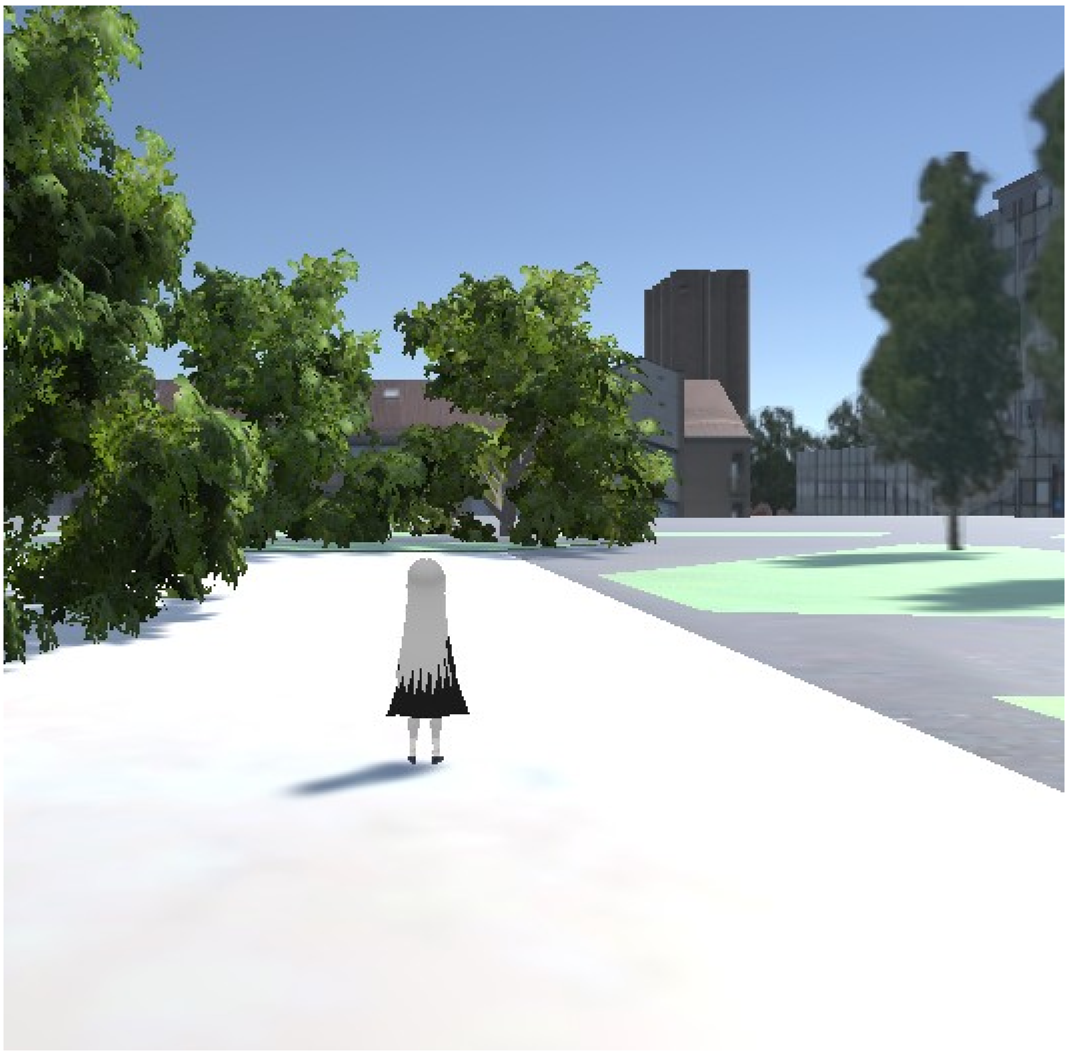
Unity Basics

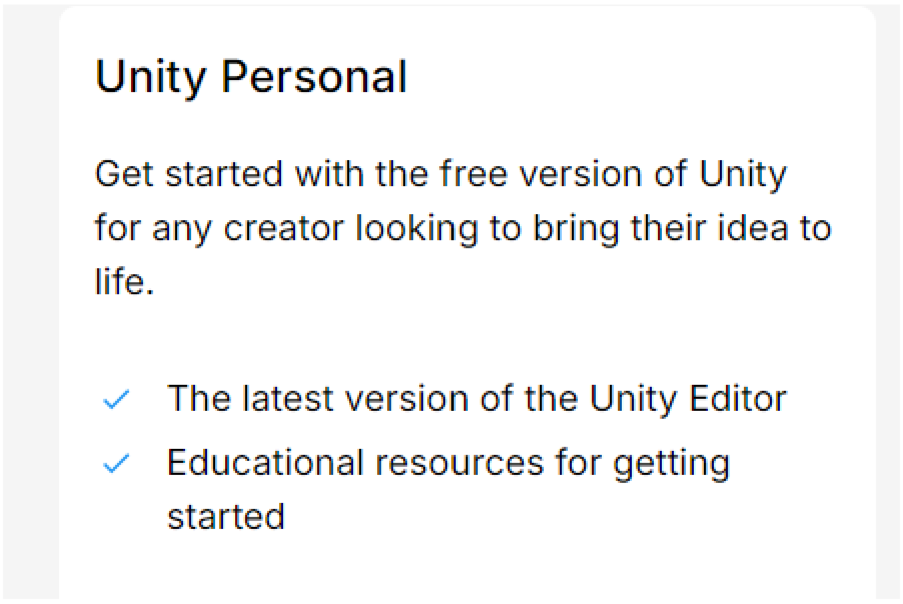


Free eBook by
Adam Higherstein

Unity tool: Installation

Go here
https://unity.com/products

And choose Unity version, Personal

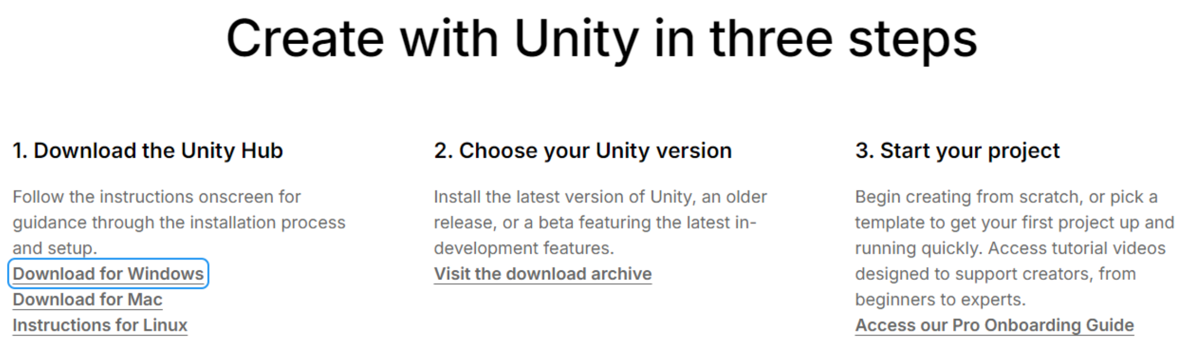


Unity Personal

Get started with the free version of Unity for any creator looking to bring their idea to life.

- ✓ The latest version of the Unity Editor
- ✓ Educational resources for getting started

Here are steps



Create with Unity in three steps

1. Download the Unity Hub

Follow the instructions onscreen for guidance through the installation process and setup.
Download for Windows
Download for Mac
Instructions for Linux

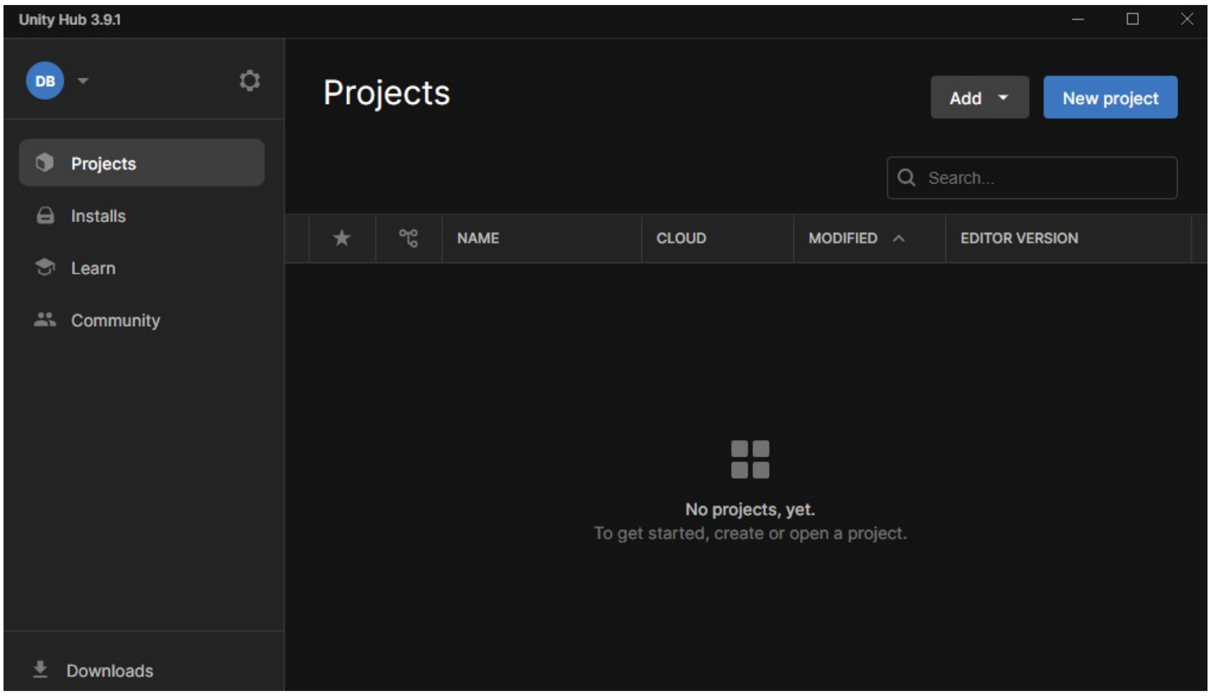
2. Choose your Unity version

Install the latest version of Unity, an older release, or a beta featuring the latest in-development features.
Visit the download archive

3. Start your project

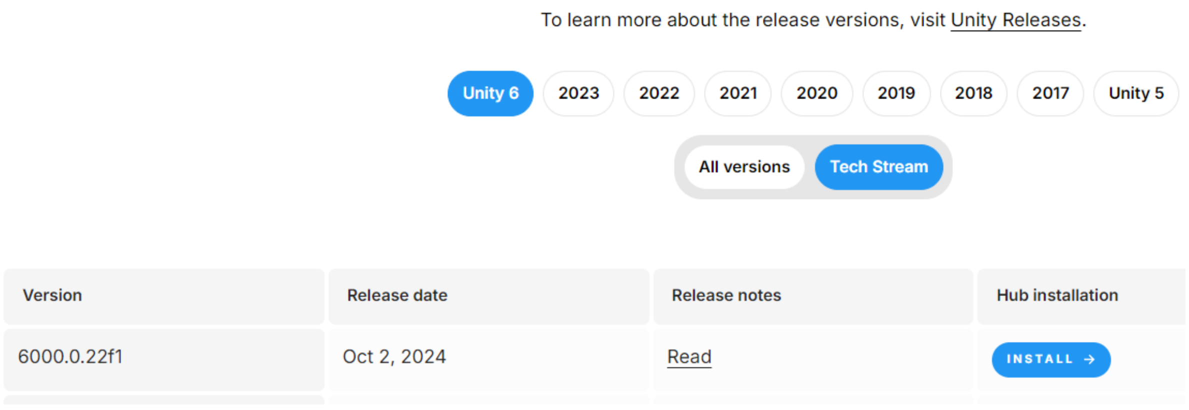
Begin creating from scratch, or pick a template to get your first project up and running quickly. Access tutorial videos designed to support creators, from beginners to experts.
Access our Pro Onboarding Guide

Install first Unity Hub



No projects, yet.
To get started, create or open a project.

Then Unity (we use version 6000, but you can install and use also 2022- or 2023-versioner...


To learn more about the release versions, visit Unity Releases.

Version	Release date	Release notes	Hub installation
6000.0.22f1	Oct 2, 2024	Read	INSTALL →

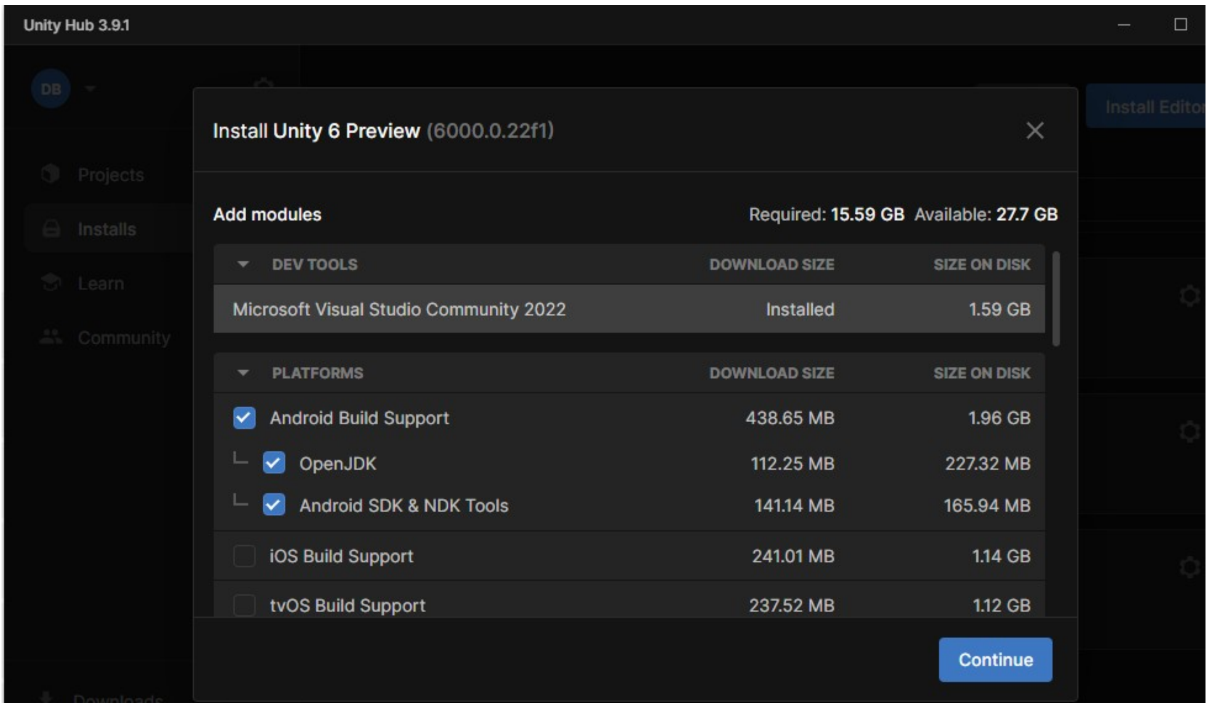
Add basic modules

e.g these are often popular

android (mobile games)

webgl (web games)

windows (standalone games)



Install Unity 6 Preview (6000.0.22f1)

Add modules — Required: 15.59 GB Available: 27.7 GB

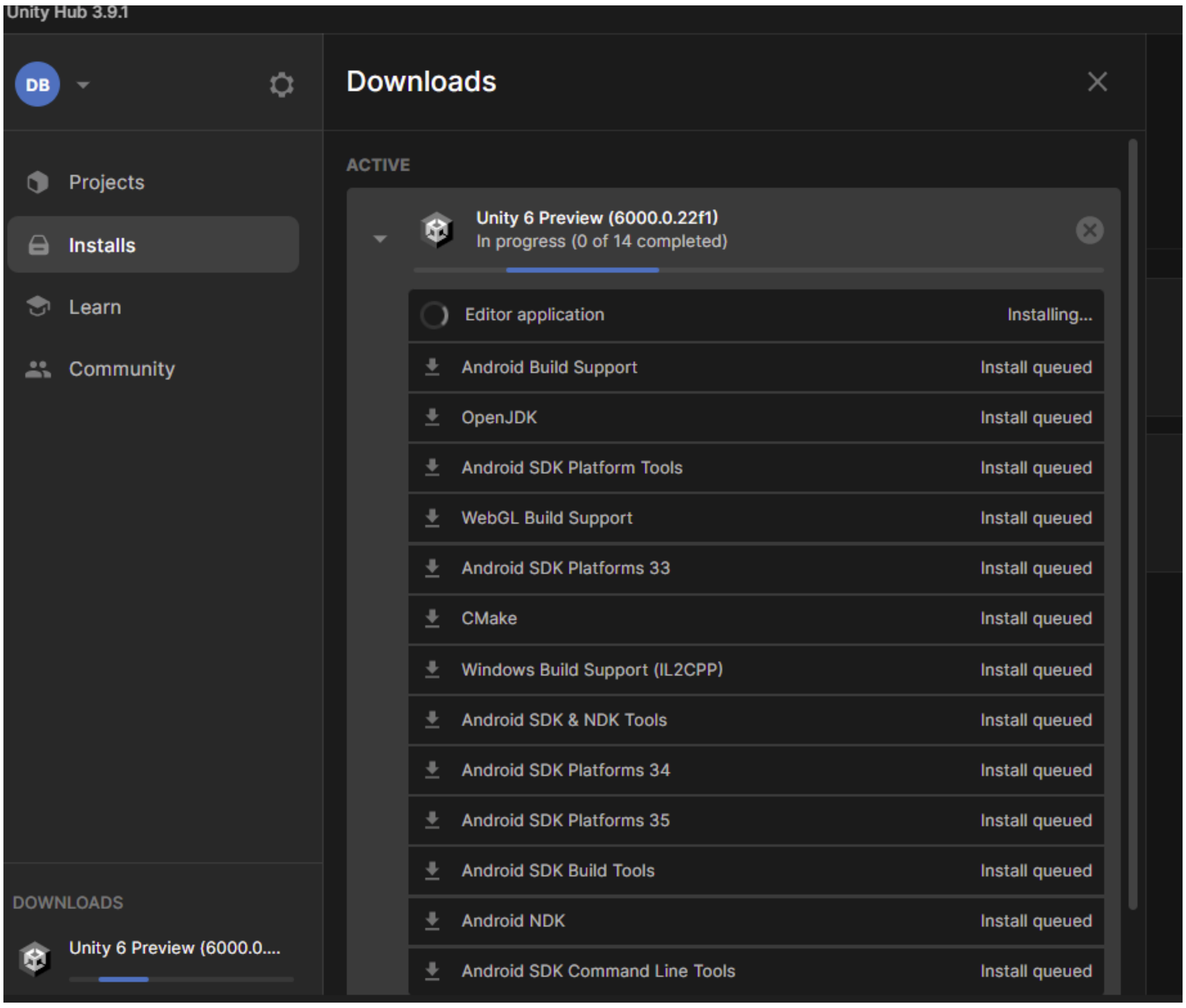
DEV TOOLS	DOWNLOAD SIZE	SIZE ON DISK
Microsoft Visual Studio Community 2022	Installed	1.59 GB

PLATFORMS	DOWNLOAD SIZE	SIZE ON DISK
✓ Android Build Support	438.65 MB	1.96 GB
✓ OpenJDK	112.25 MB	227.32 MB
✓ Android SDK & NDK Tools	141.14 MB	165.94 MB
iOS Build Support	241.01 MB	1.14 GB
tvOS Build Support	237.52 MB	1.12 GB

Add also Visual Studio

https://visualstudio.microsoft.com/downloads/

And then just install Unity editor



Downloads

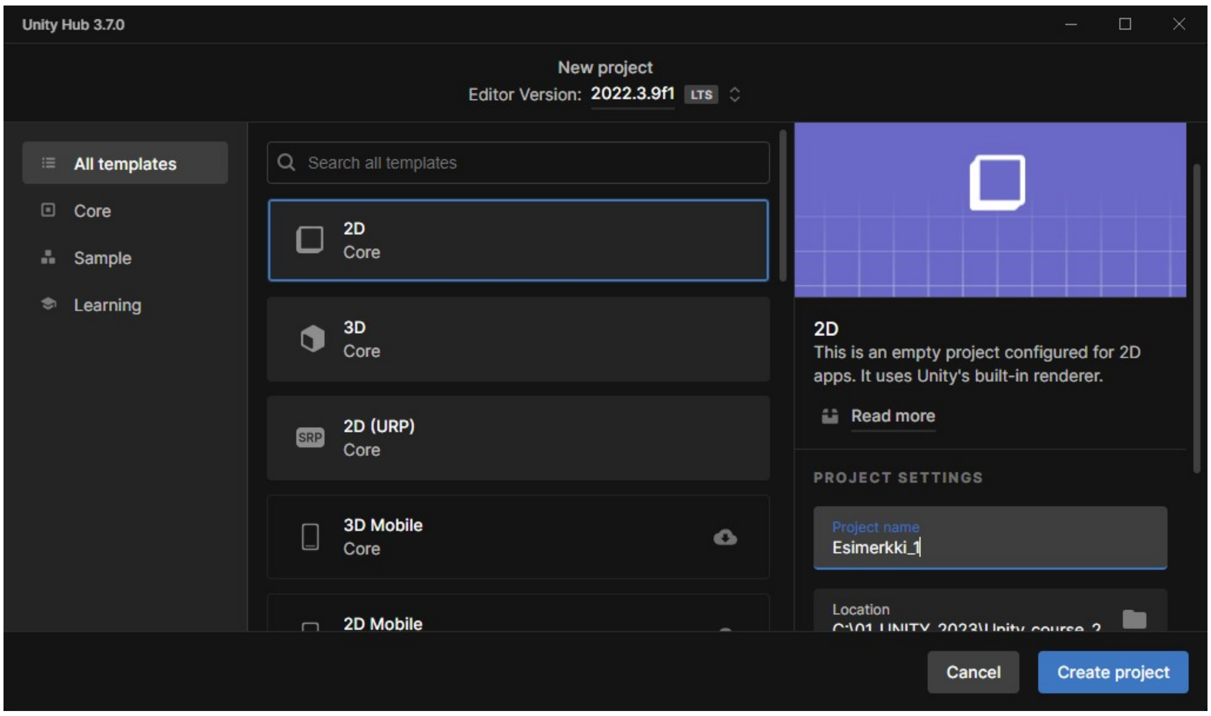
ACTIVE

Unity 6 Preview (6000.0.22f1)
In progress (0 of 14 completed)

Editor application	Installing...
Android Build Support	Install queued
OpenJDK	Install queued
Android SDK Platform Tools	Install queued
WebGL Build Support	Install queued
Android SDK Platforms 33	Install queued
CMake	Install queued
Windows Build Support (IL2CPP)	Install queued
Android SDK & NDK Tools	Install queued
Android SDK Platforms 34	Install queued
Android SDK Platforms 35	Install queued
Android SDK Build Tools	Install queued
Android NDK	Install queued
Android SDK Command Line Tools	Install queued

Start Unity Hub and a new project

Choose project type and give the name (we can start with 2D Core -version)


Unity Hub 3.7.0

New project
Editor Version: 2022.3.9f1 LTS

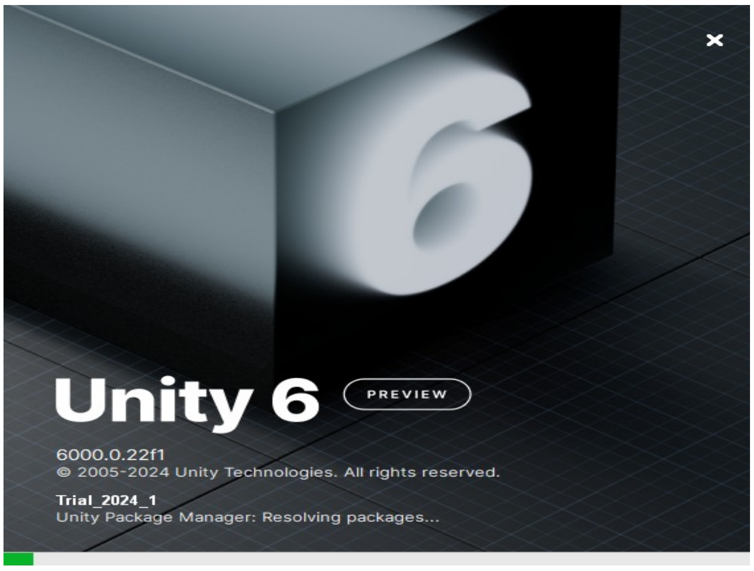
2D
This is an empty project configured for 2D apps. It uses Unity's built-in renderer.

