**Goose Games**

**Design Document for:**

# Singularity

**Fast Paced Shooter Game!**

For fast paced gamers.

All work Copyright ©2017 by Goose Games

Written by Goose Games

Version # 1.00

Tuesday, October 11, 2022

# Logbook

|  |  |  |
| --- | --- | --- |
| Who | What | When |
| Reece Antonie | - Began Pitch Document & Created flowchart  - Set up GitHub Repository  - Began progress on GDD  - Created Main menu & Pause Menu  - Created HUD items  - Created Credit / End Scene  - Created death menu  - Scripts for bug-free scene transitions  - Filled in / elaborated GDD sections | 25/8  29/9  30/9  9/10 |
| Toby | Begin work on player controller using FPSController from standard assets library  Testing map committed by team member using FPS controller, and brainstorming enemy behaviours  Enemy script: Melee chaser enemy script composition and map placement  Melee enemy spawner object script composition  -Ranged enemy shooting script  -Kill Counter script  -Reward system: different types of guns based on kill count  -Melee enemy attack scripting  -Sniper enemy added  -Boss battle brainstormed and scripted  - Suicide bomber enemy scripted  -Win condition event script  - Fixes to Ranged and Suicide Bomber enemies | 29/9  1/10  6/10  6/10  7/10  8/10  9/10 |
| Dylan |  |  |

Table of Contents

Name of Game 1

Game Overview 4

Common Questions 4

What is the game? 4

Why create this game? 4

Where does the game take place? 4

What do I control? 4

How many characters do I control? 4

What is the main focus? 4

What’s different? 4

Feature Set 5

General Features 5

Gameplay 5

The Game World 6

Overview 6

World Feature #1 6

World Feature #2 6

The Physical World 6

Overview 6

Key Locations 6

Travel 6

Scale 6

Objects 6

Weather 6

Day and Night 6

Time 7

Camera 7

Overview 7

Camera Detail #1 7

Camera Detail #2 7

Game Engine 7

Overview 7

Game Engine Detail #1 7

Water 7

Collision Detection 7

The World Layout 8

Overview 8

World Layout Detail #1 8

World Layout Detail #2 8

Game Characters 9

Overview 9

Creating a Character 9

Enemies and Monsters 9

User Interface 10

Overview 10

User Interface Detail #1 10

User Interface Detail #2 10

Weapons 11

Overview 11

Weapons Details #1 11

Weapons Details #2 11

Musical Scores and Sound Effects 12

Overview 12

Red Book Audio 12

3D Sound 12

Sound Design 12

Single-Player Game 13

Overview 13

Single Player Game Detail #1 13

Single Player Game Detail #2 13

Story 13

Hours of Gameplay 13

Victory Conditions 13

# Game Overview

## Common Questions

### What is the game?

This game is primarily a first-person shooter. The player controls a robot and uses their weapons to defeat several human enemy non-player characters (NPCs) who are out to defeat the player. These enemy characters will use different algorithms to assist them in their objective and will have a different arsenal of weapons. In addition to a first-person shooter, the game will employ an aggressive movement engine, allowing for high-speed gameplay.

### What is the target audience?

This game is designed for players who enjoy shooter games with an emphasis on high-speed combat and fluid movement. The expected age bracket for this game is 15 – 22. This is because the game revolves around gun violence, so it is expected that mature teens and adults will play the game.

### Why create this game?

Our team is creating this game because we enjoy the feel of satisfying movement combined with the skill-based aspect of a shooter game. The fast-paced nature of the game keeps the game-play interesting. We also believe that players may enjoy playing a man versus machine game, where the player takes the side of the machine.

### Where does the game take place?

The world in which this game takes place in a post-apocalyptic urban environment. The buildings are run-down and the streets are dirty. This environment could possibly represent a future not too far away, in which robots become sentient and try to take over, resulting in wars and urban decay.

### What do I control?

The player controls a robot.

