OMNI

GUUDO

Technical Design Document

Version 3.0

Last Edited by Richard Delamore
21/09/20

CONTENTS

1.0 REVISION HISTORY

Version	Description
1.0	Initial document.
2.0	Removed a Team Member. Redesigned the Game Logic. Changed the Controls. Updated the Mechanics, Systems and Graphics. Updated File and Programming Naming Standards. Updated Technical Risks.

2.0 TEAM MEMBERS

The following is a list of the Guudo Team Members:

Mitchell Sayer	Richard Delamore	Tiana Knobloch	Joshua Freeman	Daniel Iaria
Game Designer	Game Programmer	Character/ UI Artist	Environmental Artist	Prop/ Technical Artist

3.0 DEVELOPMENT ENVIRONMENT

The Development Environment can be broken down into the following subsections:

3.1 Game Engine

We are using Unreal Engine 4.23.1

There were only three possible choices when it came to choose an Engine. These choices were Unity, Unreal and my custom Engine. I decided to use Unreal Engine due to the following reasons:

Shaders

Unreal has an amazing visual shader system that enables artists to take full ownership of the shader system without requiring much input from the programmer. Unity has a strong shader system, but this system is very restrictive and requires a deep understanding of HSL shader code. Additionally, our Technical Artist, Daniel Iaria is already highly proficient in Unreal Shader shaders, so we wanted to tap into his skills. Finally, our game is very shader intensive and will need to affect the geometry of our main character, especially as he interacts with his environment.

Particles

Both Unreal and Unity have powerful particle systems, but experience tells me that Unreal's particle system is a lot stronger and has a lot more functionality. It is also visually easier to understand. Again, this enables our artists to take full ownership of the particle system without requiring much input from the programmer.

UI Elements

Unreal's UI system is a breeze to work with. In essence, it's a drag-and-drop system and can easily tap into other subsystems without much hassle. Additionally, I can also easily lock it down so the artists can work on the front end without accidently affecting the underlying logic. Considering that a major part of our game is character customizer, I wanted to make this system as logical as possible.

Physics

Unreal's physics engine is way better than that of Unity's. Our character needs the ability to grow and shrink to access different areas of the map. Their physical form also needs to morph as they interact with their environment. Considering our limited time constraints, I didn't want to develop a physics system from scratch.

3.2 IDE

We are using Visual Studio 2019.

3.3 Source Control

We are using Gitkraken with Github (URL: https://github.com/Evildea/Omni)

3.4 Source Control Procedures and Work Flow

We maintain two separate branches:

Master Branch

This is the Production branch, but it's also used by the Artists and Designer. Originally, we planned to keep the Production branch separate to the artists branch, but our Designer works very closely with the artists and they are constantly updating the assets (at least a few times a day) and these changes don't have the possibility of breaking the game, so it seemed pointless to have a separate branch for them.

Development Branch

This will be used exclusively by the programmer to develop new features. The programmer often needs to develop and test new features that require a few days to develop and test. These features can be highly unstable and crash the game. It was important that the Programmer be quarantined from the Master Branch until features were ready to be pushed out.

3.5 Source Control Naming Conventions

The following conventions are used for Source Control.

Summary

A summary must be prefixed with one of several of the following tags:

- Added
- Modified
- Removed

For example:

Modified: Guudo BP

Description

A description must be provided.

- For added, it should describe what it is and where it's expected to be used/placed in the project.
- For modified, it should describe what has been changed and if it still needs further editing.
- For removed, it should state what asset was removed and why it has been replaced, state what it has been replaced with.

3.6 Third Party Libraries

We aren’t currently using or planning to use any third-party libraries.

3.7 Asset Store or Market Place Content

No Asset Store or Market Place content is being used.

3.8 Other Software

The following external software packages will also be used:

- Gitkraken (Source control)
- Maya (3D Modelling)
- Substance Painter (Texturing)
- Substance Designer (Procedural texturing)
- ZBrush (High poly modelling and sculpture)
- Photoshop (Texture editing and concepting)
- Marmoset Toolbag 3.0 (Texture baking)
- Speed Tree (Procedural trees)
- Unreal (Game engine)
- Visual Studio (Programming)
- Microsoft Office Suite (Documentation)

4.0 GAME OVERVIEW

4.1 Genre

At its core, Guudo is a sci-fi, puzzle action game. The player is tasked with navigating a movie set environment and collecting specific objects to help them build a 'human body' that fits a specific theme. This is a fast-paced game as the player needs to race against the clock to get on set on time to receive a score displaying how well they did during that run.

4.2 Perspective

The entire game operates using a 3rd person camera.

4.3 Intended Platform

We’re developing for PC because we don’t have access to a Development kit for console. Additionally, this game wouldn’t work well on console and we wouldn’t be able to incorporate the gel features we’re aiming to develop whilst maintaining a smooth framerate.

4.4 Platform Specific Development Processes

We're only developing for PC.

4.5 Technical Goals and Features

The following technical goals have been established:

Gel body

The Guudo character has a gel / goo-like physics-based body that interacts with his environment and movements to create a bouncy, gello non-solid feeling. His body also expands, shrinks and wobbles as he pickups up objects all falls from high locations giving a sense of realism.

Character customizer

The character customizer must be both intuitive and easy for the designer to update with new models.

Smooth Framerate

A framerate of at least 60 fps must be maintained.

4.6 Recommended specs to run the game

The recommended specs for running the game are:

Operating System: Windows 10 64-bit

Processor: Quad-core Intel or AMD, 2.5 Ghz or faster

Memory: 8 GB

Graphics card: Direct X 11 or DirectX 12 compatible graphics card

Space: 850 MB

4.7 Game Objects and Logic

The following is a list of the most important Game Objects:

4.7.1 GENERAL

Map Item Manager (MapItemManager)

The Map Item Manager is responsible for scanning the map and recording a list of all the pickups available. This object is read only and is accessed by the Guudo Game Instance and the Guudo Character.

Guudo Game Mode (AGuudoGameMode)

The Guudo Game Mode is responsible for managing the in-game timer and eventual transition to the Character Customisation Screen.

Guudo Game Instance (UGuudoGameInstance)

The Guudo Game Instance holds the Player's inventory. This object is persistent and isn't destroyed between levels. The Game Instance is accessed and managed by multiple different objects such as the Guudo Character, Silhouette Widget, Customisation Widget and more.

4.7.2 GAME PLAY

Guudo Character (AGuudoCharater)

The Guudo Character is controlled by the Player controller. It can move, jump, interact, consume, grow and shrink.

Switch (ASwitch)

The switch can be activated or deactivated by the Player, thus activating something else like a door.

Push Plate (APushPlate)

The push plate can be walked on by the Player. It then propels the Player along a set path to a set destination.

Pressure Plate (APressurePlate)

The Pressure Plate can be activated by a heavy object, thus activating something else like a door.

Ball (AActor)

The Ball is essentially a boulder that can be pushed around the map and onto a pressure plate to activate it.

Door (ADoor)

The Door blocks a region of the map but can be opened through the activation of one or many other objects.

Pickup (Actor with UPickup)

A pickup can be collected by the Player and then used on the Customisation Screen to customise the final Guudo character.

4.7.3 HUD

Silhouette Widget (USilhouetteWidget)

The Silhouette Widget displays a silhouette of the character with the desired pickups to the Player's HUD.

Inventory Widget (UInventoryWidget)

A UI widget that represents all the items in the Player's inventory. This can be displayed or hidden by pressing the "Tab" key at any time.

4.7.4 CUSTOMISATION SCREEN

Modular Character (AModularCharacter)

The Modular Character is a Guudo-style mannequin whose appearance can be customised through the selection of inventory items. The objective is to make the Modular Character look like the silhouette.

Customisation Widget (UCustomisationWidget)

A UI widget that shows all the Player's inventory items and enables them to equip those items to the Modular Character. This widget tells the Modular Character what items to equip and unequip.

Item Image Widget (UItemImageWidget)

A UI element that represents one item in the Player's inventory. This is spawned by the Inventory Widget and the Customisation Widget.

Body Part Selection Tool (ABodyPartSelectionTool)

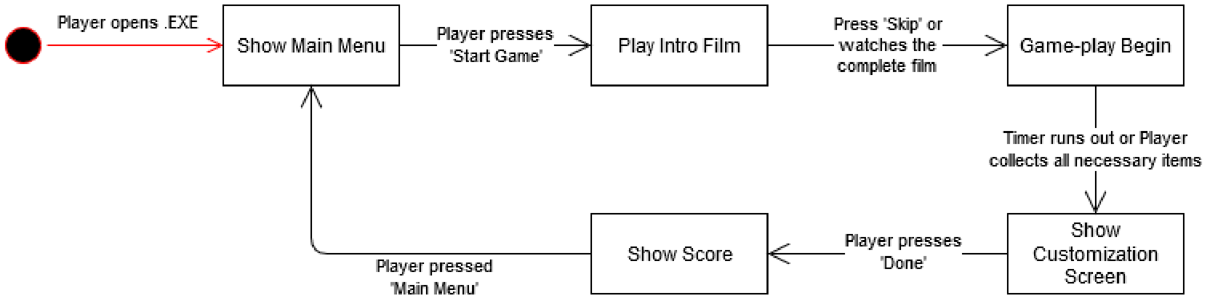
A 3D UI element that enables the Player to switch between what items to display in the Customisation Widget. The options are head items, chest items, arm items and leg items.

Score Calculator (ScoreCalculator)

The Score Calculator is responsible for calculating the final score the Player receives for their custom character design. This calculator taps into the Customisation Widget and then provides the results back to the Customisation Widget to show them.

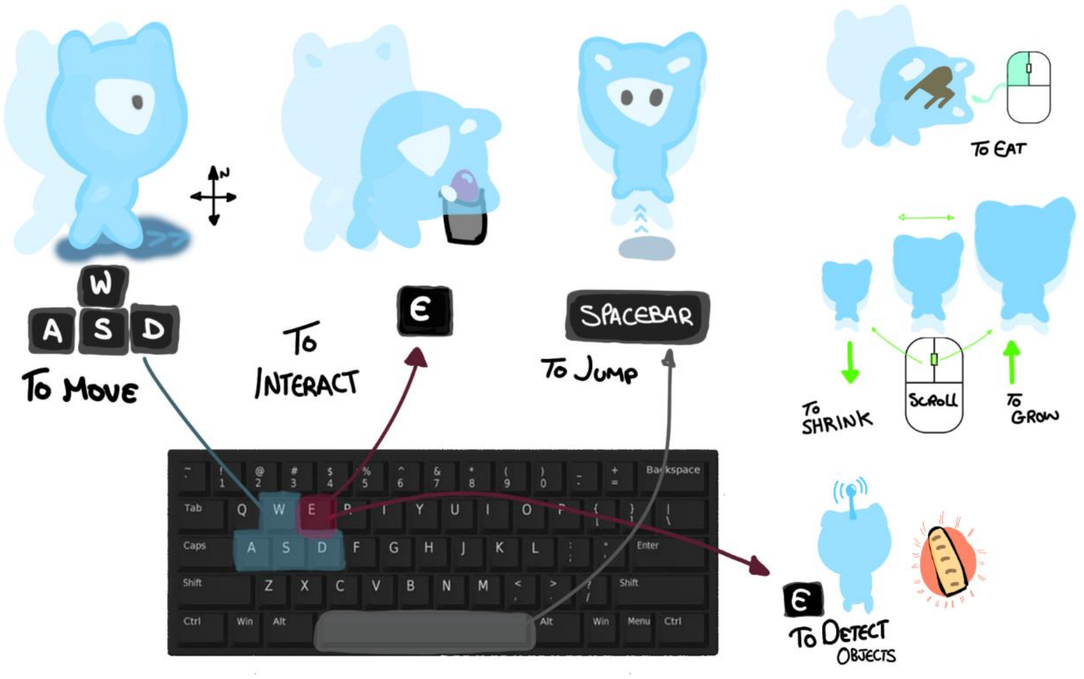
4.8 Game Flow

The basic game loop is:


Player opens .EXE
Show Main Menu
Player presses 'Start Game'
Play Intro Film
Press 'Skip' or watches the complete film
Game-play Begin
Timer runs out or Player collects all necessary items
Show Customization Screen
Player presses 'Done'
Show Score
Player pressed 'Main Menu'

5.0 CONTROLS

The following key bindings are used to control the movements of Guudo.



