



Unity Piscine - Module04

Animations and Sound

Summary: This document outlines the Module04 project for Unity Piscine.

Version: 2

Contents

I	Instructions	2
II	Foreword	3
III	Exercise 00: The Life of a Caterpillar	4
IV	Exercise 01: A Strange Environment	7
V	Exercise 02: The Sound	8
VI	Submission and Peer Evaluation	9

Chapter I

Instructions

- If you have trouble installing the required tools for your project on the 42 computers, you may use a virtual machine. In this case, you must:
 - Install the virtual machine software on your computer.
 - Install the operating system of your choice.
 - Install the necessary tools for your project.
 - Ensure that you have enough space on your session to install all of it;
 - Have everything installed before the evaluation.
- Only this page will serve as reference. Do not rely on rumors.
- Carefully read the entire document before starting.
- Your exercises will be evaluated by your fellow piscine participants.
- This document is your reference. Do not blindly trust demos or example pictures, which may include unnecessary additions.
- Got a question? Ask the peer to your right. If not, try the one on your left.
- By Odin, by Thor! Use your brain!!!



Intra shows the date and time when your repositories close. This also marks the beginning of the peer-evaluation period for that piscine day. The peer-evaluation lasts exactly 24 hours. After that, any missing evaluations will be scored as 0.

Chapter II

Foreword

A platformer game (often shortened to platformer and sometimes called a jump-and-run game) is a subgenre of action video games in which the core objective is to navigate a character through a level to reach a specific goal.

Platformers typically feature uneven terrain and suspended platforms that require jumping and climbing to traverse. Additional acrobatic actions, such as swinging from vines, using grappling hooks, wall jumping, air dashing, gliding, or bouncing on trampolines or springboards, may also be involved.

Games where jumping is entirely automated, such as some 3D entries in *The Legend of Zelda* series, are generally excluded from the genre.

The genre began with the 1980 arcade game *Space Panic*, which included ladders but lacked jumping.

Donkey Kong, released in 1981, established the template for what were initially known as climbing games. It inspired many clones and successors, such as *Miner 2049er* (1982).

During the height of the genre's popularity in the late 1980s and early 1990s, platformers accounted for between a quarter and a third of all console games. However, by 2006, their market share had declined to around 2%, down from 15% in 1998. Despite this, platformers continue to thrive commercially, with several titles achieving millions of units sold.


Source: [Platform Game - Wikipedia](#).



As with Module 02, your work in this module will be used in the next one. So, by the end of Module 04, remember to export your assets so they can be imported into Module 05.

Chapter III

Exercise 00: The Life of a Caterpillar

	Exercise :
Exercise 00: The Life of a Caterpillar	
Turn-in directory : <code>unityModule04</code>	
Required elements : The "Stage1" scene and anything relevant assets	
Forbidden functions : None	

A platformer!

Today, you will create a simple 2D platformer. The goal isn't to build a complex level but to focus on animations and sound.

Start by creating a new 2D Unity project for Module 04 and set up a basic environment in the "Stage1" scene. All necessary assets are provided in this module:

- You need a platform for your player to move on.
- Add some obstacles, but don't spend too much time on them.

Now that you have a basic scene, you can add your player:

- In the Assets folder, you will find a charming caterpillar.
- Before animating the caterpillar, create a `PlayerController` script and attach it to the caterpillar in your scene.
- For now, keep it simple — the player should be able to move horizontally and jump.
- Make sure the caterpillar cannot double-jump or wall-jump.
- Don't forget the `Rigidbody2D` and `Collider2D` components ;).

