

The

Ultimate

Programming

Guide

Kizuna Games

**Programmers:**

**Hayashi** Tensai

Lim Kheng Wei

Table of Contents

[1.0 Setting up the scene 2](#_Toc507858088)

[1.1 Hierarchy 2](#_Toc507858089)

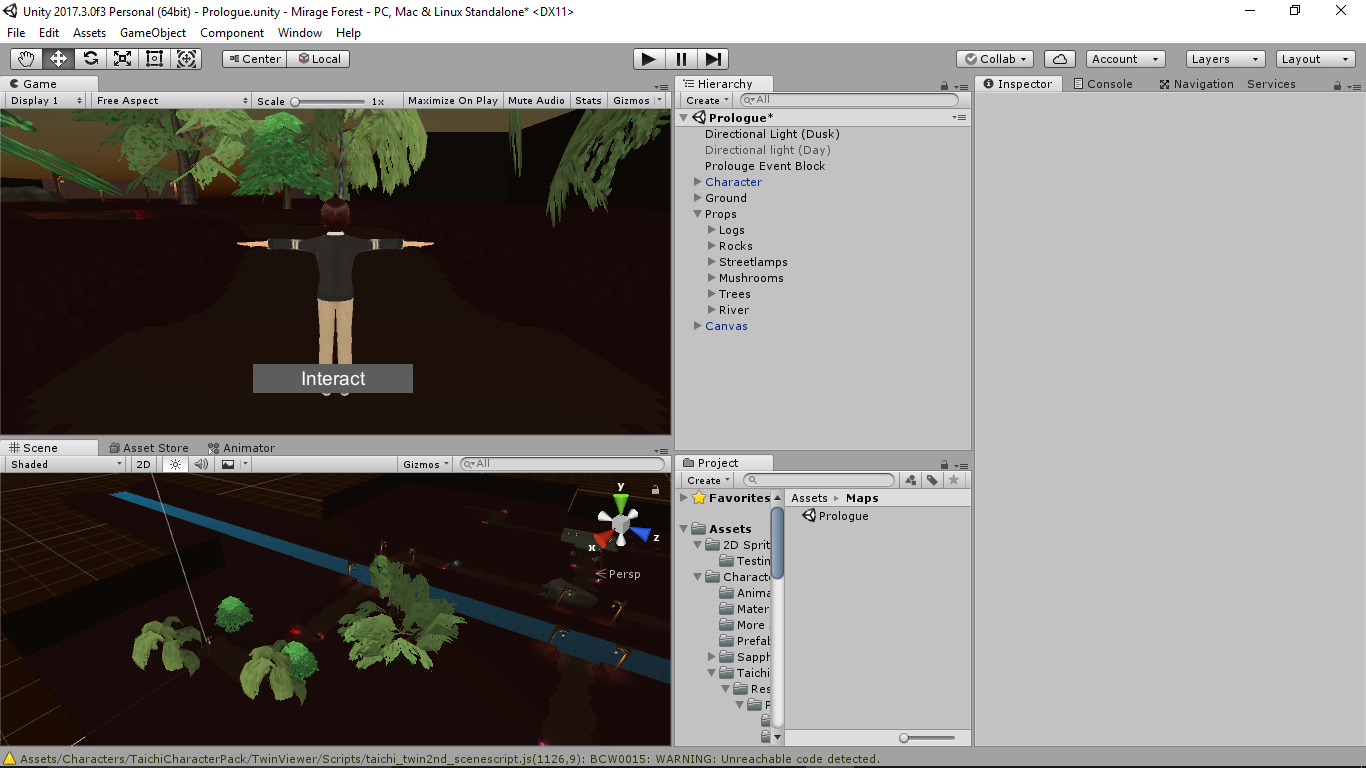
[1.2 Terrain 2](#_Toc507858090)

[2.0 Characters 3](#_Toc507858091)

[2.1 Composition 3](#_Toc507858092)

[3.0 Scripts 4](#_Toc507858093)

# Setting up the scene



## Hierarchy

These are the main sections in hierarchy:

* Ground

All the paths and ground should be all placed in this section.

* Props

Any props is placed in here and each of the prop type should have its own section, e.g.: Trees, Water, Rocks, etc.

* Canvas

Pull out the canvas prefab and place in the hierarchy.