5.1 Game Play

The controls for general game play are:

Key Binding	Result
W	Forward
A	Move left
D	Move right
S	Move backwards
E	Interact (With button or door) or set off radar
Left mouse button	Consume Item (If within range)
Scroll Wheel Up or 1	Grow
Scroll Wheel Down or 2	Shrink
Space	Jump
Scroll wheel	Zoom in / out
Tab	Open / Close the Inventory

5.2 Menus

The player needs to use the Left Mouse Button to select items or buttons on the menu screens. There are no other special mouse or keyboard inputs.

5.3 Complex Input

There are no complex inputs.

6.0 MECHANICS

The main game mechanics can be split into three categories; player mechanics, object mechanics and component mechanics.

6.1 Player Mechanics

The following is a list of Player mechanics that require further explanation. This list doesn't include standard Player mechanics like movement and jump.

Grow and Shrink Mechanics

The Player can grow and shrink. These mechanics are activated by using the 1 and 2 keys or by using the scroll wheel. These mechanics enable the Player to take on one of three forms (small, normal and large). Each of these forms provide the Player with different advantages and disadvantages.

- The small form enables the Player to fit into tight spaces but slows their movements and disables their ability to push large objects.
- The normal form enables the Player to run fast but disables their ability to fit into tight spaces.
- The large form enables the Player to push large objects but slows their movements and disables their ability to fit into tight places.

The shrink mechanic roughly halves the size of the Player's character. It can only be activated when at a large or normal size.

The grow mechanic roughly doubles the size of the Player's character. It can only be activated when at a small or normal size.

Consume

The Player can consume objects that are within range and have the pickup component applied. Once an object is consumed it is added to the Player's inventory.

Interact

The Player can interact with objects that are a switch and can be turned on and off.

Radar

The Player can fire off a radar that activates a post-processing effect which illuminates all pickups nearby. This enables the Player to find objects that are hidden.

Open Inventory

The Player can open his inventory by pressing the tab key and see a list of all pickups he has consumed.

6.2 Object Mechanics

The following is a list of Object mechanics that require further explanation. This list doesn't include standard object mechanics like physics.

Push Plate

The push plate is an object with a mechanic that pushes a player along a path (spline) when activated by the Player's weight. This is particularly useful when shooting the Player up into the air towards a target region. It simulates a jump pad with an arc the Designer can modify.

Switch

The switch is an object with a mechanics that enables it to be activated or deactivated by the Player. The activation command is then sent to a Designer specified door.

Pressure Plate

The pressure plate is an object with a mechanic that enables it to be activated or deactivated by a Designer specified heavy object such as a boulder. The activation command is then sent to a Designer specified door.

Door

The door is an object with a mechanic that enables it to receive activation commands. It can be activated by any or all the following scenarios.

- Switch activation command received.
- Pressure plate activation command received.
- The Player has collected a Designer specified minimum number of pickups.

The door activates a camera and then plays an open animation when activated. After the animation is complete it returns the camera to the Player's 3rd person view.

6.3 Component Mechanics

The following is a list of component-mechanics that require further explanation. These components can be attached to any object.

Pickup

The pickup component enables the item to be discoverable by the Item System and enables the Player to pick it up and consume it in game.

The pickup component comes with the following Designer specified details:

- The unique name.
- The shape of the silhouette.
- The general shape (for comparison purposes).
- The body position this pickup belongs to.
- Various score ratios.
- The skeletal mesh that represents this pickup.

The use of these details is clarified further in the section on items.

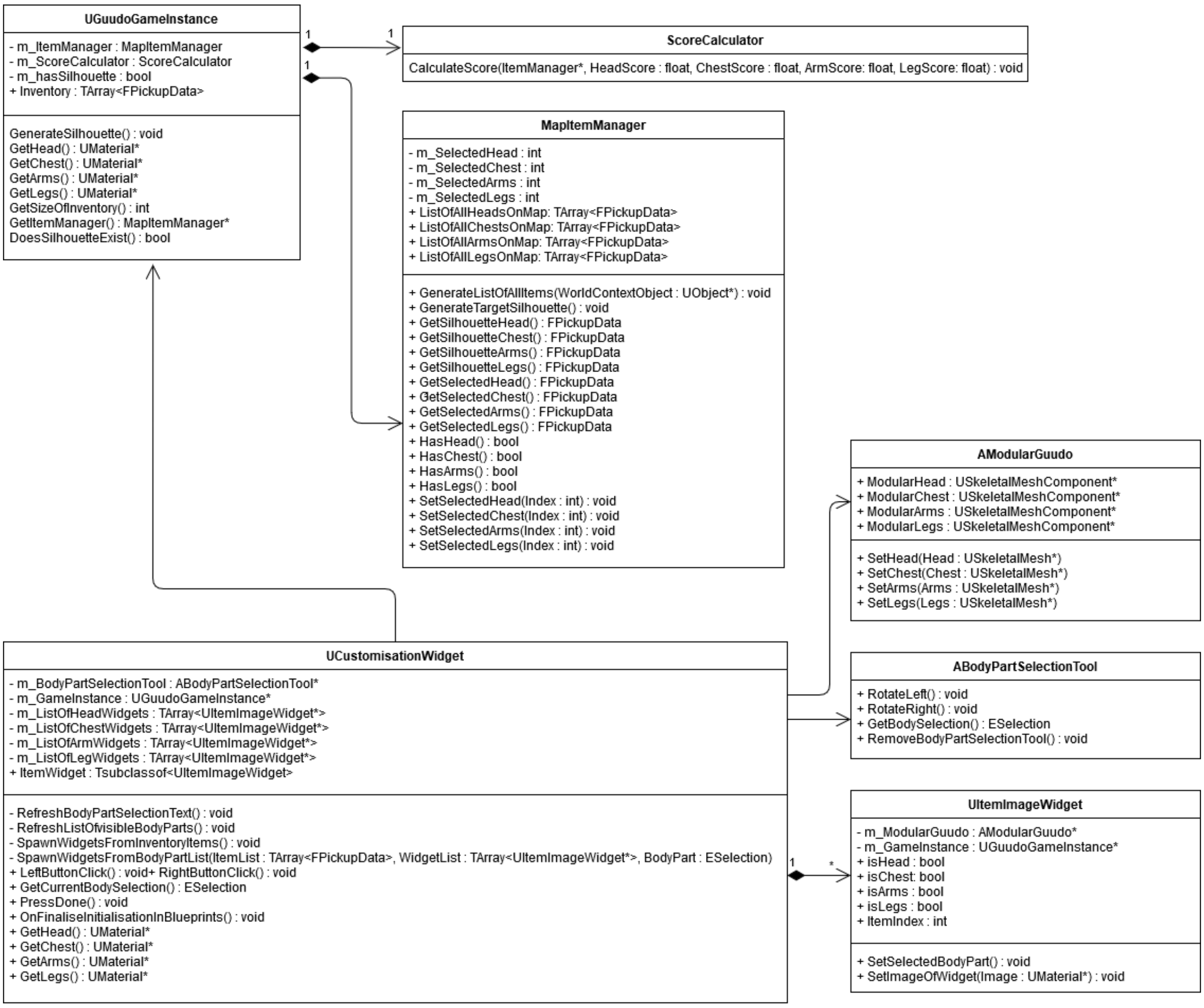
Shakeable

The shakeable component activates physics for the object and enables the Player's character to shake it when at a large form and walking past it.

This gives the Player the illusion that they are massive and have an impact on their environment around them.

7.0 SYSTEMS

The following is a class diagram of the custom game systems followed by a detailed breakdown of how they interact with one another.


UGuudoGameInstance
- m_ItemManager : MapItemManager
- m_ScoreCalculator : ScoreCalculator
- m_hasSilhouette : bool
+ Inventory : TArray<FPickupData>
GenerateSilhouette() : void
GetHead() : UMaterial*
GetChest() : UMaterial*
GetArms() : UMaterial*
GetLegs() : UMaterial*
GetSizeOfInventory() : int
GetItemManager() : MapItemManager*
DoesSilhouetteExist() : bool
1
1
1
ScoreCalculator
CalculateScore(ItemManager*, HeadScore : float, ChestScore : float, ArmScore: float, LegScore: float) : void
MapItemManager
- m_SelectedHead : int
- m_SelectedChest : int
- m_SelectedArms : int
- m_SelectedLegs : int
+ ListOfAllHeadsOnMap: TArray<FPickupData>
+ ListOfAllChestsOnMap: TArray<FPickupData>
+ ListOfAllArmsOnMap: TArray<FPickupData>
+ ListOfAllLegsOnMap: TArray<FPickupData>
+ GenerateListOfAllItems(WorldContextObject : UObject*) : void
+ GenerateTargetSilhouette() : void
+ GetSilhouetteHead() : FPickupData
+ GetSilhouetteChest() : FPickupData
+ GetSilhouetteArms() : FPickupData
+ GetSilhouetteLegs() : FPickupData
+ GetSelectedHead() : FPickupData
+ GetSelectedChest() : FPickupData
+ GetSelectedArms() : FPickupData
+ GetSelectedLegs() : FPickupData
+ HasHead() : bool
+ HasChest() : bool
+ HasArms() : bool
+ HasLegs() : bool
+ SetSelectedHead(Index : int) : void
+ SetSelectedChest(Index : int) : void
+ SetSelectedArms(Index : int) : void
+ SetSelectedLegs(Index : int) : void
AModularGuudo
+ ModularHead : USkeletalMeshComponent*
+ ModularChest : USkeletalMeshComponent*
+ ModularArms : USkeletalMeshComponent*
+ ModularLegs : USkeletalMeshComponent*
+ SetHead(Head : USkeletalMesh*)
+ SetChest(Chest : USkeletalMesh*)
+ SetArms(Arms : USkeletalMesh*)
+ SetLegs(Legs : USkeletalMesh*)
UCustomisationWidget
- m_BodyPartSelectionTool : ABodyPartSelectionTool*
- m_GameInstance : UGuudoGameInstance*
- m_ListOfHeadWidgets : TArray<UItemImageWidget*>
- m_ListOfChestWidgets : TArray<UItemImageWidget*>
- m_ListOfArmWidgets : TArray<UItemImageWidget*>
- m_ListOfLegWidgets : TArray<UItemImageWidget*>
+ ItemWidget : Tsubclassof<UItemImageWidget>
- RefreshBodyPartSelectionText() : void
- RefreshListOfvisibleBodyParts() : void
- SpawnWidgetsFromInventoryItems() : void
- SpawnWidgetsFromBodyPartList(ItemList : TArray<FPickupData>, WidgetList : TArray<UItemImageWidget*>, BodyPart : ESelection)
+ LeftButtonClick() : void+ RightButtonClick() : void
+ GetCurrentBodySelection() : ESelection
+ PressDone() : void
+ OnFinaliseInitialisationInBlueprints() : void
+ GetHead() : UMaterial*
+ GetChest() : UMaterial*
+ GetArms() : UMaterial*
+ GetLegs() : UMaterial*
ABodyPartSelectionTool
+ RotateLeft() : void
+ RotateRight() : void
+ GetBodySelection() : ESelection
+ RemoveBodyPartSelectionTool() : void
1
*
UItemImageWidget
- m_ModularGuudo : AModularGuudo*
- m_GameInstance : UGuudoGameInstance*
+ isHead : bool
+ isChest: bool
+ isArms : bool
+ isLegs : bool
+ ItemIndex : int
+ SetSelectedBodyPart() : void
+ SetImageOfWidget(Image : UMaterial*) : void

Item Manager System

The Item Manager System first activates when the Player begins gameplay. This system scans the entire map for objects that have the pickup component applied to them. (You can read more about the pickup component in the mechanics section of this document). The Item Manager System then categorises pickups into four arrays based on pickup types:

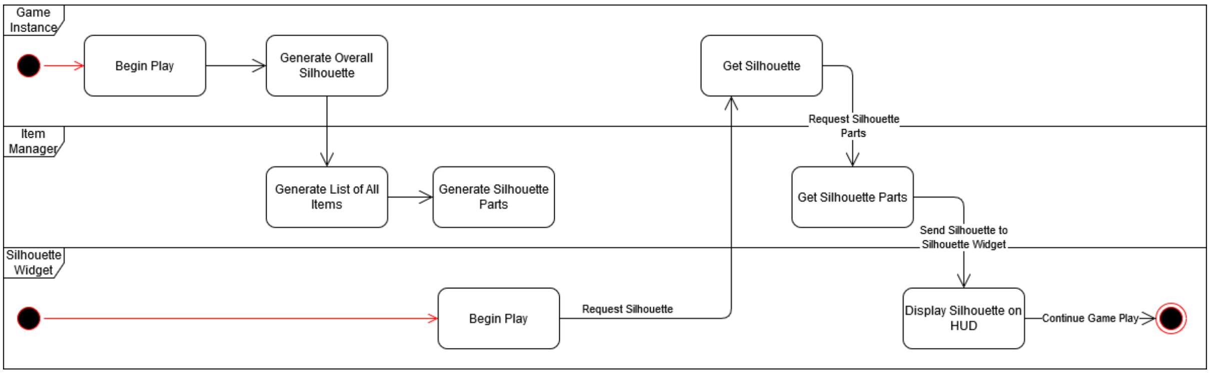
- List of head pickups
- List of chest pickups
- List of arm pickups
- List of leg pickups

Inventory System

The Inventory System is stored within the Game Instance which enables it to be transferred between levels. This system stores an array of data for all the objects the Player has collected.