PROJECT SETTINGS

Project name
Esimerkki_1

Location
C:\01 UNITY 2023\Unity course 2

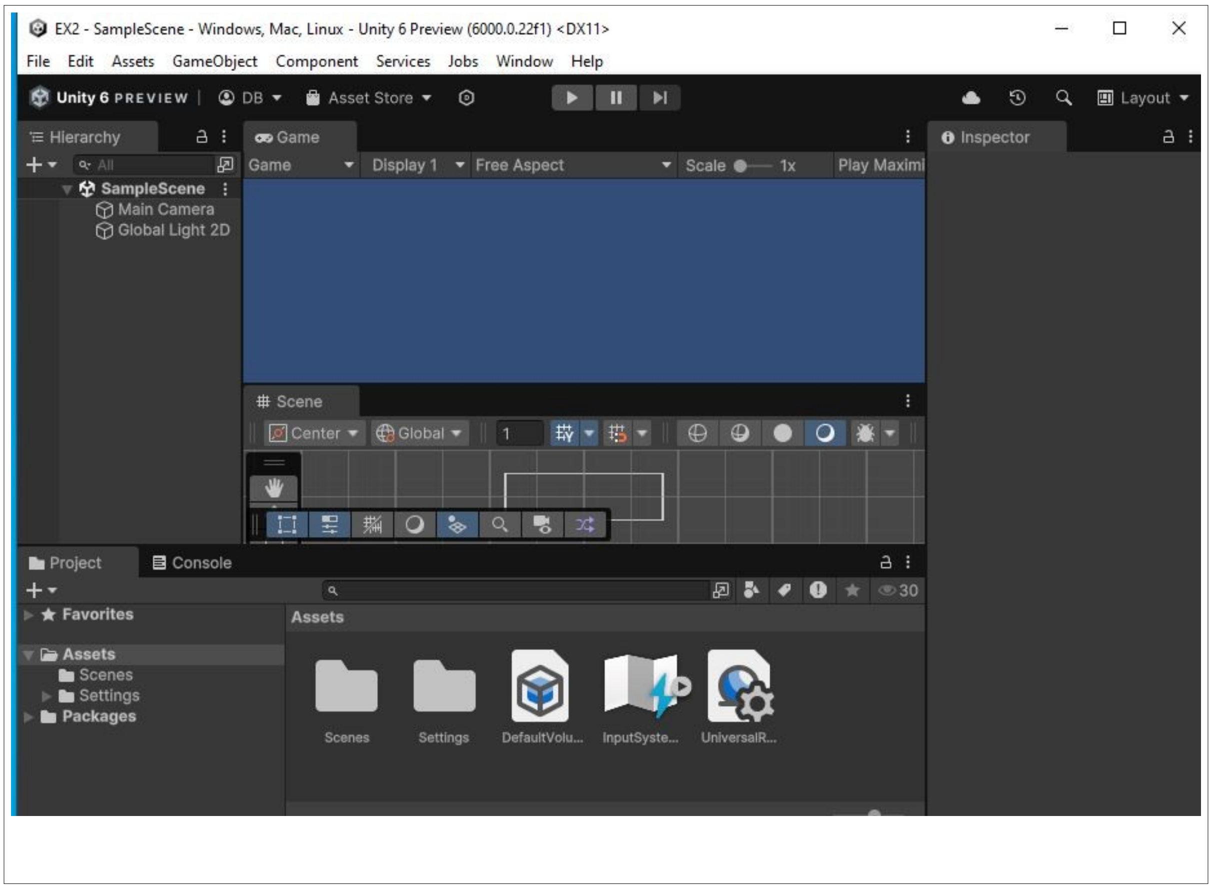
Unity version 6000.



6000.0.22f1
© 2005-2024 Unity Technologies. All rights reserved.
Trial_2024_1
Unity Package Manager: Resolving packages...

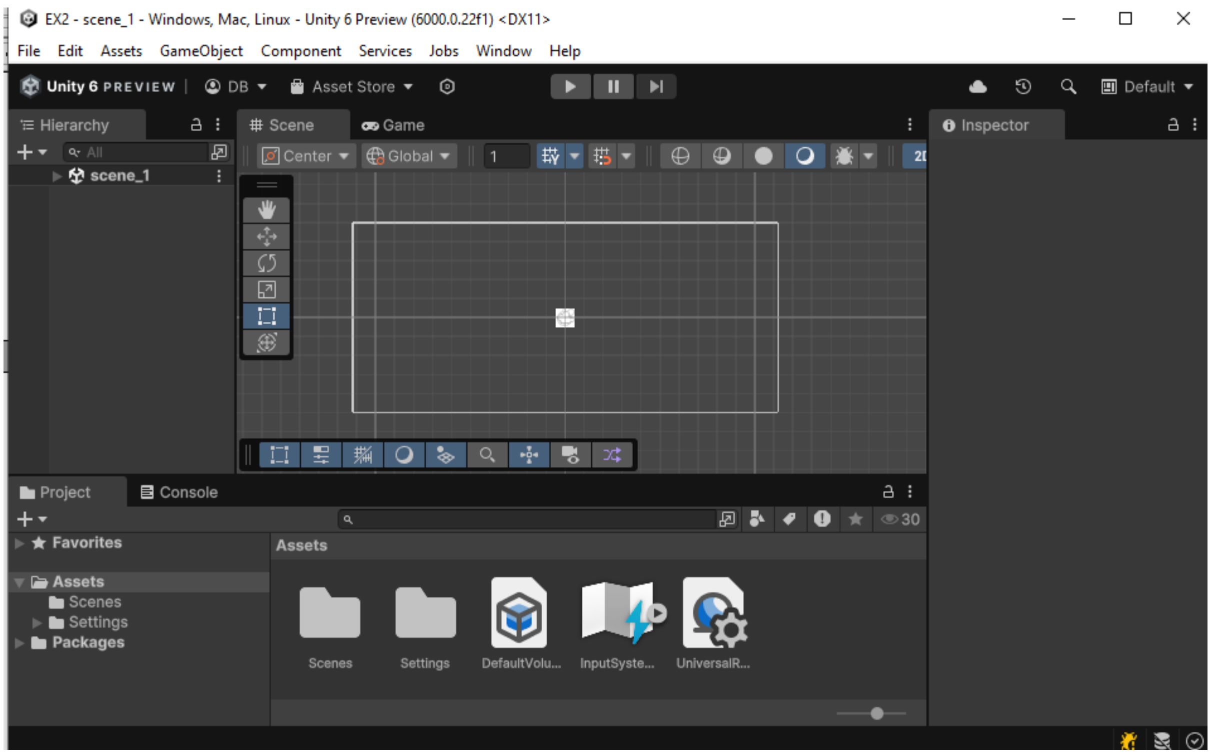
Unity is launched

New project platform is shown

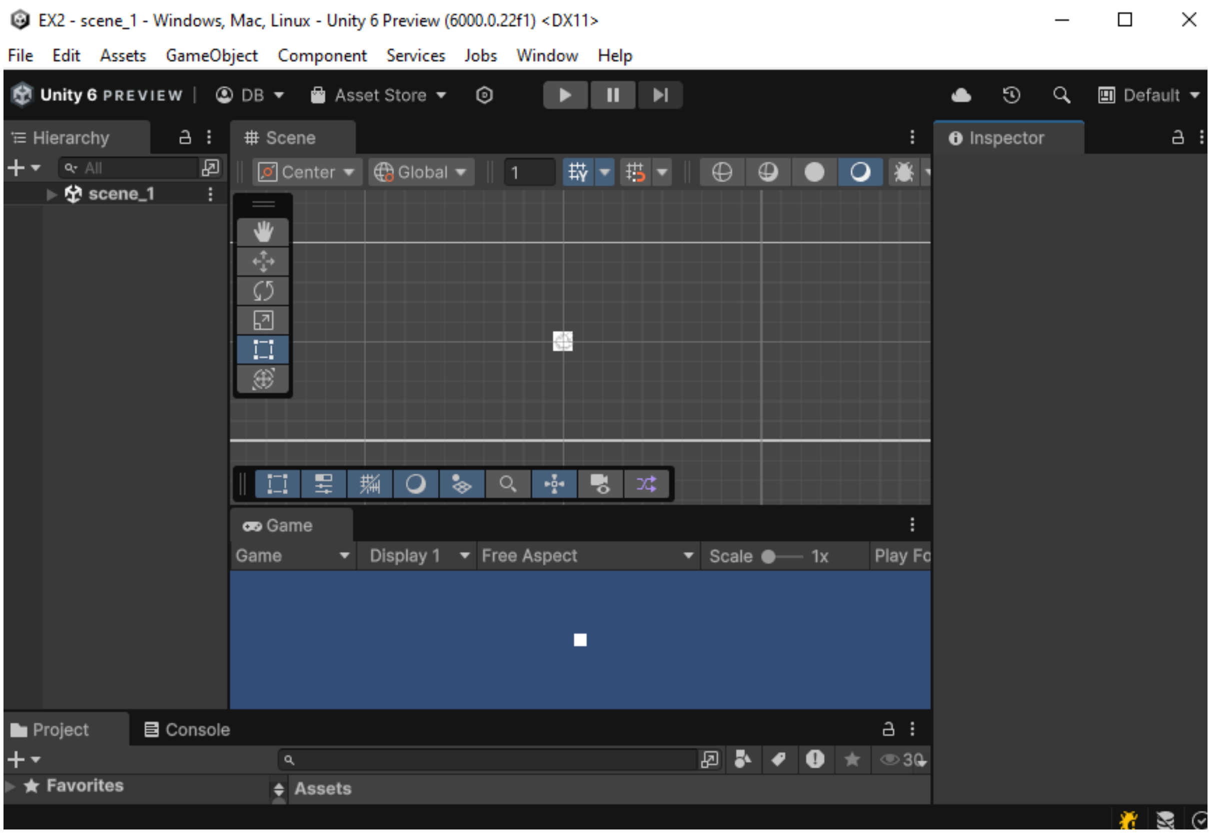


Now the video shows how to set the layout: there are several choices...

We mainly use the default layout (it can be taken into use also via Window-menu: Layouts, default):

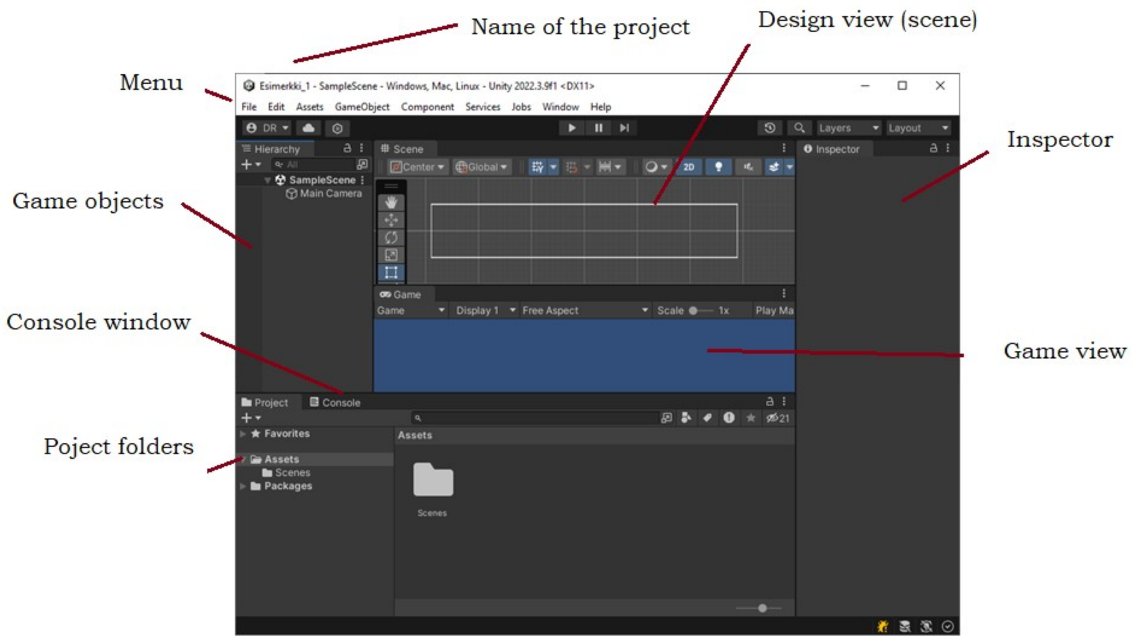


Then I normally move Game-part below Scene-part:



GUI

Main parts are shown here:



Add Game Objects to the Scene

Example 1: let add some different objects to the scene and run the app

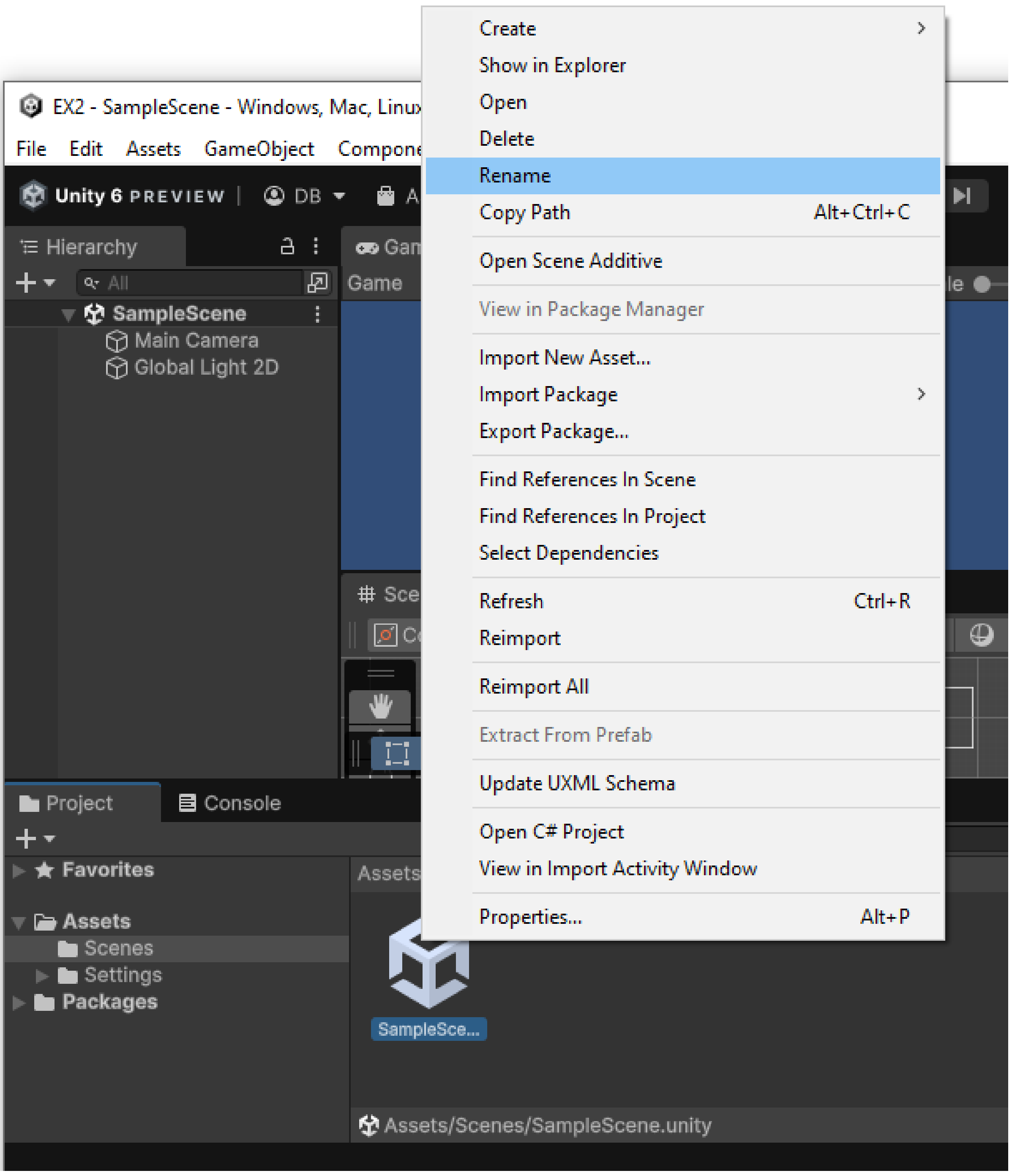
Let's rename the scene, it can be "scene_1".

We can add several scenes to one project and thus the name is important.

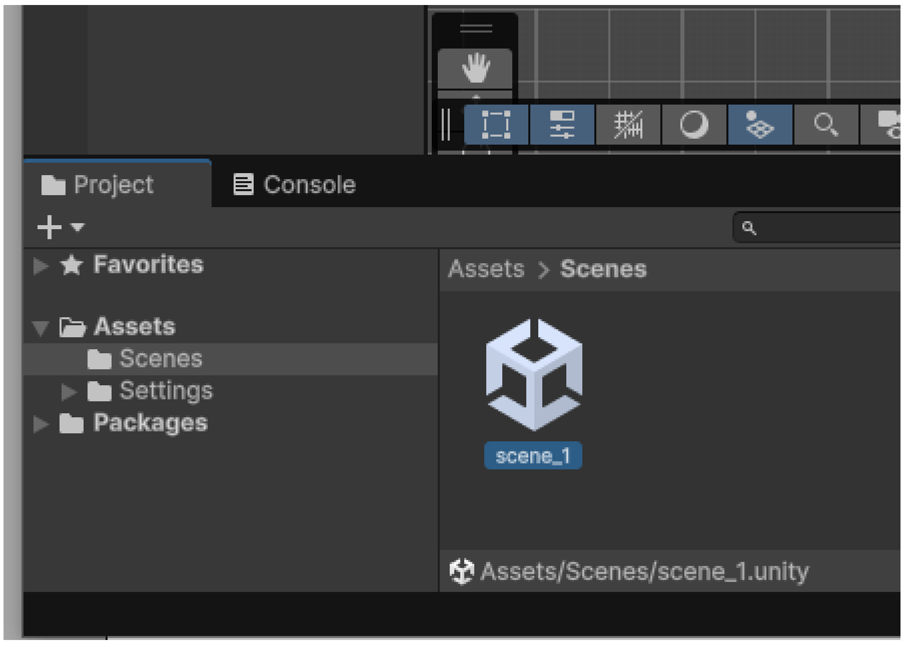
We studying Unity it is easier and faster to use several scenes instead of several new projects.

If you the want to publish your game you can choose only the scenes that belong to the project.

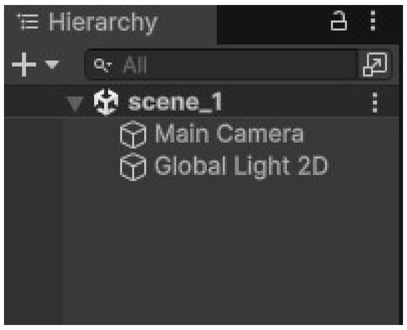
We can rename the scene by choosing it and by clicking the right mouse button: from the menu we choose ”rename”.


EX2 - SampleScene - Windows, Mac, Linux
File Edit Assets GameObject Compone
Unity 6 PREVIEW
DB
Hierarchy
SampleScene
Main Camera
Global Light 2D
Game
Project
Console
Favorites
Assets
Scenes
Settings
Packages
SampleSce...
Assets/Scenes/SampleScene.unity
Create
Show in Explorer
Open
Delete
Rename
Copy Path Alt+Ctrl+C
Open Scene Additive
View in Package Manager
Import New Asset...
Import Package
Export Package...
Find References In Scene
Find References In Project
Select Dependencies
Refresh Ctrl+R
Reimport
Reimport All
Extract From Prefab
Update UXML Schema
Open C# Project
View in Import Activity Window
Properties... Alt+P

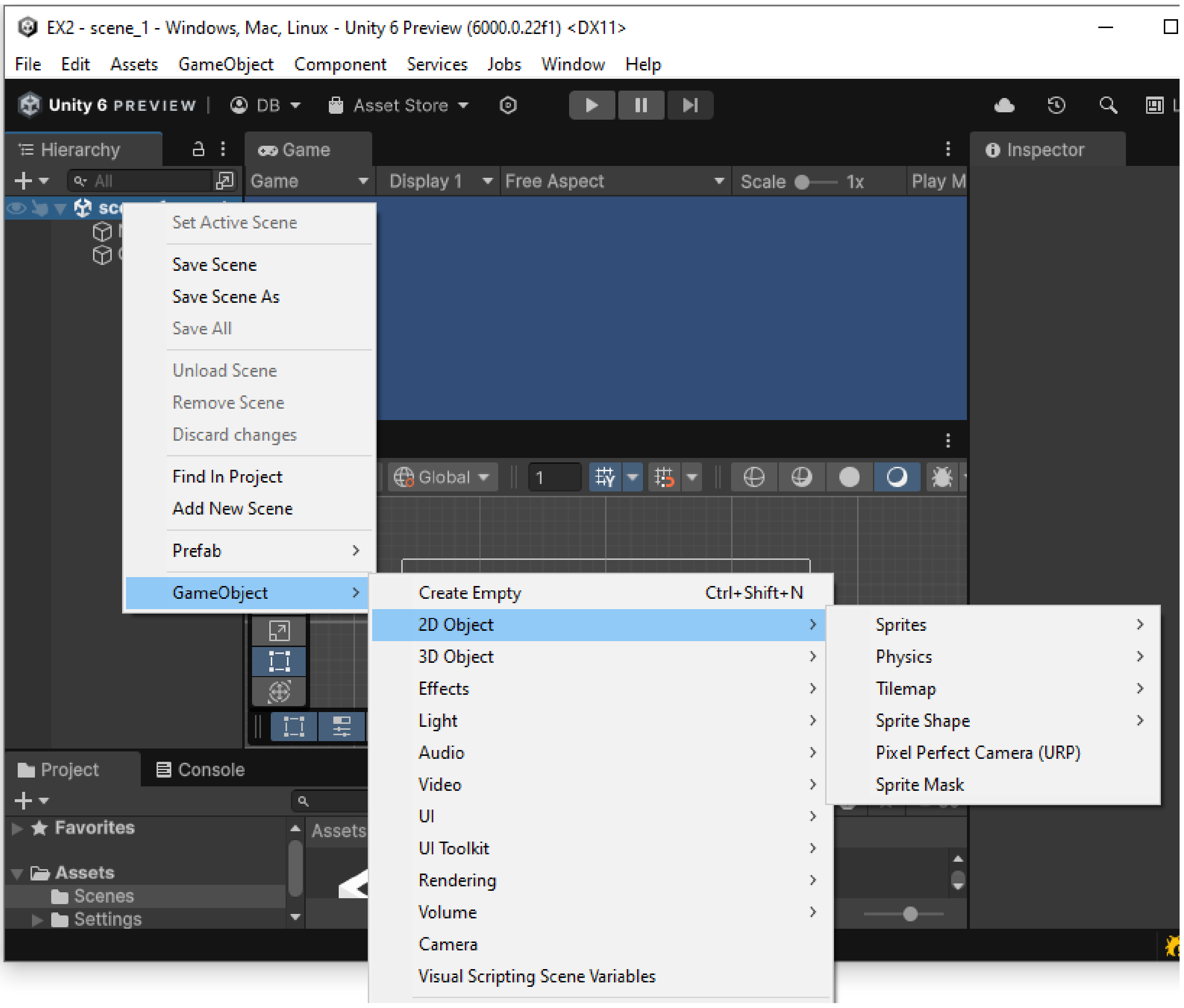
And



Now the new name is shown also on the Hierarchy part:



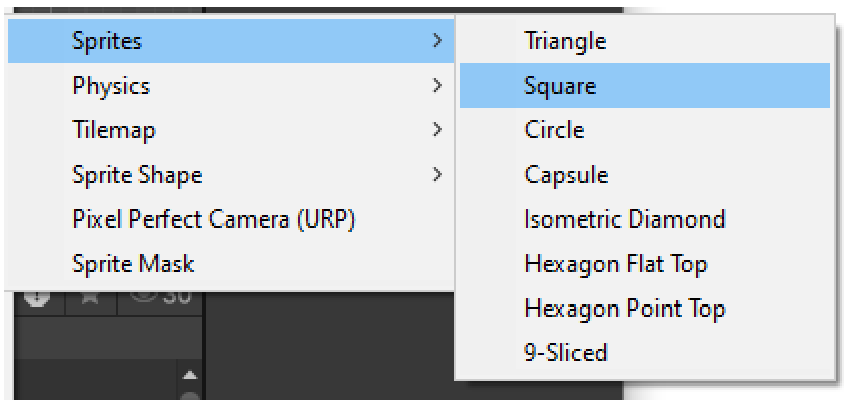
Add new GameObject



Game consists of game objects that then have one or more components. Components have fields/properties.

