**TEMASEK POLYTECHNIC**

**SCHOOL OF INFORMATICS & IT**

**DIPLOMA IN IMMERSIVE MEDIA & GAME DEVELOPMENT**

**AY2025/2026 APRIL SEMESTER**

**GADV (CGE2C25)**

**Introduction to Unity**

**2D Sprites & Animation**

To see the additional comments and resources, make sure you select **All Markup** in the **Review/Tracking** pane

A cartoon of a cheetah running

Description automatically generated

**Objectives**

In this worksheet, we take a slight breather from programming. You’ll get back to programming in the next worksheet, but rather than code with just text output, we should make use of Unity’s strengths and learn C# in the context of game development. To start with, this means 2D, so we’ll go over the basics of 2D in Unity first.

2D is also fun!

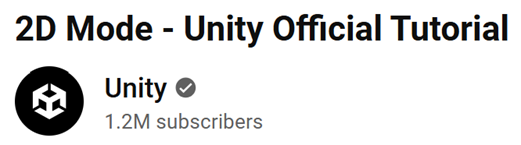
Rather than just handhold you through the concepts, you’ll watch a few official Unity videos (about 25 mins in total) and then complete some questions and practical exercises.

You might be tempted to use Chat GPT, but don’t. It’s always best to learn in context, meaning that you need to take into account how the explanation fits into the rest of the video. Chat CPT Is great for helping you learn, but it isn’t so good at understanding context from a video (yet).

Also, you should really train your brain to think for itself! 😊

**Part 1: Unity 2D Components and Tools**

By this lesson, you must have looked at the following official Unity videos:



* [2D Mode](https://www.youtube.com/watch?v=L3vLnEc7HTg) (7:17)
* [Sprite Type](https://www.youtube.com/watch?v=R0J1fNSoxcI) (3:59)
* [SpriteRenderer](https://www.youtube.com/watch?v=VfAYSWpf7gg) (3:43)
* [Sprite Editor](https://www.youtube.com/watch?v=gbgIA3pwpHc) (6:45)
* [Sorting Layers](https://www.youtube.com/watch?v=ZzcyREamMUo) (3:43)

They are about 25 minutes in total. They are a bit old now, but are still relevant, and cover most of what you need to know. Basic 2D workflows havent changed much, though additional tools have been added. These will be pointed out to you as you go through the worksheet.

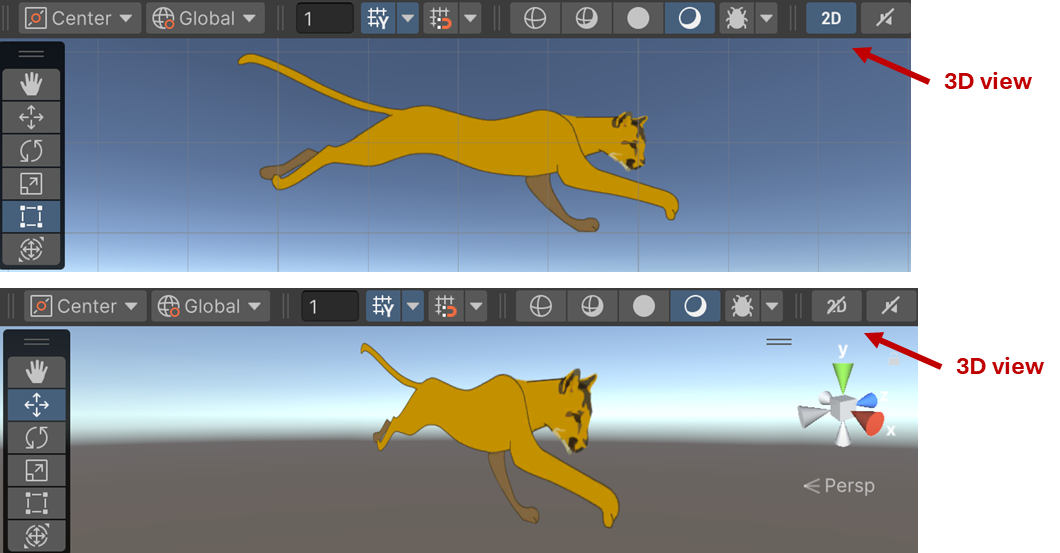
Answer all the questions below. They aren’t graded, but it’s better to get to grips with things now than wait until you start your project!