Silhouette Creation System

The Silhouette Creation System is responsible for accessing the Item Manager and generating a random silhouette and displaying that silhouette to the user in both the HUD and Character Customisation Screens.



Character Creation System

The Character Customisation System first activates when the Player transitions to the Character Customisation Screen. This system accesses the Inventory System and scans the entire system. It then categorises the items into four arrays based on pickup types:

- List of head pickups
- List of chest pickups
- List of arm pickups
- List of leg pickups

The Character Creation System then displays the correct list of pickups based on what the Player has selected. It also accesses the data for these pickups when the Player presses apply and sends that data to the modular character to display.

Score Calculation System

The Score Calculation System accesses the Character Creation System and Silhouette Creation System and calculates a score based on how similar the Player's creation is to the Silhouette. It then provides this score as feedback to the Player.

8.0 GRAPHICS

8.1 Perspective

Our game will primarily be from the 3rd person perspective.

The perspective of an FPS or isometric 3D Game would be an awkward angle that wouldn't fit the feel of the game we were aiming for. With games like I am Bread, Goat simulator and Animal crossing as references, they present their games through a 3rd person perspective that allows them to explore the world around them with a free roaming camera, immersing the player better in the world that was intended for the target audience.

8.2 Art Style

We are using a stylised PBR (Physically Based Rendering).

8.3 Graphic Features

We don't employ any additional graphic features beyond the post-processing and shaders.

8.4 Expected Polycount and Texel Density

The artists are working under the following constraints to keep the game optimized and running smoothly:

Texel density: 1024x1024

Polycount: On average 500-1000 polys depending on the size and importance of the model.

8.5 Post-Processing

We are only using the following post-processing features:

Scan Effect:

Highlights to show where pickup items are to help the Player identify pickups from other static models laying around the map. This is used by the radar feature and is activated when the Player presses ‘E’.

We've decided to include an outline shader because this is a very fast-moving game and we want players to be able to easily identify pickups even at a distance.

Volumetric Height Fog:

This will be used to create light shafts which will give depth to the level. Without the fog effect, the levels would lack depth and would appear quite bland.

8.6 Shaders

The following shaders are being use:

Guudo’s Material Shader:

This shader will give Guudo a liquid/gel-like appearance that sways, moves, and deforms as the Player traverses the level. This is required because the main character is an alien who has an outer shell of goo. This goo won't look natural if it's completely static, the goo shader will make it look more real.

Water Material Shader:

This shader is used in the ship scene for the water around the ship.

Foliage Shader:

This shader is used to create the movement of grass as well as create a subsurface scattering of light as it passes through leaves. We've decided to include a Foliage shader because foliage will be used to hide the seams within the map and to also create an environment that is more alive.

8.7 Lighting

We are using dynamic lighting in the Space / Inside ship scene. However, for the outdoor jungle and ship scene we are using directional lighting with sky dome features.

8.8 Particle effects

We’re using the following particle effects with more effects possibly being added in the future:

Footprints:

Guudo generates footprint particles that indicate where he has been. This will help the player avoid taking paths they’ve already treaded.

Goo Splat:

This particle effect will be generated when he shrinks and grows. This particle exists simply to make Guudo more liquid like.

9.0 PHYSICS

We are using the built physics engine that comes with Unreal Engine.

The primary features we’re using are standard collision detection, collision resolution, gravity and impulse, however, we’re also using physics as part of the Guudo gel workflow.

9.1 Physics used

Gravity

Gravity is used in the following situations:

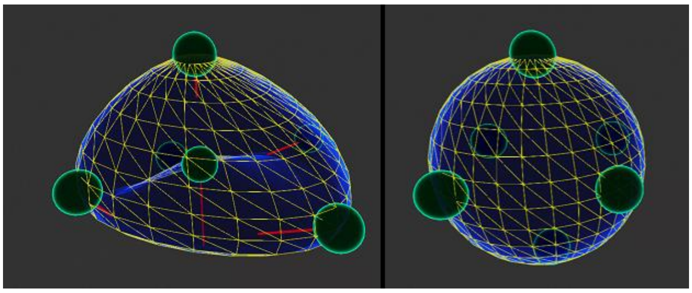
- The shakeable component uses gravity to enable the object to bounce up and down when the Player walks past it in large form.
- Various game mechanics like the pressure-plate require objects to be pushed onto them to activate. Gravity is required to ensure the ball doesn't bounce off into the sky.

Impulse

Impulse is used in the following situations:

- The shakeable component uses impulse to enable the objects to shoot up into the air slightly when the Player walks past them in large form.
- The grow mechanics applies an impulse against everything around it to ensure the Player doesn't get stuck under something when growing.

Shaders



Physics is used to create the gel-like appearance of the Player's character. The shader accesses several physics enabled sockets that bounce around as the Player moves. This is then used to stretch the characters skin.

9.2 Potential slowdowns identified

The only concern we have related to potential slowdowns is if heaps of objects are near the Player and they all have the shakeable component.

We have tested with 20 separate objects utilising the shakeable component at once without any noticeable slowdowns. We don't intend on ever having that many objects with the shakeable component next to each other at once, so this shouldn't be an issue.

The second biggest issue is that on rare occasions Guudo's skin, which is physics controlled, gets caught on various objects thus causing a shearing effect.

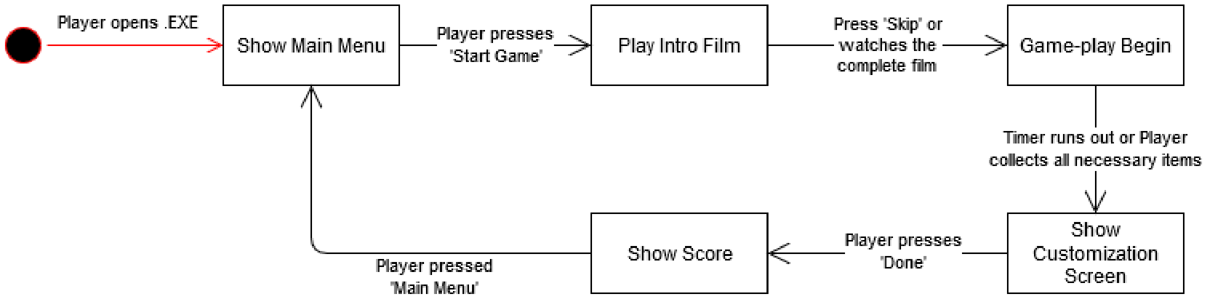
The tech artist is constantly tweaking the values of the skin to avoid this.

10.0 GAME FLOW

10.1 Mission / Level Structure

The basic game loop is:

Show Main Menu → Play Intro Film → Game-play Begin → Show Customisation Screen → Show Score → Show Main Menu.



Player opens .EXE
Show Main Menu
Player presses 'Start Game'
Play Intro Film
Press 'Skip' or watches the complete film
Game-play Begin
Timer runs out or Player collects all necessary items
Show Customization Screen
Player presses 'Done'
Show Score
Player pressed 'Main Menu'

<u>Show Main Menu</u>

When the Player opens the .EXE, the first thing they see is the Main Menu.

<u>Play Intro Film</u>

When the Player presses ‘Start Game’, the intro film starts playing. They have the option of watching the complete film or skipping it by pressing the skip button.

<u>Game-play Begin</u>

When Game-play begins, the Item Manager and Silhouette Creation system activate and update the HUD with the silhouette the Player needs to collect. The Player then needs to navigate around the environment to find items (pickups) to make their silhouette by platforming, shrinking and knocking things over.

<u>Customise the Character</u>

When the Player collects all the necessary items, or the timer runs out, they’ll be able to use the items they've collected along the way to build their custom character.

Receive Score

When the Player has confirmed their choice, they will be scored on how accurate they did when constructing their final form.

10.2 PLAYER OBJECTIVES

For the tutorial part of the level, the Player starts the game in the learning pit.

Learning Pit

The Player will be taught how to move and jump around the pit. They'll need to jump out of this area to complete the objective.

Leaving the Nest

The Player will then be taught how to Shrink and Grow. They'll need to use these abilities to proceed to the next objective.

Make the Jumps

The Player will need to perform basic platforming to proceed to the next objective.

Open Wide

The Player will learn how to eat objects and fire off their radar. They'll need to use these abilities to proceed to the next objective.

Learning to Run

The Player will start the actual gameplay experience.

10.3 EVALUATION OF PLAYER PROGRESS

Levels aren't stored in memory. There is currently only one level, so this level is loaded completely into memory upon Game Start.

The Game State maintains a copy of the Player's Inventory across levels. This is to ensure that the Inventory isn't destroyed when transitioning to the Character Customisation Screen.

11.0 LEVELS

The Guudo game only features one level. This level is split up into three main sections. Each section forces the Player to perform different actions.

Space:

The space scene revolves around platforming to collect objects for the silhouette, specifically jumping from platform to platform whilst using shrink and grow to knock things over to reach certain places.

Jungle:

The jungle scene revolves around puzzle solving, where the player needs to traverse mazes, push objects to open new paths and platform to reach higher places.

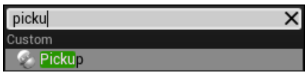
Caribbean:

The Caribbean scene also revolves around physic manipulation, where again the player will need to push objects to a place to open a new path, use platforming to reach new places, and using grow and shrink inside the boat to access objects to open locked places and progress the game.

12.0 ITEMS

12.1 Item Functionality

Any item that has the Pickup Component added can be collected by the Player.



The Pickup component comes with the following variables that can be tweaked by the Designer:

Variable	Comment
Name	A unique name. Items need unique names as these names are stored by the inventory.
Silhouette	A Silhouette image that represents the item. This is used by the HUD and Character Customisation screen.
General Shape	This is an enumerator that represents the general shape of the object. If the player uses the wrong item in the Character Customisation screen, but it has the correct shape, they'll get the **Similar Score**. The possible shapes are Box, Sphere, Cylinder or Stick.
Is Head, Chest, Arms or Legs	The part of the body this item can be applied to.
Similar Score	How many points the Player will get out of 100 if their pickup matches the shape but not the name in the Character Customisation screen.
Wrong Score	How many points the Player will get out of 100 if their pickup doesn't match any of the requirements for the Character Customisation screen.
Mesh	A Skeletal Mesh that represents the item. This is used by the Character Customisation screen.

