

# Abandon

## 1. Everything comes to life!

### 1.1. Introduction

You find yourself alone in an empty world, no idea who you are and why you are here. As you reach out to feel the environment, you realise that the objects near you awaken and come chasing after you but they dropped dead and start to fade pitch black upon contact with you. The next moment you realise that the environment around you seems to dim out suddenly. In the distance, you see a shining portal...

### 1.2. Character Biography

You play as a mysterious little girl who is alone in an empty world. You have no idea why but the objects around comes alive and starts chasing after you. At the same time, you discover that the objects that have come to life will die when they touch you. You want to make more friends so you came up with a plan.

## 2. Core Gameplay

### 2.1. Core Mechanics

The core mechanics in Alone basically consist of three elements:

- **The Awakening** that makes all the objects the player touches come to life.
- **Darkness** that consumes the environment and the avatar.
- **The Portal of Light** that shines the way out of the looming darkness that threatens to consume the player's avatar.

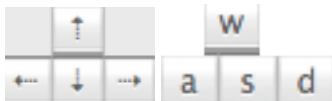
### 2.2. Avatar

#### 2.2.1 Overview

The player's Avatar has the following abilities:

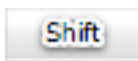
- **Move** around the environment. The avatar moves relatively to the directional button pressed by the player. The player will be able to toggle between the default running and walking for the avatar's movement.
- **Awaken** the objects in the environment. When the avatar is near to an object, the object will wake up and chase after the avatar.
- **Pushing** objects that are dead. This is a countermeasure to prevent the character from getting stuck when cornered by dead objects.
- **Shining halo** that is required before you enter the portal of light.

### 2.3. Controls



#### Directional Keys

*The avatar moves in the corresponding direction of the directional key that is pressed.*



#### Speed Modifier

*The avatar's movement changes when this button is pressed.*

#### 2.3.1 Speed Modifier

When the key "Shift" is pressed, the speed of the character will be modified, resulting in a walking action by the avatar.

#### 2.3.2 Awaken

When the player moves into the detection range of an object, the object will be awakened.

#### 2.3.3 Push

All objects in the game world are push-able. This is a counter-measure to get the player out of situation whereby the player will be stuck in a corner by objects that he has killed.

## **2.4. Player's Objective**

The player's objective is to escape to the portal before the darkness fills the environment.

### **2.4.1 Shining Halo**

The player requires the shining halo before access is granted into the portal of light. On certain levels, the player may be required to search for their missing halo before proceeding to the portal.

### **2.4.2 Game End State**

The level will end when the environment and the player's avatar is consumed by darkness.

### **2.4.3 Light and Dark**

The game level starts with an initial amount of light that allows the player to see the environment. This amount of light decreases each time an object touches the avatar and fades into black death, a certain amount of light will be taken away from the avatar

## 3. Environment

### 3.1. Exit Portal

The main objective of the player is to get to the exit portal. The exit portal is sometimes locked and has to be triggered before the player can enter the portal.

### 3.2. Portal Trigger

The portal trigger is a single activation mechanism that allows the player to unlock the exit portal.

### 3.3. Objects Types

Objects are placed all around the game environment that reacts to the player. Different objects will have varied behaviour based on the following types:

#### 3.3.1 Regular Objects

These objects will:

- Chase the player after they are awakened
- They have varying chasing speeds
- Take away a moderate amount of light from the environment when they die

#### 3.3.2 Deadly Objects

These objects will:

- Take away all your light instantly and causing instant loss to the game

#### 3.3.3 Do-not-run-pass-me Objects

These objects will:

- Wake up instantly when they are awoken
- Player will not be able to run pass the objects without getting their light taken away

#### 3.3.4 “Fat” Objects

These objects will:

- Be too “fat” to fit in through tight corridors
- Act as clogging objects at choking points in the map