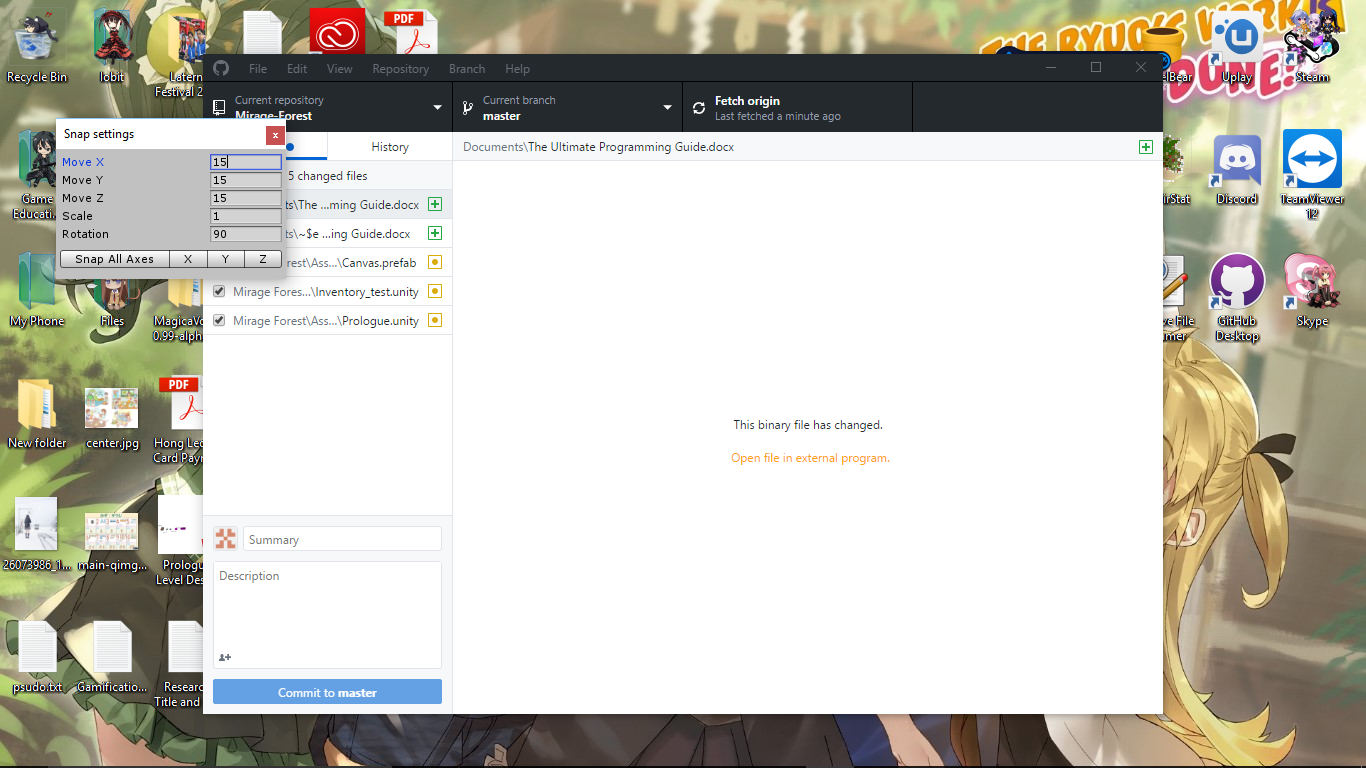
## Terrain

Tagging

Make sure all the ground prefabs are tag “Ground” before use, the tag is used to detect for jumping.

Building the Terrain

As stated earlier, all the ground prefab needs to be place in the “Ground” section of the Hierarchy.

Now let’s get into building the terrain, the first block should be placed at (0, 0, 0) as a reference point for the first block. For next block use the control key to snap the terrain one after another.

Use these snap settings when building the terrain as the terrain is 15 units wide:

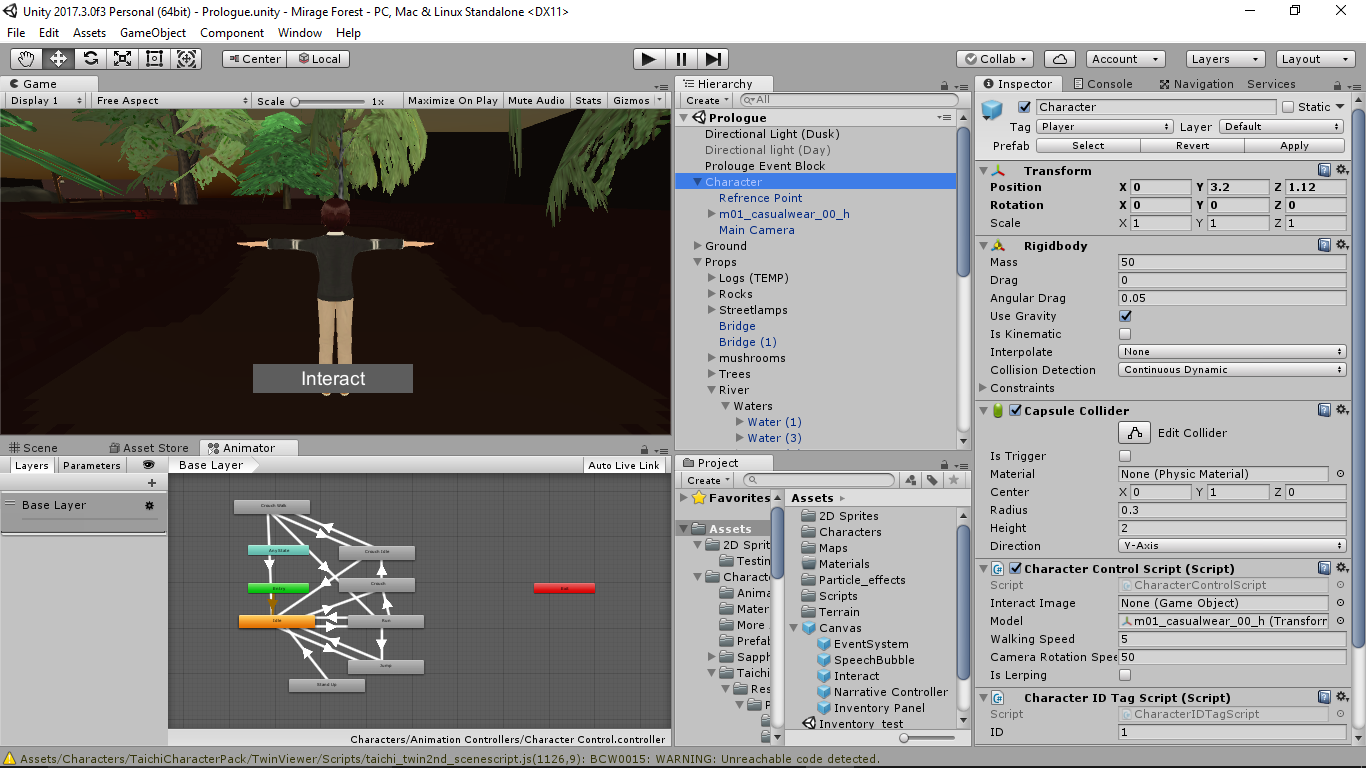
\*\*\*Additional Info:

The snap settings for the water is 21.5 units

# Characters

## Composition

These are parts that the character is composed of, I’ll break this down as we go along.



Character

There will be a few components in this part:

* Capsule Collider

These is of course used for collision purposes so that the character doesn’t just drop to eternity.

* Rigidbody

This is there so that gravity and collision works :)

* Character related scripts

All the scripts for the character should be placed here and NOT in the children.

The scripts involved are as follows:

* CharacterAnimationScript
  + CharacterAnimation enum
    - This determines the current animation that should be played for the model
  + ChangeAnimation( ) function
    - This function is called every time a change of animation is requested.
    - “if(charaAnim != characterAnimation && !anim.IsInTransition(0))”

This line is placed so that if it won’t restart the animation if it’s the same animation

The part where the code is not allowed to run during transition needs to be reviewed.

* CharacterControlScript
  + This script controls the following
    - Movement (WASD)
    - Rotation (Mouse)
    - Camera Reset (Q)
    - Collision Detection for ground and intractable objects
* CharacterIDTagScript
  + This script keeps only one int value for the script to keep the ID of the character to be identified by the narrative scripts.
  + This script needs review as it inherits MonoBehaviour and doesn’t use it at all (Memory Wastage).

Reference Point

This will act as reference point to determine where the speech bubble will spawn.

m01\_casualwear\_00\_h

* Name will vary for different characters
* This will be where the actual mesh is.

Main Camera

The camera that follows the character, only Kaizen will have the main camera.

# Sound

|  |  |  |  |
| --- | --- | --- | --- |
| Type of Sound | Global Sound | Area Based Sound | Local (Character) Sounds |
| Method of execution | Control by a global controlling script | TBC | Run by a script attached to the character |
| Examples | BGM  UI Buttons | Water Sounds  Wind  Sounds created by other objects | Character Movement Sounds  Character Collision Sounds |

# Scripts

This section is placed here for each of jumping in to specific script

* [CharacterAnimationScript](The%20Ultimate%20Programming%20Guide.docx#CharacterAnimationScript)
* [CharacterControlScript](#CharacterControlScript)
* [CharacterIDTagScript](#CharacterIDTagScript)
* EventTriggerScript
* InteractionScript
* InventoryScript
* ItemDatabaseScript
* ItemInteractionScript
* NarrativeControlScript
* NarrativeDatabaseScript
* ProlougeEventScript
* SpiritInteractionScript