If a question requires you to complete some practical task, add a screenshot to your answer.

1. [**2D Mode**](https://www.youtube.com/watch?v=L3vLnEc7HTg) **video in Unity**

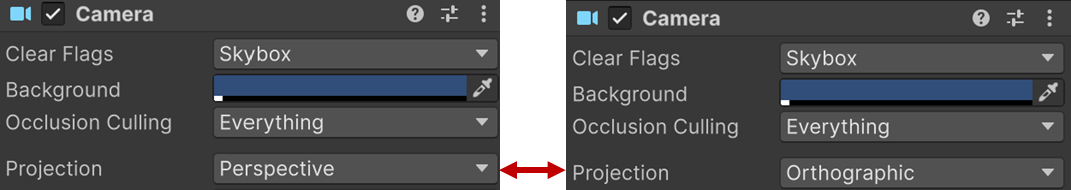
**Question 1**

Briefly describe TWO key difference between 2D and 3D mode.



**Question 2**

At 01.48, the video Introduces the idea of a **perspective** or **orthographic** camera. What does this mean?



**Question 3**

At 04.15, the video covers the various tools available for use in the Scene View. What is the difference between the **hand** tool and the **translate** tool?



1. [**Sprite Type**](https://www.youtube.com/watch?v=R0J1fNSoxcI) **video**

**Question 4**

What is the main difference between a **texture** and a **sprite**?

Look at this reference (you must get used to using the Unity documentation!):

<http://docs.unity3d.com/Manual/Textures.html>

A picture containing mollusk

Description automatically generated



**Question 5**

At (1:00) the video mentions the **Multiple** sprite import option. This is for when multiple sprite images are stored in a single image file, e.g. for a character animation, as in the zombie image above where each sprite image is a single frame in the animation.

When multiple sprites are in a single image file, this is called a **spritesheet**. Unity also has something called a **sprite atlas**. We won’t look at sprite atlases here, but you should explore these independently.

1. What is a **spritesheet**?
2. Why bother using a spritesheet – why not just use lots of single image files?

**Question 6**

Explain in your own words what the **Pixels To Units** property of a sprite represents (see 01.30 in the video).

A picture containing text, clipart, tableware, dishware

Description automatically generated

1. [**SpriteRenderer**](https://www.youtube.com/watch?v=VfAYSWpf7gg) **video**

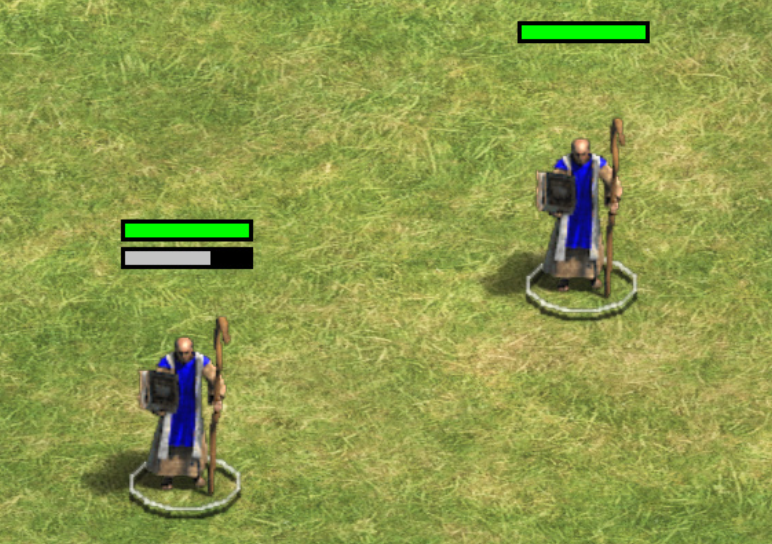
**Question 7**

At the beginning of the video, the narrator says:

“The sprite renderer component displays images that have been imported as the sprite type for use in both 2D and 3D scenes.”

Give an example of when a 2D sprite might be used in a 3D scene (not including the GUI).

Here’s a hint:



**Question 8**

What is an **alpha channel** (00.39 video time), and why is it important for a sprite?