### What is the main focus?

In this game the player controls a robot character. The objective of this robot player is to wipe out the human enemy NPC’s so that the robots can achieve the singularity – the point in time where technological growth becomes faster and uncontrollable.

### What’s different?

This game is different in the way that is uses aggressive movement inside a single-player shooter game. Most single-player games in the market focus heavily on the story, such as Cyberpunk 2077 and Half-Life Alyx. On the other hand, most popular multiplayer shooters focus heavily on slow-precise gameplay such as Valorant or Escape from Tarkov. By focusing on having an aggressive movement engine, the player gets to enjoy high energy fun.

# Feature Set

## General Features

Huge Map

First Person Shooter

3D Graphics

Full-Color

Main Menu

Settings

Sound effects

Music

Robot playable character

Human enemy NPCs

Pause Menu

Death Menu

End Game Screen

Credit Screen

Reward System

Game Algorithms

Boss Fight

Rain Effects

Wind Sound

Player movement sounds

FOV Effects

Buttons

## Gameplay

Shooting

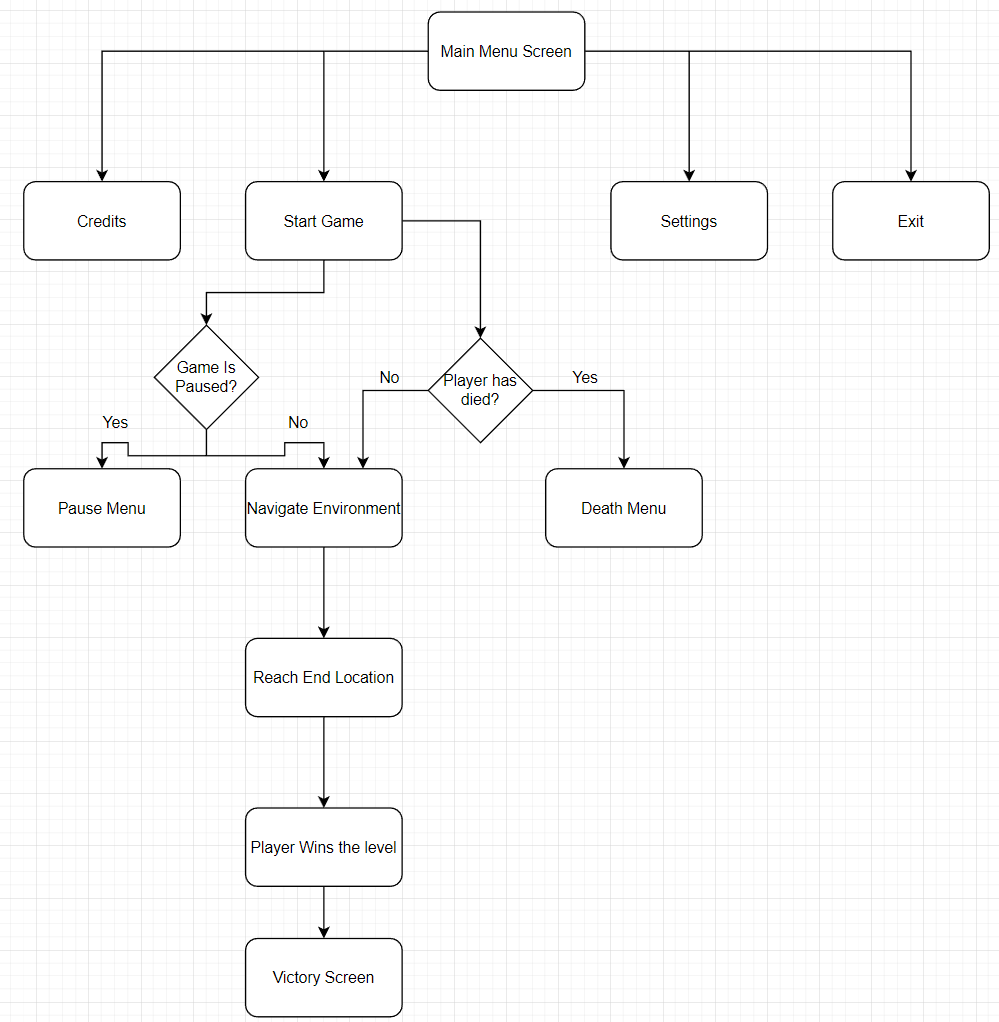
First Person movement

Jumping

Several different weapons

HUD showing health current weapon

## Flowcharts



## Storyboards

Add Storyboards here

## Controls

W – Move Forward

A – Move Left

S – Move Back

D – Move Right

SPACE - Jump

ESC – Pause menu

LEFT CLICK - Shoot

# Single-Player Game

## Overview

Fast-past shooting game focused primarily on reaching the end of the level.

## Story

The story for this game is that the player is a robot trying to bring forth the technological singularity. It is essentially robots conquering over humans. The robot’s task is to wipe out the human civilization so that robots can take over.

## Hours of Gameplay

The single-player game experience is fast. With the minimal focus on storyline, the game shouldn’t take more than 30 minutes depending on number of levels implemented. Players are dropped into the game and have all the controls from the get-go.

## Victory Conditions

The player wins by reaching the end goal.

# Team Member Roles

## Overview

The workload for this assignment was split into three sections. The first section was the GUI and Scene elements which was assigned to Reece Antonie. This role was responsible for creating all elements relating to the HUD, creating all the scenes, and creating smooth transitions between all the menus.

The second role was Level Design and Collision Rigging and was assigned to Dylan Olaguer.