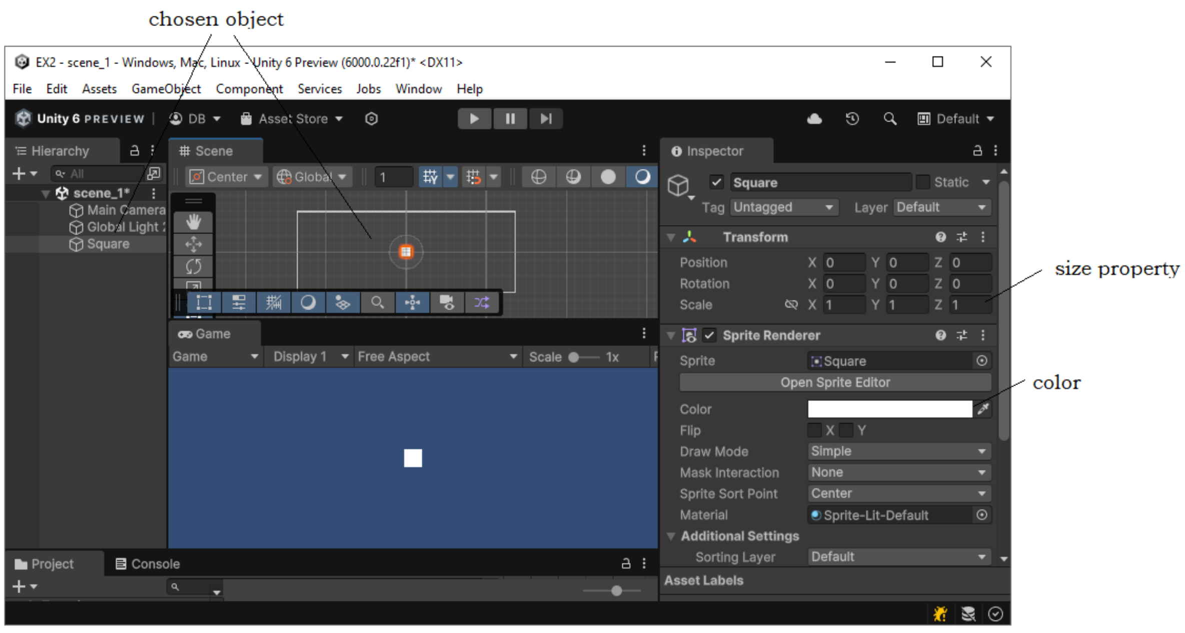
You can add components, remove them and edit them. You can see components on the Inspector window.

Adding a square object



Let's change the color and size:

Component named Transform makes it possible to change location, size and so on.



Square Static
Tag Untagged Layer Default

Transform

Position	X	0	Y	0	Z	0
Rotation	X	0	Y	0	Z	0
Scale	X	1	Y	1	Z	1

Sprite Renderer

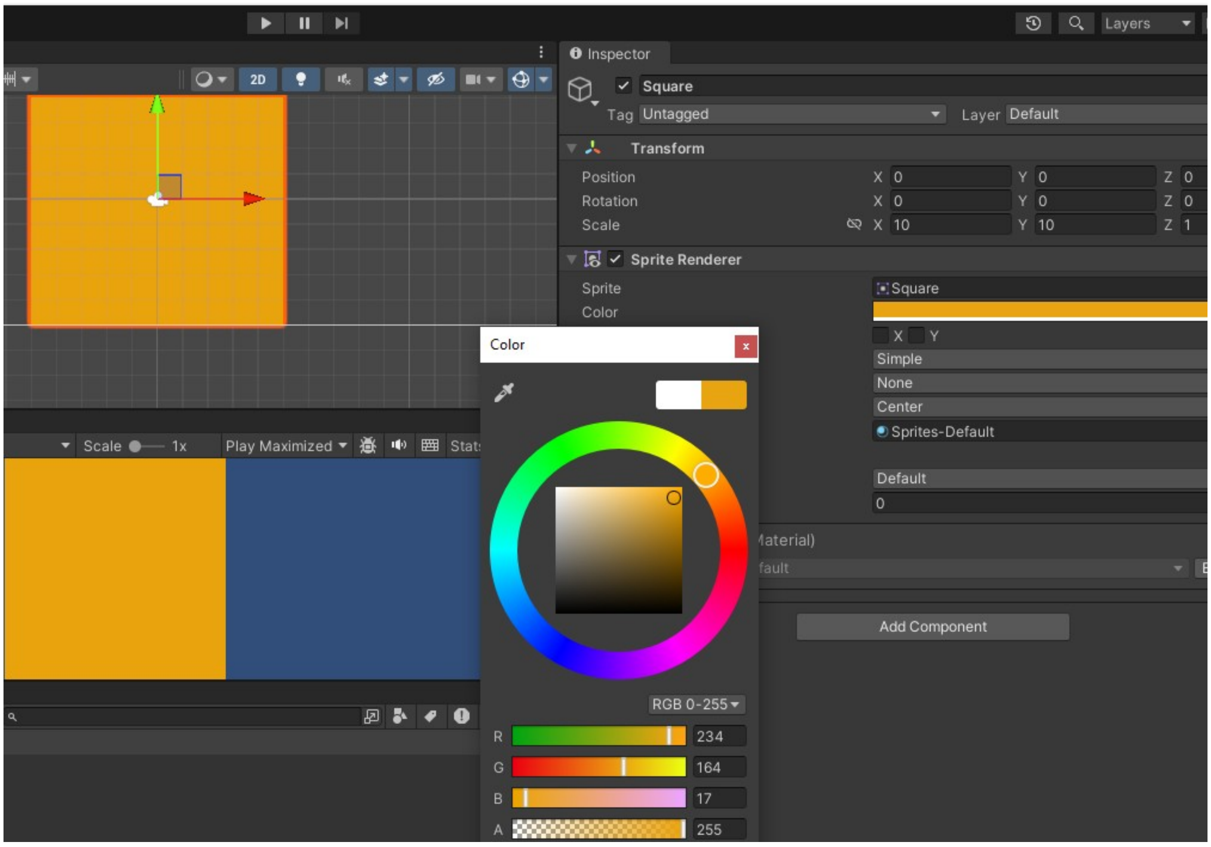
Sprite	Square
	Open Sprite Editor
Color	
Flip	X Y
Draw Mode	Simple
Mask Interaction	None
Sprite Sort Point	Center
Material	Sprite-Lit-Default

Additional Settings

Sorting Layer	Default

Asset Labels

And Sprite Renderer has a field (property) that is used to change the colour:



Square
Tag Untagged Layer Default

Transform

Position	X	0	Y	0	Z	0
Rotation	X	0	Y	0	Z	0
Scale	X	10	Y	10	Z	1

Sprite Renderer
Sprite Square
Color
X Y
Simple
None
Center
Sprites-Default
Default
0

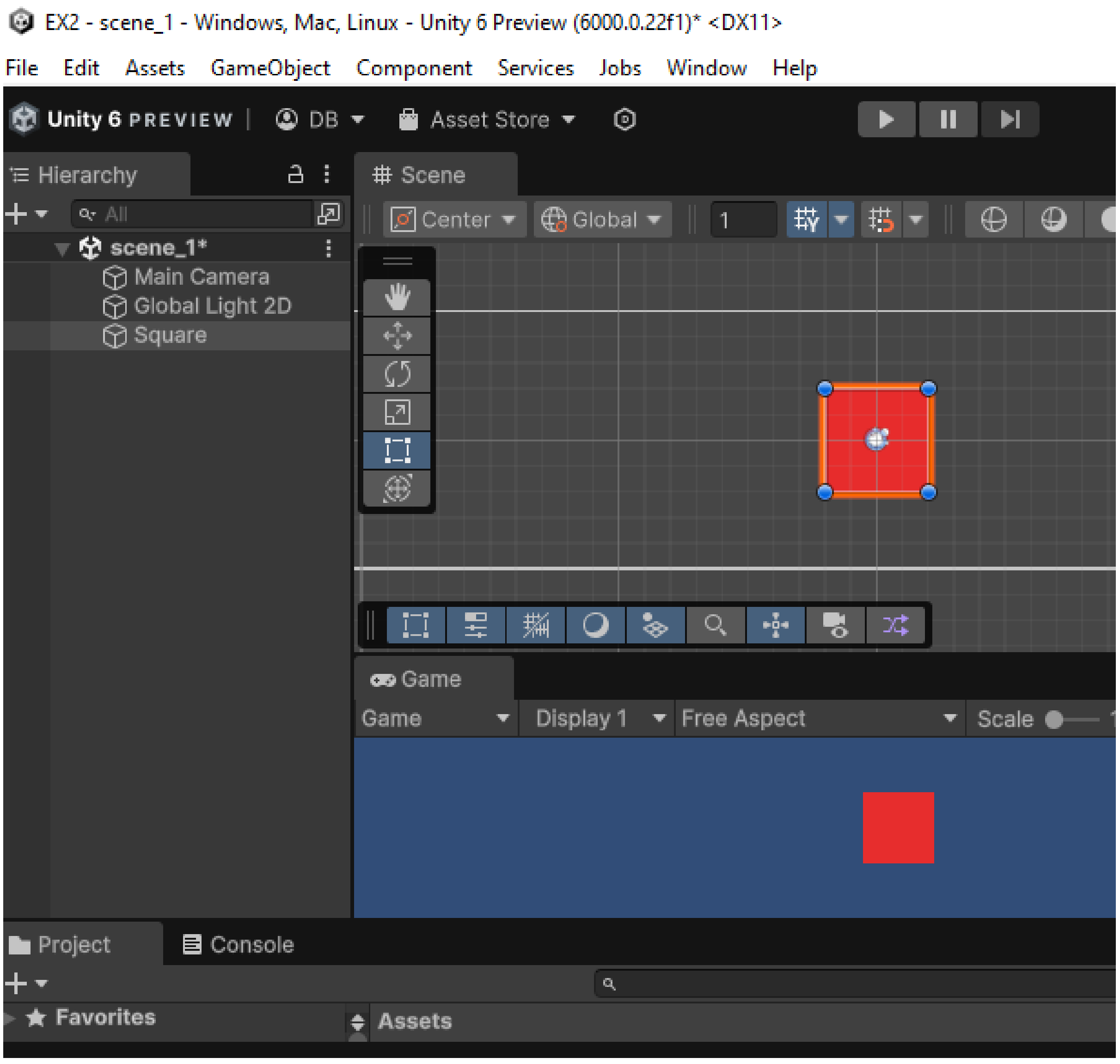
RGB 0-255

R	234
G	164
B	17
A	255

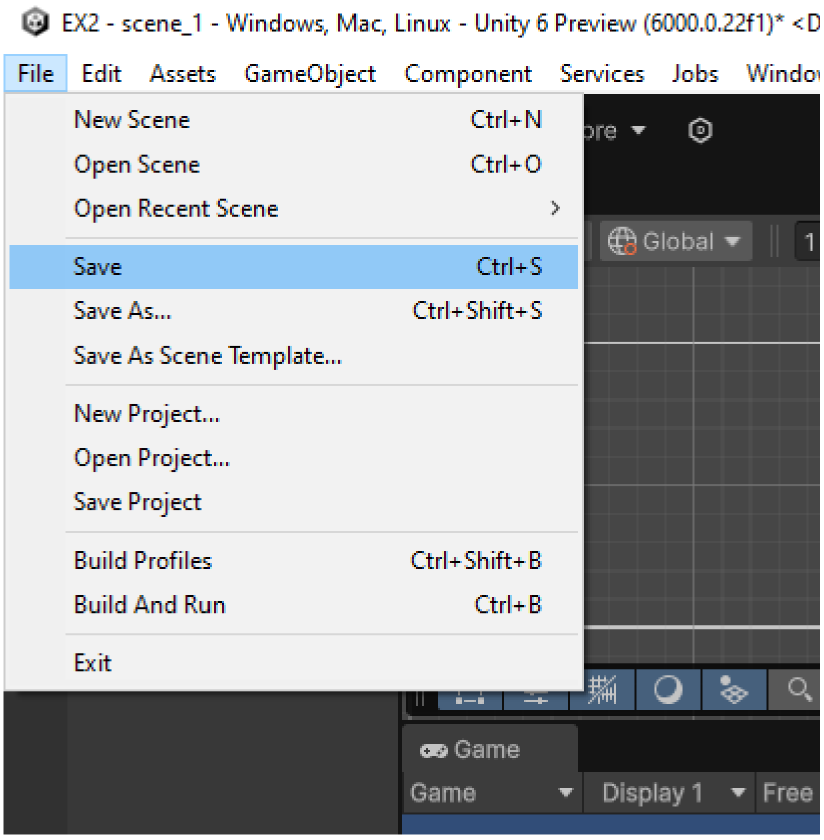
Try to make changes!!

Note!

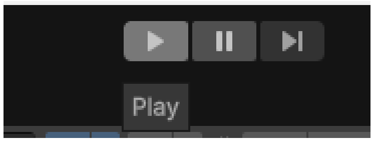
You can resize the design window by using mouse wheel.

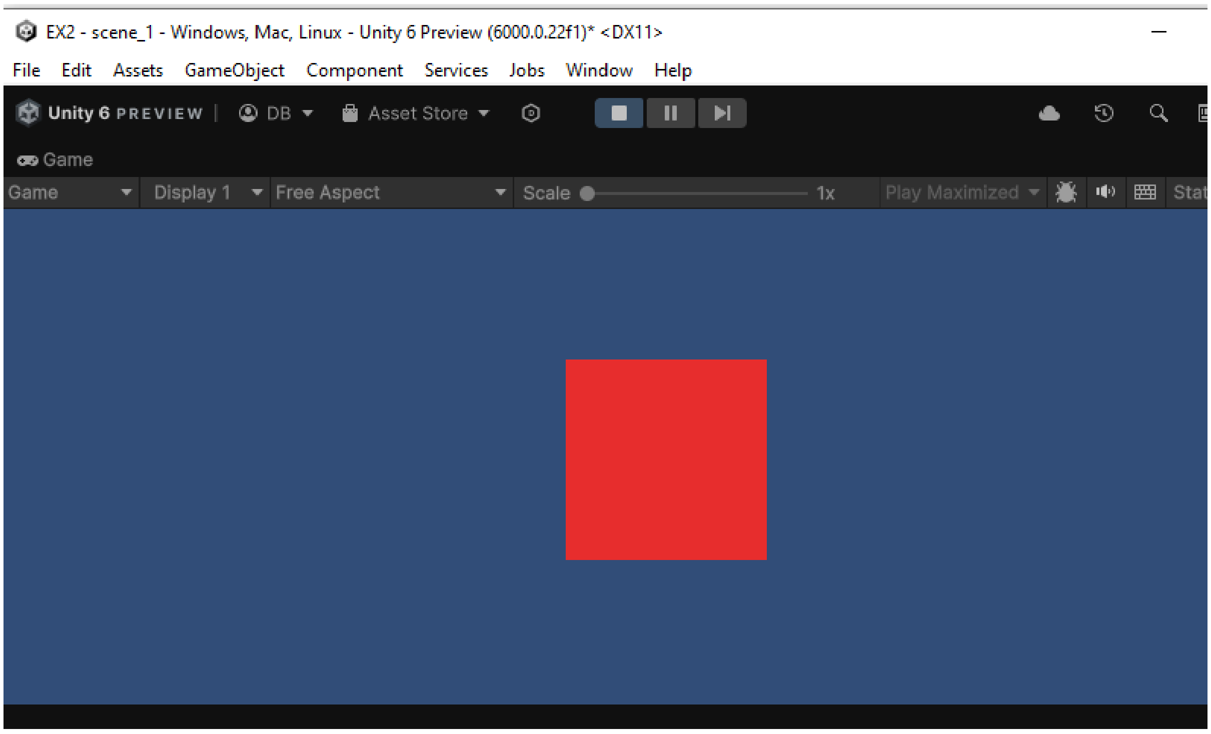

EX2 - scene_1 - Windows, Mac, Linux - Unity 6 Preview (6000.0.22f1)* <DX11>
File Edit Assets GameObject Component Services Jobs Window Help
Unity 6 PREVIEW
DB
Asset Store
Hierarchy
All
scene_1*
Main Camera
Global Light 2D
Square
Scene
Center
Global
1
Game
Game
Display 1
Free Aspect
Scale
Project
Console
Favorites
Assets

When you have made changes, an asterisk (*) is shown after the scene name to remind you to save changes! Save the project by choosing Ctrl + S or File menu.

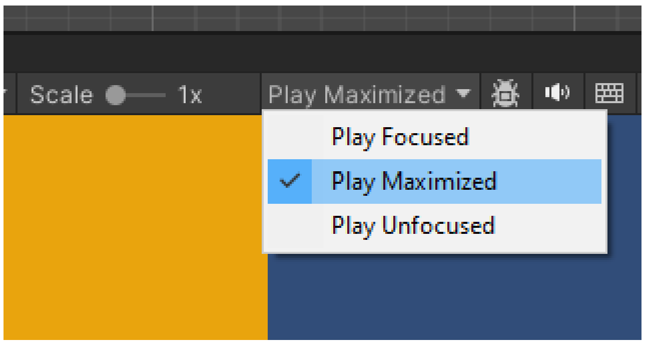


Run the project




EX2 - scene_1 - Windows, Mac, Linux - Unity 6 Preview (6000.0.22f1)* <DX11>
File Edit Assets GameObject Component Services Jobs Window Help
Unity 6 PREVIEW
DB
Asset Store
Game
Game
Display 1
Free Aspect
Scale
1x
Play Maximized
Stat

Different ways to run the project



Play focused -mode shows the game in run mode all the time.

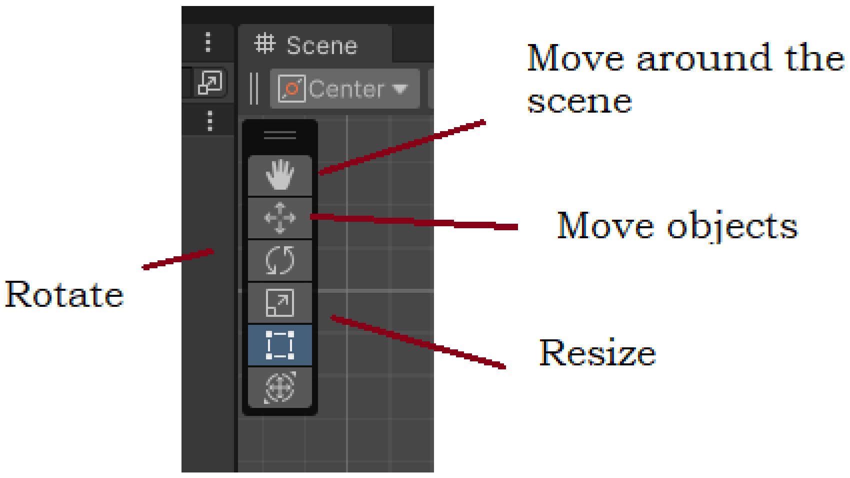
Now you have created your first Unity app!!

The procedure is: create, save, test run...

Let's start learning new things now!!

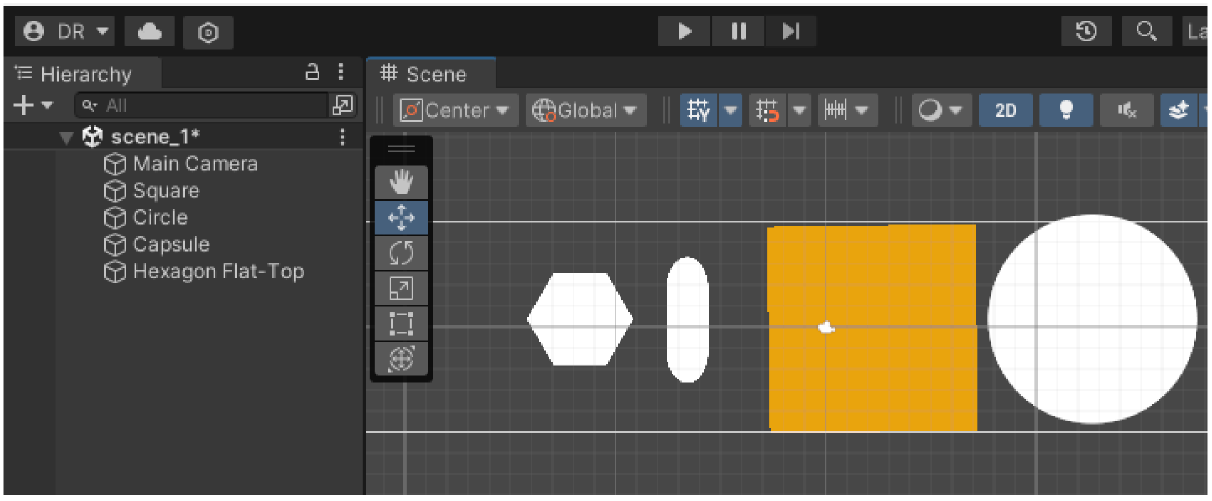
Next steps are
manipulate game objects
add new game objects
add functionality: we need to start coding!!