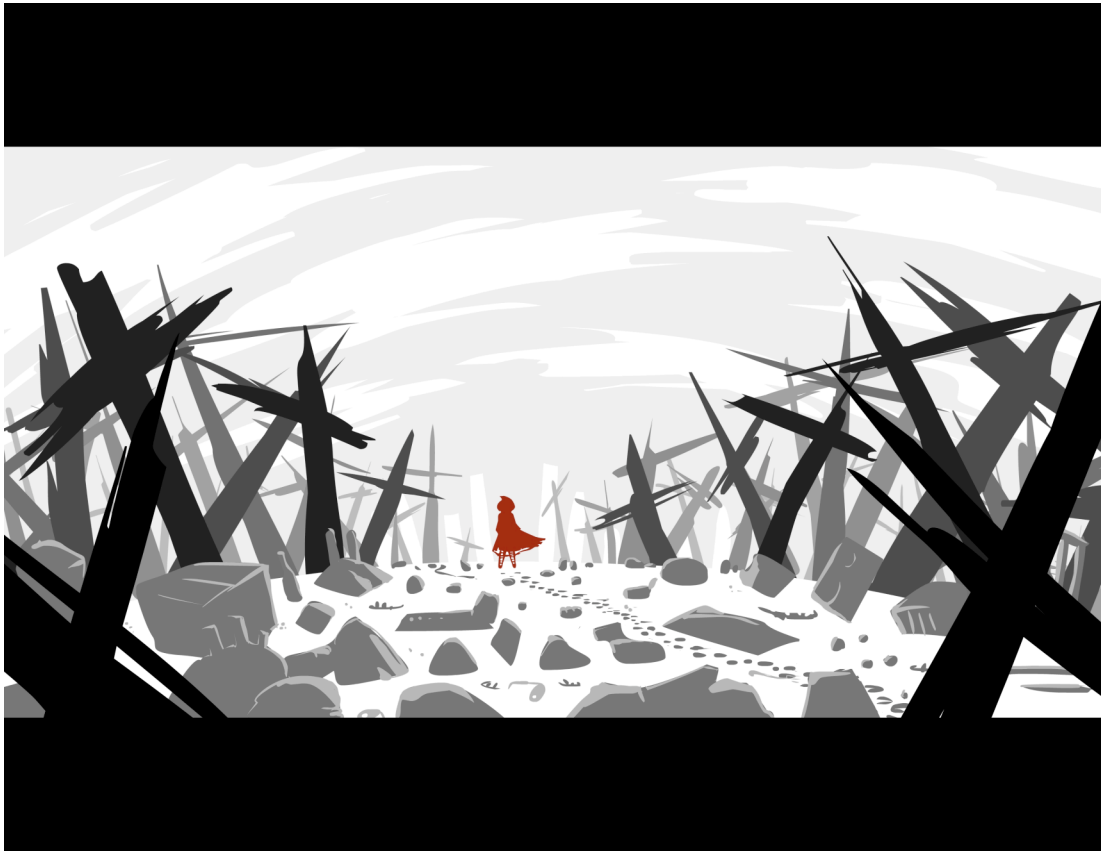
#### 3.3.5 Object End State

Objects will die and fade to black after they come into contact with the player, taking away some of the light of the environment with them.

## 4. Artistic style outline

### 4.1. Setting

A nightmarish world that is void of life with the little girl alone in this world.



#### 4.1.1 3D World environment

### 4.2. Character

#### 4.2.1 The Little Girl

A little girl that is alone in an empty world. She has the ability of giving life but also causing death. She wants to make friends by using her ability to bring objects to life but was unable to stay near them, as she would render them dead.



The Little Girl

#### 4.2.2 3D Character model

### 4.3. Tile-based Level

With the tile-based level editor, artists are required to create the tiles that make up the level. 3D environment assets have to be created with the tile-based level editor in mind. Walls and obstacles form part of the levels and they have to be distinguished from the objects that can be brought to life.