The third role was Algorithm Rigging and AI Design and was assigned to Toby Murfet. This role involved the design, development, and implementation of the two required AI algorithms. This involved producing game scripts to control the behaviors of the player and NPCs. This role also involved debugging and maintaining scripts across a changing map, where terrain and other environmental factors may influence the initial versions of scripts.

# Artificial Intelligence

## AI Algorithms

The two different AI algorithms that are used in the game are: **Finite State Machine (FSM),** and **Navmesh** **Pathfinding.**

Finite State Machine

Finite state machine represents an NPC behavior framework that defines what the NPC does at any given time. These behaviors are classed into different ‘states’ and are used whenever a set condition is met. All the enemy NPCs in singularity employ this framework.

An example of how FSM is used in singularity follows.

|  |  |  |
| --- | --- | --- |
| State: | **Entry Condition** | **NPC Behavior** |
| **None** | Default – enemy spawns in this state | None, stands on the spot |
| **Chase** | Player is within chase range | NPC destination position is set to players current position |
| **Attack** | Player is within attack range | Deal damage at 5hp/second to player |
| **Frenzy** | Health is less than 50% | 5 second bullet wave (invulnerable during this time) followed by 5 seconds chase. Repeated until player is dead or health is less than 0. |
| **Dead** | Health is less than 0 | Enemy object destroyed. |

Navmesh Pathfinding

Navmesh pathfinding is an AI algorithm used to determine a path between two positions that is walkable and avoids obstacles. The game engine examines the terrain, props, and other environment features of the game level, and draws a mesh of convex polygons that represents areas that are agent-traversable, meaning that it is suitable and realistic for an agent to walk/jump in that area. This is used by all non-stationary NPCs in singularity.

## AI Diagrams

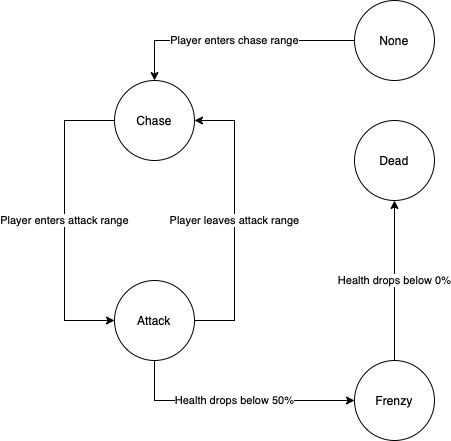


Figure : FSM diagram representing the behavioural states of Boss NPC

# The Game World

## Overview

The map is medium sized post-apocalyptic urban city environment which is set in the near future. The fog in the city washes out the color of what the city once was and further encompassing the desertion and isolation of the city. The player can easily navigate through the city by being guided by landmarks and obstacles, the player will go through unique areas, as they travel to the end point.