Edit gameobjects

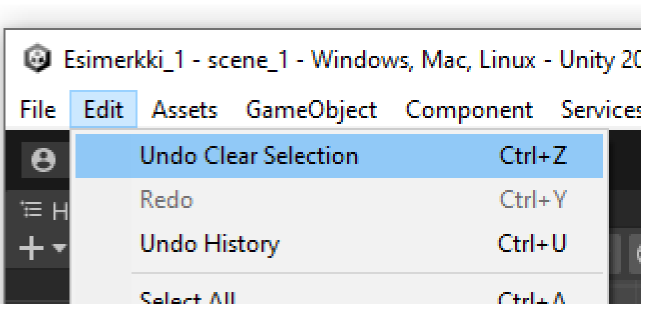


Video: move place, size

Try now to add different kinds of Sprites to you scene and edit their properties:



If you do mistakes, you can easily cancel by choosing Edit – Undo or by using key combination Ctrl - Z.



Add functionality

We need to add codes to our game to achieve functionality.

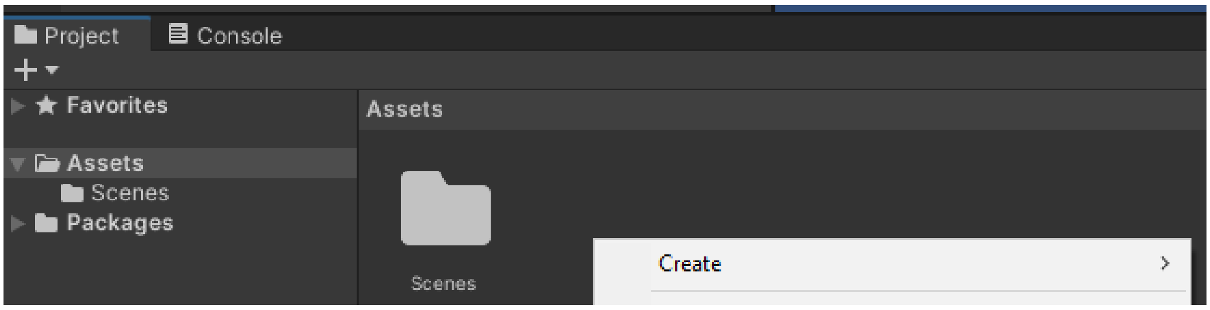
We create now a short experiment where we can move some object on the scene at runtime.

For codes it is good to create an own subolder to Assets-folder. The structure of different items and folders remain clear.

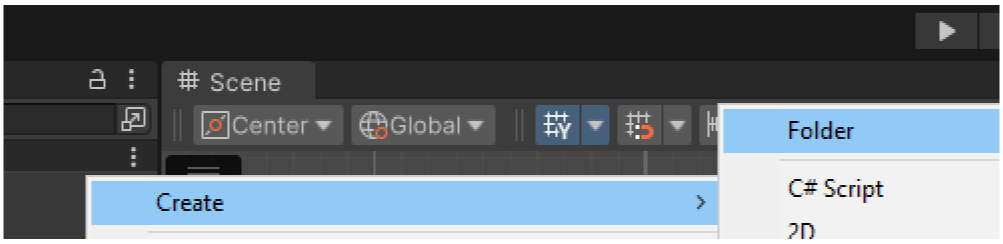
Note!
Have you installed Visual Studio to you machine to be used together with Unity?
If not, go to Microsoft's Visual Studio -page and install a community version!

It is good to create folders to store different kinds assets.

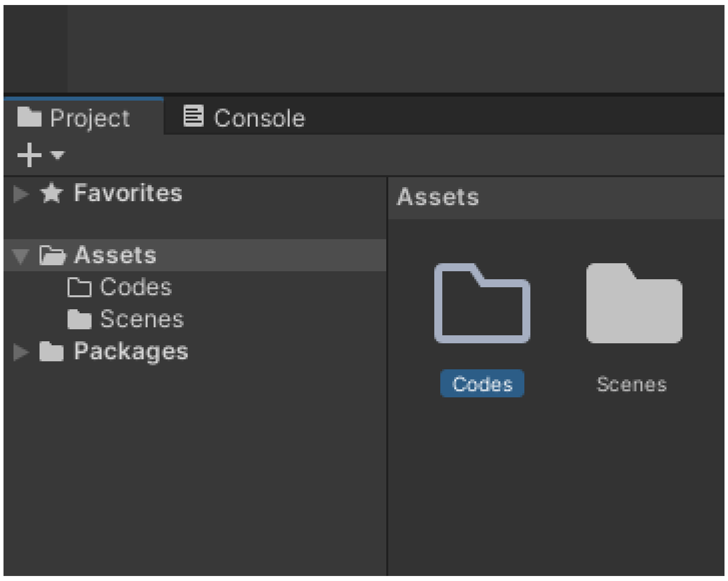
Click Project Assets and rightclick the area on the right.



From the Create-menu choose folder.

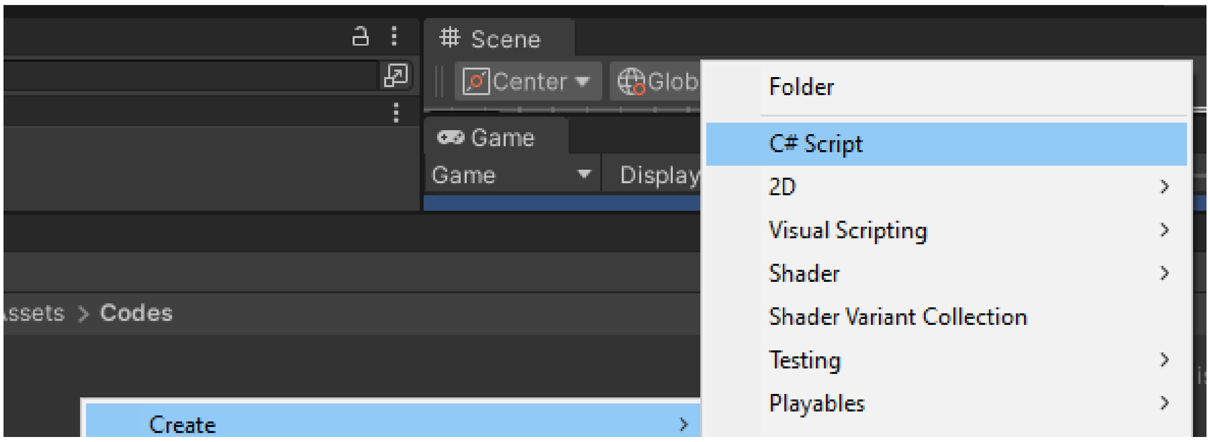


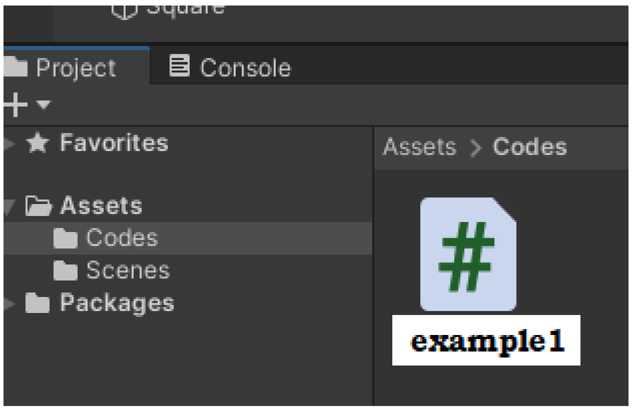
Create a new folder named Codes.



Now you can create a new script to Codes folder.

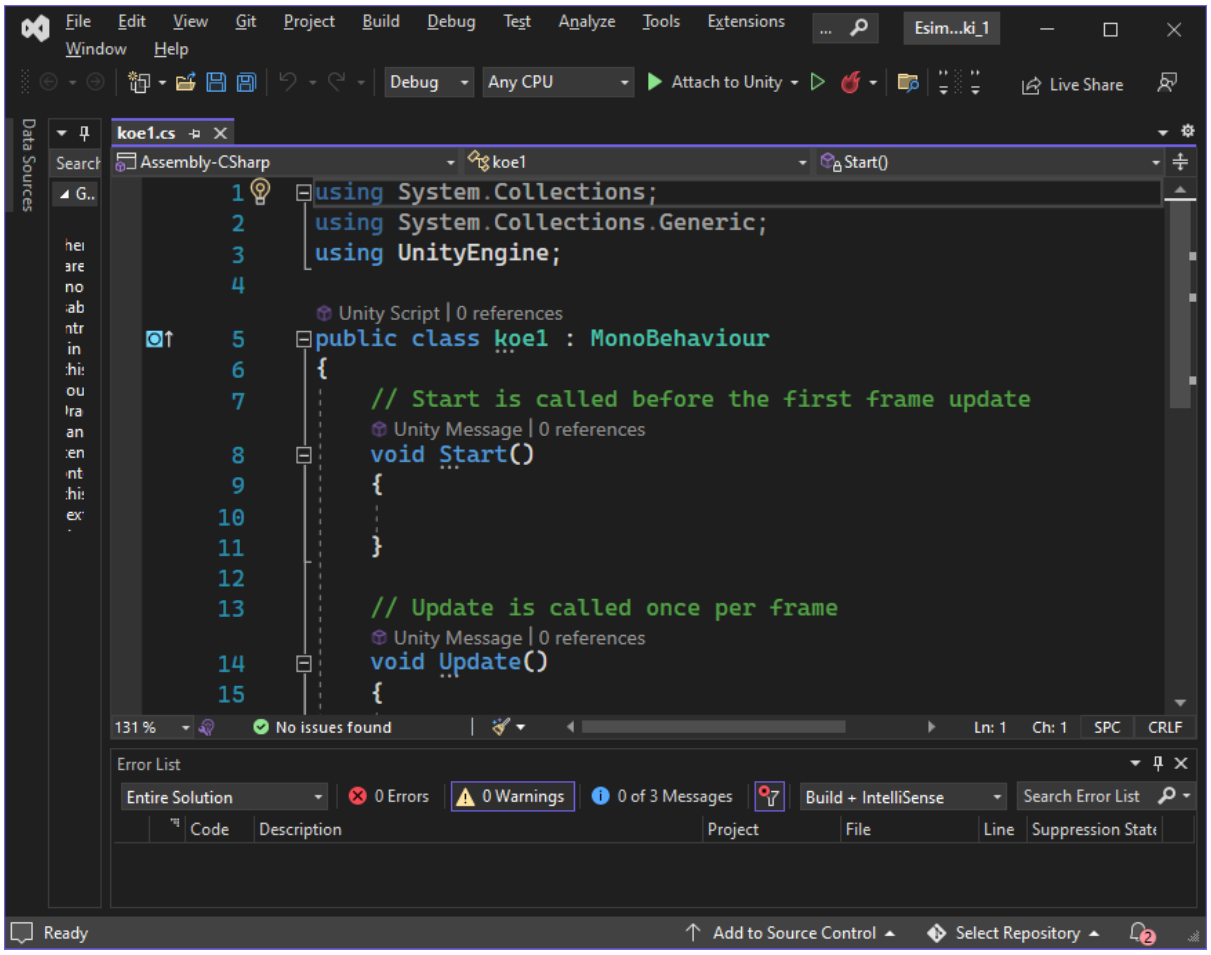
The name can now be example1.


Scene
Center
Glob
Game
Game
Display
Assets > Codes
Create
Folder
C# Script
2D
Visual Scripting
Shader
Shader Variant Collection
Testing
Playables


Square
Project
Console
Favorites
Assets
Codes
Scenes
Packages
Assets > Codes
example1

Visual Studio: basic usage

We can now edit our script. Doubleclick the filename: Visual Studio opens.



```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class koe1 : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {
```

Every script is a class. In the code we have automatically as default methods Start() ja Update().

Start() is executed when we start the application.

Update() is executed when a new frame is shown (refresh rate defines how often it is executed).

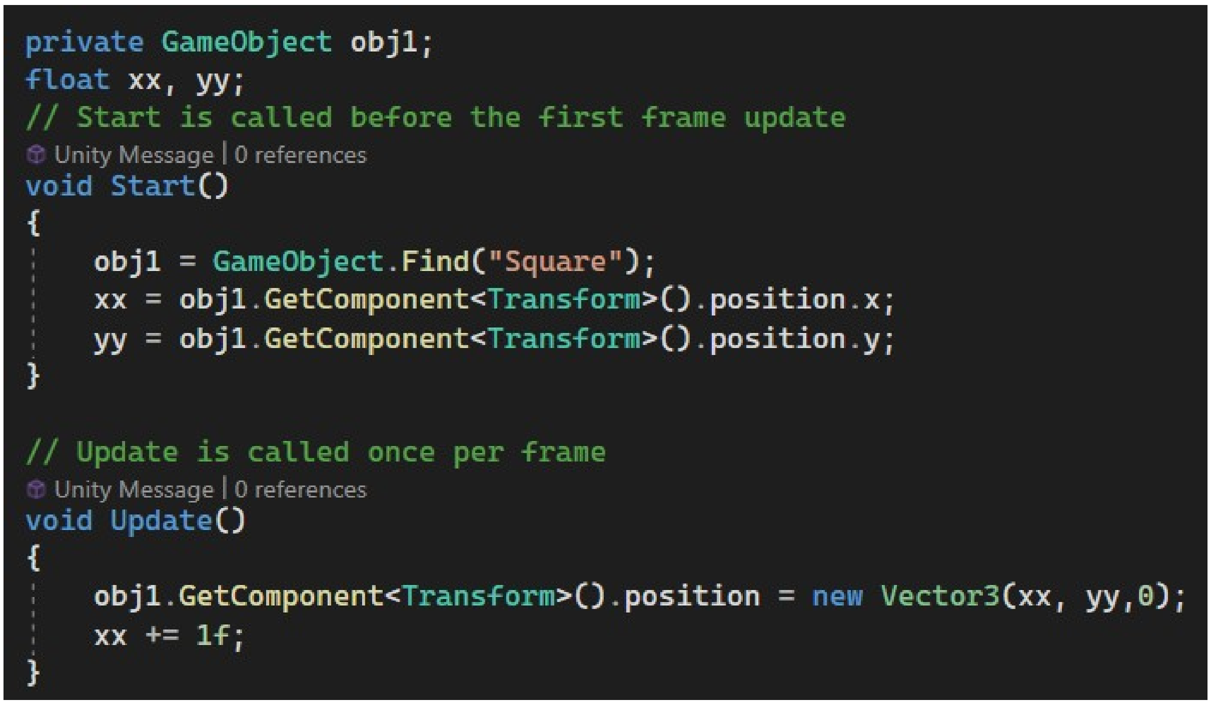
You do not need to add code inside those methods, but quite often they are needed...

Example 1: moving circles

Goal
We write a script that can find our objects, now our square object (name is simply ”Square”).
Then we want that the object moves **back and forth** on the screen.

Plan
Create a connection (reference) to the object
Find the Transform-component of the object
Manipulate the properties (now x coordinate) in Update() method (dynamically)

Version 1
Object moves only to the right having some fixed speed



```
private GameObject obj1;
float xx, yy;
// Start is called before the first frame update
Unity Message | 0 references
void Start()
{
    obj1 = GameObject.Find("Square");
    xx = obj1.GetComponent<Transform>().position.x;
    yy = obj1.GetComponent<Transform>().position.y;
}

// Update is called once per frame
Unity Message | 0 references
void Update()
{
    obj1.GetComponent<Transform>().position = new Vector3(xx, yy,0);
    xx += 1f;
}
```

The reference called ”obj1” refers to GameObject that is the base of all objects. All items we add to the project are basically of type GameObject. They normally have a spesific name (e.g. Button, Sprite, etc.).

Access specifier (private, protected, public) tells how other classes can access the item...

```
private GameObject obj1;
```

In start() method we set the obj1 to refer to our Square.

```
obj1 = GameObject.Find("Square");
```

Then we write down the current values of x and y.

```
xx = obj1.GetComponent<Transform>().position.x;
yy = obj1.GetComponent<Transform>().position.y;
```

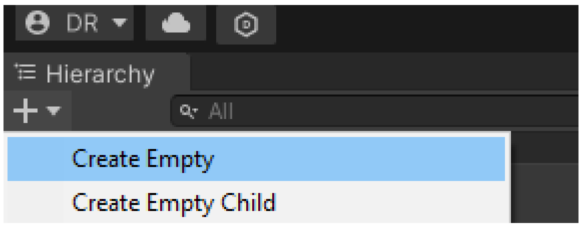
Then, in Update()-method we add value 1 to x: so the position of the object changes when a new frame is shown:

```
obj1.GetComponent<Transform>().position = new Vector3(xx, yy,0);
```

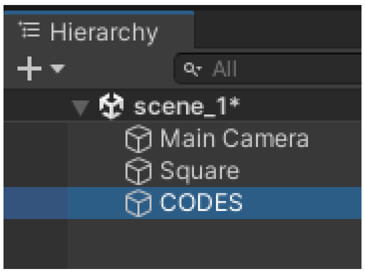
```
Save the code and return to Unity.
```

It is a good principle to put different kinds of items to their own subfolders and also to own gameobjects inside the hierarchy.

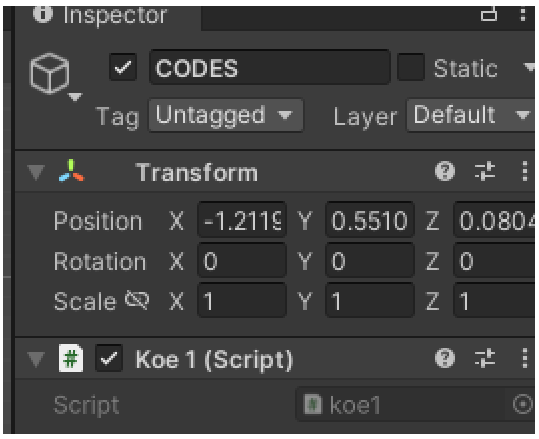
So, create an empty gameobject, name it codes and add the script there. Now the script is seen in every part of the project when we run the project.



AND



It is easy to just drag and drop the codefile to the CODES-object. Then it is seen in Inspector part as a component.


Inspector

CODES Static

Tag Untagged Layer Default

Transform

Position X -1.2119 Y 0.5510 Z 0.0804

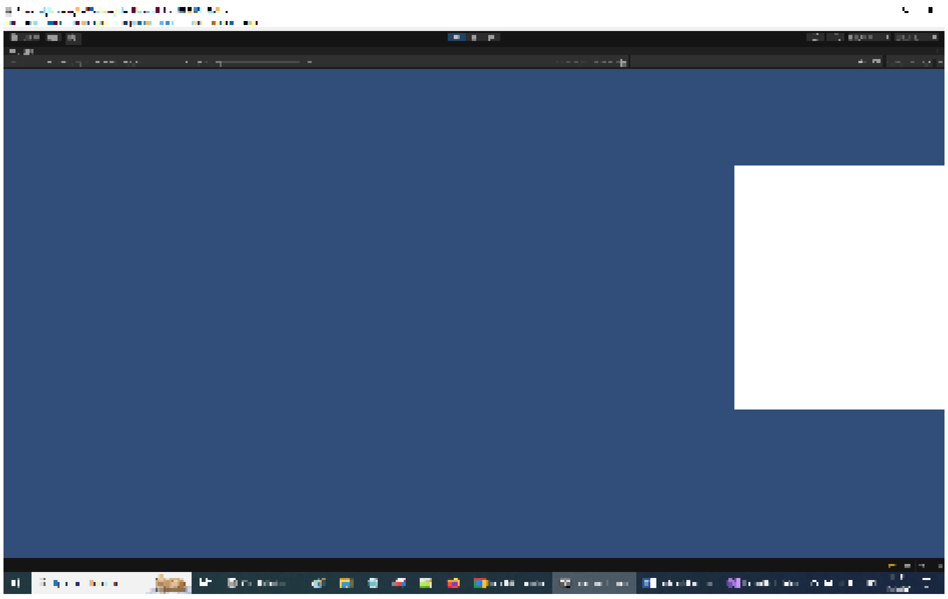
Rotation X 0 Y 0 Z 0

Scale X 1 Y 1 Z 1

Koe 1 (Script)

Script koe1

Test run



We note that the velocity is too hight.

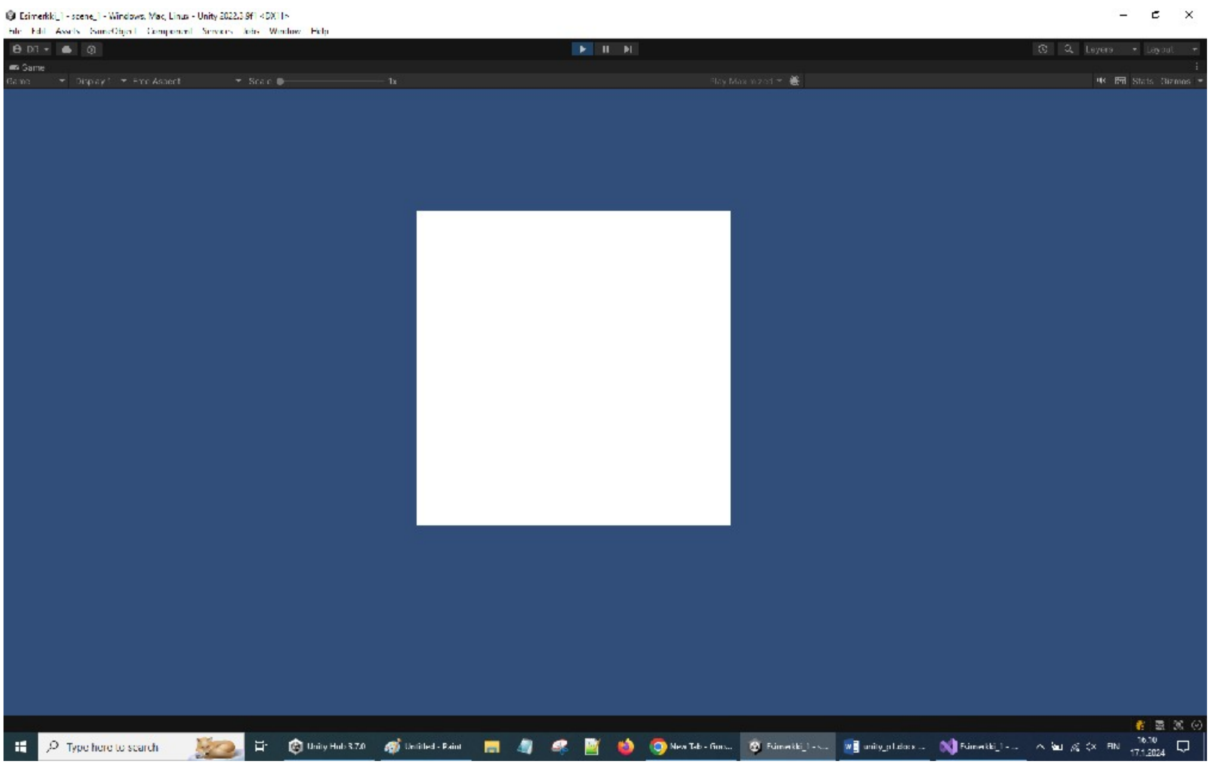
Let's create a new version!

