Odera, Ben, Christos: Studio 2

Technical Design Document

Section 1 - List of Features Captured from GDD

1.1 List of Features Based on the Game Design Document

* Top Down Isometric
* Game world with
* 3D objects
* Characters
* Weapons
* Forest
* Windows platform deployed
* Story
* Cutscenes
* Audio and sound effects
* Enemies (Free Asset)
* Enemy with group AI with A\* path finding
* Comprehensive/informative HUD
* Menu
* Interactive objects
* Farm
* Traders (Likely)
* Mine Trees for wood, Stone for stone, etc.

Section 2 - Choice of Game Engine

The game engine that has been chosen is Unity 3D, Unity is a development tool set. Some of the features that Unity possesses are:

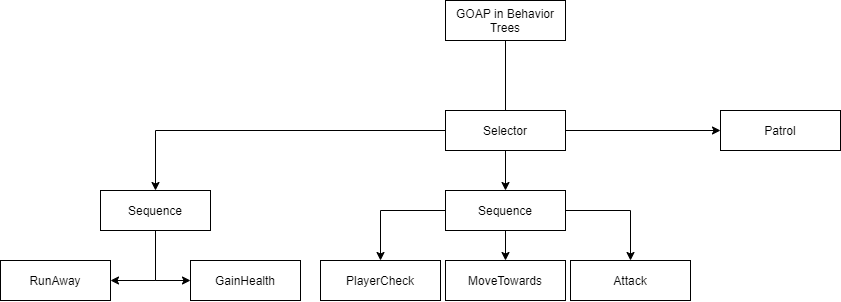
* Animation
* AI
* Audio system
* Editing of the game environment
* C# scripting
* Physics support
* Rendering

