

Polyart Zombies v.0.9.0



Thank you for purchasing Polyart Zombies, a collection of unique polygonal zombie characters with limbs dismemberment, ragdoll and animations.

What is included?



Themed zombies: Chef, cop, construction worker, doctor, hooker, military, nun, teen boy, teen girl and waiter



Generic or casual zombies

Asset	Count
Zombies Characters	20
Average Triangle Count	2k
Animations	10
Materials	1
Textures	2
Scripts (.cs)	2
Scenes	2
Custom Shaders	0

Demo scenes overview

Demo_1_Scene.unity

This scene showcases all characters with few animations. Over time their limbs fall apart and eventually the main body fall to the ground with ragdoll physics.

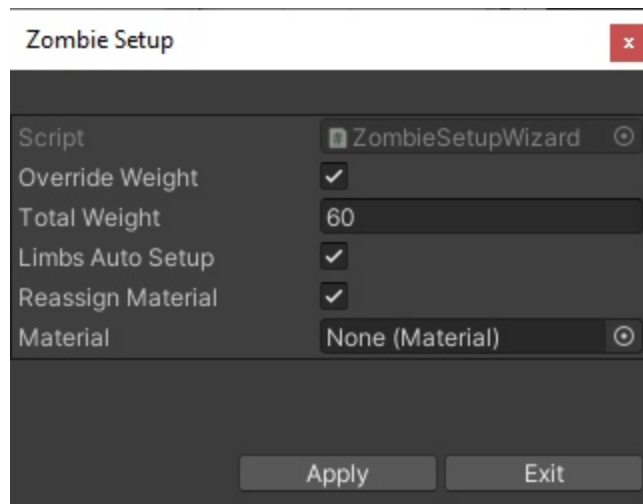
The **DemoZombieGroup.cs** script controls this scene. Our function calls of interest here are found inside the **UpdateInterval()**. There we access the **zombie.limbs** list items and a simple call to **...limbs[i].Detach()** will separate the part from the main body and become part of the physics simulation. Then by calling **zombie.EnableRagdoll()** the ragdoll physics will be activated for the whole character. In this example script the body will fall if one of the legs or the head is detached.

Demo_2_animations.unity

This scene focuses on showcasing all animations. Here you'll find a character for each existing animation (10 at this moment). **DemoZombieAnimations.cs** is a simple script that applies a different animation state to each character.

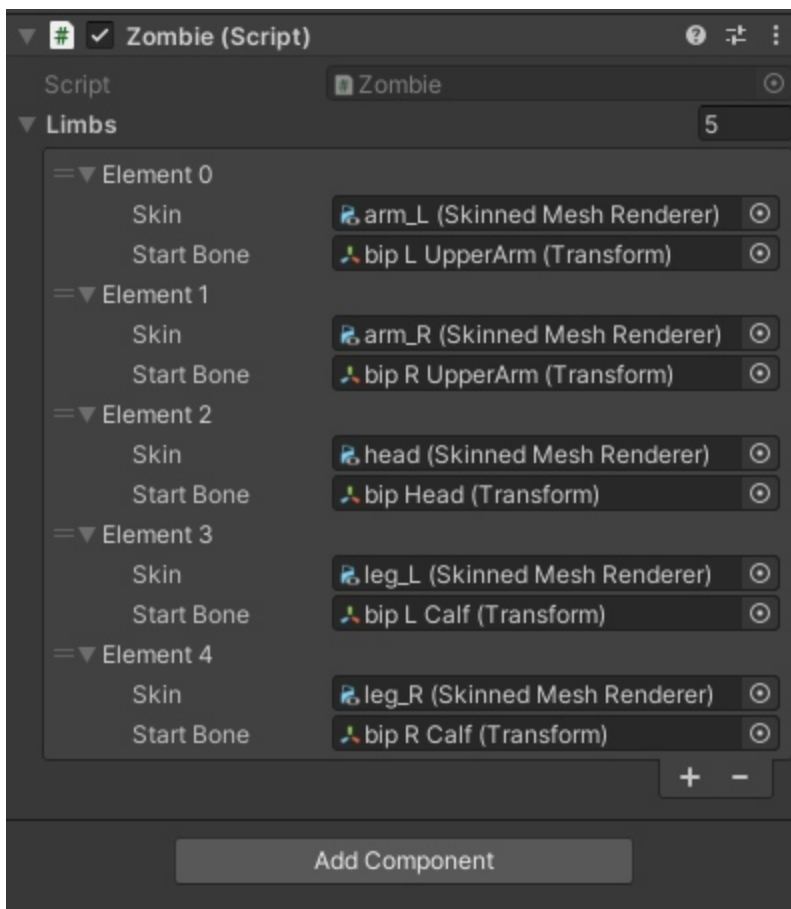
The Scripts

Editor/ZombieSetupWizard.cs



Zombie Setup is accessed via the main menu **Tools/DA/Zombie Setup**. This tool helps with adjusting the zombie properties or configuring a new or modified character that will be prepared to work with the **Zombie** component.

Zombie.cs



The **Zombie** component is the (necrotic?) heart of our zombies. Here is kept the list of detachable limbs, information that is composed by a mesh part and a bone where it starts.

Then with handy functions you can control how and when to trigger the following features:

Zombie.limbs[i].Detach() - By calling *Detach()* from one of the items of the *limbs* list make that part fall apart and get ragdoll physics activated, while the rest of the main body keeps its current animation state.

Zombie.EnableRagdoll() - Activates ragdoll physics for the whole character.

Zombie.DisableRagdoll() - Deactivates ragdoll physics for the whole character.

Starting Guide

The only complex task the scripts do is to pre-process the skinned mesh to prepare for the dismemberment system to work. Everything else, at this moment, is up to you depending on the type of game you are building.

- To start, drag any of the character prefabs to the scene.
- Configure the Animator component to satisfy your needs.
- Code your own scripts that interact with the Zombie component to trigger limbs detachments or body ragdolls when you decide.

Roadmap

- Add a new set of ~10 special zombies, like "tank", "boomer", "witch", etc... I can listen to suggestions.
- Hit detection approach. Facilitate projectile and explosion detection while knowing the part of the body affected.
- Add ~10 additional animations.
- Skinned Mesh pre-optimization joining all parts into a single mesh.
- Helper scripts to easily change eye color.
- Basic NPC AI examples.
- Simple example game.

Support and Feedback

This project is evolving, anything could be changed to improve or add more features.

Your feedback is important! If you have any comments, questions, or need assistance, please reach out at denys.almaral@gmail.com.

Let's get ready to zzzombieeeee!