Version 2

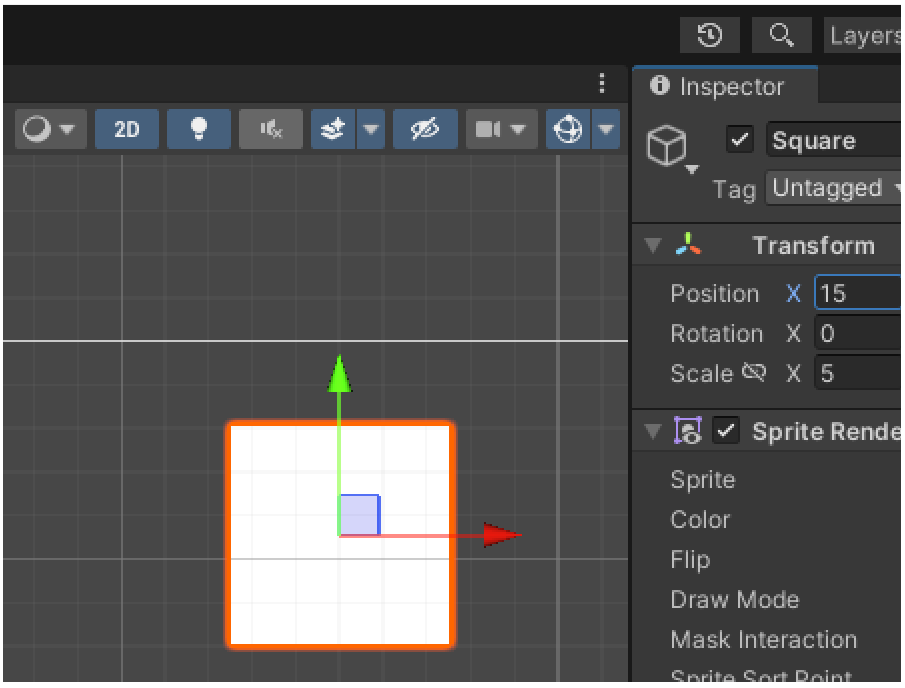
Let's try to use a small step:

```
xx += 0.001f;
```

Now it moves too slowly...



So, now limits -12 – 012 are ok.



The square could move between limits -12 - +12.

For changing the direction we can use if statements and a special variable,,,

For example:

float stepX = 0.01f;

It is good to adjust the velocity: thus we use now a new variable stepX for that purpose.

Now the sign of the variable stepX changes when limits are reached:

```
        if (obj1.GetComponent<Transform>().position.x >= 5f)
        {
            stepX = -0.05f;
        }
        if (obj1.GetComponent<Transform>().position.x <= -5f)
        {
            stepX = 0.05f;
        }
```

Whole update() methods

```
  void Update()
    {
        obj1.GetComponent<Transform>().position = new Vector3(xx, yy,0);
        xx += stepX;
        if (obj1.GetComponent<Transform>().position.x >= 5f)
        {
            stepX = -0.05f;
        }
        if (obj1.GetComponent<Transform>().position.x <= -5f)
        {
            stepX = 0.05f;
        }
    }
```

Edit, save and try!

Some important parts in our example:

Start()
Update()
From the code we can make a connection to game objects and then there components and of course to component fields (properties) and manipulate them.

Video

Example 2: changing colors

Goal
We change the colour of the object randomly after some spesific interval.

We can use a square here, too.

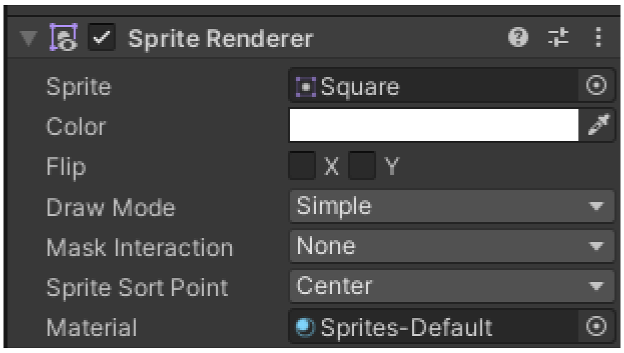
We create a new script (at the same time we repeat some earlier steps).

Let the name of the script be example2.cs.

Here we start:
```
private GameObject obj1;

    // Start is called before the first frame update
    void Start()
    {
        obj1 = GameObject.Find("Square");
    }
```

Now we have to access the colour of the object: it is inside the component called SpriteRenderer..

Component



Sprite Renderer

Sprite	Square
Color	
Flip	X Y
Draw Mode	Simple
Mask Interaction	None
Sprite Sort Point	Center
Material	Sprites-Default

Now we chance the color: there we can use Color class and create the color obects.

Version 1

Color is to be changed to red.

Here is the beginning of the code.

```
private GameObject obj1;

    // Start is called before the first frame update
    void Start()
    {
        obj1 = GameObject.Find("Square");
        obj1.GetComponent<SpriteRenderer>().color = Color.red;
    }
```

Here we refer to color field (property) of the component

```
obj1.GetComponent<SpriteRenderer>().color
```

Some basic constant colors can be created easily, e.g. Red now;

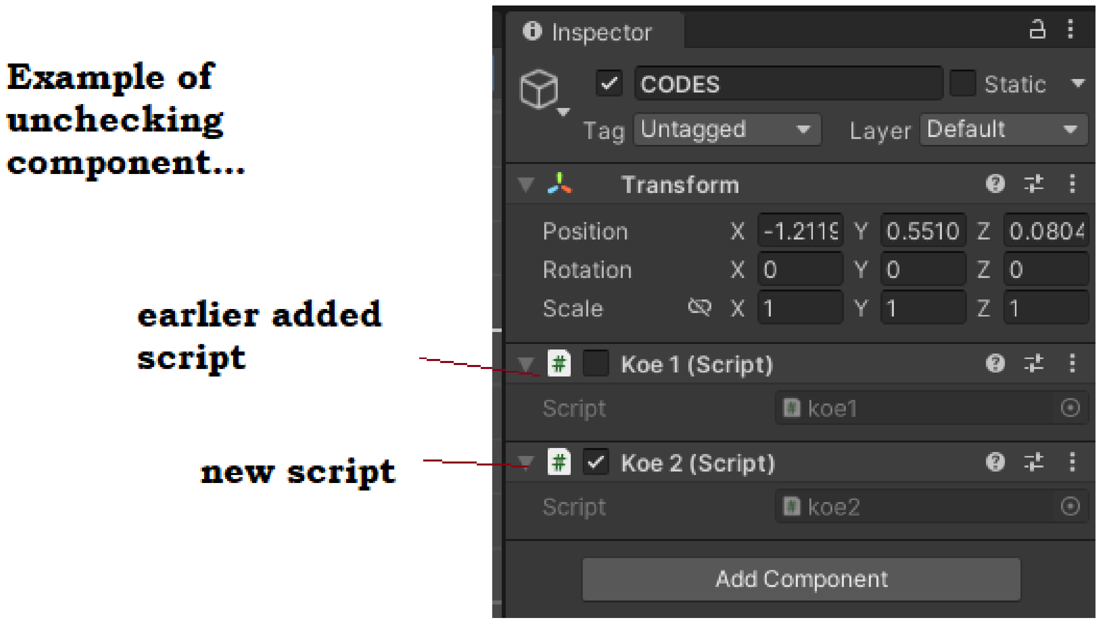
Color.red

And this code changes the color of our square object:

```
obj1.GetComponent<SpriteRenderer>().color = Color.red;
```

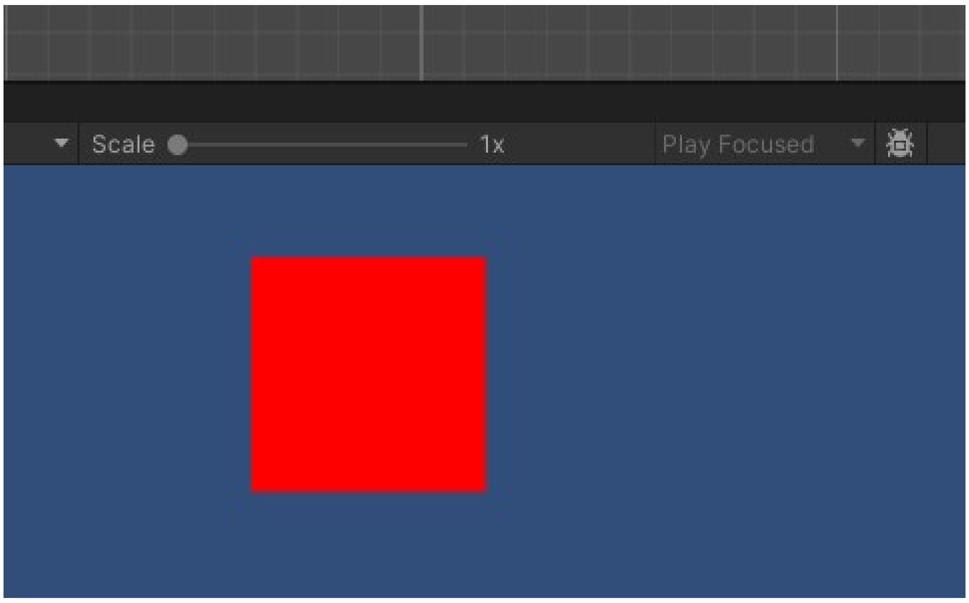
Sijoita koodi CODES-objektiin ja koeaja. Voit ottaa koodin koe1.cs pois päältä tilapäisesti:

We put the codefile to CODES object and run it. If we already have files there we can uncheck codes that we do not need to run now.



[✓] CODES Static
Tag Untagged Layer Default
Transform
Position X -1.2119 Y 0.5510 Z 0.0804
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
[ ] Koe 1 (Script)
Script koe1
[✓] Koe 2 (Script)
Script koe2

Test run



You can run the project either by using mode "Play Focused", or "Play Maximized"...

Version 2

Now we change the color 10 times. May be we just use only red color and change its tone step by step...

We can use the Color class and create colors using components R, G and B. There values are between 0 and 255.

For example RGB-values (255, 0, 0) mean red color.

We can also use floating point values that are between 0 and 1.

Version 1 of the code

```
private Color newColor;
    // Start is called before the first frame update
    void Start()
    {
        obj1 = GameObject.Find("Square");
        newColor = new Color(1.0, 0, 0);
        obj1.GetComponent<SpriteRenderer>().color = newColor;
    }
```

Red color is added to our square as before.

Try it!

Final Update()-code:

```
    float redColor = 0.1f;
    int n = 0;
    private Color newColor;

    void Update()
    {
          n++;
          if (n >= 100)
          {
              newColor = new Color(redColor, 0, 0);
              obj1.GetComponent<SpriteRenderer>().color = newColor;
              redColor += 0.05f;
              n = 0;
          }
   }
```

When n is 100 we change the color: we can use also other ways to make changing process slower. Later we speak a bit about Time.deltatime that means the interval in seconds from the last frame to the current one.

Animations

Animations

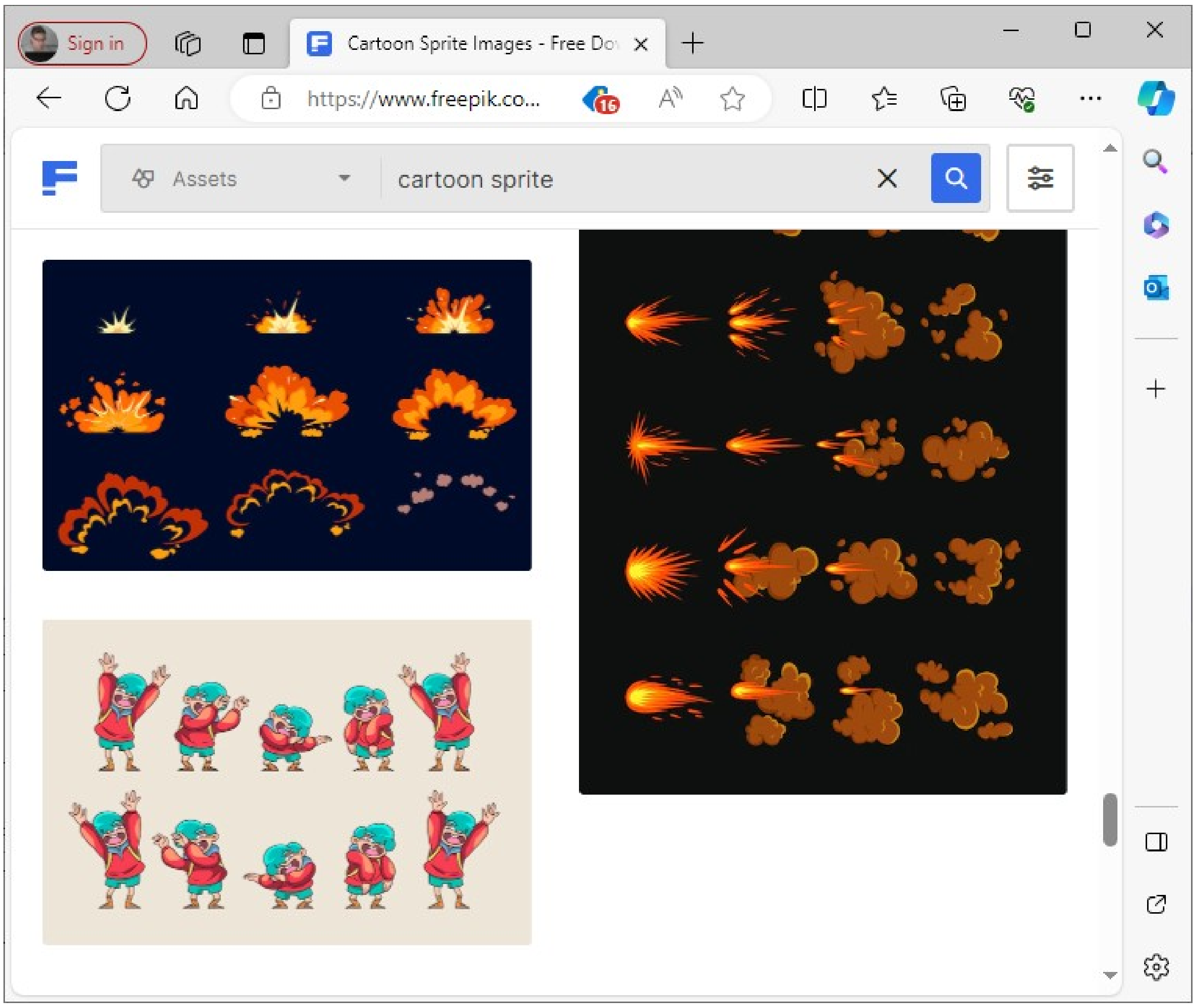
There are several ways to create animations. Now we use Unity's own tool.

Animation consists of different pictures (series of pictures). Background has to be transparent. There are several tools (also online) that camnbe used to remove background color.

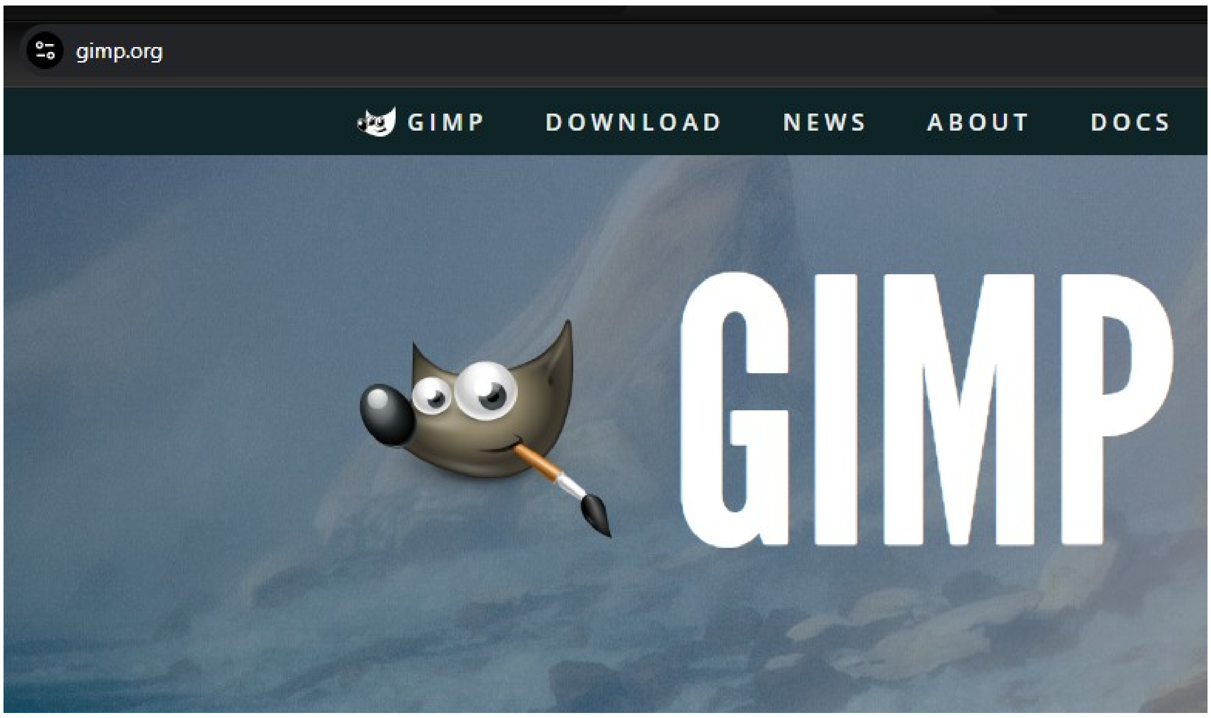
Let's try!

We can google a bit and try to find free picture series: e.g. Searching "sprite cartoon" item.s

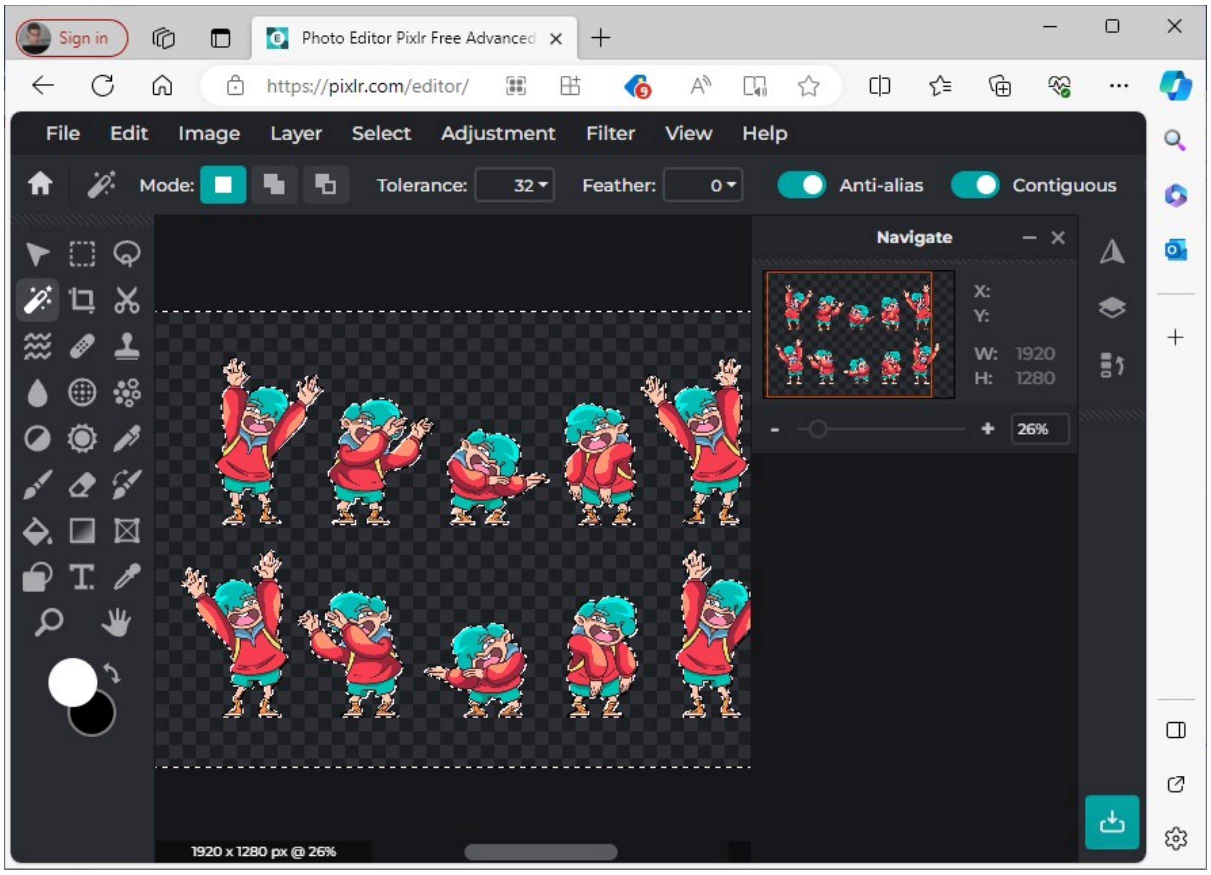
For example this place was found:



Good tool to edit pics is e.g. GIMP



To make background transparent we can also use online tools e.g. https://pixlr.com/editor/



Mode:

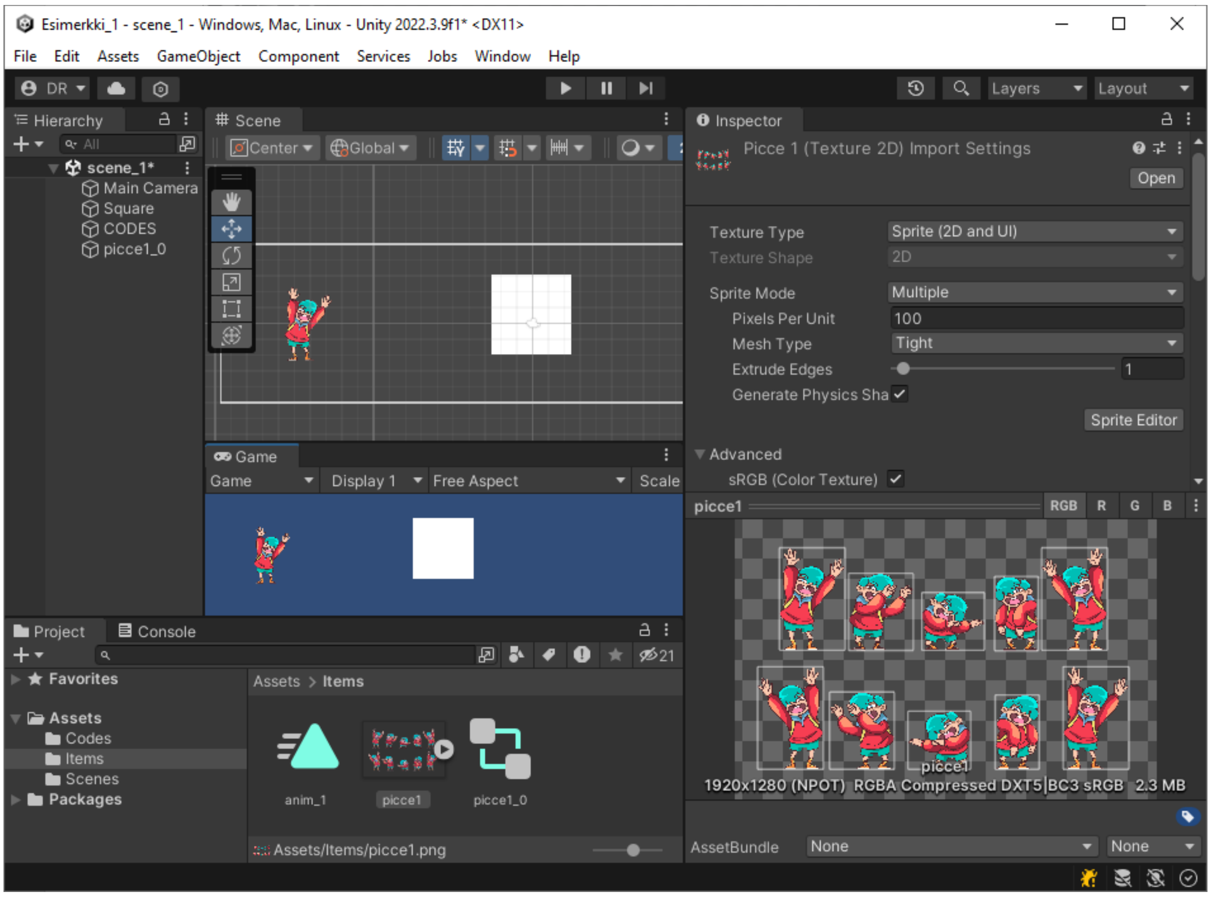
Tolerance: 32

Feather: 0

Anti-alias

Contiguous

Then you can create a folder for animations and move the pic there.