12.2 List of Items

The following is a List of Items (pickups) that will exist in the game:

Heads:

- UFO
- Alien Head
- Relic
- Mushroom
- Coconut
- Camera
- TV

Torso:

- Planet
- Brain Jar

- Ruins
- Crate
- Keg Barrel
- Fan
- Computer Tower

Arms:

- Alien Gun
- Cluster of Crystals
- Rubble
- Pirate Sword
- Map
- Boom Mic
- Light Stand

Legs:

- Robot Leg
- Log
- Bundle of vines
- Pirate leg
- Rifle
- Telescope
- Baguette

12.3 Player Effect

The Pickups don't give the Player any specific benefit until they reach the Customisation Screen.

12.4 Influence on Gameplay

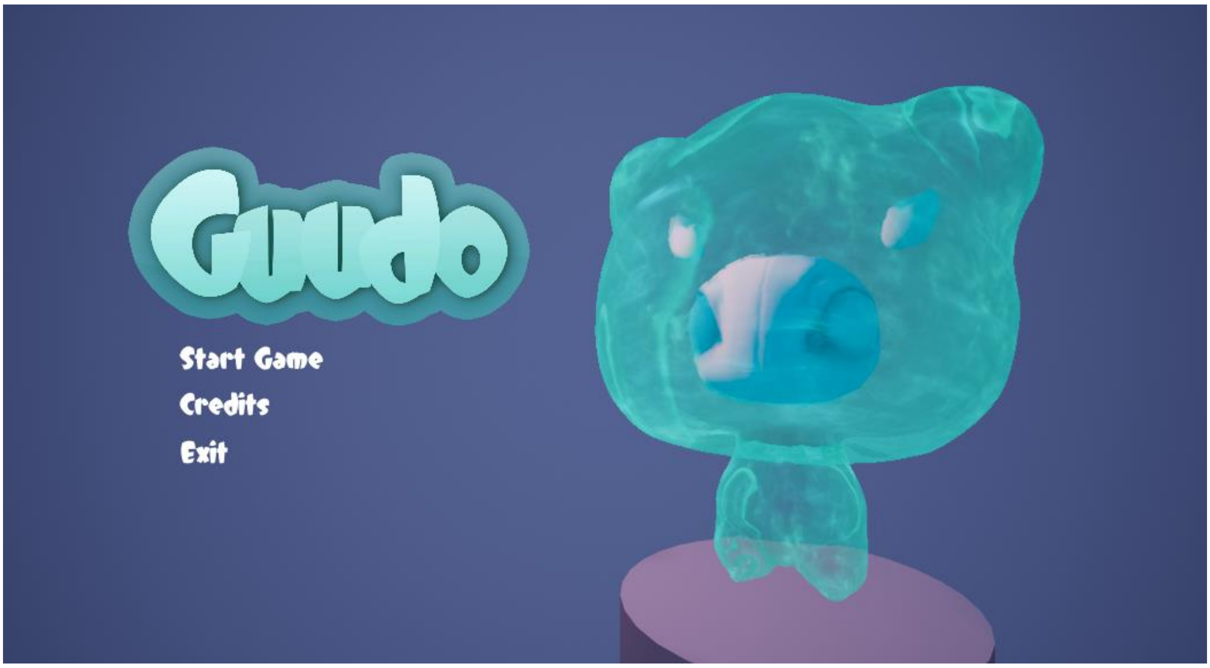
When the Player reaches the Customisation Screen, they'll need to apply the pickups to build their custom character. The goal is to make their custom character match as closely as possible to the silhouette they've been provided. The closer their design is, the more points they'll get.

13.0 INTERFACE

13.1 Menus

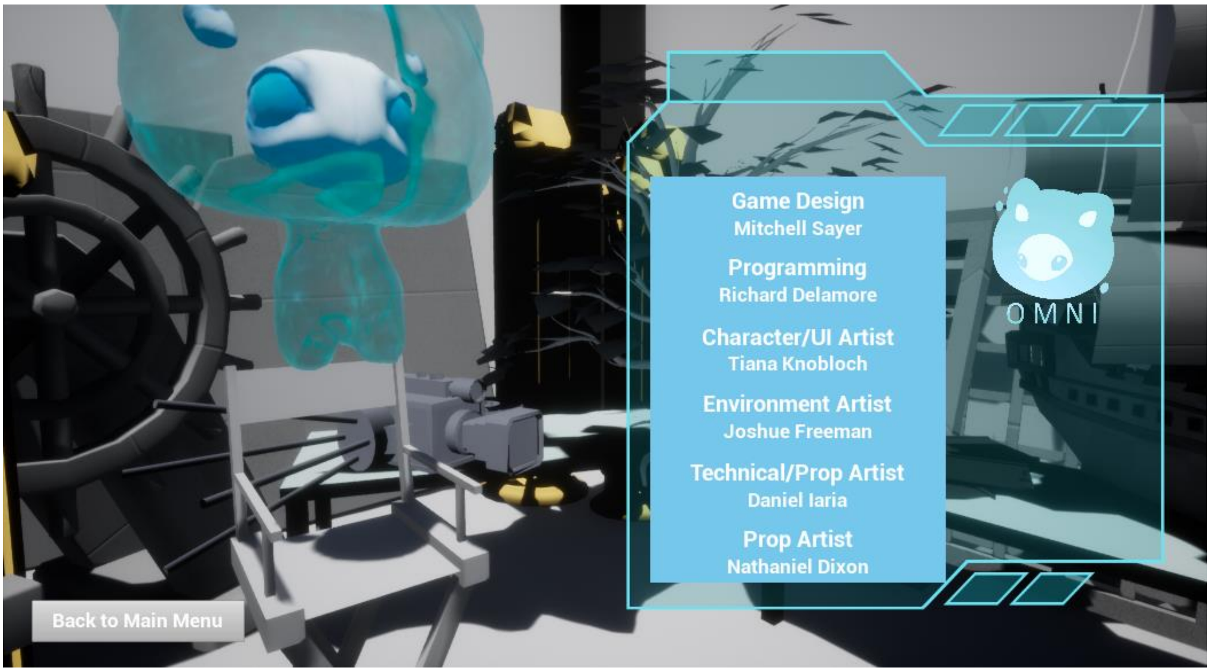
Main Menu:

This is the main Entry Point into the game. The "Start Game" button takes the Player to the Introduction Video. The "Credits" button takes the Player to the Credits Screen.



Credits Screen:

This screen simply lists the people involved in creating the project. There is nowhere to go but back to the Main Menu.



Game Design
Mitchell Sayer

Programming
Richard Delamore

Character/UI Artist
Tiana Knobloch

Environment Artist
Joshue Freeman

Technical/Prop Artist
Daniel Iaria

Prop Artist
Nathaniel Dixon

Intro Video

The Intro Video presents the background story for the Guudo game. It tells how Guudo crashed on earth and how he needs to collect items. The Player can choose to watch the whole cinematic or press the "Skip" button to jump ahead to the main level.

13.2 UI / HUD

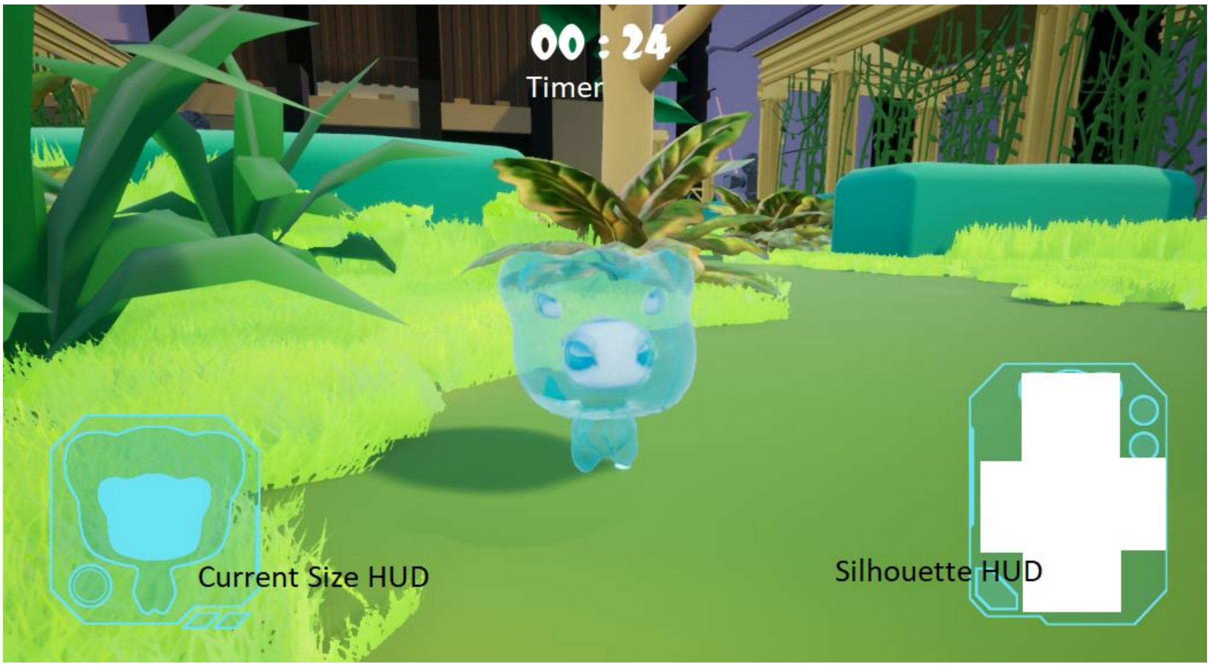
Gameplay HUD:

There are three main parts to the Gameplay HUD:

The **Timer** tells the Player how long they have left to complete the level. This is managed by the Game Mode.

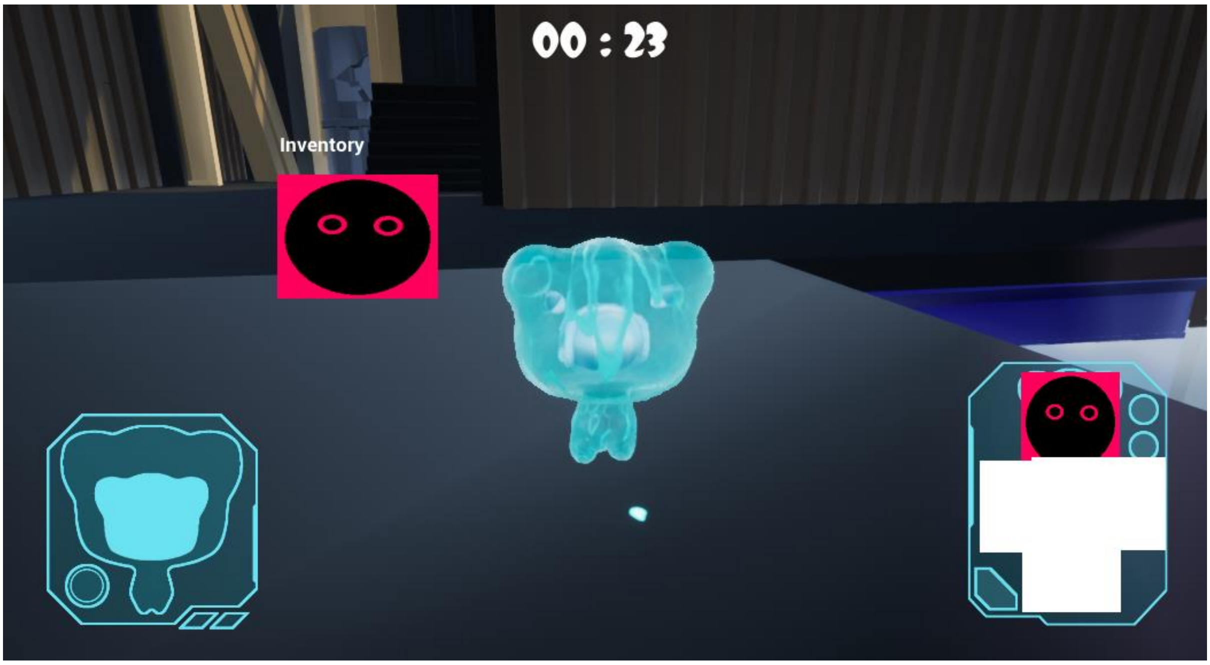
The **Current Size HUD** visually indicates to the Player their current Growth State. It also flashes red when they attempt to grow in an area where they can't grow. This is managed by the Playable Guudo Character.