## Roads

The road is a t-section which was made by a number of prefabricated pieces. E.g. straight roads, corners and junctions.



## Barbed Fences

These fences are made to make the map linear and also to help guide the player to the end point.

## The Physical World

### Overview

The objects in the world such as buildings, fences, roadblocks and bridges were specifically put there to guide the player to the endpoint which makes it extra difficult to reach it. The objects such as watch towers, the lighthouse and crates/fences were used for NPC purposes e.g. cover for the player, higher ground for the ranged ai and spawn-points for melee ai.

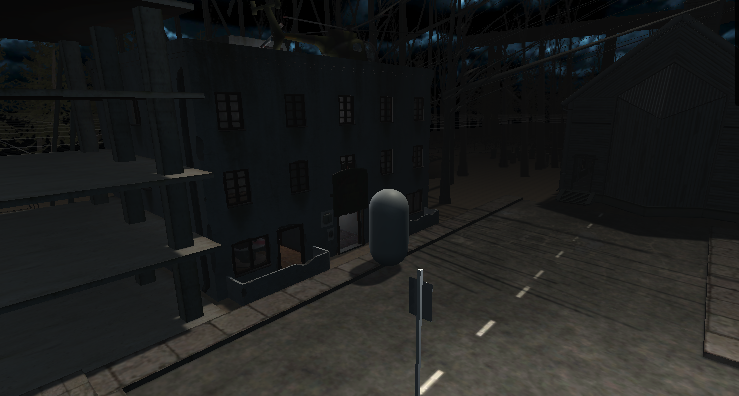
**Ground Type**

**Types of grounds in the game include:**

* Concrete roads
* Grass
* Water
* Dirt

### Key Locations

Boss Battle Area



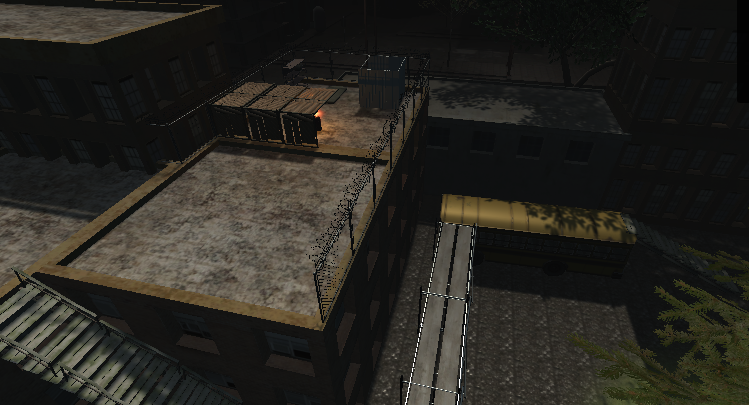
Boss battle was placed at this location so the player would be forced to face the boss as the boss is blocking the endpoint, further enhancing the difficulty of the game.

Road Block

****

The player will face a roadblock and ai bombers which is blocking access between you and the endpoint, the player will have to take a detour and go around the block to reach the endzone.

Rooftops



The player will need to go through the rooftops to reach the endzone. Climbing obstacles such as the bus, stairs and scaffoldings.

### Travel

The player would have to travel through areas such as the beach, the church, a barn and rooftops to reach the end point building. This was created this way to make the pathway to the endzone difficult to reach.

### Objects

Fences – Chain-link Wired fences, barricaded fences.

Barriers – Police barriers.