Picce 1 (Texture 2D) Import Settings

Texture Type Sprite (2D and UI)
Texture Shape 2D
Sprite Mode Multiple
Pixels Per Unit 100
Mesh Type Tight
Extrude Edges 1
Generate Physics Sha ✓
Advanced
sRGB (Color Texture) ✓

picce1
1920x1280 (NPOT) RGBA Compressed DXT5|BC3 sRGB 2.3 MB

Create the animation

Open editor

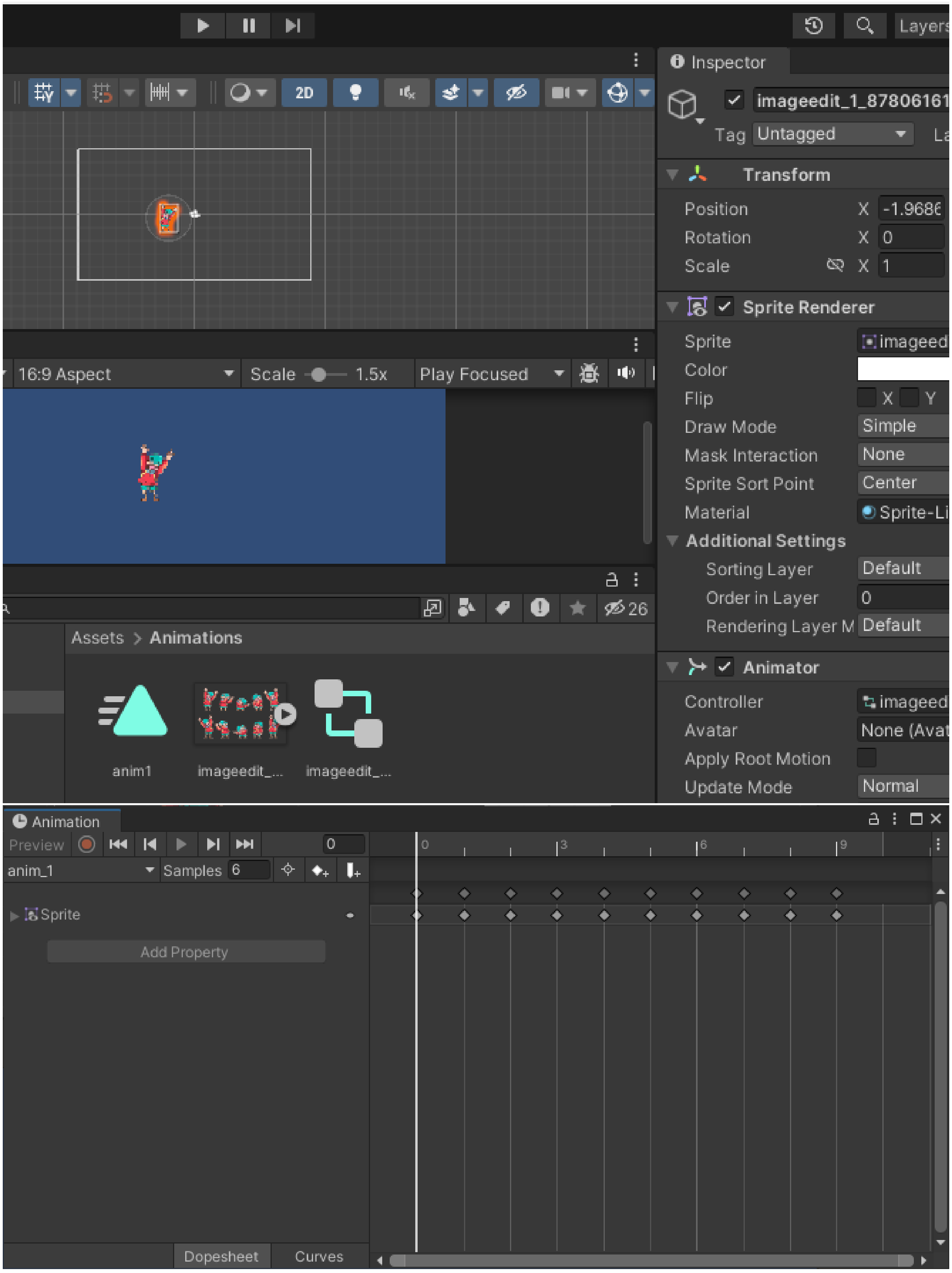
Choose Multiple

Accept by clicking Apply

Now the Animation Controller is saved

Speed of animation can be controlled with this tool

Put the animation to your project and run the app



imageedit_1_87806161
Tag Untagged

Transform
Position X -1.9686
Rotation X 0
Scale X 1

Sprite Renderer
Sprite imageed
Color
Flip X Y
Draw Mode Simple
Mask Interaction None
Sprite Sort Point Center
Material Sprite-Li
Additional Settings
Sorting Layer Default
Order in Layer 0
Rendering Layer M Default

Animator
Controller imageed
Avatar None (Avat
Apply Root Motion
Update Mode Normal

16:9 Aspect
Scale 1.5x
Play Focused

anim_1 Samples 6
Sprite

To make animation slower you can decrease amount of Samples.

Also timings can be edited.

With this tool you can also record an animation.

Try!!

Own Animation

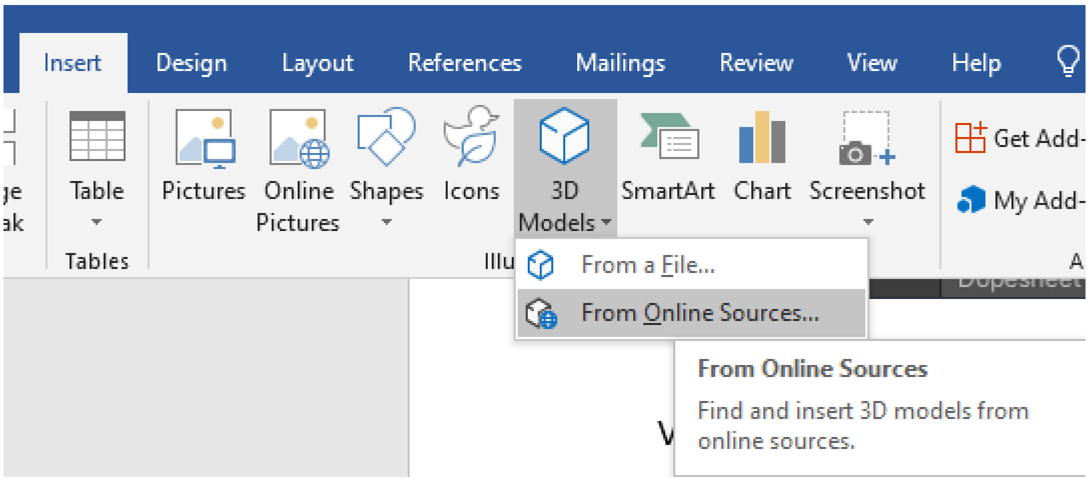
We can use Microsoft's Paint tool

You can choose your own tool freely!

About animation

We want to create a girl face: girl rolls her eyes

You can also use e.g. Ms Word galleries to download a picture!



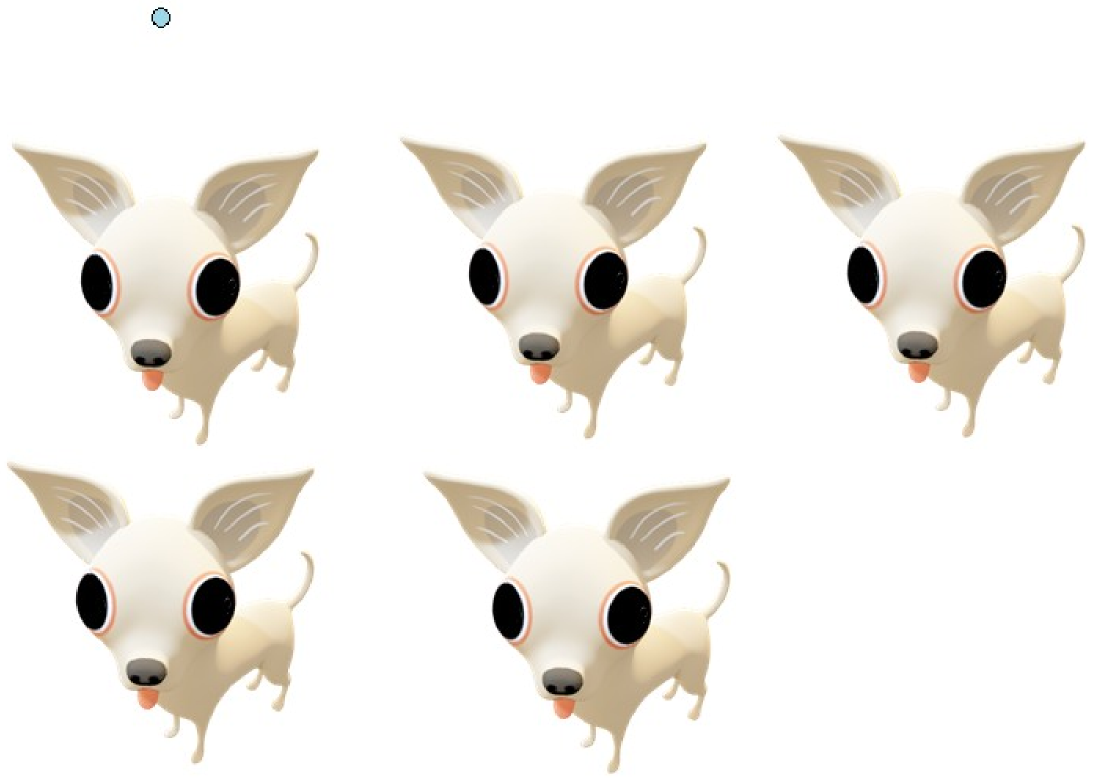
From Online Sources

Find and insert 3D models from online sources.

I take this picture

I take it to Paint

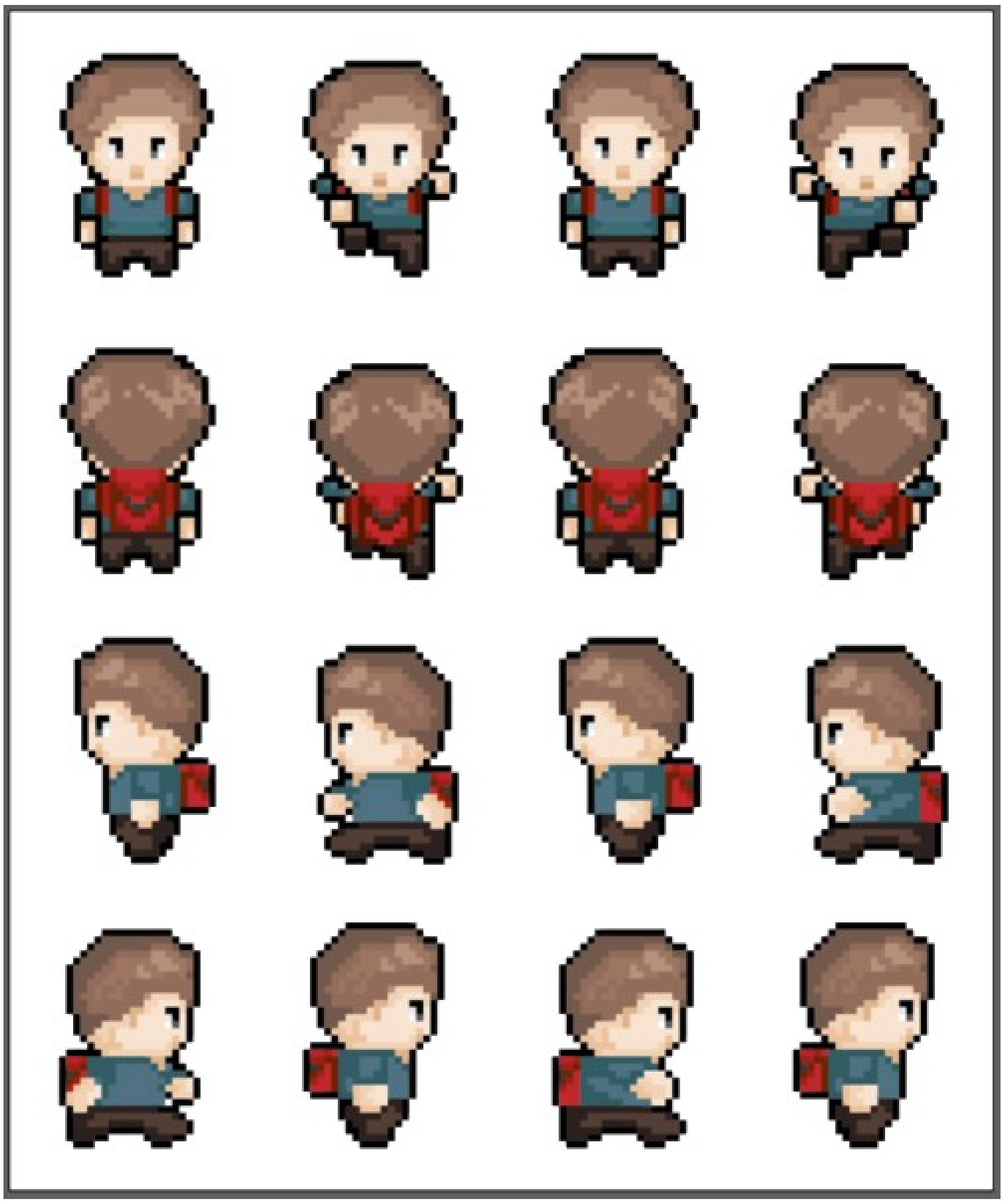
I cut the eyballs from the dog-
Now I copy 5 different dogs and add the eyeballs to each dog to different places.

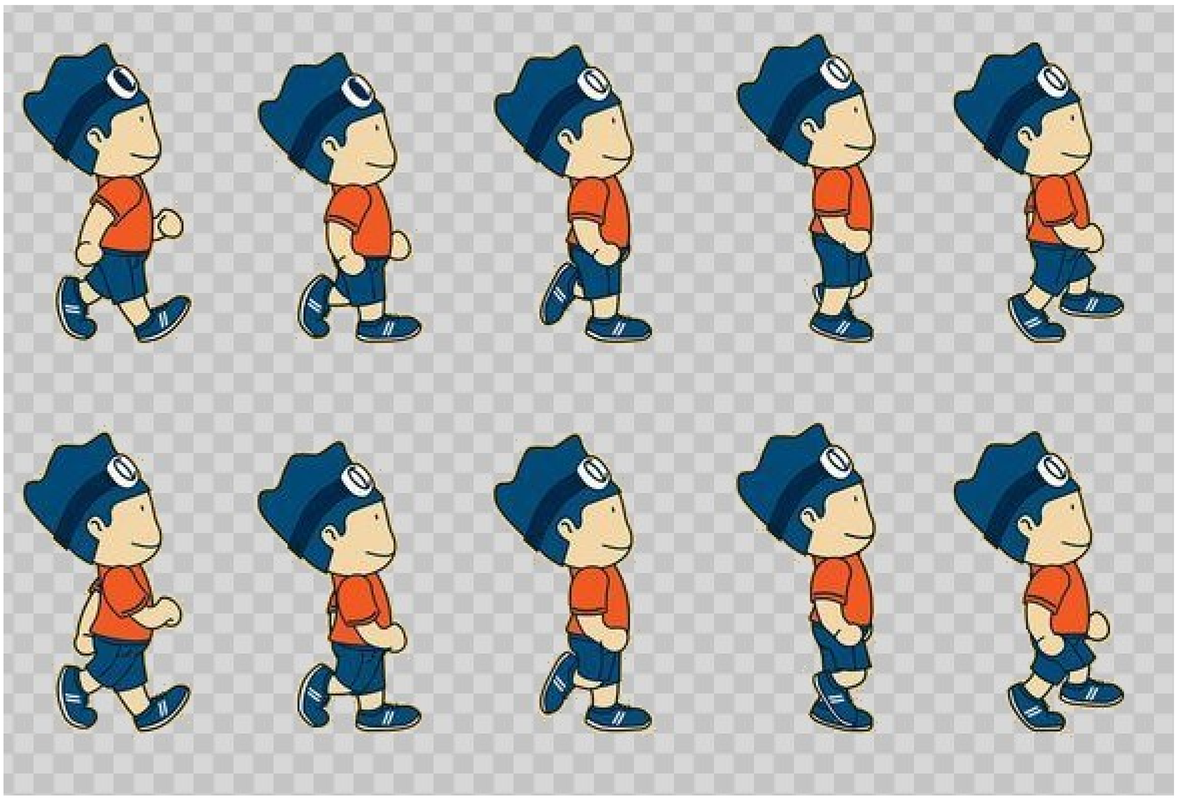


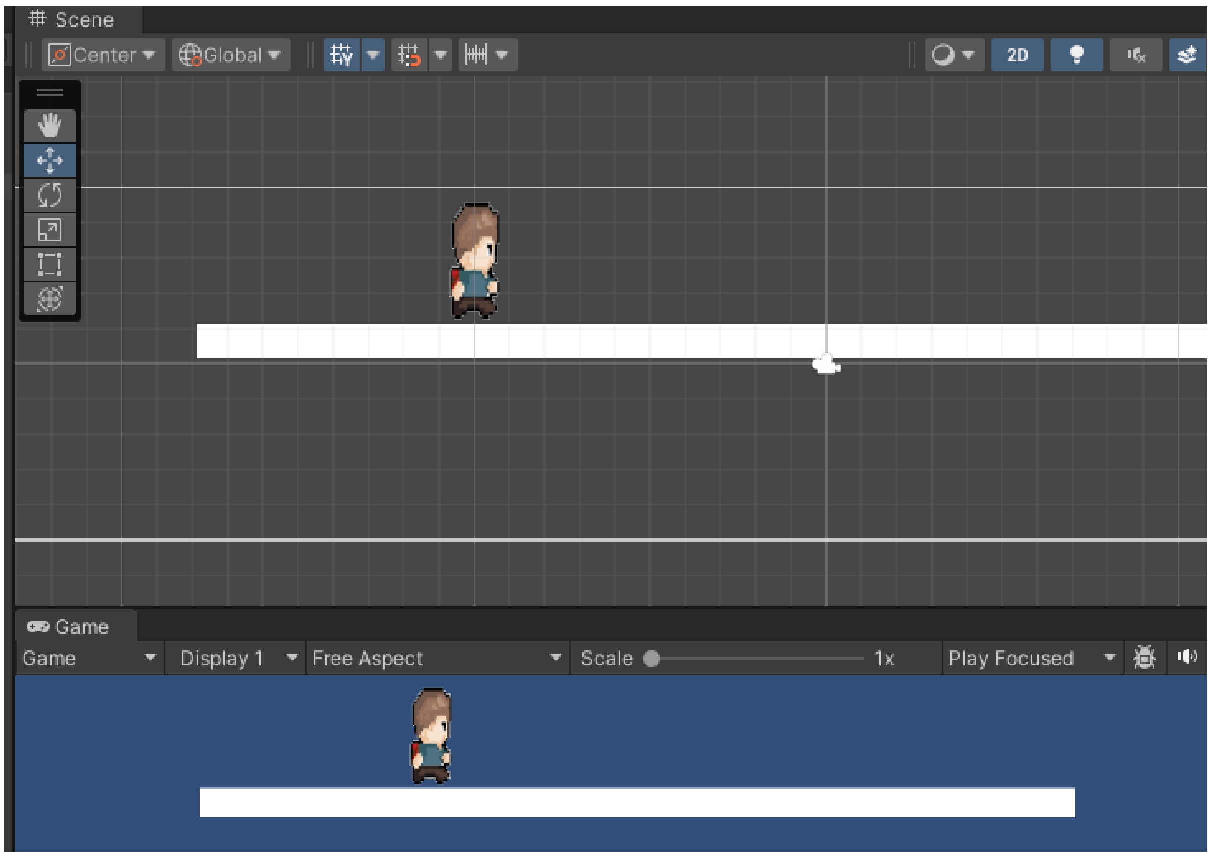
And

I have really seen better ones ;)

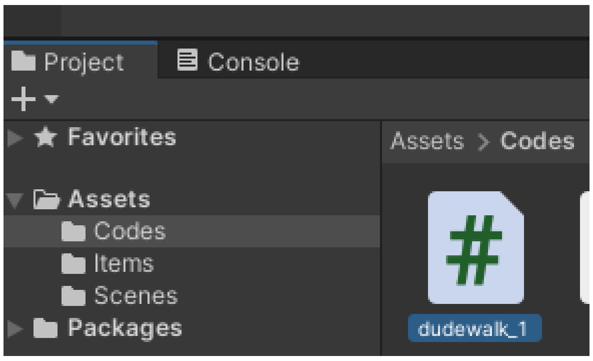
And after the process:







To the folder Codes we create a new script, dudewalk_1.cs:



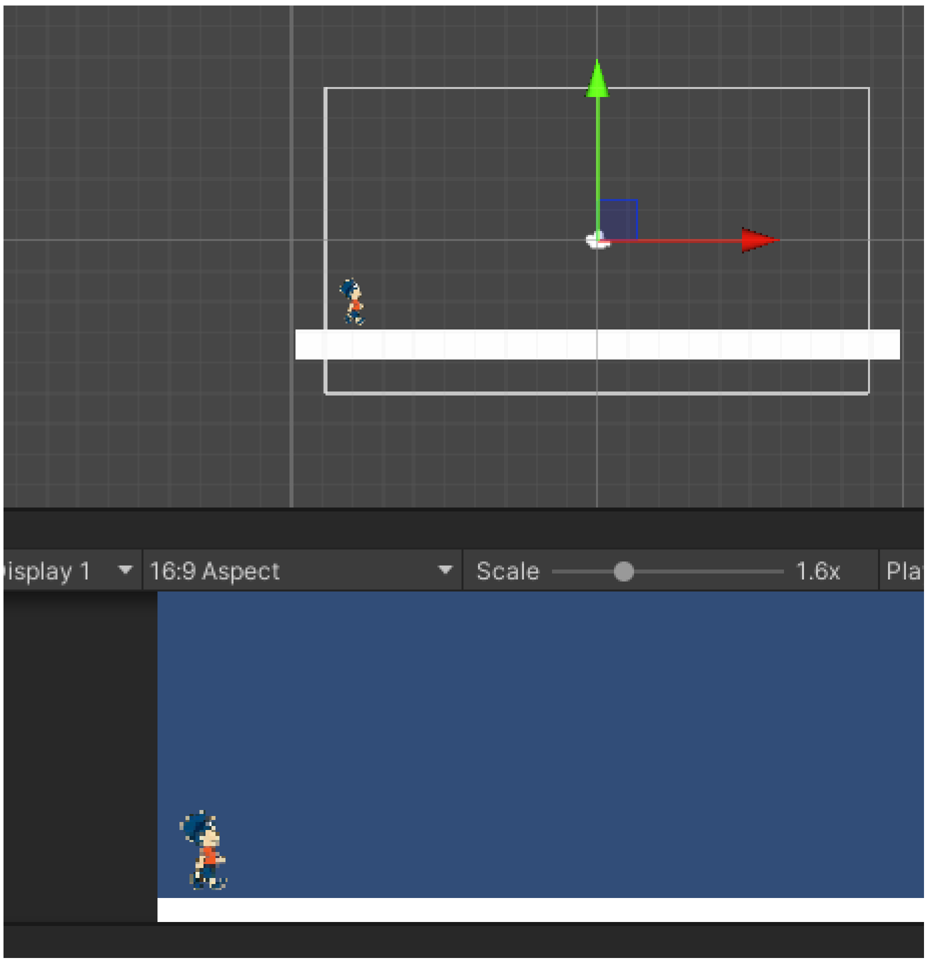
We put the guy to walk back and forth. When the guy reaches some sprecific position, we rotate it 180 degrees.