The **Silhouette HUD** visually indicates to the Player the type of body parts that they need to try and collect. Currently, there are no silhouette graphics, so these look like white squares. This information is gathered from the Game Instance. For more information on how the Silhouette is generated, please see the section on Systems.



Inventory HUD

The inventory HUD appears when the Player presses the "tab" button. It lists all the items the Player has collected so far.



13.3 Character Customisation

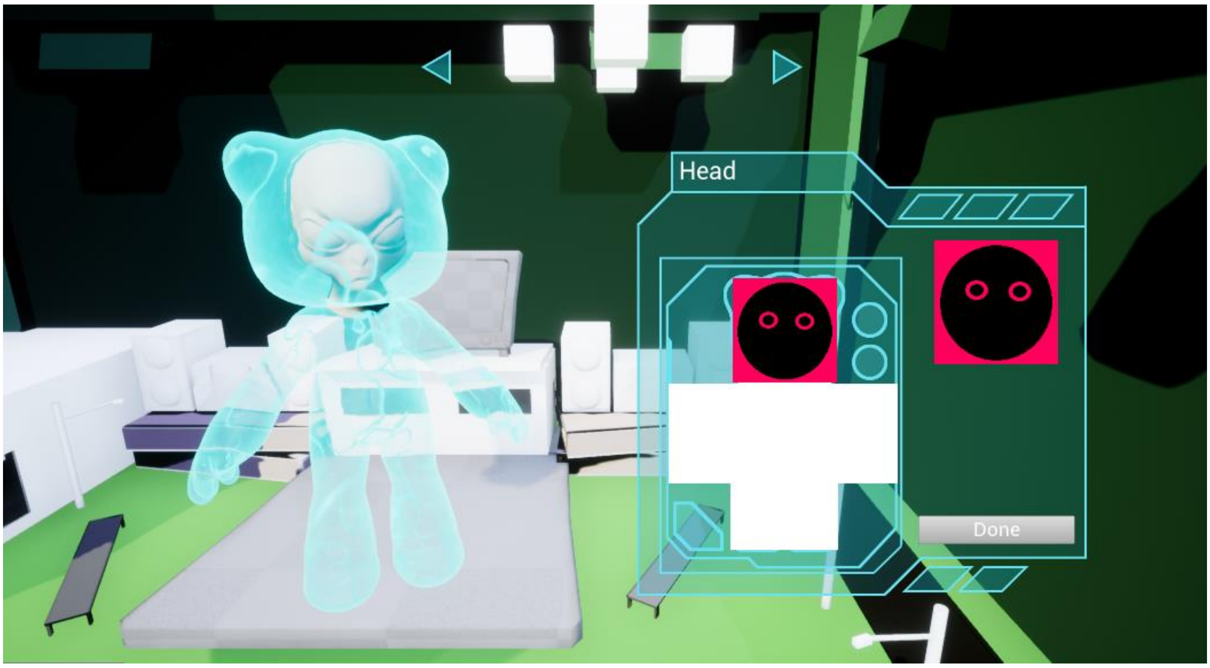
The following two screens are used to score the Player's character customisation. To understand how they work in depth, please see the section on Systems.

Character Customisation Screen

The Character Customisation screen displays a list of items the Player has in their Inventory.

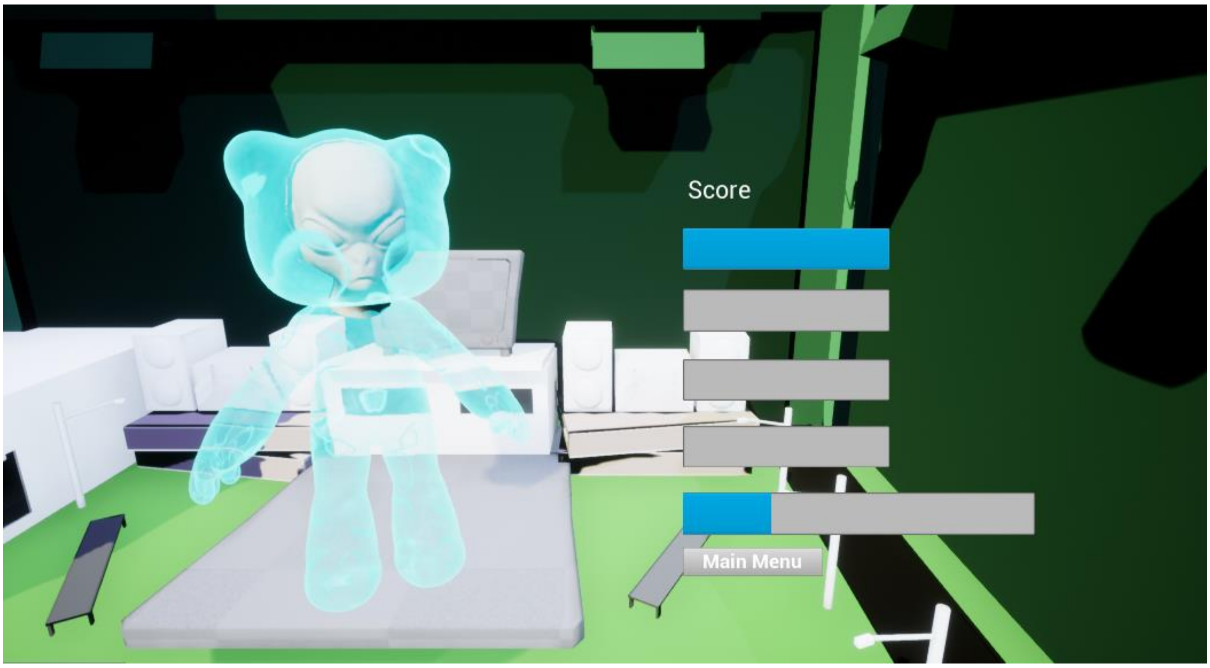
In the top-middle of the screen is the Body Part Selection Tool. Clicking on the Left or Right arrows updates which part of the body the Player can update. In the example graphic below, the Player has selected the Head part of the body.

Clicking on any of the items in the Character Customiser will apply it to the Player's body.



Score Screen

The Score Screen indicates how many points the Player got for each part of the body and overall.



13.4 Camera Operation

The entire game operates using a 3rd person camera attached to the end of a camera arm that trails behind the main Guudo character.

The camera arm automatically pushes the camera out of the way when scene objects block the camera's view of the Player.

The only time when the camera changes its view target is when the Player activates a door. The camera then quickly sweeps across to look at the door as it opens and then sweeps back to the original position behind the Player when the opening animation is completed.

Finally, the camera also influences the Guudo character's material. If the camera gets too close to the Guudo character, the character starts to fade out to enable the Player to look through the character easily and see the environment.

13.5 Player Controls

The various interfaces simply tap into the Player Controller. There are no custom controls used.

To learn about the controls in depth, see the section titled Controls.

14.0 FILE AND PROGRAMMING STANDARDS

14.1 Folder Structure

The following file are to be adhered to:

Folder (level 1)	Folder (level 2)	Comment
Assets		
	Decals	Decals such as the footprints.
	M	Material and Material Instances.
	Particles	Particle Effects.
	PP	Post Processing Effects.
	Silhouettes	Silhouette UI Images.
	SK	Skeletal Meshes.
	SM	Static Meshes.
	SpeedTree	Speed Tree imports.
	T	Raw Textures.
	Test	Test Folder.
Blueprints		General Blueprints derived from C++ classes.
	Customisation Screen	Blueprints dedicated to just the Customisation Screen. Some of these are derived from C++ classes.
CameraShake		Camera Shake Effects such as the pickup effect and the heavy walk effect.
HUD		Various HUD widgets such as the Loading screen, Credits and Media Player

	Cusomisation Screen	Widgets derived from C++ classes that are dedicated to just the Customisation Screen.
	UI	Raw textures used just by the HUD.
Levels		
	Level Templates	The Menu and Game Play Levels.
Manager		Game Instance (This holds the inventory which is global)
Movies		Video and Media Player for the Intro Cinematic
Sound		Various sounds such as background music.
	Footprints	Sounds dedicated to just the footprints.
	UI	Sounds dedicated to just the UI.

14.2 Naming Conventions

The following affixes are added to assets to specify their purpose:

Prefix	Suffix	Use
M_		Material or Material Instance.
T_		Raw Texture
	_A	Alpha Map
	_N	Normal Map
	_ORM	Occlusion Roughness
	_E	Emissive Map
P_		Particle System
SK_		Skeletal Mesh
SM_		Static Mesh
BP_		Blueprint Implementation
	_Greybox	Grey box Level
	_BGM	Background Music

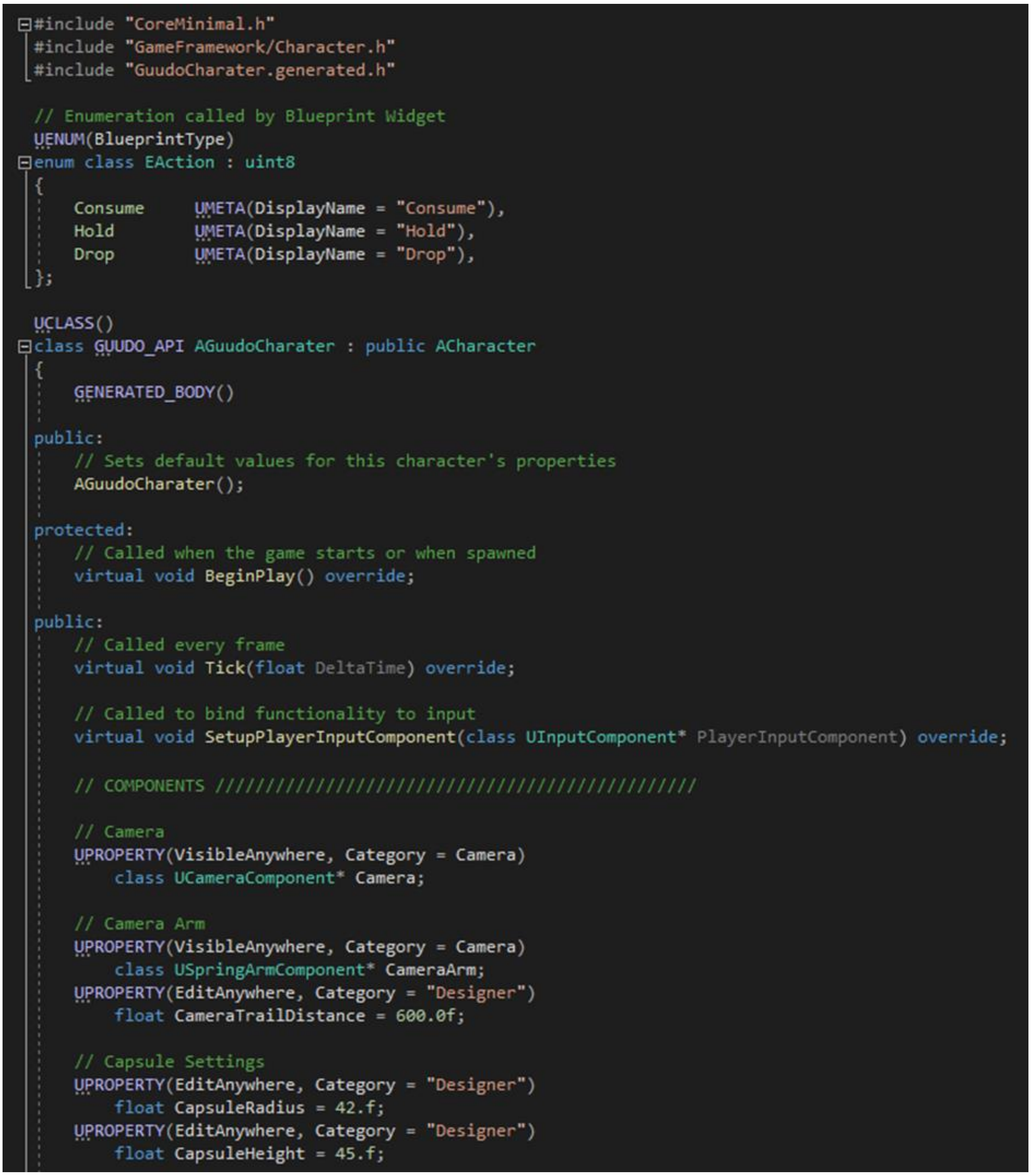
14.3 Script Conventions

The following prefixes are added to variables and classes to specify its purpose:

Prefix	Use
M_	Private Variable
E	Enumerator. This is a requirement set by Unreal.
F	Struct. This is a requirement set by Unreal.
A	Actor. This is a requirement set by Unreal.