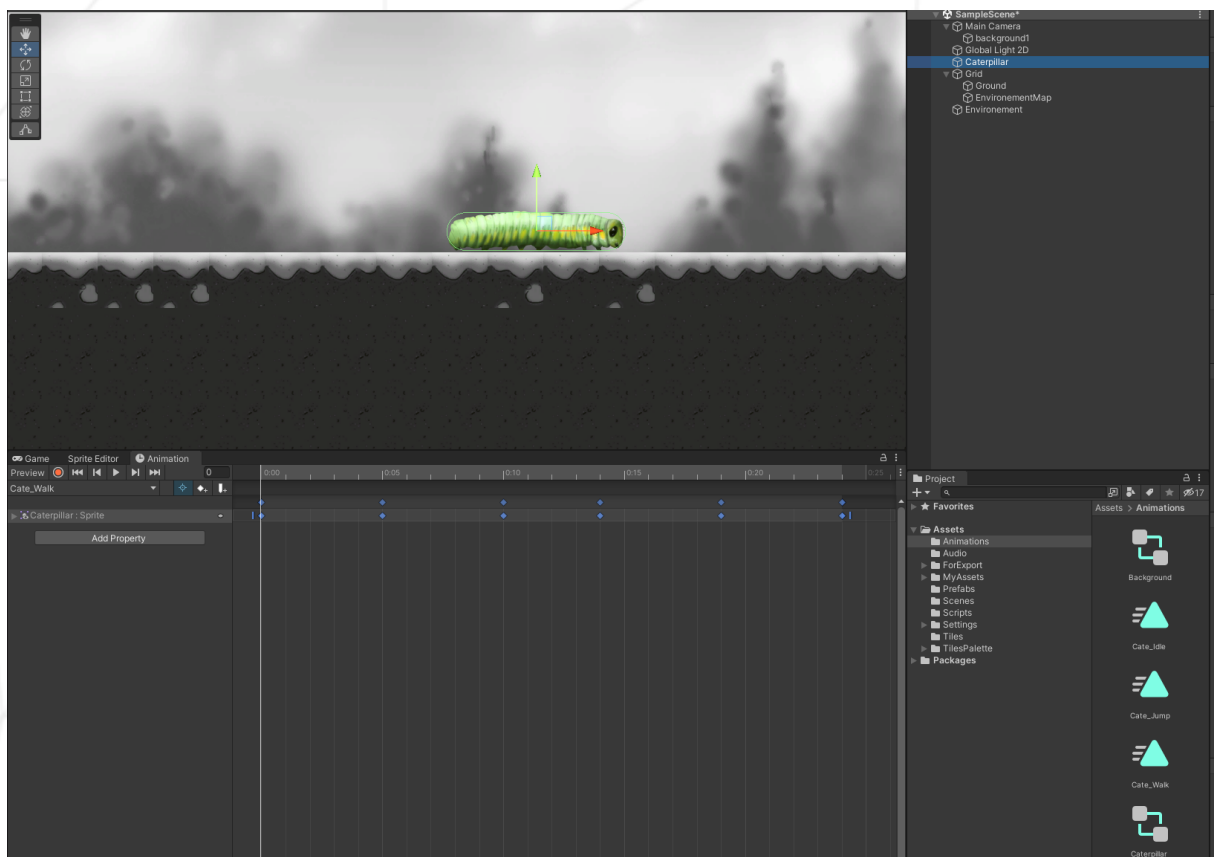
Now it's time to animate the caterpillar!

The caterpillar has six animations:

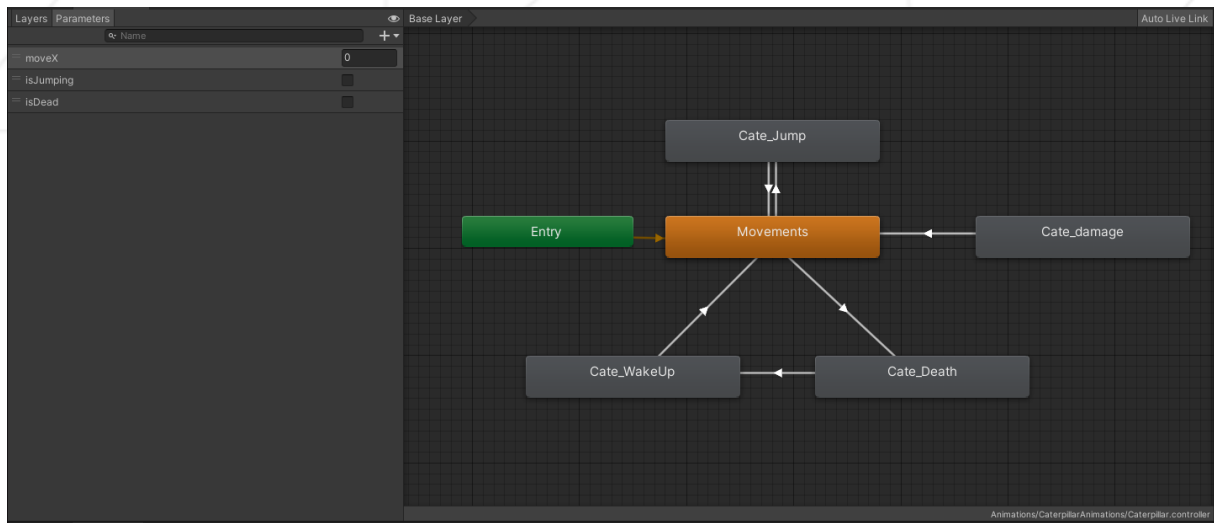
- Idle
- Walk
- Jump
- Take Damage
- Defeated
- Respawn

Create a new “Animations” folder to store all your animation clips.