Remember Transform component that can now be used to rotate an item.

Code

```
GameObject anim1;
    float xx;
    void Start()
    {
        anim1 = GameObject.Find("walking_a");
        anim1.GetComponent<Transform>().position = new Vector3(-5, 3, 0);
        xx = anim1.GetComponent<Transform>().position.x;
    }

    float stepX = 0.05f;

    void Update()
    {
        int s = 1;
        anim1.GetComponent<Transform>().position = new Vector3(xx, 3, 0);
        xx += stepX;
        if (anim1.GetComponent<Transform>().position.x >= 5f)
        {
            stepX = -0.05f;
            anim1.GetComponent<Transform>().rotation =
            new Quaternion(0, 1, 0, 0);
            if (s == 2)
            {
                anim1.GetComponent<Transform>().rotation =
                new Quaternion(0, 0, 0, 0);
                s = 1;
            }

        }
        if (anim1.GetComponent<Transform>().position.x <= -5f)
        {
            stepX = 0.05f;
            if (s == 1)
            {
                anim1.GetComponent<Transform>().rotation =
                 new Quaternion(0, 0, 0, 0);
                s = 2;
            }
        }

    }
```

Test run



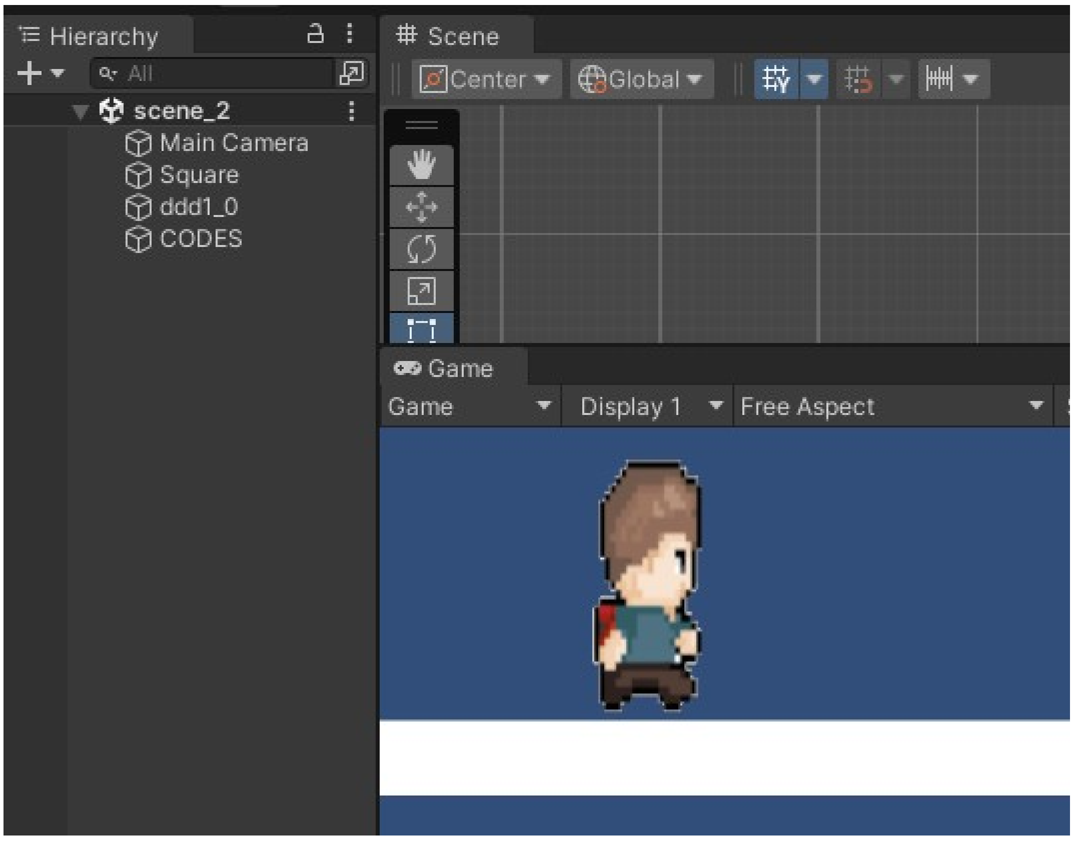
Try yourself!!!

Character control using mouse

Take a look at the Update() method that is shown here below:

```
GameObject anim1;

    void Start()
    {
        anim1 = GameObject.Find("ddd1_0");
    }

    float stepX = 0.01f;
    float xx = -5;
    // Update is called once per frame
    void Update()
    {
        anim1.GetComponent<Transform>().position = new Vector3(xx, 3, 0);
        xx += stepX;
        if (Input.GetMouseButtonDown(0))
        {
            stepX = 0.01f;
            anim1.GetComponent<Transform>().rotation = new Quaternion(0, 0, 0, 0);
        }

        if (Input.GetMouseButtonDown(1))
        {
            stepX = -0.01f;
            anim1.GetComponent<Transform>().rotation = new Quaternion(0, 1, 0, 0);
        }
    }
```

screen copy



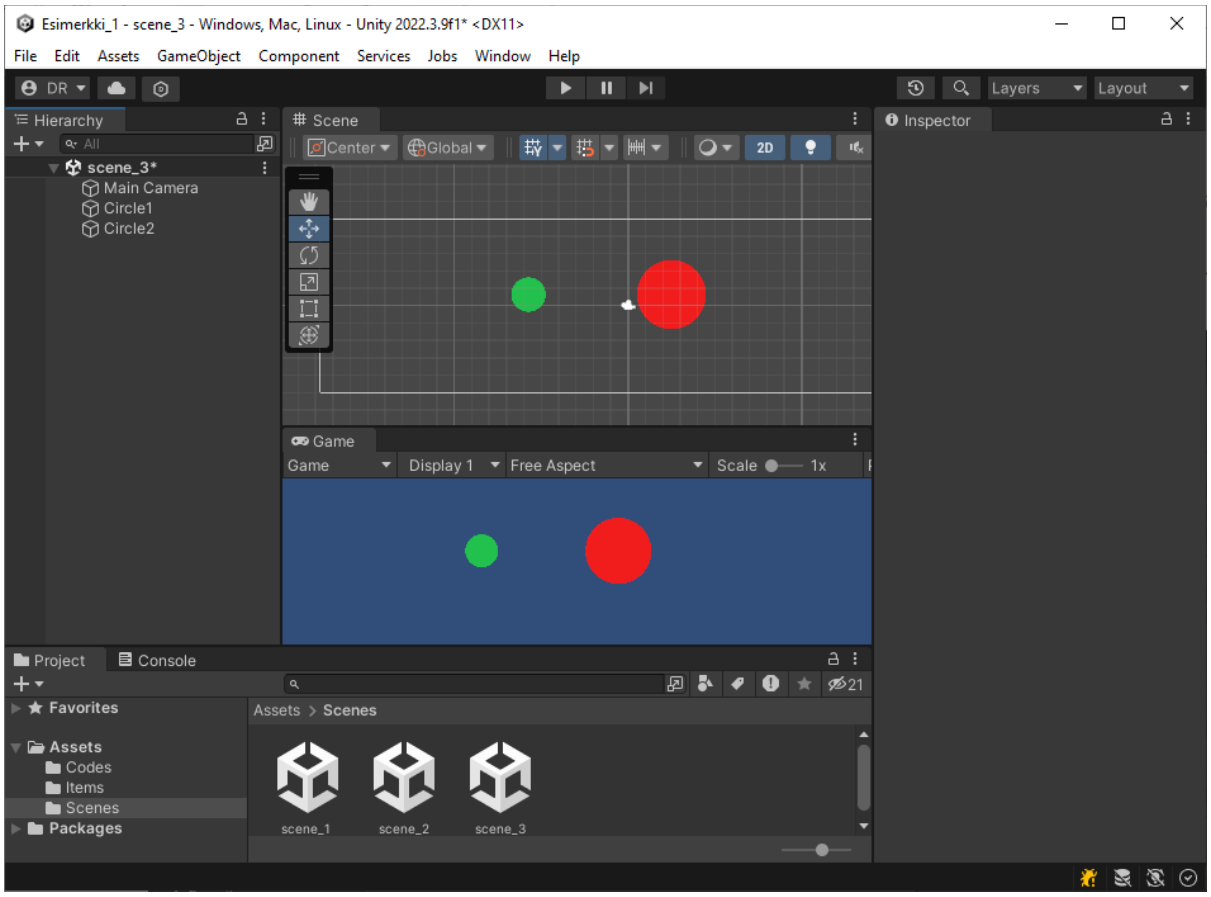
Exercise
Try to put character to walk in 4 directions!

Collisions

When objects touch each other, collision occurs. Collision functions have been added to objects, there are different kinds of collisions. When collision occurs we can by code get to know collision participants...

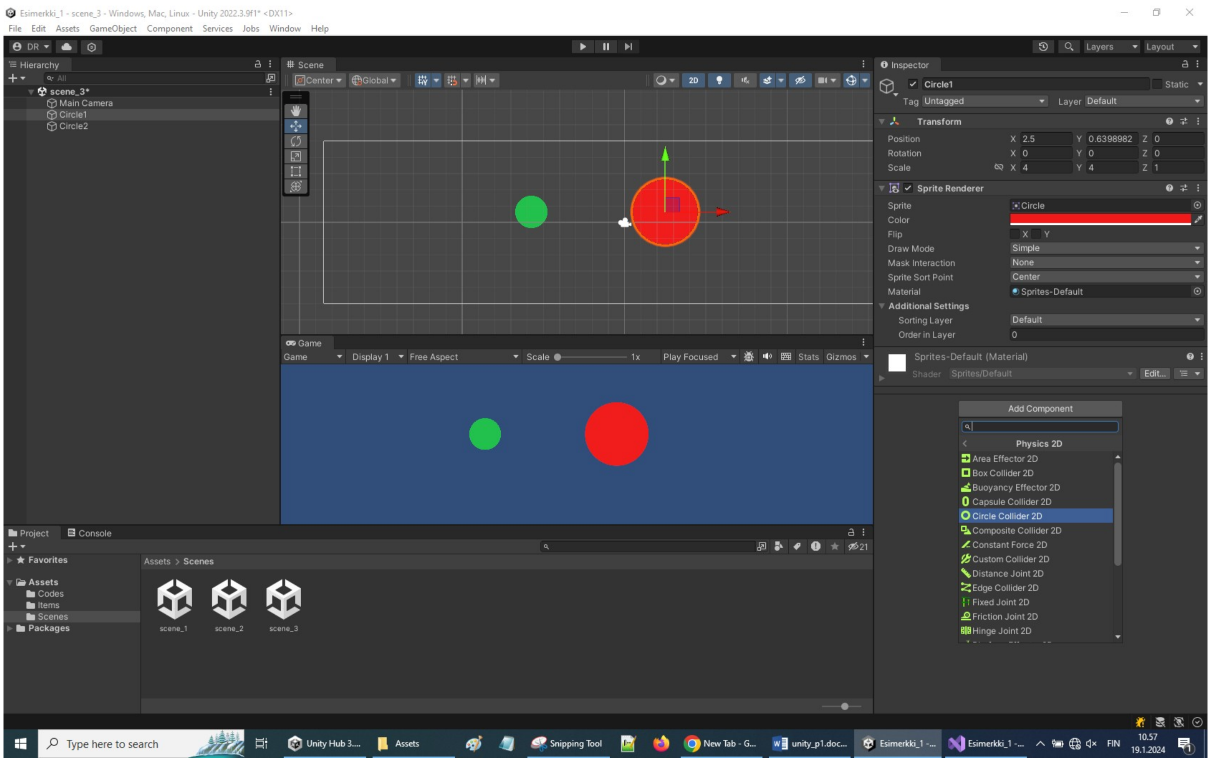
Example: collision

We add there 2 circle sprites first. When collision occurs we play a sound.



Now circles have names Circle1 and Circle2.

Below you can see how to add collider - now to the bigger circle.

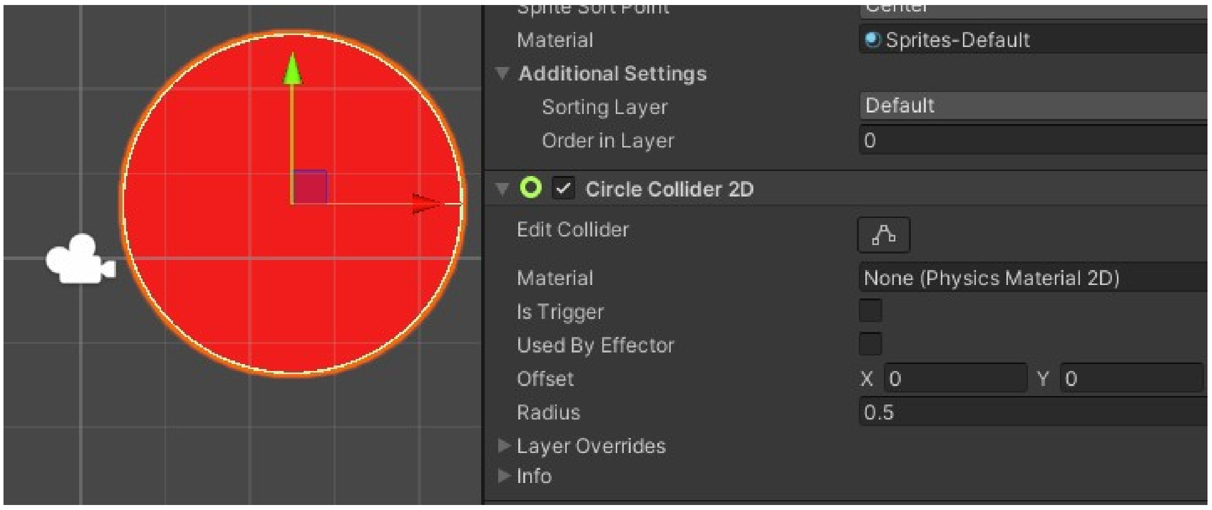


Inspector
Circle1 Static
Tag Untagged Layer Default

Transform
Position X 2.5 Y 0.6398982 Z 0
Rotation X 0 Y 0 Z 0
Scale X 4 Y 4 Z 1

Sprite Renderer
Sprite Circle
Color
Flip X Y
Draw Mode Simple
Mask Interaction None
Sprite Sort Point Center
Material Sprites-Default
Additional Settings
Sorting Layer Default
Order in Layer 0

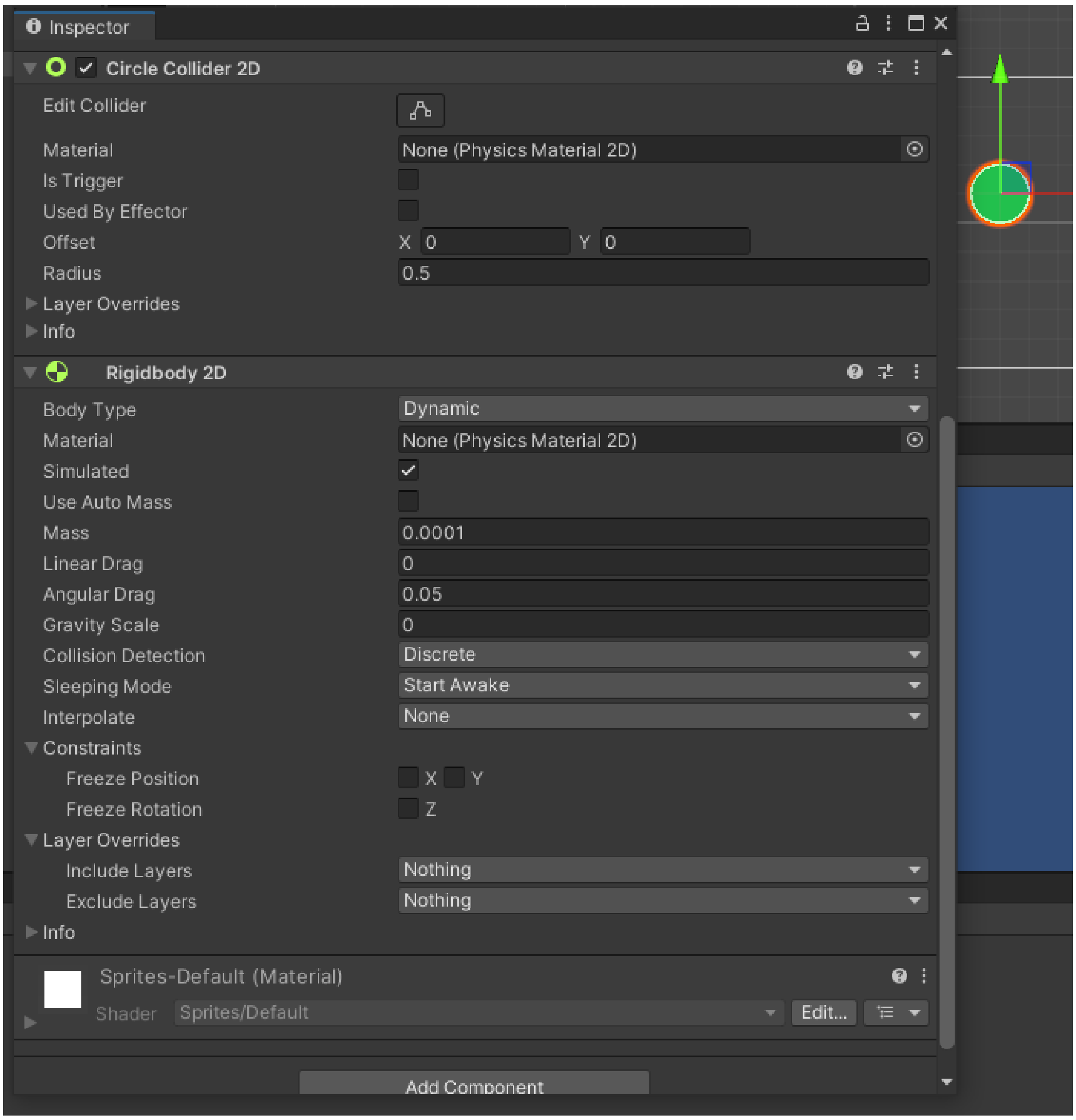
Sprites-Default (Material)
Shader Sprites/Default


Material Sprites-Default
Additional Settings
Sorting Layer Default
Order in Layer 0

Circle Collider 2D
Edit Collider
Material None (Physics Material 2D)
Is Trigger
Used By Effector
Offset X 0 Y 0
Radius 0.5
Layer Overrides
Info

You can change the circle collider are when needed.

We have added these components and values to smaller circle.



Circle Collider 2D

Material: None (Physics Material 2D)
Is Trigger: ☐
Used By Effector: ☐
Offset: X 0 Y 0
Radius: 0.5
Layer Overrides
Info

Rigidbody 2D

Body Type: Dynamic
Material: None (Physics Material 2D)
Simulated: ✓
Use Auto Mass: ☐
Mass: 0.0001
Linear Drag: 0
Angular Drag: 0.05
Gravity Scale: 0
Collision Detection: Discrete
Sleeping Mode: Start Awake
Interpolate: None
Constraints
Freeze Position: ☐ X ☐ Y
Freeze Rotation: ☐ Z
Layer Overrides
Include Layers: Nothing
Exclude Layers: Nothing
Info

Sprites-Default (Material)
Shader: Sprites/Default

To smaller circle we add RigidBody- ja Circle Collider 2D -components.

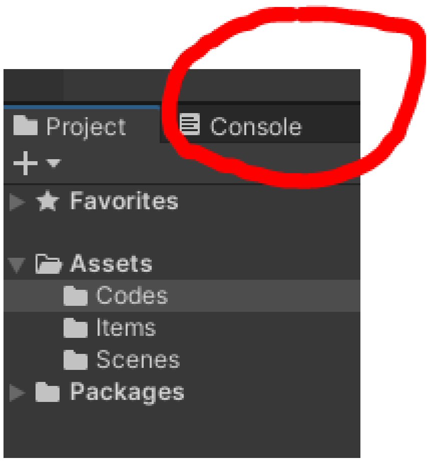
Then write this code:

```
  private void OnTriggerEnter2D(Collider2D collision)
    {
        if (collision.gameObject == circ2)
        {
            Debug.Log("Collision1");
        }

    }
```

Note line;
`Debug.Log("Collision1");`

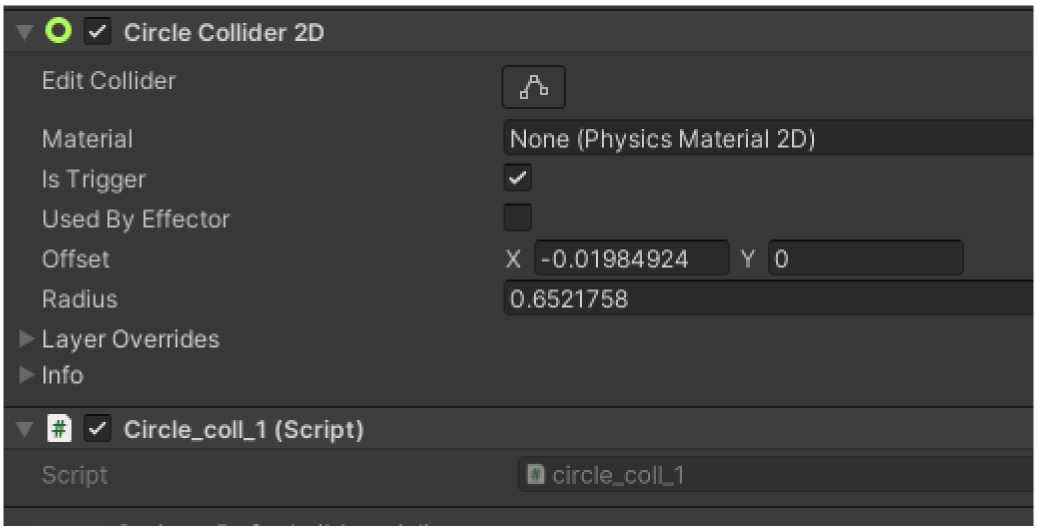
When creating the 1. version of some application, we can use `Debug.Log() method to write output to the console.`

`Console window is a part of the Unity GUI.`

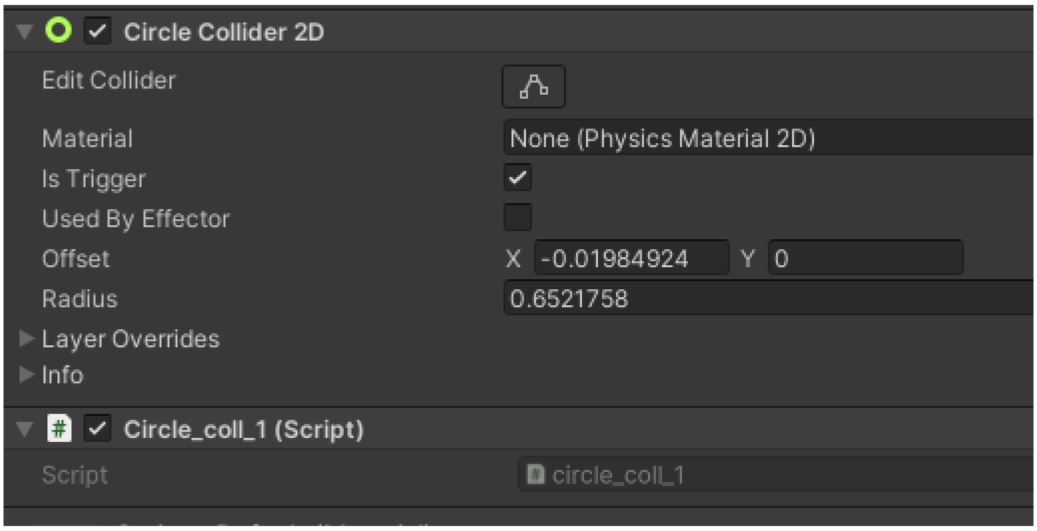


Just for testing can Debug.log() used and we can remove the line later.