Public variables won’t include any prefix because these are directly visible in the Unreal Editor.

14.4 Good Coding Conventions



```
#include "CoreMinimal.h"
#include "GameFramework/Character.h"
#include "GuudoCharater.generated.h"

// Enumeration called by Blueprint Widget
UENUM(BlueprintType)
enum class EAction : uint8
{
	Consume		UMETA(DisplayName = "Consume"),
	Hold		UMETA(DisplayName = "Hold"),
	Drop		UMETA(DisplayName = "Drop"),
};

UCLASS()
class GUUDO_API AGuudoCharater : public ACharacter
{
	GENERATED_BODY()

public:
	// Sets default values for this character's properties
	AGuudoCharater();

protected:
	// Called when the game starts or when spawned
	virtual void BeginPlay() override;

public:
	// Called every frame
	virtual void Tick(float DeltaTime) override;

	// Called to bind functionality to input
	virtual void SetupPlayerInputComponent(class UInputComponent* PlayerInputComponent) override;

	// COMPONENTS ///////////////////////////////////////////////

	// Camera
	UPROPERTY(VisibleAnywhere, Category = Camera)
		class UCameraComponent* Camera;

	// Camera Arm
	UPROPERTY(VisibleAnywhere, Category = Camera)
		class USpringArmComponent* CameraArm;
	UPROPERTY(EditAnywhere, Category = "Designer")
		float CameraTrailDistance = 600.0f;

	// Capsule Settings
	UPROPERTY(EditAnywhere, Category = "Designer")
		float CapsuleRadius = 42.f;
	UPROPERTY(EditAnywhere, Category = "Designer")
		float CapsuleHeight = 45.f;
```

15.0 TECHNICAL RISK MANAGEMENT

15.1POTENTIAL TECHNICAL RISKS

Time constraints

This is one of the biggest concerns as we are working under a tight schedule. To mitigate this issue, I focussed on programming the absolute necessary components first such as the inventory system and the customisation screen.

Source Control

One of the main issues we have faced so far is source control. Twice I’ve lost considerable time due to people using source control incorrectly. The only way to mitigate this issue is to train team members in how to correctly use source control.

Rating System

The rating system, which is part of the character customisation screen and pickups, was one of the most complex systems in the game to develop. This system is crucial to the success of the game, so I needed to focus my efforts on ensuring I finished it first to test whether it worked as intended.

The rating system is currently in place, but I won't know if there will be any issues with the rating system until the Designer finishes building the levels and sprinkles them with items (pickups).

Gel Body

This was the second biggest potential headache; however, we successfully implemented a gel-like body system by using sockets, physic objects and shaders.

16.0 ASSET LIST

16.1 Script Assets

The following is a list of Script Assets

Guudo.Target.cs	
GuudoEditor.Target.cs	
Guudo.Build.cs	
Guudo.cpp	
Guudo.h	
GuudoCharater.cpp	
GuudoCharater.h	Guudo Character
GuudoGameMode.cpp	
GuudoGameMode.h	
GuudoGameModeBase.cpp	
GuudoGameModeBase.h	
PickupData.cpp	
PickupData.h	Struct: Pickup Data
BodyPartSelectionTool.cpp	
BodyPartSelectionTool.h	Body Part Selection Tool (Customisation)
CustomisationWidget.cpp	
CustomisationWidget.h	Customisation HUD
InventoryWidget.cpp	
InventoryWidget.h	Inventory HUD
ItemImageWidget.cpp	
ItemImageWidget.h	Customisation or Inventory Item HUD
ModularGuudo.cpp	
ModularGuudo.h	Modular Character (Customisation)
Ball.cpp	

Ball.h	Ball or Boulder
Door.cpp	
Door.h	Door
Pickup.cpp	
Pickup.h	Pickup Component
PressurePlate.cpp	
PressurePlate.h	Pressure Plate
PushPlate.cpp	
PushPlate.h	Push Plate
Shakeable.cpp	
Shakeable.h	Shakeable Component
Switch.cpp	
Switch.h	Switch
GuudoGameInstance.cpp	
GuudoGameInstance.h	Game Instance
MapItemManager.cpp	
MapItemManager.h	Item Manager
ScoreCalculator.cpp	
ScoreCalculator.h	Score Calculator (Customisation)
SilhouetteWidget.cpp	
SilhouetteWidget.h	Silhouette HUD

16.2 General Assets

The following is a list of the General Assets:

GreyBox_Assets_Boom_Mic.uasset
GreyBox_Assets_Computer_Tower1.uasset
GreyBox_Assets_Computer_Tower3.uasset
GreyBox_Assets_Curved_Monitor2.uasset
GreyBox_Assets_Curved_Monitor3.uasset
GreyBox_Assets_Curved_Monitor4.uasset
GreyBox_Assets_Curved_Monitor5.uasset
GreyBox_Assets_Curved_Monitor6.uasset
GreyBox_Assets_Curved_Monitor7.uasset
GreyBox_Assets_Desk.uasset
GreyBox_Assets_Keyboard.uasset
GreyBox_Assets_Keyboard1.uasset
GreyBox_Assets_Keyboard2.uasset
GreyBox_Assets_Mouse.uasset
GreyBox_Assets_Mouse1.uasset
GreyBox_Assets_Mouse2.uasset
GreyBox_Assets_Scifi_Door.uasset
GreyBox_Assets_Set_Light1.uasset
GreyBox_Assets_SM_BreakroomWall_End.uasset
GreyBox_Assets_SM_SoundproofWall_Lrg.uasset
GreyBox_Assets_SM_SoundproofWall_Sml.uasset
lambert1.uasset
M_Crystal.uasset
M_CrystalTree.uasset
M_Floor.uasset
M_PropsAtlas.uasset