1. [**Sprite Editor**](https://www.youtube.com/watch?v=gbgIA3pwpHc) **video**

**Question 9**

What does it mean to *slice* a spritesheet image?

A picture containing doll, dancer, toy

Description automatically generated

**Question 10**

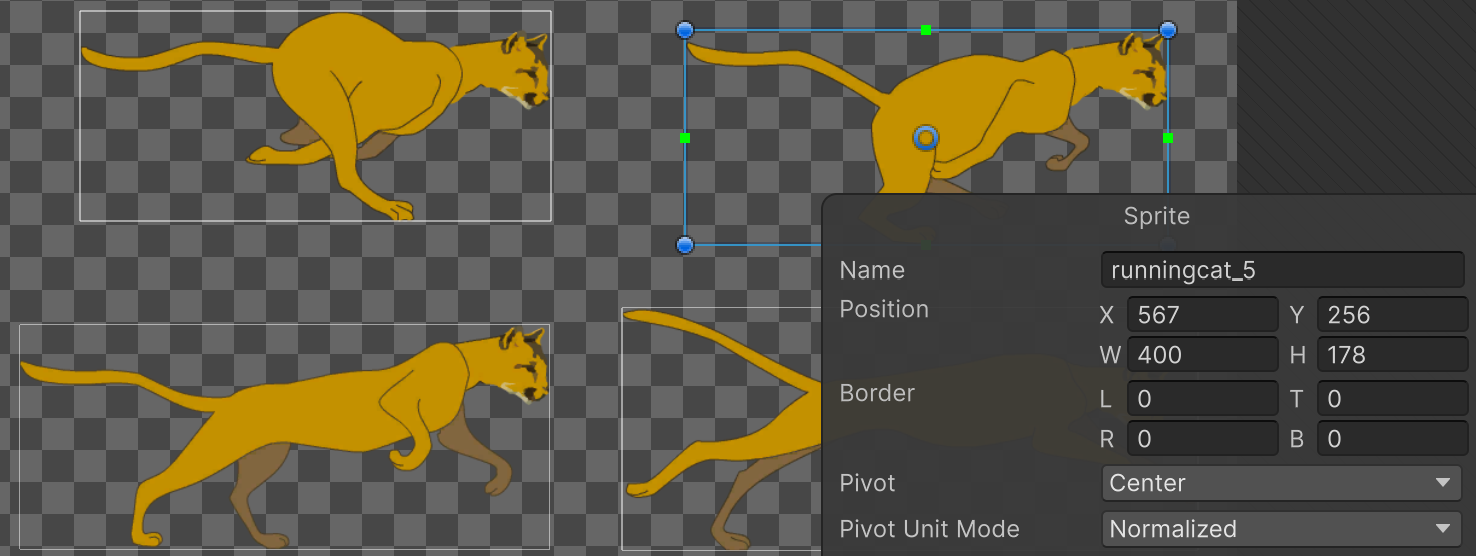
Slicing a sprite sheet is important for animation. Why?

**Part 2: Create a 2D Animation**

**2.1 Setting up the Assets**

IMPORTANT!

For newer version of Unity, sprite editing has been moved into a separate 2D Sprite package. This contains all the sprite editing tools, such as the Sprite Editor:



If you see this button, click it to install the 2D Sprite package:



1. Download the **Spritesheets.zip folder from LMS**.

The folder contains multiple sprite sheets for you to choose from. Use them to practise creating an animation of each characters in Unity.

It’s up to you to figure out how to do this! Learning how to do things by yourself (with a little bit of help, perhaps) is a critical skill in game development.



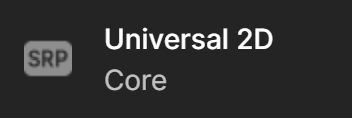
1. For this project, you will be creating a 2D Project, with the Project name **Sprite Animation**.

Open Unity Hub, and click **New project**.

Icon

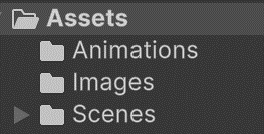
Description automatically generated

Select the **Universal** **2D Core** template, and set an appropriate project folder. Wait for the project to be created.



1. Under the **Project** View in Unity, click on the Assets Folder. Mouse right-click, select **Create** and **Folder**, and name the new folder **Images**. Then create another folder called **Animations**.