Section 3 - Task Breakdown

These are the tasks that must be completed to finish the game

Figure 1 AI Task Breakdown

**pls help ☺.**



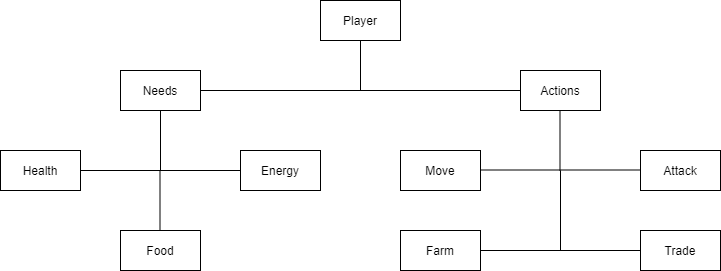
Figure 2 Player Task Breakdown

Figure 3 Gameplay Task Breakdown

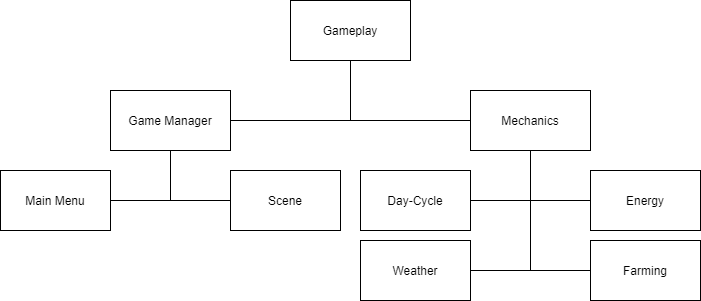
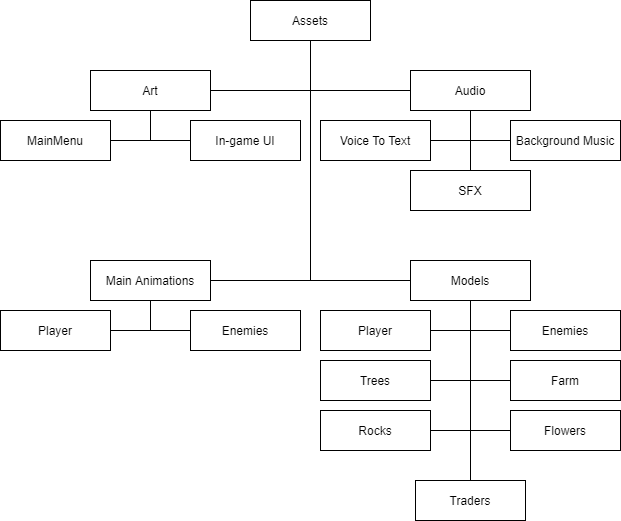
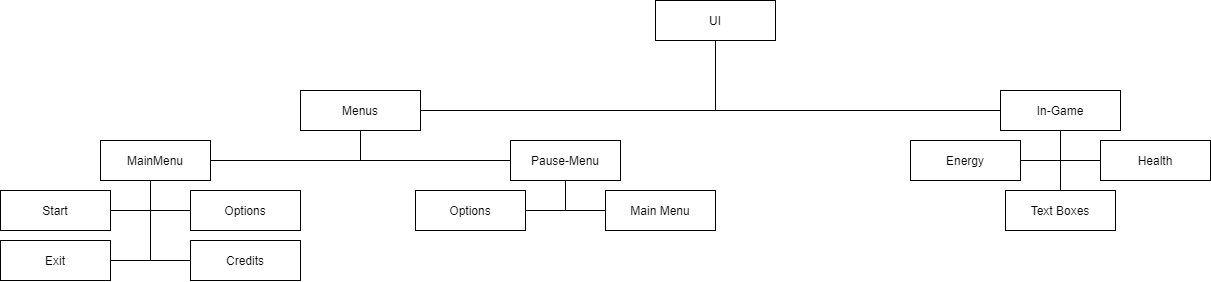
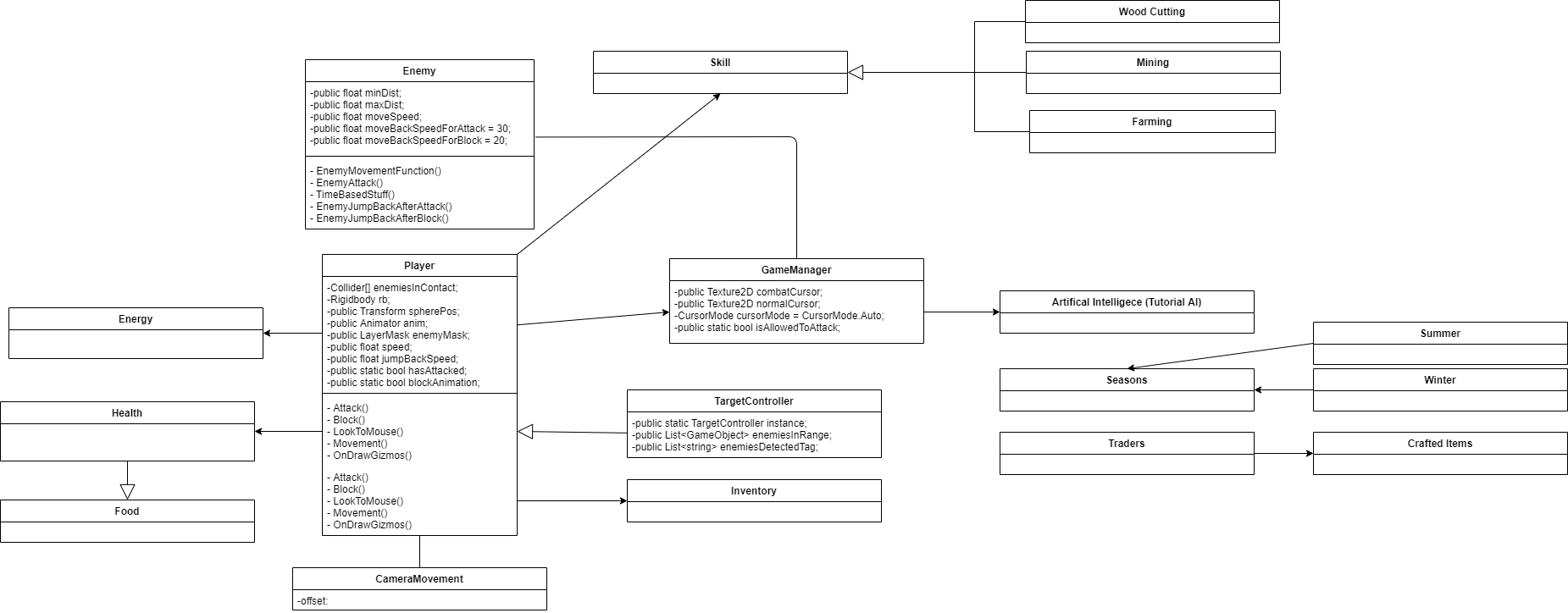