#### 4.3.1 Wall tiles

This set of tiles forms the walls of the game environment. They have to be very distinctive from the 3D object models so that the player does not confuse them with the objects that are interactive.

#### 4.3.2 Floor tiles

This set of tiles forms the floor of the game environment, which depicts where the player's avatar and the objects will be able to move around in. The floor tiles are also important to contribute pathfinding data for the objects to find their path to the player's avatar.

### 4.4. Objects

Objects in the game are everyday common objects that can be found in homes, offices and on the streets. These objects are to be distinguishable from the environment to allow the player to know that they are interactive.



Everyday Objects

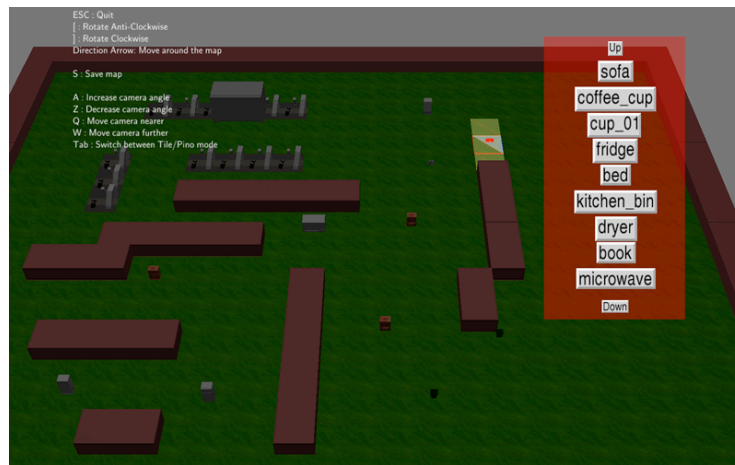
## 5. Systematic breakdown of components

### 5.1. 3D Renderer with Pinocchio integrated

This allows the artists to input their art assets into the framework of the game system with the auto-rigging system "Pinocchio" integrated to the assets.

### 5.2. Level editor

This allows the game designer to design levels and able to make changes to the levels easily. Objects can also be placed on the level with the level editor.



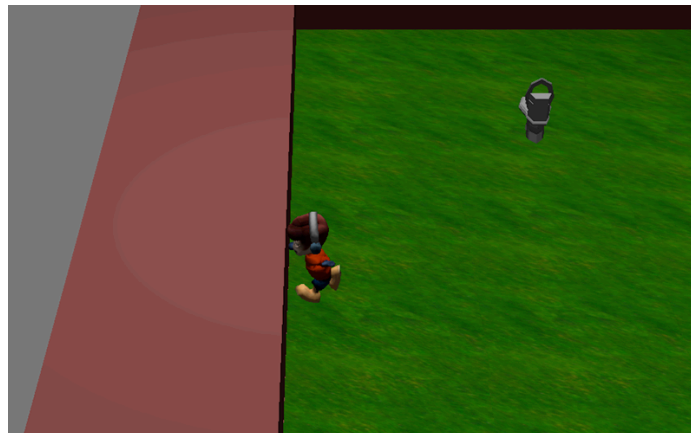
Level Editor

### 5.3. Collision system

A collision system is required for interaction between the player and the objects and obstacles in the game environment.

#### 5.3.1 Obstacles/ Walls

The player avatar and objects in the game world will not be able to pass through these obstacles and walls. The player will not be able to bring these obstacles to life.