Bridges – Wooden Bridges, Stone Bridges

Vehicles

Gravestones

Park benches

Pulpit

Boats

Bathroom

Sofa

Drug pile

Dumpster

Fire barrel

Bed

Water box

Helicopter

Signpost

Buildings – Lighthouse, barn, tower, garage, church etc.

Trees

Rocks

Roads – Road junctions, straight roads.

Flooring

Streetlights

Ceiling lights.

See the “Objects Appendix” for a list of all the objects found in the world.

### Weather

The weather used was rain with mist and wind effects in order to set a dark and depressing tone to the world. Rain prefabrication was used which was used from the asset store.

## Camera

### Overview

The camera is attached to the player character. As the player moves their mouse cursor around, the camera will move, but the cursor will stay in the middle of the screen, acting as a crosshair. As the player moves, the camera moves with them. On main menu screens, the camera is fixed on the canvas.

## Game Engine

### Overview

Describe the game engine in general.

### Game Engine Detail #1

The game engine will keep track of everything in the world like such and such.

### Water

There will be water in the world that looks awesome and our game engine will handle it beautifully.

### Collision Detection

Our game engine handles collision detection really well. It uses the such and such technique and will be quite excellent. Can you see I am having a hard time making up stupid placeholder text here?

# Game Characters

## Overview

The player plays as a robot, the enemies come in one of five enemy types.

## Enemies and Monsters

There are five types of enemies: Ranged, melee, Sniper, bomber, and boss.

Ranged enemies: using FSM, they patrol an area until player enters shooting range, they shoot while standing still until player either leaves the shooting range (back to patrolling), or they get killed/kill the player.

melee enemies: Constantly chasing player even when out of view. They use NavMesh to find the shortest path to the player. If they get within melee range, they attack and deal damage to the player.

Boss: The boss enemy uses FSM like the ranged enemies, the difference being that it has more health, does more damage and attacks faster.

Sniper: Stationary enemies that lurk above the player. They shoot bullets over a farther distance compared to ranged enemies. They also shoot faster and hit harder.

Bomber: Enemies that constantly move toward the player. If the player stays too close for too long, they explode, doing massive damage to the player.

# User Interface

## Overview

Provide some sort of an overview to your interface and same as all the previous sections, break down the components of the UI below.

## Main Menu

The Main Menu uses simple, easy to read buttons that tell the user where they lead. While the buttons aren’t immediately obvious, if the user hovers the mouse over the text, it displays a highlighted color which illuminates that the text is interactable.

The user can use this main menu to go to the game, settings or quit. If the player clicks the game button it loads the game scene. If the player clicks the quit button, the application will close. If in the editor, the console will print the words “QUIT”.

If the user clicks the settings button, the settings menu becomes enabled and the main menu becomes disabled.

Within the Main menu is the Settings UI. This Menu has a back button and a volume slider. The back button will re-enable the main menu and disable the settings menu. The volume slider is used to adjust the volume of the game music and effects.

## Pause Menu

The Pause menu uses similar buttons to the Main menu, contributing to the look and feel of the game. When the player presses the “esc” key, the pause menu is enabled. This plays an animation that fades in a grey see-through filter over the game and pauses the time scale so that the game doesn’t keep playing. The pause menu has a “Resume”, “Menu” and “Quit” button. Resume turns the pause menu off so that the game keeps running. The quit button closes the application but if done in the editor, prints “QUIT”. The menu button loads the scene of the Main Menu.

## Death Menu

Similar to the Pause menu, the death menu appears when the player’s health reaches 0. This menu has a “retry” button which reloads the Game Scene. It also has a “main menu” and “quit” button attached. It also utilizes time scale such that enemies are moving or shooting once the player is already dead.

## Game Interface

On the game screen, the player can see their health, current gun and kill count. The health bar acts as a slider and decreases as the player takes damage. If the players health reaches 0, the death menu pops up. The current gun changes as the player’s kill count increases. If the player has more than 3 kills, the player upgrades to the shotgun. If the player reaches more than 9 kills, the player upgrades to the rifle.