Figure 4 Assets Task Breakdown

Figure 5 UI Task Breakdown

Section 4 - UML Class Diagram

Figure 6 UML Diagram

Section 5 - High-Level Diagrams to Explore Software Design

5.1 Layout Diagrams:

The layout diagram illustrates the layout of the level in the game. This is shown in Figure 6 below:

Figure 6 Layout Diagram

(Place Holders)

Farm🡪

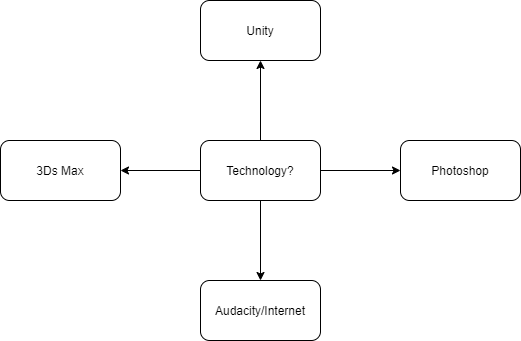
****

Combat🡪



5.2 Technology Diagram:

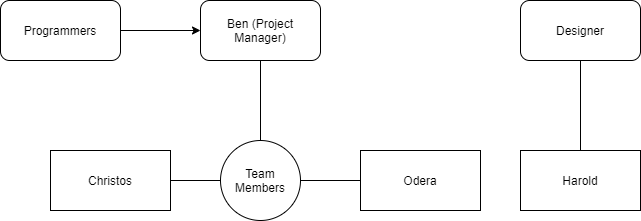
The technology diagram indicates the technology that is going to be used for different aspects of the game and is displayed in Figure 7

Figure 7 Technology Diagram 🡪

5.3 Team Diagram:

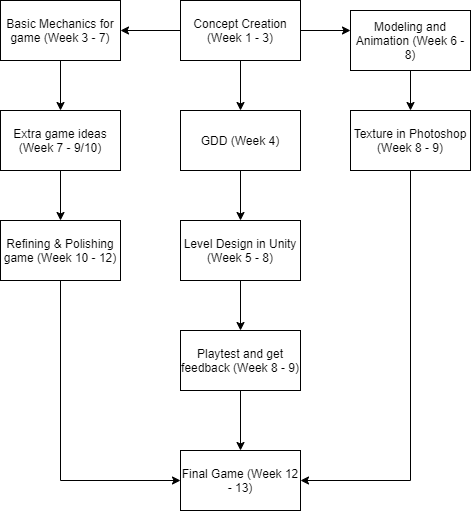
The team diagram represents the teams and their members, and is shown in Figure 8.