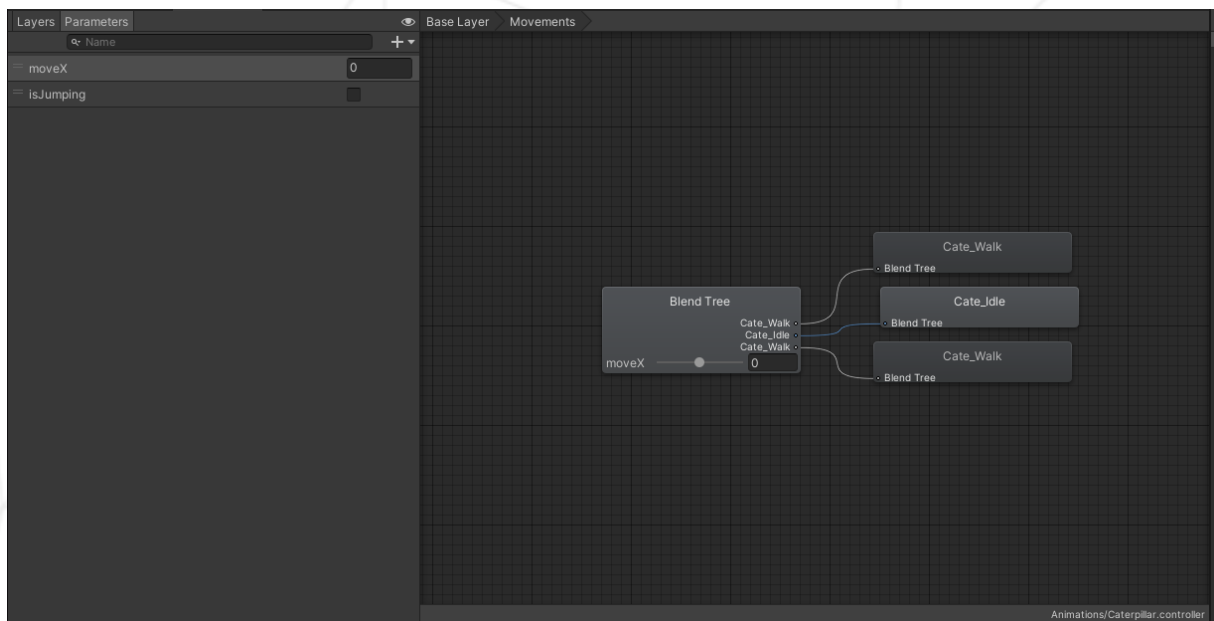
For example, your Walk AnimationClip might look like this:



For the BaseLayer in the Animator, you might have something like this:



And the movement transitions in your BaseLayer:




Of course, these are just examples, there are many ways to configure your Animator.



Refer to the Animation View and Animator sections of the Unity documentation for help.

Chapter IV

Exercise 01: A Strange Environment

	Exercise :
Exercise 01: A Strange Environment	
Turn-in directory : <code>unityModule04</code>	
Required elements : The "Stage1" scene and anything relevant assets	
Forbidden functions : None	

Now that your little caterpillar is ready, it's time to bring the environment to life.

You must include at least two types of animated background objects:

- Trees that move, falling leaves, etc.

You must also include at least two objects that animate when interacting with the caterpillar:


- A vine that moves when the caterpillar gets close.
- A cactus that throws poisonous jelly when approached.

Your caterpillar should also respond to the environment:

- When hit by a vine or jelly, play the damage animation.
- Give your caterpillar 3 HP. When its HP reaches 0, play the defeat animation.
- The defeat animation should be accompanied by a fade from transparent to black.
- After that, the caterpillar should respawn at the beginning of the level with a wake-up animation and a fade from black to transparent.

Chapter V

Exercise 02: The Sound

	Exercise :
Exercise 02: The Sound	
Turn-in directory : <code>unityModule04</code>	
Required elements : The "Stage1" scene and anything relevant assets	
Forbidden functions : None	

Sound is a crucial component of a video game. When chosen well, sound and music help immerse the player deeper into the game world.

For this exercise, visit the Unity Asset Store and download the following asset:

[RPG Essentials Sound Effects - FREE](#)

This package contains all the sound effects you'll need for the game — except music.

Add sound throughout your project:

- Background music (you will need to find your own).
- Jump sound for the caterpillar.
- Damage sound for when the caterpillar takes a hit.
- Defeat sound.
- Respawn sound.
- Attack sound for the vine.
- Jelly throwing sound for the cactus.

Chapter VI

Submission and Peer Evaluation

Submit your assignment in your `Git` repository as usual. Only the work inside your repository will be evaluated during the defense. Don't hesitate to double-check the names of your folders and files to make sure everything is correct.



You should not upload the entire Unity project to `Git`, as this can unnecessarily increase the size of your repository.

- Make sure Unity saves as many files as possible in text format rather than binary. In Unity, go to `Edit > Project Settings > Editor`. Under `Asset Serialization`, set it to `Force Text`.
- Ensure that the `.gitignore` file automatically generated by Unity is present.



The evaluation will take place on the computer of the learner or group being evaluated.