### Technical Design Document Template

#### 1.0 Revision History

<As you revise the document, list what was changed and when it was changed>

|  |  |
| --- | --- |
| Version | Description |
| 1.0 | Initial document |

#### 2.0 Development Environment

**2.1 Game Engine**

Raylib

**2.2 IDE**

Visual Studio 2019

**2.3 Source Control procedures**

Git

**2.4 Third Party Libraries**

Raylib.dll

**2.5 Other Software**

N/A

#### 3.0 Game Overview

**3.1 Technical Goals**

- 2D top-down perspective

- Utilise basic shapes and minor sprite work

- Have an Agent seek to a point

- Have an Agent flee from another agent

- Have an Agent that attacks other agents

- Have agents that harvest resources

- Allow the player to use resources to place down different nodes

**3.2 Game Objects and Logic**

- Helper agent: The player commands these agents to perform specific tasks

\* Harvester agent: This agent is used to seek harvest nodes and collect resources

\* Fighter agent: This agent will fight when other agents get attacked, you can even set paths for them to follow.

\* Builder agent: This agent creates nodes on the players command

- Enemy agent: They attack your Helper agents

- Harvest Node: This holds resources and can be destroyed when nodes are lost, Resources are harvested at a rate.

- Pile Node: Holds resources, resources are picked up instantly to the agents max inventory

**3.3 Game Flow**

You play a tribe trying to survive in a cruel world. You can control/manage the tribe to ensure their survival, by telling the tribespeople to collect resources that are imperative for their survival such as food and water. They can also collect resources which can be used to create huts to live in to keep them cozy, but they will also need them to fend off the monsters that attack in the night. You have to manage your tribe by setting workers to harvest, guard or build in order to survive and beat the game. You lose if your tribe is overrun by the hoard, but if you manage to survive for 7 days, and beat the big boss, you will beat the game. The game is pause-able at any time so you can quit the game when ever you like. (saving option might be available at some point).

Programming stuff to be considered includes:

* Minimising code
* Minimising files size
* Commenting \*
* Optimisation
* Organising code into correct classes

<description of what the player can do (actions) from the start menu to playing the game, through to hitting quit. Include how to win, how to lose, how the player is moved, and what programmer things might need to be considered>

#### 4.0 Mechanics

CORE GAME MECHANICS:

* Recourse harvesting
* Agent control
* Agent autonomy
* Survival
  + HP
  + Thirst
  + Hunger
  + Happiness
  + Temp meter
* Defence
* Hoard combat

<A list of the core game mechanics. I.e., what the player can do and how they achieve this, and what this triggers in the game. For example, shooting enemies is a core mechanic in an FPS>

#### 5.0 Graphics

Top-down 2D, minimal sprite work, quaint simplistic style, lots of colour. Inspired by dwarf fortress

<Describe graphics features here. I.e., is your game top-down 2D? >

#### 6.0 Artificial Intelligence

**Behaviour types:**

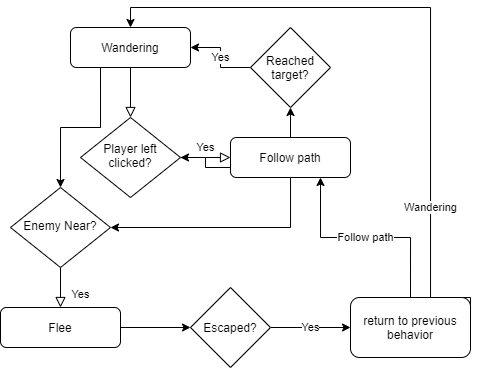
* Seeking: The agent has a target and will seek toward it
* Wandering: The target has no specific direction and will roam. (go into more detail)
* Fleeing: When approached by or approaching a target the agent will run away
* Pursuit:
* Combat:
* Harvesting:

**The Agents:**

Tribespeople:

The Tribes people will have the basic behaviours of:

* Feeing: from enemies
* Wandering: around home base
* Path following: To homes at night





The Harvesters will the addition behaviours:

* Path following: To selected harvest nodes
* Harvesting: Collecting resources from harvest nodes

The Miners will the addition behaviours:

* Path following: To selected harvest nodes
* Harvesting: Collecting resources from harvest nodes