Your Assets Folder should now contain 3 folders.

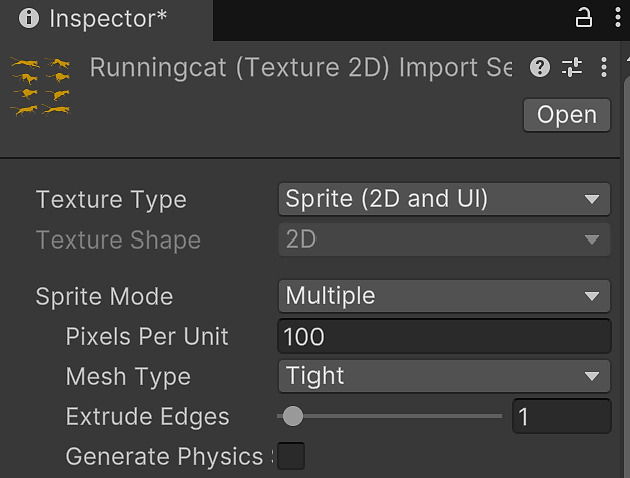


1. Unzip the **Spritesheet.zip** folder which you downloaded from LMS, and choose a sprite sheet you’d like to create an animation for.

Drag that image to the **Sprites** folder which you created earlier. Now, you can start to generate individual 2D images from the Sprite Sheet.

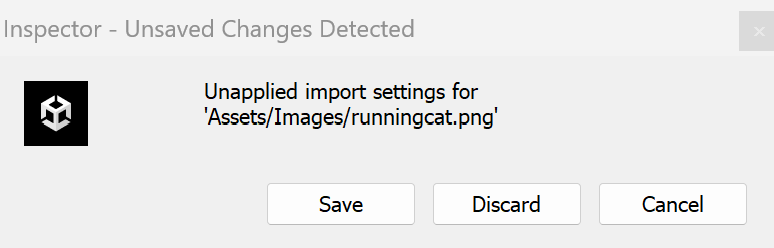
1. Click on the sprite sheet in the **Sprites** folder. The **Inspector** View will show the various fields to edit.

Ensure that the fields are setup as shown below in the **Inspector**. Without setting the fields, you won’t be able to slice the individual 2D Images.



* 1. Texture Type: **Sprite (2D and UI)**
  2. Sprite Mode: **Multiple**

1. Click **Apply**. If you don’t apply your changes, you’ll get this warning:



1. In the Inspector, click on the **Open** **Sprite Editor** button.



1. After clicking the **Sprite Editor** button, the **Sprite Editor** view will appear.

Click on the **Slice** button and select:

1. Type: **Automatic**
2. Pivot: **Center**



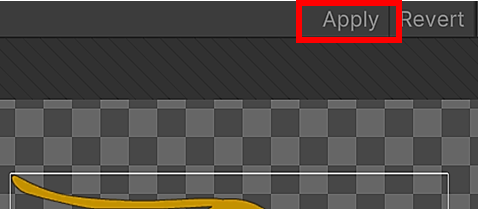
1. Now click on the **Slice** button.

You’ll notice that Unity has sliced all the individua images in the spritesheet.

Each image is given unique name for easy identification.



Click on the **Apply** button near the top.

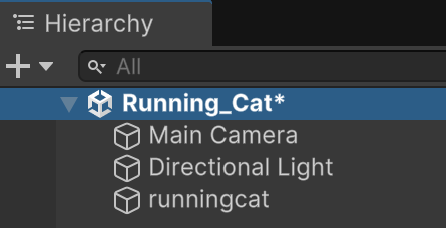


Your spritesheet is now ready for animation!

We’ll do that next.