SM_Book.uasset
SM_Boom_Mic.uasset
SM_BreakroomWall_Long.uasset
SM_BreakroomWall_nd.uasset
SM_Camera.uasset
SM_Clipboard.uasset
SM_Column.uasset
SM_Computer_Tower1.uasset
SM_Computer_Tower3.uasset
SM_CrystalTree.uasset
SM_Curved_Monitor2.uasset
SM_Curved_Monitor3.uasset
SM_Curved_Monitor4.uasset
SM_Curved_Monitor5.uasset
SM_Curved_Monitor6.uasset
SM_Curved_Monitor7.uasset
SM_Desk.uasset
SM_DirectorChair.uasset
SM_Fern.uasset
SM_GuudoUFO.uasset
SM_HangLights.uasset
SM_Keyboard.uasset
SM_Keyboard1.uasset
SM_Keyboard2.uasset
SM_LongDesk.uasset
SM_MiniHouse.uasset
SM_MiniLightPole.uasset
SM_MiniSetBG.uasset
SM_MiniStreet.uasset
SM_MiniStreetCorner.uasset
SM_Mouse.uasset
SM_Mouse1.uasset
SM_Mouse2.uasset
SM_MushroomGroup.uasset
SM_Pen.uasset
SM_RoundTable.uasset
SM_Scaffold.uasset
SM_ScaffoldCorner.uasset
SM_Scifi_Corner.uasset
SM_Scifi_Door.uasset
SM_Scifi_Pillar.uasset
SM_Scifi_Wall_5.uasset
SM_Scifi_Wall_Long.uasset
SM_Scifi_Wall_Mid.uasset
SM_Scifi_Wall_Short.uasset
SM_SetUFO.uasset
SM_Set_Light1.uasset
SM_SoundproofWall_Lrg.uasset
SM_SoundproofWall_Sml.uasset
SM_Stereo.uasset
SM_TV.uasset
Soundbooth_SoundBooth.uasset
Soundbooth_Stairs.uasset
Soundbooth_Stairs1.uasset
WorldGridMaterial.uasset
Assets\C_Mesh\C_JungleTree.uasset
Assets\Decals\M_Footprint.uasset

```
Assets\Decals\M_Shrink_Area.uasset
Assets\Decals\M_Shrink_Area_Inst.uasset
Assets\M\M_AlienHead.uasset
Assets\M\M_Baguette.uasset
Assets\M\M_BoomMic.uasset
Assets\M\M_cannonball.uasset
Assets\M\M_Coconut.uasset
Assets\M\M_Column.uasset
Assets\M\M_Crystal.uasset
Assets\M\M_CustomizerGuudo.uasset
Assets\M\M_Dirt.uasset
Assets\M\M_Doors.uasset
Assets\M\M_ElephantLeaf.uasset
Assets\M\M_Fern.uasset
Assets\M\M_Floor.uasset
Assets\M\M_Greenscreen.uasset
Assets\M\M_Greybox.uasset
Assets\M\M_Guudo.uasset
Assets\M\M_GuudoHead.uasset
Assets\M\M_Guudo_Inst.uasset
Assets\M\M_Hallway.uasset
Assets\M\M_Hedge.uasset
Assets\M\M_House.uasset
Assets\M\M_JungleFloor.uasset
Assets\M\M_KrakenTentacle.uasset
Assets\M\M_Lilypads.uasset
Assets\M\M_Log.uasset
Assets\M\M_Master.uasset
Assets\M\M_MasterFlat_Inst.uasset
Assets\M\M_MenuPlinth.uasset
Assets\M\M_Mushroom.uasset
Assets\M\M_MushroomPickup.uasset
Assets\M\M_Pavillion.uasset
Assets\M\M_PirateShip.uasset
Assets\M\M_Road.uasset
Assets\M\M_Scaffold.uasset
Assets\M\M_SoundproofWall.uasset
Assets\M\M_Table.uasset
Assets\M\M_Tallfern.uasset
Assets\M\M_VinePickup.uasset
Assets\M\M_Wood.uasset
Assets\M\Planets\M_AlienPlanet.uasset
Assets\M\Planets\M_BarrenPlanet.uasset
Assets\M\Planets\M_ColdPlanet.uasset
Assets\M\Planets\M_Hotplanet.uasset
Assets\M\Planets\M_Planet.uasset
Assets\M\SciFi\M_Crystal.uasset
Assets\M\SciFi\M_CrystalTree.uasset
Assets\M\SciFi\M_ScifiWall.uasset
Assets\M\SciFi\M_Scifi_Fern.uasset
Assets\M\SciFi\M_Scifi_Shroom.uasset
Assets\Particles\JellySplash.uasset
Assets\Particles\M_Guudo_Bubble.uasset
Assets\Particles\M_JellySplash.uasset
Assets\Particles\P_Bubble.uasset
Assets\Particles\P_Footstep.uasset
Assets\PP\OcclusionMasking.uasset
```

```
Assets\PP\Outline.uasset
Assets\PP\PC_Scan.uasset
Assets\PP\PP_Outline.uasset
Assets\PP\PP_Outline_Inst.uasset
Assets\PP\PP_Scan.uasset
Assets\PP\PP_Scan_Inst.uasset
Assets\PP\SampleOffsetPixels.uasset
Assets\PP\StencilBitMask.uasset
Assets\Silhouettes\test.uasset
Assets\Silhouettes\test_Mat.uasset
Assets\Silhouettes\test_Mat_2.uasset
Assets\SK\GuudoMenu_BP.uasset
Assets\SK\SK_Guudo.uasset
Assets\SK\SK_Guudo_Anim.uasset
Assets\SK\SK_Guudo_PhysicsAsset.uasset
Assets\SK\SK_Guudo_Skeleton.uasset
Assets\SK\SK_KrakenTentacle.uasset
Assets\SK\SK_KrakenTentacle_Anim.uasset
Assets\SK\SK_KrakenTentacle_PhysicsAsset.uasset
Assets\SK\SK_KrakenTentacle_Skeleton.uasset
Assets\SK\Animations\Anim_Idle01.uasset
Assets\SK\Animations\Anim_Idle02.uasset
Assets\SK\Animations\Anim_Idle03.uasset
Assets\SK\Animations\Dance_Anim.uasset
Assets\SK\Animations\GuudoShadow_BP.uasset
Assets\SK\Animations\Guudo_AnimBP.uasset
Assets\SK\Animations\Guudo_BS.uasset
Assets\SK\Animations\Idle_Anim.uasset
Assets\SK\Animations\InAir_Anim.uasset
Assets\SK\Animations\JumpEnd_Anim.uasset
Assets\SK\Animations\JumpStart_Anim.uasset
Assets\SK\Animations\MiniSprint_Anim.uasset
Assets\SK\Animations\NormalSprint_Anim.uasset
Assets\SK\Animations\Pickup_Anim.uasset
Assets\SK\Animations\Walk_Anim.uasset
Assets\SK\Customiser\SK_GuudoCustomizer.uasset
Assets\SK\Customiser\SK_GuudoCustomizer_Skeleton.uasset
Assets\SK\Customiser\Arms\SK_BoomMic.uasset
Assets\SK\Customiser\Arms\SK_Crystals.uasset
Assets\SK\Customiser\Head\SK_AlienHead.uasset
Assets\SK\Customiser\Head\SK_Camera.uasset
Assets\SK\Customiser\Head\SK_Coconut.uasset
Assets\SK\Customiser\Head\SK_Curved_Monitor.uasset
Assets\SK\Customiser\Head\SK_SetUFO.uasset
Assets\SK\Customiser\Legs\SK_Baguette.uasset
Assets\SK\Customiser\Legs\SK_BoomMic.uasset
Assets\SK\Customiser\Legs\SK_Log.uasset
Assets\SK\Customiser\Torso\SK_Barrel.uasset
Assets\SK\Customiser\Torso\SK_RingedPlanet.uasset
Assets\SK\Guudo_Shadow\SK_Guudo_shadow.uasset
Assets\SM\greyboxue4.uasset
Assets\SM\JungleTree_Branch_Mat.uasset
Assets\SM\JungleTree_Leaf_Mat.uasset
Assets\SM\lambert3.uasset
Assets\SM\M_Carribean_Atlas.uasset
Assets\SM\SM_AlienHead.uasset
Assets\SM\SM_Baguette.uasset
```

```
Assets\SM\SM_Barrel.uasset
Assets\SM\SM_Book.uasset
Assets\SM\SM_BoomMic.uasset
Assets\SM\SM_BoomMicPickUp.uasset
Assets\SM\SM_Breakroom_Wall.uasset
Assets\SM\SM_Breakroom_WallLong.uasset
Assets\SM\SM_CabinDoor.uasset
Assets\SM\SM_CageDoor.uasset
Assets\SM\SM_Camera.uasset
Assets\SM\SM_CameraPickup.uasset
Assets\SM\SM_Cannon.uasset
Assets\SM\SM_CannonBall.uasset
Assets\SM\SM_Capstan.uasset
Assets\SM\SM_Chain.uasset
Assets\SM\SM_ChainShot.uasset
Assets\SM\SM_Clipboard.uasset
Assets\SM\SM_Coconut.uasset
Assets\SM\SM_Column.uasset
Assets\SM\SM_ComputerTower.uasset
Assets\SM\SM_Crystal.uasset
Assets\SM\SM_CrystalGrp.uasset
Assets\SM\SM_CrystalTree.uasset
Assets\SM\SM_CurvedMonitor.uasset
Assets\SM\SM_Desk.uasset
Assets\SM\SM_DirectorChair.uasset
Assets\SM\SM_ElephantLeaf.uasset
Assets\SM\SM_Flintlock.uasset
Assets\SM\SM_Fountain.uasset
Assets\SM\SM_GooFootprint.uasset
Assets\SM\SM_GreenScreen.uasset
Assets\SM\SM_GroundFern.uasset
Assets\SM\SM_GuudoUFO.uasset
Assets\SM\SM_HangingVine.uasset
Assets\SM\SM_HangLights.uasset
Assets\SM\SM_Hedge.uasset
Assets\SM\SM_JungleTower.uasset
Assets\SM\SM_JungleTree.uasset
Assets\SM\SM_Keyboard.uasset
Assets\SM\SM_Lilypads.uasset
Assets\SM\SM_Lilypad_Lrg.uasset
Assets\SM\SM_Lilypad_Med.uasset
Assets\SM\SM_Lilypad_Sml.uasset
Assets\SM\SM_Log.uasset
Assets\SM\SM_LongDesk.uasset
Assets\SM\SM_MazeCorner.uasset
Assets\SM\SM_MazeHall.uasset
Assets\SM\SM_MazePressurePad.uasset
Assets\SM\SM_MediaPlayer.uasset
Assets\SM\SM_MiniHouse.uasset
Assets\SM\SM_MiniLightPole.uasset
Assets\SM\SM_MiniSetBG.uasset
Assets\SM\SM_MiniStreet.uasset
Assets\SM\SM_MiniStreetCorner.uasset
Assets\SM\SM_Mouse.uasset
Assets\SM\SM_MushroomGroup.uasset
Assets\SM\SM_MushroomPickup.uasset
Assets\SM\SM_PantheonTop.uasset
```

```
Assets\SM\SM_Pen.uasset
Assets\SM\SM_Pillar.uasset
Assets\SM\SM_PirateShip.uasset
Assets\SM\SM_Planet.uasset
Assets\SM\SM_Plate.uasset
Assets\SM\SM_PressurePad.uasset
Assets\SM\SM_RingedPlanet.uasset
Assets\SM\SM_RockPath_Flat.uasset
Assets\SM\SM_Rockpath_Incline.uasset
Assets\SM\SM_Rockpath_Steep.uasset
Assets\SM\SM_RoundTable.uasset
Assets\SM\SM_ScaffoldCorner.uasset
Assets\SM\SM_Scaffolding.uasset
Assets\SM\SM_ScifiPillar.uasset
Assets\SM\SM_Scifi_Corner.uasset
Assets\SM\SM_Scifi_Door.uasset
Assets\SM\SM_Scifi_Wall5.uasset
Assets\SM\SM_Scifi_WallMid.uasset
Assets\SM\SM_Scifi_WallShort.uasset
Assets\SM\SM_Scifi_Wall_Long.uasset
Assets\SM\SM_Scifi_Wall_Short.uasset
Assets\SM\SM_Setlight.uasset
Assets\SM\SM_SetUFO.uasset
Assets\SM\SM_SoundproofWall_Lrg.uasset
Assets\SM\SM_SoundproofWall_Sml.uasset
Assets\SM\SM_SoundStage.uasset
Assets\SM\SM_Stair.uasset
Assets\SM\SM_SteeringWheel.uasset
Assets\SM\SM_Stereo.uasset
Assets\SM\SM_TallFern.uasset
Assets\SM\SM_Telescope.uasset
Assets\SM\SM_Tower.uasset
Assets\SM\SM_TowerDoor.uasset
Assets\SM\SM_Trapdoor.uasset
Assets\SM\SM_TreePlatform.uasset
Assets\SM\SM_TV.uasset
Assets\SM\SM_VineNarrow.uasset
Assets\SM\SM_VinePickup.uasset
Assets\SM\SM_VineWide.uasset
Assets\SM\WorldGridMaterial.uasset
Assets\SM\Pirate_Ship\SM_AnchorBeam.uasset
Assets\SM\Pirate_Ship\SM_Beams.uasset
Assets\SM\Pirate_Ship\SM_BowSprit.uasset
Assets\SM\Pirate_Ship\SM_CageFront.uasset
Assets\SM\Pirate_Ship\SM_CanonWindow.uasset
Assets\SM\Pirate_Ship\SM_CaptainsQuarters.uasset
Assets\SM\Pirate_Ship\SM_CrowsNest.uasset
Assets\SM\Pirate_Ship\SM_Forecastle.uasset
Assets\SM\Pirate_Ship\SM_ForeMast.uasset
Assets\SM\Pirate_Ship\SM_ForeMastCross.uasset
Assets\SM\Pirate_Ship\SM_ForeMastLadder.uasset
Assets\SM\Pirate_Ship\SM_ForeMast_Sails.uasset
Assets\SM\Pirate_Ship\SM_FrontRail.uasset
Assets\SM\Pirate_Ship\SM_FrontWall.uasset
Assets\SM\Pirate_Ship\SM_Grate.uasset
Assets\SM\Pirate_Ship\SM_GunDeck.uasset
Assets\SM\Pirate_Ship\SM_Hull.uasset
```

```
Assets\SM\Pirate_Ship\SM_Keel.uasset
Assets\SM\Pirate_Ship\SM_LowerDeck.uasset
Assets\SM\Pirate_Ship\SM_Maindeck.uasset
Assets\SM\Pirate_Ship\SM_MainDeckStairs.uasset
Assets\SM\Pirate_Ship\SM_MainMast.uasset