The Warrior will the addition behaviours:

* Path following: To selected enemies
* Patrol: pathing around selected areas



The Builder will the addition behaviours:

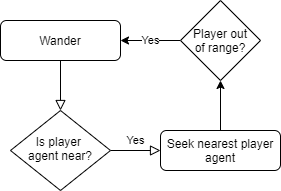
* Path following: To construction sites
* Building: Increased the construction rate of building by flat amount

The Enemy has a different behaviour to the tribesman and behaviour is revolves around sieges. When not sieging:

* Wander: aimlessly
* Pursue: tribesman

When sieging:

* Path: to base
* Pursue: tribesman





The Boss is going to be a variant of enemy that will only spawn on the last day and when it is killed will result in a win. It will only spawn during the siege so its wander behaviour won’t be utilised, but it will have unique pathing and attacking features:

* Pathing: ignores obstacles
* Attacking: does damage in a wide area

<Describe how AI works, i.e. state machine, fuzzy logic, GOAP. Describe the various behaviours and how they change behaviour, how do the ‘creatures’ in the game evaluate the world> <include diagrams/flowcharts showing decision making processes>

#### 7.0 Items/Resources

This is the following are tables of items and resources.

<List of items you can pick up that can affect the player, and what they will affect, like ‘picking up the hammer (refer collisions above) adds 5 to the players attack attribute’. Include details on how items influence gameplay or AI logic.>



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Player | Harvest | Attack | Building | Description |
| Default | 1 | 1 | 2 | The humble tribesperson, well rounded. Can, do anything |
| Axe | 5 | 3 | 0 | The Axe wielding tribesman only focuses on harvesting nodes. Their axe gives them a means of self-defence but can’t build. |
| Pickaxe | 5 | 3 | 2 | Pickaxe, Axe same thing. Just don’t get too stoned ;D |
| Sword | 2 | 7 | 0 | The Sword Gives marginally better harvesting and far better attack. Can’t build though |
| Hammer | 2 | 2 | 5 | The hammer Gives an all-around boost to stats but mostly focus on building |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Node | Resources | Storage | Tool | Description |
| Tree | Wood  Food | 20  4 | Axe/Sword  NA |  |
| Rock | Stone  Ore | 25  2 | Pickaxe |  |
| Iron | Ore  Stone | 10  2 | Pickaxe |  |
| Bush | Food  Wood | 10  1 | N/A |  |

#### 8.0 Game Flow

**8.1 ‘Mission’ / ‘Level’ structure**

<Are all levels stored in memory? what data is saved across levels, are levels loaded synchronously to prevent pauses?>

This game only has the single level in it, that level is the great plains of oodle. Oodle is home to many wonderous creatures and resources. The elements that are saved in this level are the Harvest nodes (position, resources types, tool required, sprites and refresh timers), pathing nodes (position, cost to traverse), the players starting point and the siege info which includes the numbers of enemies that spawn over the course of the game and the type of enemies it chooses to spawn.

**8.2 Objectives**

<What does the player try to accomplish on each level/mission? How is the players progress evaluated?>

Notes:

* Player collects resources
* Crafts equipment
* Creates tribespeople,
* Protects tribespeople from enemies

#### 9.0 Interface

**9.1 Menu**

<What are the menu options and what do they do?>

**9.2 Camera**

<Describe the camera, how it moves, perspective/orthographic, can it switch? How? Does it need to render-to-texture? does it prevent itself going through walls, use flowcharts to document behaviour>

The camera is a top-down camera. The player can control the camera so it moves up, down, left and right, allowing the player to explore the map as well as the option to follow the tribesman or track the enemy.

**9.3 Controls**

<Keyboard, tablet touch/swipe/tilt, joystick, mouse etc. record double taps, multi touch, use mouse smoothing/ scale mouse for aiming etc.>

The player can control the camera using the arrow-keys or WASD

#### 10.0 Asset List

<List all files needed, along with known attributes >

#### 11.0 Technical Risks

In this game I want there to be:

* Resource harvesting
* Crafting
* Building
* Agent selecting
* AI Behaviours: (seek, peruse, flee, pathfind)
* Wave system
* Agents have different behaviours at different times
* Agents have different behaviours when equipping different tools