**2.2 Creating the Animation**

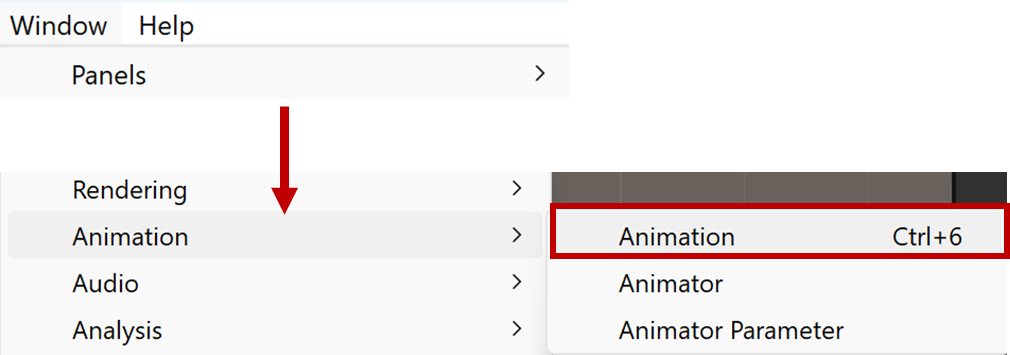
1. Create a new empty GameObject.