Assets\SM\Pirate_Ship\SM_MainMastCross.uasset
Assets\SM\Pirate_Ship\SM_MainMastLadder.uasset
Assets\SM\Pirate_Ship\SM_MainMastSails.uasset
Assets\SM\Pirate_Ship\SM_MizzenMast.uasset
Assets\SM\Pirate_Ship\SM_MizzenMastCross.uasset
Assets\SM\Pirate_Ship\SM_MizzenMastLadder.uasset
Assets\SM\Pirate_Ship\SM_Mizzenmast_Sails.uasset
Assets\SM\Pirate_Ship\SM_Railing.uasset
Assets\SM\Pirate_Ship\SM_RearWall.uasset
Assets\SM\Pirate_Ship\SM_Rigging.uasset
Assets\SM\Pirate_Ship\SM_Rudder.uasset
Assets\SM\Pirate_Ship\SM_SideWall.uasset
Assets\SM\Pirate_Ship\SM_Stairs.uasset
Assets\SM\Pirate_Ship\SM_UpperDeck.uasset
Assets\SM\Pirate_Ship\SM_UpperdeckRear.uasset
Assets\SM\Pirate_Ship\SM_Windows.uasset
Assets\SpeedTree\DeadTree\Broadleaf_Desktop_Forest.uasset
Assets\SpeedTree\DeadTree\Broadleaf_Desktop_Forest_Billboard_Billboard_Mat.uasset
Assets\SpeedTree\DeadTree\Broadleaf_Desktop_Forest_Billboard_Color_Tex.uasset
Assets\SpeedTree\DeadTree\Broadleaf_Desktop_Forest_Billboard_Normal_Tex.uasset
Assets\SpeedTree\DeadTree\Broadleaf_Desktop_Forest_Branch_Mat.uasset
Assets\SpeedTree\DeadTree\Broadleaf_Desktop_Forest_Color_Tex.uasset
Assets\SpeedTree\GrassClump\GrassClump_Billboard_Billboard_Mat.uasset
Assets\SpeedTree\GrassClump\GrassClump_Billboard_Color_Tex.uasset
Assets\SpeedTree\GrassClump\GrassClump_Billboard_Normal_Tex.uasset
Assets\SpeedTree\GrassClump\GrassClump_Billboard_Subsurface_Tex.uasset
Assets\SpeedTree\GrassClump\GrassClump_Color_Tex.uasset
Assets\SpeedTree\GrassClump\GrassClump_Leaf_Mat.uasset
Assets\SpeedTree\GrassClump\GrassClump_Normal_Tex.uasset
Assets\SpeedTree\GrassClump\GrassClump_Subsurface_Tex.uasset
Assets\SpeedTree\GrassClump\Grass_Clump_Lrg.uasset
Assets\SpeedTree\GrassClump\Grass_Clump_Sml.uasset
Assets\SpeedTree\GrassClump\JungleGrass.uasset
Assets\SpeedTree\GrassClump\Sci_fi_Billboar.uasset
Assets\SpeedTree\GrassClump\Sci_fi_Grass.uasset
Assets\SpeedTree\GrassClump\SM_Grass.uasset
Assets\SpeedTree\GrassClump\SM_JungleGrass.uasset
Assets\SpeedTree\JungleTree\JungleTree.uasset
Assets\SpeedTree\JungleTree\JungleTree_Billboard_Billboard_Mat.uasset
Assets\SpeedTree\JungleTree\JungleTree_Billboard_Color_Tex.uasset
Assets\SpeedTree\JungleTree\JungleTree_Billboard_Normal_Tex.uasset
Assets\SpeedTree\JungleTree\JungleTree_Branch_Mat.uasset
Assets\SpeedTree\JungleTree\JungleTree_Color_Tex.uasset
Assets\SpeedTree\JungleTree\JungleTree_Leaf_Mat.uasset
Assets\SpeedTree\JungleTree\JungleTree_Normal_Tex.uasset
Assets\T\T_Bubble.uasset
Assets\T\T_Footprint_N.uasset
Assets\T\T_Footprint_O.uasset
Assets\T\AlienHead\AlienHead_Normal.uasset
Assets\T\Baguette\Baguette_Normal.uasset
Assets\T\BoomMic\BoomMic_Normal.uasset
Assets\T\Coconut\Coconut_Normal.uasset
```

```
Assets\T\Crystal\Crystal_A.uasset
Assets\T\Crystal\Crystal_E.uasset
Assets\T\Crystal\Crystal_N.uasset
Assets\T\Crystal\Crystal_ORM.uasset
Assets\T\CrystalPickup\N_CrystalPickup.uasset
Assets\T\CrystalTre\CrystalTree_A.uasset
Assets\T\CrystalTre\CrystalTree_N.uasset
Assets\T\CrystalTre\CrystalTree_ORM.uasset
Assets\T\Dirt\M_CaveDirt_AORM.uasset
Assets\T\Dirt\M_CaveDirt_basecolor.uasset
Assets\T\Dirt\M_CaveDirt_normal.uasset
Assets\T\ElephantLeaf\ElephantLeaf_D.uasset
Assets\T\ElephantLeaf\ElephantLeaf_N.uasset
Assets\T\ElephantLeaf\ElephantLeaf_ORM.uasset
Assets\T\Guudo\Guudo_A.uasset
Assets\T\Guudo\Guudo_E.uasset
Assets\T\Guudo\Guudo_N.uasset
Assets\T\Guudo\Guudo_ORM.uasset
Assets\T\Guudo_Goo\Guudo_Goo_A.uasset
Assets\T\Guudo_Goo\Guudo_Goo_N.uasset
Assets\T\JellySplat\T_JellySplat.uasset
Assets\T\JellySplat\T_JellySplat_N.uasset
Assets\T\KrakenTentacle\KrakenTentacle_normal.uasset
Assets\T\Lilypads\Lilypads_Normal.uasset
Assets\T\Lilypads\Lilypads_OcclusionRoughnessMetallic.uasset
Assets\T\Lilypads\LilyPads_Opacity.uasset
Assets\T\Log\Log_Normal.uasset
Assets\T\MushroomPickup\N_MushroomPickup.uasset
Assets\T\VinePickup\N_VinePickup.uasset
Blueprints\Ball_BP.uasset
Blueprints\Door_BP.uasset
Blueprints\GameMode_BP.uasset
Blueprints\Goo_Footprint_BP.uasset
Blueprints\Guudo_BP.uasset
Blueprints\PirateShip_BP.uasset
Blueprints\PressurePlate_BP.uasset
Blueprints\PushPlate_BP.uasset
Blueprints\Switch_BP.uasset
Blueprints\CustomisationScreen\BodySelectionTool_BP.uasset
Blueprints\CustomisationScreen\CharacterCustomisationHUD.uasset
Blueprints\CustomisationScreen\CustomisationGameMode_BP.uasset
Blueprints\CustomisationScreen\ModularGuudo_BP.uasset
CameraShake\Consume.uasset
CameraShake\WalkShake.uasset
EasterEggs\greyboxue4.uasset
Geometry\Meshes\1M_Cube.uasset
Geometry\Meshes\1M_Cube_Chamfer.uasset
Geometry\Meshes\CubeMaterial.uasset
Geometry\Meshes\TemplateFloor.uasset
HUD\BasicHUD.uasset
HUD\Credits.uasset
HUD\InventoryWidget_BP.uasset
HUD\LoadingScreen.uasset
HUD\MainMenuWidget.uasset
HUD\MediaPlayer.uasset
HUD\SilouetteWidgetHUD.uasset
HUD\Customisation\CustomisationWidget_BP.uasset
```

```
HUD\Customisation\ItemImageWidget_BP.uasset
HUD\HUDImages\Face.uasset
HUD\UI\Button_01.uasset
HUD\UI\Guudo.uasset
HUD\UI\Guudo_Font.uasset
HUD\UI\UI_ArrowButton_L.uasset
HUD\UI\UI_ArrowButton_R.uasset
HUD\UI\UI_ArrowHover_L.uasset
HUD\UI\UI_ArrowHover_R.uasset
HUD\UI\UI_ArrowPress_L.uasset
HUD\UI\UI_ArrowPress_R.uasset
HUD\UI\UI_CantGrow.uasset
HUD\UI\UI_CustomisationPanel.uasset
HUD\UI\UI_GuudoTitle.uasset
HUD\UI\UI_InnerCircle.uasset
HUD\UI\UI_LargeGrow.uasset
HUD\UI\UI_LoadEars.uasset
HUD\UI\UI_LoadingBase.uasset
HUD\UI\UI_LrgGrowBG.uasset
HUD\UI\UI_MidCircle.uasset
HUD\UI\UI_NormalGrow.uasset
HUD\UI\UI_NormGrowBG.uasset
HUD\UI\UI_OmniLogo.uasset
HUD\UI\UI_OuterCircle.uasset
HUD\UI\UI_SilhouetteBG.uasset
HUD\UI\UI_SmallGrow.uasset
HUD\UI\UI_SmlGrowBG.uasset
HUD\UI\Silhouette\M_AlienHeadSilhouette.uasset
HUD\UI\Silhouette\M_AlienHead_Silhouette.uasset
HUD\UI\Silhouette\M_PlanetSilhouette.uasset
HUD\UI\Silhouette\UI_AlienHead_Silhouette.uasset
HUD\UI\Silhouette\UI_Camera_Silhouette.uasset
HUD\UI\Silhouette\UI_Coconut_Silhouette.uasset
HUD\UI\Silhouette\UI_Planet_Silhouette.uasset
HUD\UI\Silhouette\UI_TV_Silhouette.uasset
HUD\UI\Silhouette\UI_UFO_Silhouette.uasset
Levels\00_LevelTemplates\03_ProgrammingLevels\RichardsTestMap_BuiltData.uasset
Levels\00_LevelTemplates\04_Menus\CharacterCustomisation_BuiltData.uasset
Levels\00_LevelTemplates\04_Menus\MainMenu_BuiltData.uasset
Levels\00_LevelTemplates\05_OpenWorld\Guudo_AdvGreybox_BuiltData.uasset
Levels\00_LevelTemplates\05_OpenWorld\JungleTree.uasset
Levels\00_LevelTemplates\05_OpenWorld\Maze_BP.uasset
Levels\00_LevelTemplates\05_OpenWorld\MitchMap_BuiltData.uasset
Movies\Guudo_MediaPlayer.uasset
Movies\Guudo_MediaPlayer_Video.uasset
Movies\Guudo_MediaPlayer_Video_Mat.uasset
Movies\Guudo_OP.uasset
Sound\Background_Music.uasset
Sound\Consume.uasset
Sound\Customisation_BGM.uasset
Sound\MainMenu_BGM.uasset
Sound\Selection1.uasset
Sound\Selection2.uasset
Sound\Selection3.uasset
Sound\00_CantGrow\CantGrow.uasset
Sound\01_Grow\GrowCue.uasset
Sound\01_Grow\NGrow1.uasset
```

```
Sound\01_Grow\NGrow2.uasset
Sound\01_Grow\NGrow3.uasset
Sound\01_Grow\SGrow1.uasset
Sound\01_Grow\SGrow2.uasset
Sound\02_Shrink\BShrink1.uasset
Sound\02_Shrink\BShrink2.uasset
Sound\02_Shrink\NShrink1.uasset
Sound\02_Shrink\NShrink2.uasset
Sound\02_Shrink\ShrinkCue.uasset
Sound\03_Jump\JumpCue.uasset
Sound\03_Jump\Jump_1.uasset
Sound\03_Jump\Jump_2.uasset
Sound\03_Jump\Jump_3.uasset
Sound\03_Jump\Jump_4.uasset
Sound\03_Jump\Jump_5.uasset
Sound\03_Jump\Jump_6.uasset
Sound\04_Footprints\FootstepCue.uasset
Sound\04_Footprints\Footsteps-10.uasset
Sound\04_Footprints\Footsteps-11.uasset
Sound\04_Footprints\Footsteps-12.uasset
Sound\04_Footprints\Footsteps-2.uasset
Sound\04_Footprints\Footsteps-3.uasset
Sound\04_Footprints\Footsteps-4.uasset
Sound\04_Footprints\Footsteps-5.uasset
Sound\04_Footprints\Footsteps-6.uasset
Sound\04_Footprints\Footsteps-7.uasset
Sound\04_Footprints\Footsteps-8.uasset
Sound\04_Footprints\Footsteps-9.uasset
Sound\04_Footprints\Footsteps.uasset
Sound\05_Eating\Eating1.uasset
Sound\05_Eating\Eating_Cue.uasset
Sound\UI\GABBY\Audio_Track__15_.uasset
Sound\UI\JOSH\Audio_Track-2__1_.uasset
Sound\UI\JOSH\Audio_Track-3.uasset
Sound\UI\JOSH\Audio_Track-3__1_.uasset
Sound\UI\JOSH\Audio_Track-4.uasset
Sound\UI\JOSH\Audio_Track__1_.uasset
Sound\UI\JOSH\Audio_Track__2_.uasset
Sound\UI\MIKA\Audio_Track-10.uasset
Sound\UI\MIKA\Audio_Track-11.uasset
Sound\UI\MIKA\Audio_Track-12.uasset
Sound\UI\MIKA\Audio_Track-13.uasset
Sound\UI\MITCH\Audio_Track-3-2.uasset
Sound\UI\MITCH\Audio_Track-4__1_.uasset
Sound\UI\MITCH\Audio_Track-5__1_.uasset
Sound\UI\MITCH\Audio_Track-7__1_.uasset
Sound\UI\MITCH\Audio_Track-8__1_.uasset
Sound\UI\MITCH\Audio_Track-9__1_.uasset
Sound\UI\PACO\Audio_Track-15.uasset
Sound\UI\PACO\Audio_Track-16.uasset
Sound\UI\PACO\Audio_Track-17.uasset
Sound\UI\PACO\Audio_Track-20.uasset
Sound\UI\PACO\Audio_Track.uasset
Sound\UI\PACO\Paco.uasset
Sound\UI\PACO\PacoConstruction.uasset
Sound\UI\WARNING\Audio_Track-14.uasset
Sound\UI\WARNING\Audio_Track-19.uasset
```

Sound\UI\WARNING\Audio_Track-6.uasset
Sound\UI\WARNING\Audio_Track-7.uasset
Sound\UI\WARNING\Audio_Track-8.uasset
Sound\UI\WARNING\Audio_Track-9.uasset
Sound\UI\WARNING\JWarning.uasset
Sound\UI\WARNING\Warning.uasset