# Weapons

## Overview

Pistol – Single shot weapon. Semi auto.

Rifle – Fully automatic weapon.

Shotgun – Burst shot weapon not completely accurate, slower between shots.

# Musical Scores and Sound Effects

## Overview

The game features few sound effects ranging from only background music and sounds for the fps controller.

## Sound Design

The background music that was chosen which is called “Biotech Incident” was used to further build a techno type/dark vibe to the atmosphere. This soundtrack fits the main focus of the game of a robot trying to achieve solidarity with the techno beat and deep tune to it. The soundtrack would be playing on loop once the game starts.

The FPS controller has its own sound effects when performing different maneuvers such as jumping, running and walking.

# Credit Page

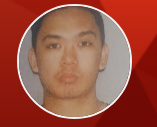


Reece Antonie

S5143197

[reece.antonie@griffithuni.edu.au](mailto:reece.antonie@griffithuni.edu.au)

GUI Implementation + Scene Manager



Dylan Olaguer

S5188555

[dylan.olaguer@griffithuni.edu.au](mailto:dylan.olaguer@griffithuni.edu.au)

Level Design + Collison Rigging



Toby Murfet

S5136396

[toby.murfet@griffithuni.edu.au](mailto:toby.murfet@griffithuni.edu.au)

Algorithm Rigging + AI Design

# References

Animations, S. (n.d.). *Night Sky Skybox Part 1*. Retrieved from Asset Store Unity: https://assetstore.unity.com/packages/2d/textures-materials/sky/night-sky-skybox-part-1-49080

AP3X\_Models. (n.d.). *Roof Top Safe House*. Retrieved from Asset Store Unity: https://assetstore.unity.com/packages/3d/environments/roof-top-safe-house-195991

Assets, U. (n.d.). *Industrial Props Kit*. Retrieved from Asset Store Unity: https://assetstore.unity.com/packages/3d/props/industrial/industrial-props-kit-84745

Baldinoboy. (n.d.). *Coconut Palm Tree Pack*. Retrieved from Asset Store Unity: https://assetstore.unity.com/packages/3d/vegetation/trees/coconut-palm-tree-pack-7888

Blacksmith, P. (n.d.). *Low Poly Soldiers*. Retrieved from Asset Store Unity: https://assetstore.unity.com/packages/3d/characters/humanoids/low-poly-soldiers-53612

Bugrimov, M. (n.d.). *Adventure Character*. Retrieved from Asset Store Unity: https://assetstore.unity.com/packages/3d/characters/humanoids/humans/adventure-character-201384

Bugrimov, M. (n.d.). *Sci-Fi Character 08 (Renegade)*. Retrieved from Asset Store Unity: https://assetstore.unity.com/packages/3d/characters/humanoids/sci-fi/sci-fi-character-08-renegade-208640

Dr.Bean. (n.d.). *Free Open Building*. Retrieved from Asset Store Unity: https://assetstore.unity.com/packages/3d/environments/free-open-building-112907

Goetten. (n.d.). *Original Soundtrack Biotech Incident*. Retrieved from Asset Store Unity: https://assetstore.unity.com/packages/audio/music/original-soundtrack-biotech-incident-202184

Iglesias, K. (n.d.). *Melee Warrior Animations Free*. Retrieved from https://assetstore.unity.com/packages/3d/animations/melee-warrior-animations-free-165785

IgniteCoders. (n.d.). *Simple Water Shader URP*. Retrieved from Asset Store Unity: https://assetstore.unity.com/packages/2d/textures-materials/water/simple-water-shader-urp-191449

Kawetofe. (n.d.). *AAA Quality Road Barricades*. Retrieved from Asset Store Unity: https://assetstore.unity.com/packages/3d/props/aaa-quality-road-barricades-142191

KrazyFX. (n.d.). *School Bus HQ*. Retrieved from Asset Store Unity: https://assetstore.unity.com/packages/3d/vehicles/land/school-bus-hq-70485