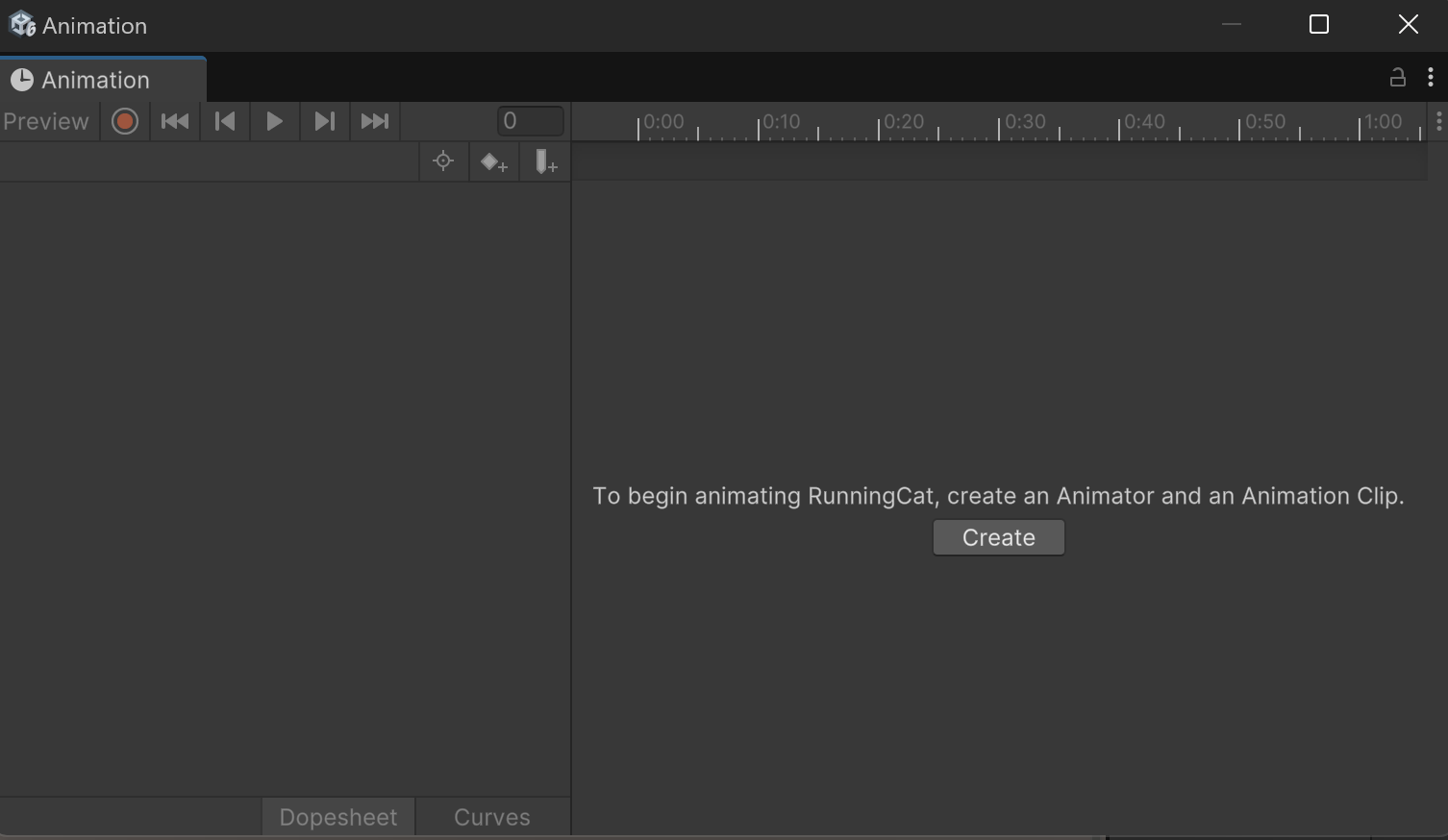


1. Rename the GameObject to **RunningCat**.

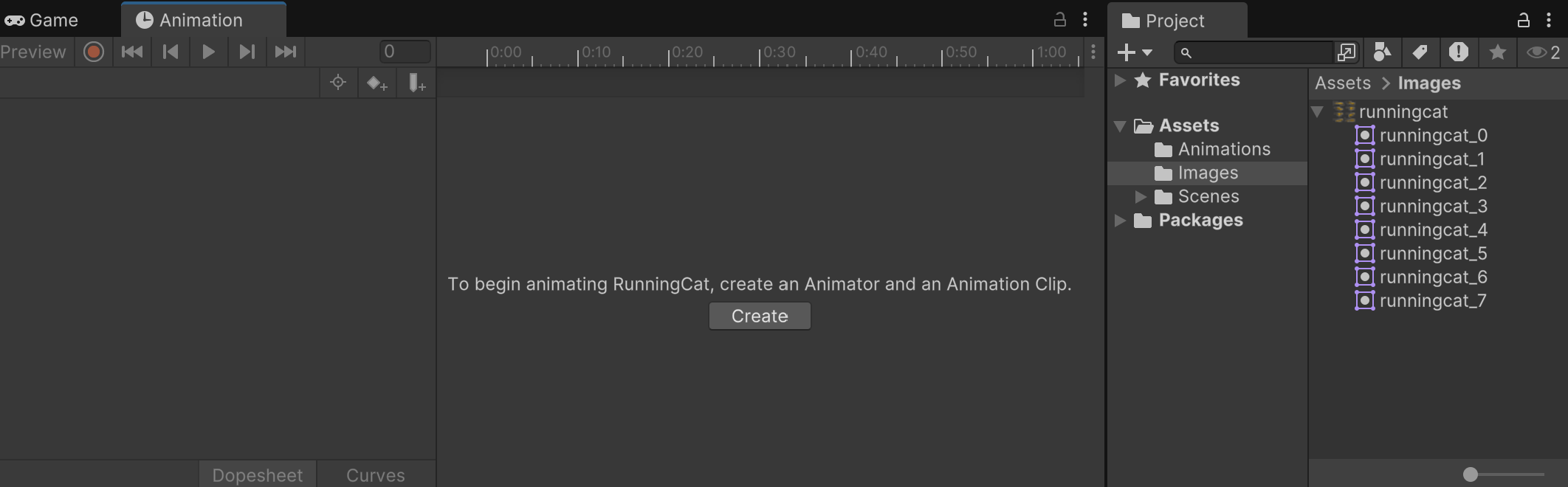
Reminder: the **\*** symbol next to the scene name means you haven’t saved your changes yet, so make sure you select **File/Save** or press **Ctrl+s** to save your work!

1. To create a 2D Animation from a spritesheet, open the **Animation** panel.





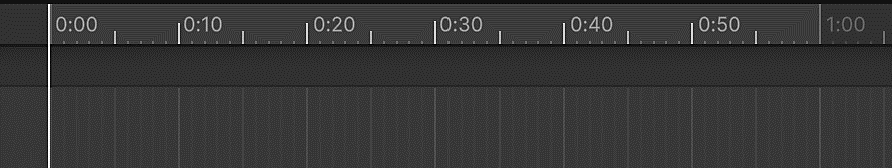
1. Set the Unity layout as shown below, to have the **Project** and **Animation** panels side by side.