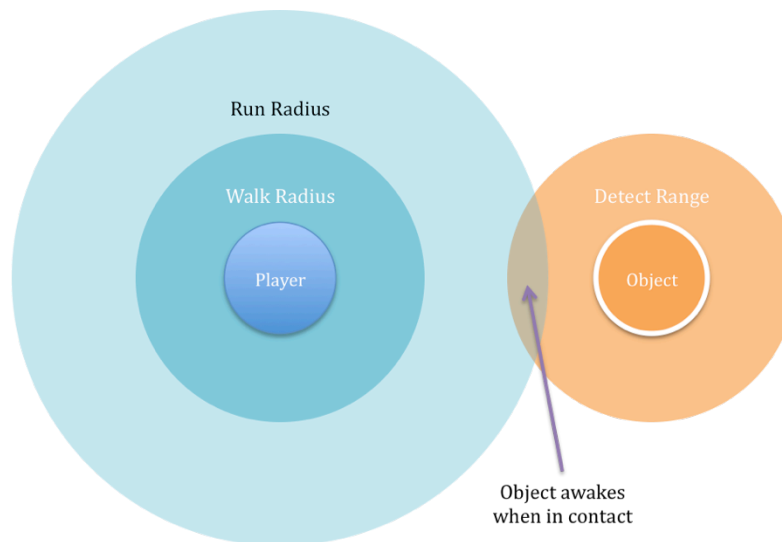


Wall Collision

### 5.3.2 Player

The player will be able to move around the world but not able to move past walls and obstacles. When the player goes near to objects in the environment, objects will awaken and starts to chase after the player.

#### 5.3.2.1 Player Awaken Mechanic Graph

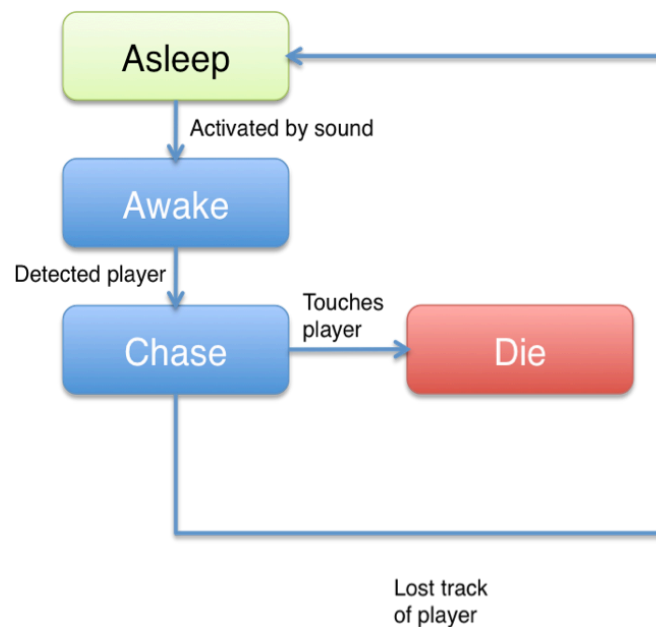


### 5.3.3 Objects

Objects in the environment are initially asleep and when they detect the player, they will awaken and start to chase after the player.

The diagram below will illustrate the different states the object will be in.

#### 5.3.3.1 Objects State Machine





## **5.4. Pathfinding algorithm**

The pathfinding algorithm allows the objects to actively search for the position of the player's avatar and chase after the player.

### **5.4.1 Flocking algorithm**

To achieve greater level of interactivity and challenges to the player, the flocking algorithm allows the objects to search for possible alternate routes towards the player's avatar to outflank the player.

## **5.5. Library integration system**

This system allows the easy implementation and addition of the various assets required for the game.

### **5.5.1 Art assets library**

Various different art assets are required for the game to load. The art assets library allows artists to update and add more art assets into the game easily.

The various different art assets are as follows:

- 3D Objects models
- Terrain tiles for the level editor
- User Interface screens

### **5.5.2 Audio assets library**

Allows audio assets to be stored in a library ready for the game to load as and when required while playing the game.

The two types of audio assets required for the game are:

- BGM
- SFX

### **5.5.3 Level list**

This list allows the game to load levels created using the level editor. Levels can easily be added or removed from the list.

## **5.6. Camera interface**

A camera interface that allow the artists and designer to try out different camera angles and distances in the game. This also allows possible camera views toggle to be available for the player to invoke varied game experience.

## 6. Suggested Game Flow Diagram