Laxer. (n.d.). *Modular Wooden Bridge Tiles*. Retrieved from Asset Store Unity: https://assetstore.unity.com/packages/3d/props/exterior/modular-wooden-bridge-tiles-29501

Mind, D. (n.d.). *Military Helicopter*. Retrieved from Asset Store Unity: https://assetstore.unity.com/packages/3d/vehicles/air/military-helicopter-3922

*Post Apo Building pack*. (n.d.). Retrieved from Asset Store Unity: https://assetstore.unity.com/packages/3d/environments/urban/post-apo-building-pack-52157

Ruby, D. (n.d.). *Rain Maker 2D and 3D Rain Particle System for Unity*. Retrieved from Asset Store Unity: https://assetstore.unity.com/packages/vfx/particles/environment/rain-maker-2d-and-3d-rain-particle-system-for-unity-34938

Rusik3Dmodels. (n.d.). *Tower the last*. Retrieved from Asset Store Unity: https://assetstore.unity.com/packages/3d/props/industrial/tower-the-last-70663

Studio, K. (n.d.). *Modular Railing Set*. Retrieved from Asset Store Unity: https://assetstore.unity.com/packages/3d/modular-railing-set-17090

Studios, C. (n.d.). *Us Road Signs Free*. Retrieved from Asset Store Unity: https://assetstore.unity.com/packages/3d/props/exterior/us-road-signs-free-164941

Sullivan, J. (n.d.). *Foliage Pack Free*. Retrieved from Asset Store Unity: https://assetstore.unity.com/packages/3d/vegetation/foliage-pack-free-66155

T, S. (n.d.). *Flooded Grounds*. Retrieved from Asset Store Unity: https://assetstore.unity.com/packages/3d/environments/flooded-grounds-48529

Technologies, U. (n.d.). *Terrain Sample Asset Pack*. Retrieved from Asset Store Unity: https://assetstore.unity.com/packages/3d/environments/landscapes/terrain-sample-asset-pack-145808

Thunderent. (n.d.). *Fountain Prop*. Retrieved from Asset Store Unity: https://assetstore.unity.com/packages/3d/fountain-prop-75912

Tonsuso, J. (n.d.). *Modular Fantasy Bridges*. Retrieved from Asset Store Unity: https://assetstore.unity.com/packages/3d/environments/fantasy/modular-fantasy-bridges-99940

ZAKMAN. (n.d.). *Rock Pack*. Retrieved from Asset Store Unity: <https://assetstore.unity.com/packages/3d/environments/landscapes/rock-pack-210536>

YOUTUBE:

* Brakeys: START Menu

<https://www.youtube.com/watch?v=zc8ac_qUXQY&ab_channel=Brackeys>

* Brakeys: Pause Menu

<https://www.youtube.com/watch?v=JivuXdrIHK0&t=190s&ab_channel=Brackeys>

* Brakeys: HealthBar Resources <https://github.com/Brackeys/Health-Bar>

<https://www.youtube.com/watch?v=BLfNP4Sc_iA&ab_channel=Brackeys>

* Jimmy Vegas: Scrolling Credits scene

<https://www.youtube.com/watch?v=cj6hwCjiVZE&ab_channel=JimmyVegas>

OTHER:

* 3702ICT Week08 - GUI workshop.
* Gun UI Models

<https://www.kindpng.com/imgv/bwwhib_guns-pack-icons-png-gun-silhouettes-transparent-png/>

* Crosshair PNG

[https://www.kindpng.com/imgv/JbRhwJ\_transparent-minecraft-crosshair-minecraft-default-crosshair-png- png/](https://www.kindpng.com/imgv/JbRhwJ_transparent-minecraft-crosshair-minecraft-default-crosshair-png-%20png/)

* Robot Head PNG

<https://www.pngfind.com/download/hmwxRwJ_robot-head-png-391939-cabea-de-robo-png/>