1. Select the **RunningCat** GameObject in the **Hierarchy**, then click the **Create** button in the Animator panel..

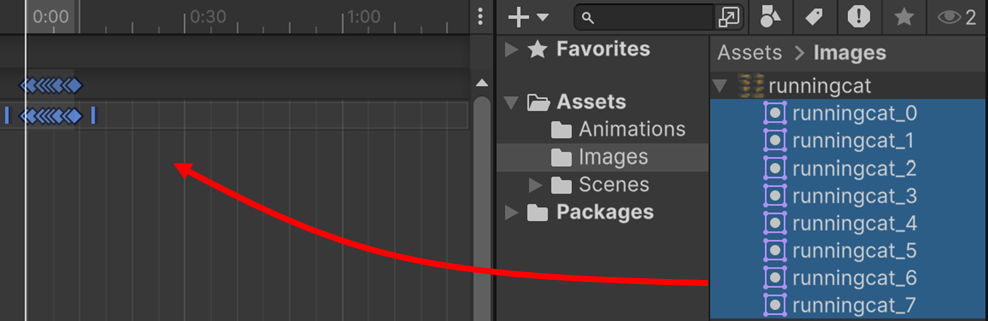
Save the new animation to the Animation folder which you created earlier. Name the animation **RunningCat**. A file called **Running\_Cat.anim** is created inside the folder.

You should see a *timeline* appear in the Animator window. This is where you will specify the keyframes for the animation.



Inside the Images folder, click the arrow next to the runningcat spritesheet and select all the sprites. (See the screenshot for step 6, below).

1. Drag the selected sprites to the Animation view.



Make sure that the **RunningCat** GameObject is selected and run the project.

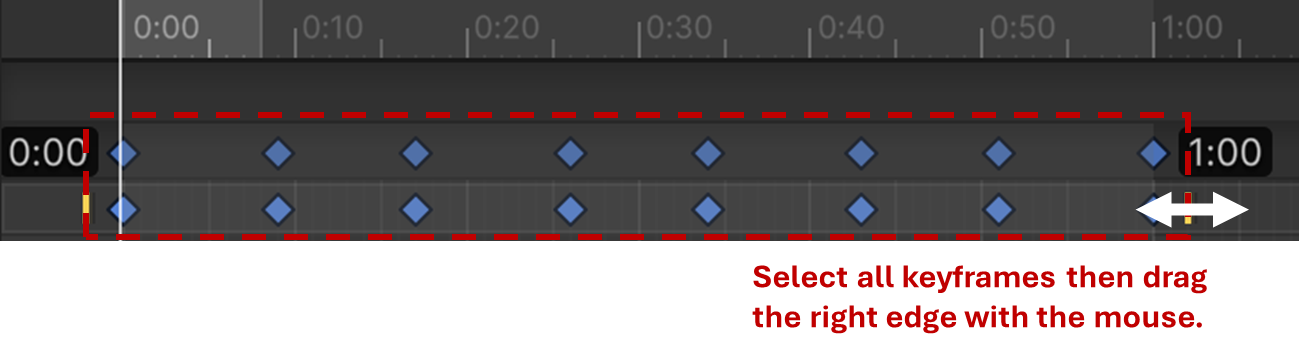
The cat runs too fast! Why?

A cartoon of a cheetah running

Description automatically generated

If you look at the timing at the top of the Animation panel, you can see that the animation lasts for only 0:10 seconds.

You can change this by selecting all the keyframes, then dragging the right edge with the mouse to the time that you want, e.g. 1:00 second.



The cat now runs at a more appropriate speed.

