360 degree control over, the sword will take up the right hand side of the viewport

## Controls

* WASD – Directional Movement
* Mouse X & Y – Camera 360 control
* Space – Jump
* Left Click – Attack/UI Interact
* Right Click – Block
* E – Interact
* Esc – Pause

## Play Space

*This section was not completed during the Jam However I’ve elected to leave the basic framework of the section in.*

The game will have 5 Game Spaces, The starting farm & Village, and then 3 surrounding Biomes

### Farm

This is the players home and where the player can grow and harvest crops. It is their home and should be designed around being a safe place for the player to be.

*[Map with break down. Elements such as plot locations, respawn/spawn point(s) breakable crate placement, shop location, Collectibles(?)]*

### Village

This is where the player goes to buy and sell items from the farm or for adventuring, its hold out against the enemy infested areas outside the village, something the player will aim to deal with

*[Map with break down]*

### Swamp Biome

A swamp biome infested with rotting skeletons

*[Map with break down]*

### Cave Biome

A cave biome filled with goblin enemies

*[Map with break down]*

### Crags Biome

A cragy biome with a big tower filled with wraith enemies

*[Map with break down]*

## Game Entities

Include, name and describe all the game entities that you will include in your prototype.

|  |  |  |  |
| --- | --- | --- | --- |
| **Game Entity** | **Description**  | **How does the player use/interact/control this entity?** | **How does this entity work?**  |
| Player Character | The player pawn | The player will directly control this character. | This entity contains all the basic logic and inputs that the player has access to. |
| Farm Plots | The plots for planting crops | The player can interact with these to set the desired crop to grow, to harvest the grown crop, or rip out the crop to switch which is being grown. | The player selects a crop and through the magic of data assets it sets its 9 children meshes to the appropriate one and begins a timer to count down to the crops grown state at which point the player can harvest the crops. |
| Shop | A shop for buying/Selling items | The player can interact with this and then either sell items from their inventory or buy items from the shops | Both the shop and player have an inventory which contains elements, the player can drag from theirs into the sell slot to sell an item/item stack or from the shops inventory into their own to buy from it. |
| Enemies | Enemies for the player to kill | The player can kill this entity or it can kill them | The enemy runs off of a behaviour tree which governs their behaviour. They are always hostile and will attack the player on site, continuing to attack until on of them is dead or they lose track of the player. |
| Items | Things that go in the inventory | The player can collect these and they are stored in the inventory | He data for all items is stored as a data asset which could contain logic for anything, from the crops or to armour/weapons or even collectibles to sell for additional gold. |

## Systems and Mechanics

Include, name and describe all the systems and mechanics that you will include in your prototype.

|  |  |  |  |
| --- | --- | --- | --- |
| **System/Mechanic** | **Description**  | **How does the player use/interact/control this System/Mechanic?** | **How does this system and mechanics work?**  |
| Core Player Movement | The core movement systems of the player | The player inputs directly control this system | Input data is taken and turned into relevant movement within the game |
| Player Attack | The player attack system | The player can use this to attack with their equipped weapon | The player character has a child class weapon. When pressing the relevant button this swings in an ark in front of the player dealing damage to any entities in the arcs swing |
| Player Block | A block to prevent an incoming attack | When pressing the relevant button the player can use this to negate an incoming attack | The child weapon will have a secondary function that puts it in a blocking position, if an attack is forthcoming it will be interrupted by the players block |
| Inventory System | A system for storing various items | The player can open this using the relevant key and view stored items, as well as move them around within or drag over to the shop if within proximity. | [See Jamie, data assets, loops] |
| Armour System | A system of lowering damage received | The player can equip items which directly effect this system. | The player can equip items in the inventory by moving them into an equip slot from the inventory slot, equipping a piece of armour will reduce the amount of damage received by the player via the health component, equipping more around increases the amount negated, as does wearing a set. |
| Enemy Manager | A manager for guiding enemies through encounters | The player has no baring over this system beyond triggering it | This system is responsible for managing spawning enemies, initialising them and keeping a combat engagement populated with enemies. |
| *Enemy Variety* | A quick and dirty way of creating enemies | Player can fuck off | The idea is to use Data assets to add false variety to enemies, either through hard coded values or random, having data assets change things such as a hue/glow/size of the enemy as well as health and damage allowing us to make the 3 enemies we have feel diverse and engaging beyond “oh this enemy again” |

##