Figure 8 Team Diagram 🡪



5.4 Implementation Diagram:

Figure 9 Implementation Diagram 🡪



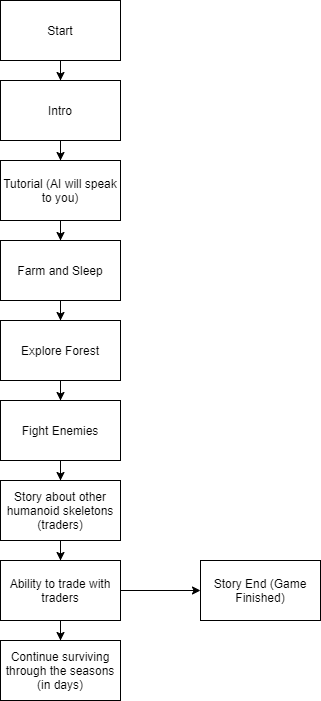
● The implementation diagram illustrates the flow of the different stages of the creation of the game

● It also includes what weeks each stage will take place at as seen above

5.5 Gameplay Diagram:

This gameplay diagram is a simple flowchart of how to play the game, which is shown in Figure 10

Figure 10 Gameplay Diagram



Section 6 - Art Tools

6.1 3Ds Max

6.1.1 Version:

* 2017

6.1.2 About 3Ds Max:

* Design software providing integrated 3D modelling, animation and rendering tools

6.1.3 What will it be Used For:

* 3D modeling

6.2 Photoshop

6.2.1 Version

* 2020

6.2.2 About

* Texture the model that we have created

6.2.3 What will it be Used For:

* Texturing
* Drawing

6.3 Audacity:

6.3.1 Version

* 2.33

6.3.2 About Audacity

* Audio editor for recording, slicing and mixing audio
* Allows live audio to be recorded and tapes and records to be converted into digital recordings

6.3.3 What will it be Used For:

* Possibly music, or sound effects

Section 7 - 3D Objects & Terrain

7.1 3D Objects:

* Farmers Shelter
* This will be where the farm is, as well as the farmers house
* Forest Battlegrounds

- This will be where the farmable trees are, as well as stone, and the enemies will be placed here too

* Fishing Area (Probably)

- Fish swimming around inside, and a mini game to catch them for food

7.2 Terrain:

* Farmers Shelter

- Soil- Flowers

* Forest Battlegrounds

- Dirt Road - Trees - Rocks - Bushes

* Fishing Area (Probably)

- Mini pool with fish swimming.

Section 8 - Collision Detection, Physics & Interaction

8.1 Collision Detection:

* This is where there will need to be detection of the intersection of two or more objects within the game
* Characters cannot walk through objects - objects must remain solid - with the exception of a few items like pickups.
* Collision Detection will be needed for when:
* Character gets hit by enemies and vice versa
* Enemy and player collides
* Enemies colliding with each other
* Enemies colliding with the environment
* Farmer house colliding with the character
* To enforce collision detection in the game:
* A\* algorithm will be used
* Unity’s physics system can be used
* Steering behaviours Collision Avoidance can possibly be used

8.2 Physics

* This is the component that makes the game have similarities to real life
* Objects and players must react to player input and player decision (movement for example)
* Must be realistic
* Friction and gravity
* Physics will be needed for:
* Character and Enemy movement
* To enforce physics in the game:
* Unity will be used

8.3 Interaction:

* This is how characters interact with the game world
* Player Interaction with objects/items:
* Interacting with enemies
* Interacting with the farm
* Interacting with forest (eg. mining trees for wood)
* Enemy interaction:
* Enemy remain following a path unless triggered by character presence/ or they do not spawn until player is near
* Once triggered, enemies will begin attacking the character until the character goes a set distance from the path.
* Once the player is out of range, the enemy will return to their path and cease to exist, or just continue on their path
* To enforce interaction in the game:
* Unity can be used with triggers and colliders

Section 9 - Game Logic & Artificial Intelligence

9.1 Game Logic:

* C# in Unity using visual studio 2019

9.2 Artificial Intelligence:

* A\* for movement, and group AI for the enemies

Section 10 - Audio

10.1 Audio Effects:

* Record sound using Audacity or free sounds from the internet.