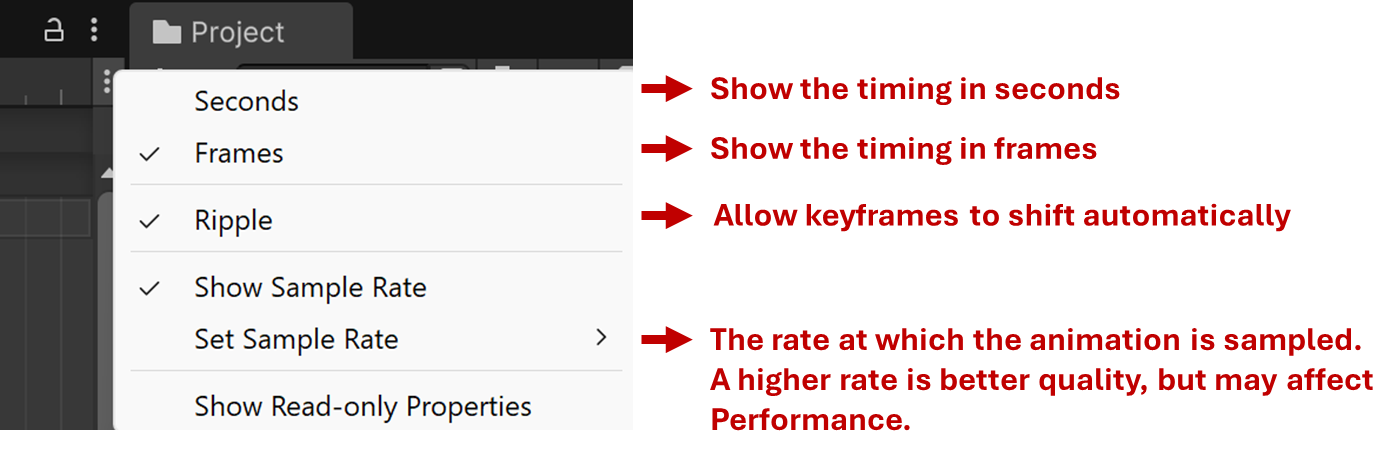
A yellow cat running with orange lines

Description automatically generated

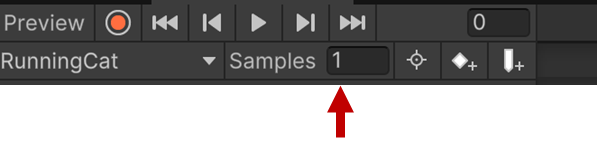
1. Click on the **properties** button in the Animation View.



Make sure that **Frames**, **Ripple** and **Show Sample Rate** are ticked.



1. Change the sample number to 1 and run the project.



What happens?

A yellow cat on a hill

Description automatically generated

This illustrates the effect of the sampling rate on the animation.

**Exercises**

1. Now complete the official Unity Learn spritesheet animation tutorial [here](https://learn.unity.com/tutorial/introduction-to-sprite-animations). Again, it’s a bit old, but still applies to the current version of Unity. It will take about 30-45 minutes.
2. Download the **Knight\_player\_1.4.zip** file from LMS. Unzip the file, which contains many different spritesheets for different character animations. Implement at least 2 of these animations.

A cartoon character with a weapon

Description automatically generated

Have fun! ☺

**Summary**

In this worksheet, you explored the fundamentals of 2D animation in Unity, focusing on sprites, keyframes, and animation workflows. Through a mix of video tutorials, guided questions, and hands-on exercises, you learned how to import spritesheets, slice sprites, and create animations using Unity's built-in tools.

By working with keyframe timing and sample rate adjustments, you gained control over animation speed and fluidity. The exercises encouraged independent problem-solving, an essential skill for game development.

Going forward, you should continue experimenting with sprite animations, Animator Controllers, and animation scripting to deepen your understanding. These concepts will also serve as a foundation for 3D animation workflows later in your learning journey.

**References**

Some additional references are given below. These will be a great hep if you use 2D animation in your project. (Some are for 3D, but the same techniques apply.)

*In this tutorial, you'll import 2D assets into Unity to create Sprites, both one at a time and from Sprite Sheets (Sprite Atlases) containing multiple assets that you will configure in the Sprite Editor.*

<https://learn.unity.com/tutorial/importing-2d-assets-into-unity-2019-3>

*The animation view allows you to build and customize your own animations within Unity. In this video you will learn how to make your own animations and use them to build creative behaviours within a scene. An older tutorial from Unity, but still relevant.*

<https://www.youtube.com/watch?v=8jDvh3hbzv4>

*Sprite Animations are animation clips that are created for 2D assets. There are various ways to create Sprite Animations. One way is to create them from a Sprite Sheet, a collection of Sprites arranged in a grid. The Sprites are then compiled into an Animation Clip that will play each Sprite in order to create the animation, much like a flipbook. Sprite Animations can also be created by using keyframe animation in Unity, via the Animation window.*

<https://learn.unity.com/tutorial/introduction-to-sprite-animations>

*Scripting is what allows us to really bring a model's animations to life. In this video you will learn how to utilize the animator class in code to control awesome animations. An older tutorial from Unity, but still relevant.*

<https://www.youtube.com/watch?v=s7EIp-OqVyk>

Search the Unity Learn website, and YouTube, for many more tutorials! Brackey’s is always a great place to start. We’ll look at this in more detail when we move on to 3D animation.