Circle1 info:


Circle Collider 2D
Edit Collider
Material: None (Physics Material 2D)
Is Trigger: ✓
Used By Effector:
Offset: X -0.01984924 Y 0
Radius: 0.6521758
Layer Overrides
Info
Circle_coll_1 (Script)
Script: circle_coll_1

We can see that the code belongs now to Circle1.


Circle Collider 2D

Edit Collider

Material: None (Physics Material 2D)

Is Trigger: ✓

Used By Effector

Offset: X -0.01984924 Y 0

Radius: 0.6521758

Layer Overrides

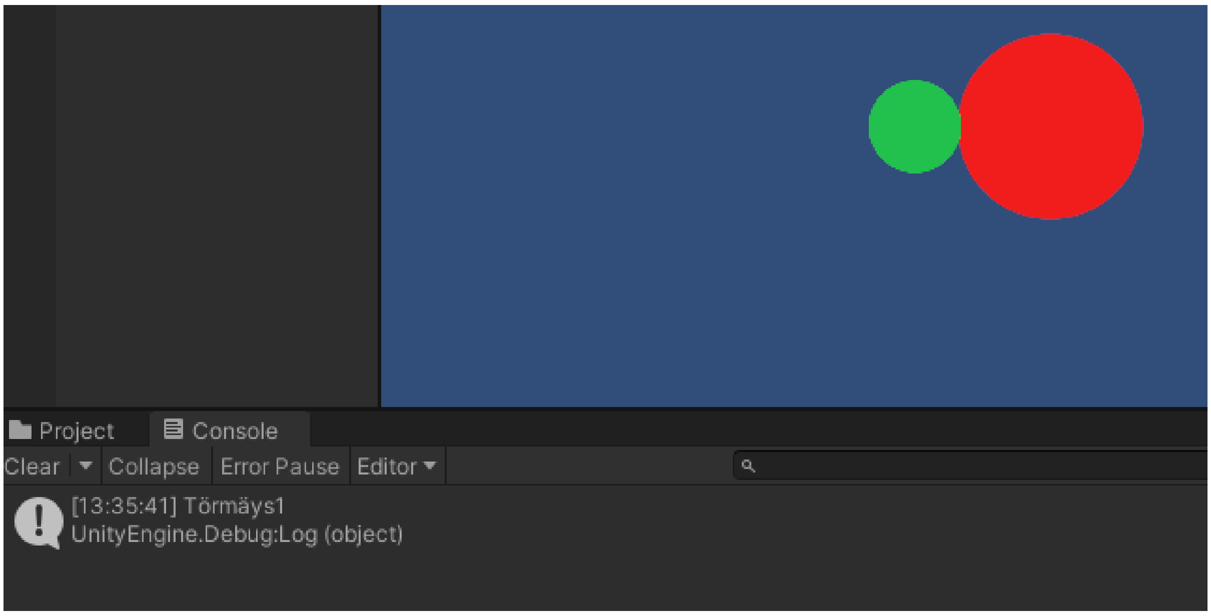
Info

Circle_coll_1 (Script)

Script: circle_coll_1

Trigger property is on and thus Circle 1 follows collisions.

In test run collision was noticed:



```
[13:35:41] Törmäys1
UnityEngine.Debug:Log (object)
```

Add more properties to the example:

add sound

hide bigger circle

...

Add audio

How to add a sound?

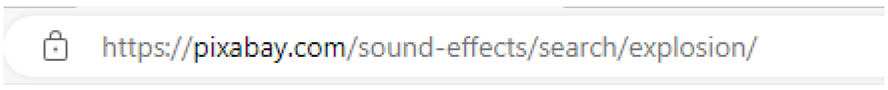
AudioSource is needed and also AudioListener that is as default in Camera object.

Then an audio clip is needed.

We add now AudioSource to Circle2 and search a mp3-clip from internet-

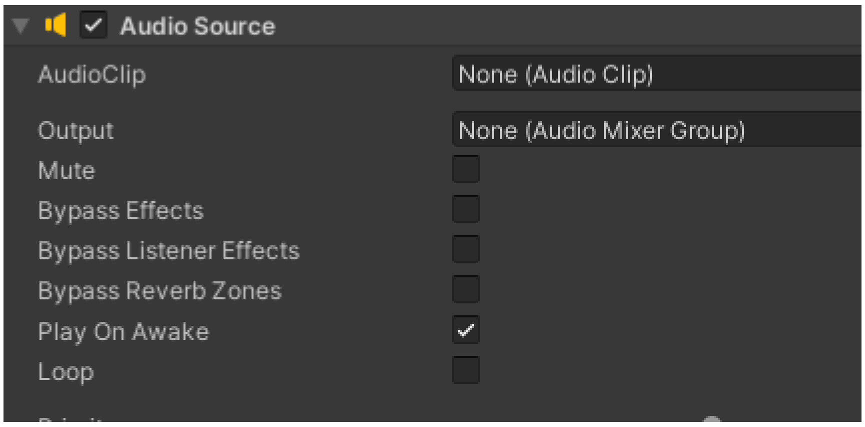
Here is one site:

Kokeilen tällaista sivustoa, josta otan linkissä näkyvän äänitiedoston:


https://pixabay.com/sound-effects/search/explosion/

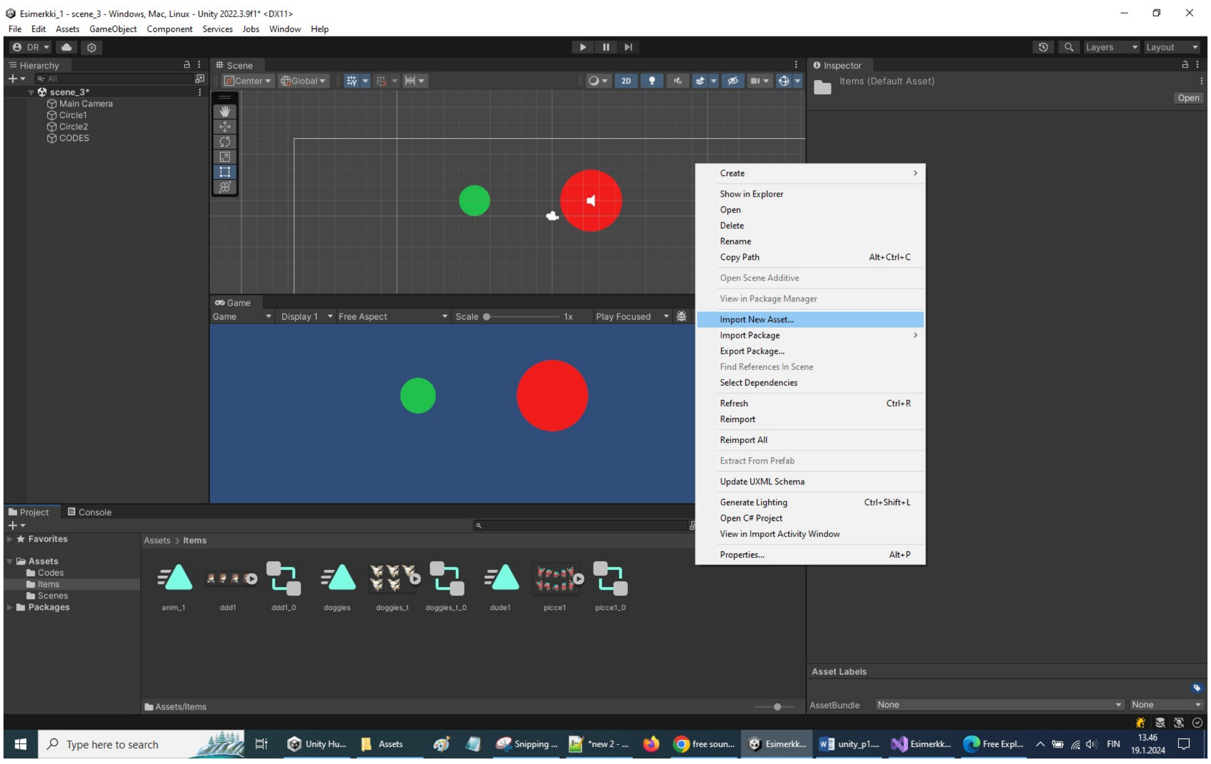
[Free Explosion Sound Effects Download - Pixabay](https://pixabay.com/sound-effects/search/explosion/)

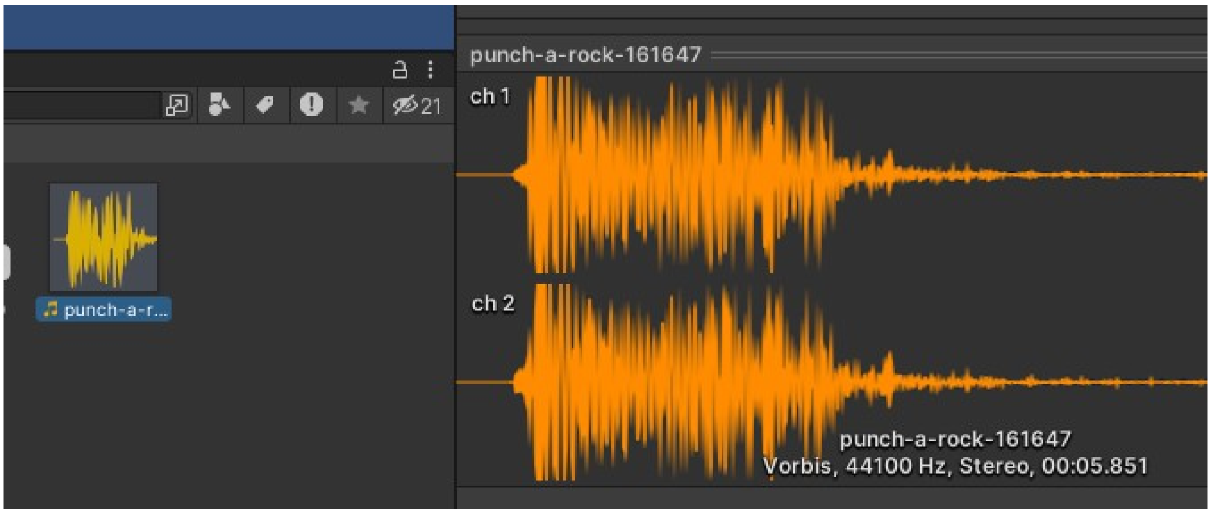
And then


Audio Source

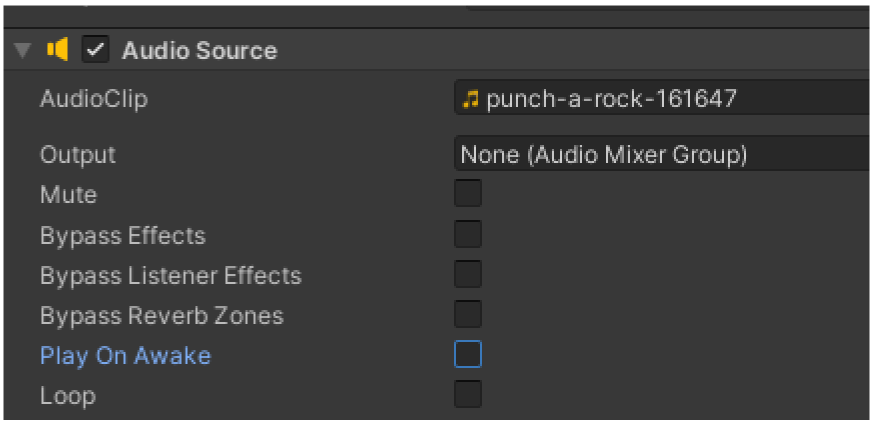
AudioClip None (Audio Clip)
Output None (Audio Mixer Group)
Mute
Bypass Effects
Bypass Listener Effects
Bypass Reverb Zones
Play On Awake ✓
Loop

Load the audioclip to Items folder:





Drag and drop the clip to AudioSource component called AudioClip.


Audio Source

AudioClip	punch-a-rock-161647
Output	None (Audio Mixer Group)
Mute	☐
Bypass Effects	☐
Bypass Listener Effects	☐
Bypass Reverb Zones	☐
Play On Awake	☐
Loop	☐

The the code:

```
    GameObject circ1, circ2;

    AudioSource aus1;
    void Start()
    {
        circ1 = GameObject.Find("Circle1");
        circ2 = GameObject.Find("Circle2");

        aus1 = circ2.GetComponent<AudioSource>();
    }

    private void OnTriggerEnter2D(Collider2D collision)
    {
        if (collision.gameObject == circ2)
        {
            aus1.Play();
            circ1.SetActive(false);
        }

    }
```

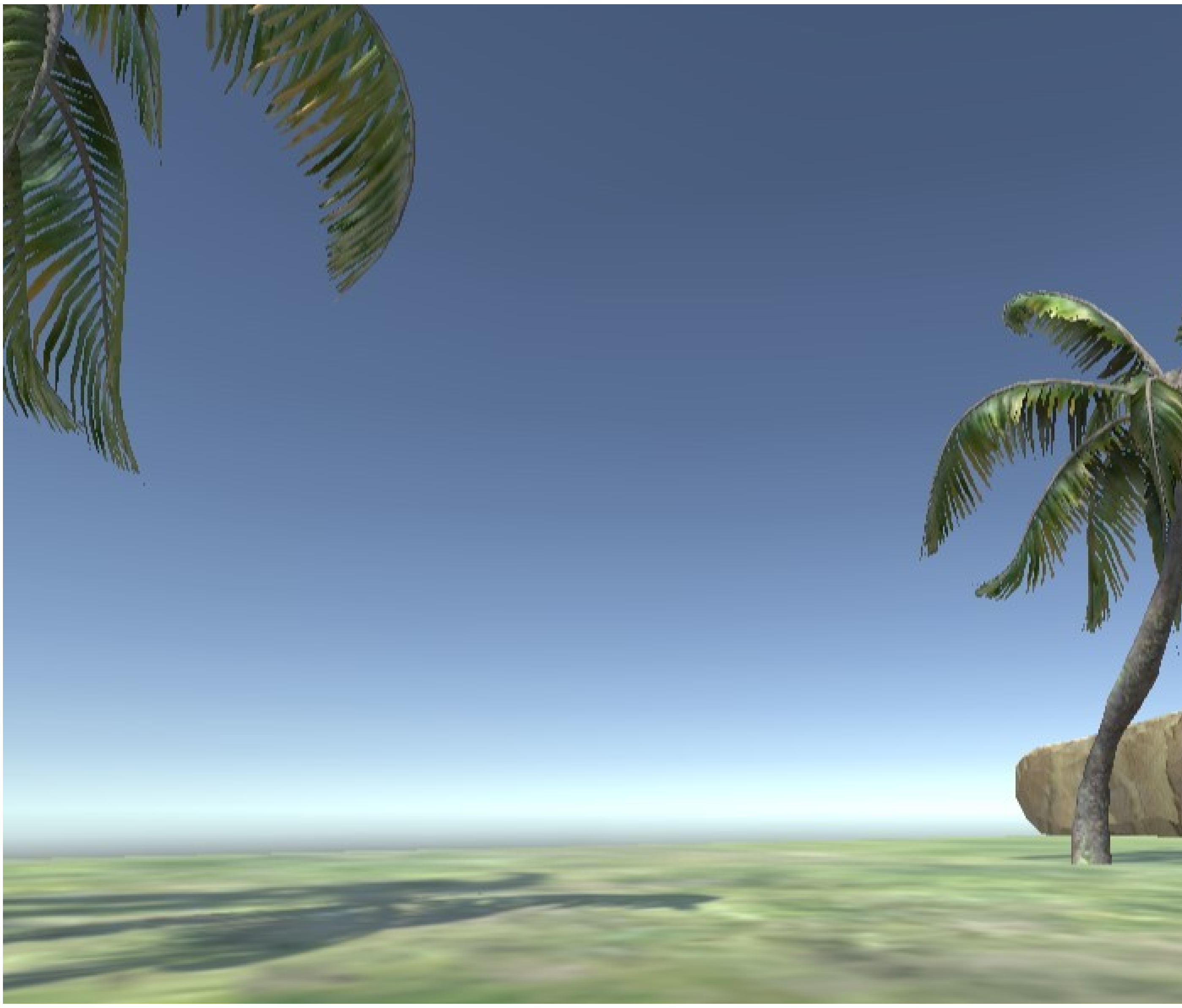
Try!!

Example: background image and movements

Let's create a new scene.

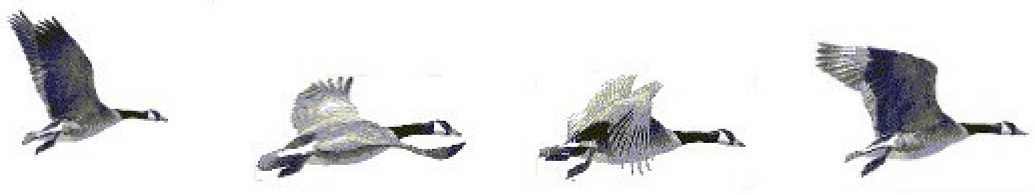
Then images, here is one place:

https://www.rawpixel.com/



Or

For animations I have these bird images.

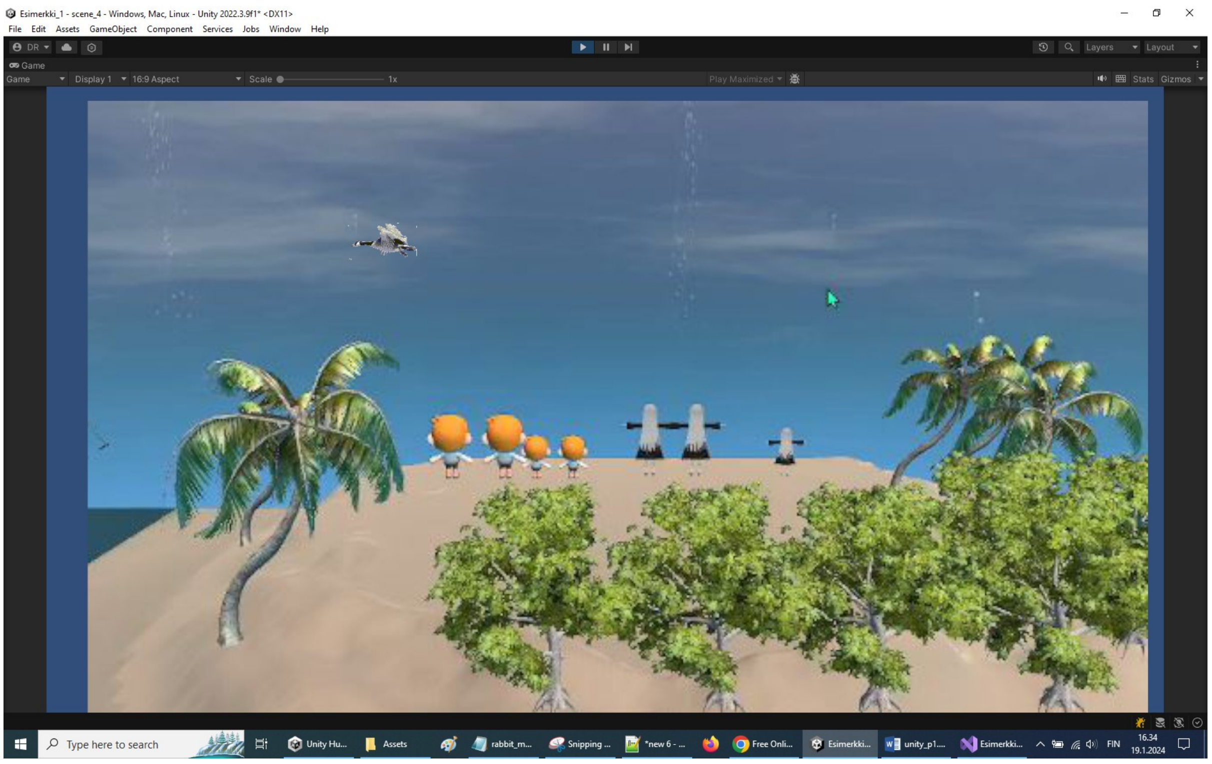


You can remove the background color e.g. with this tool.

https://www.online-image-editor.com/

Then image is imported to Unity: it will be an animation.

Test run


Esimerkki_1 - scene_4 - Windows, Mac, Linux - Unity 2022.3.9f1* <DX11>
File Edit Assets GameObject Component Services Jobs Window Help
Game
Display 1
16:9 Aspect
Scale
1x
Play Maximized
Stats Gizmos
Layers
Layout
Type here to search

This code is connected to bird:

```
float speed = 2f;
    void Start()
    {
    }

    // Update is called once per frame
    void Update()
    {
        this.GetComponent<Transform>().Translate(speed * Time.deltaTime, 0f, 0f);
        if (this.GetComponent<Transform>().position.x > 9f ||
            this.GetComponent<Transform>().position.x < -9f)
        {
            this.GetComponent<Transform>().Rotate(0f, 180f, 0f);
        }
    }
```

Place is updated using Translate-() method, that has parameters x, y and z. we of course could have been able to use Vector3 definition.

When bird reaches some defined position, direction is changed.

Exercise

Put some animal springing back and forth on the ground.

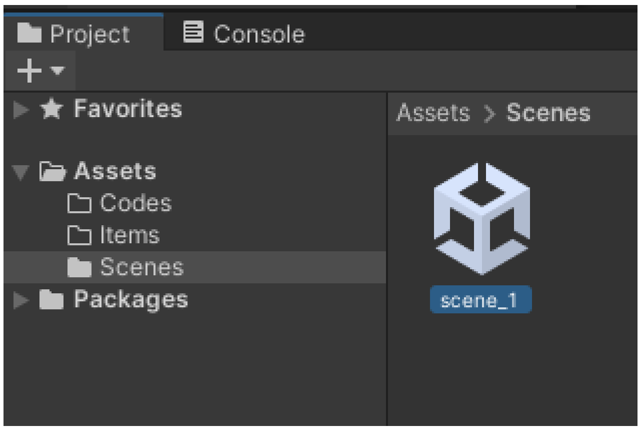
Example: Calculator

Learn basics of GUI: buttons, textfields, input field and how to use code to with them

GUI

We start a new 3D project.

Create folders Items and Codes. Rename the scene.



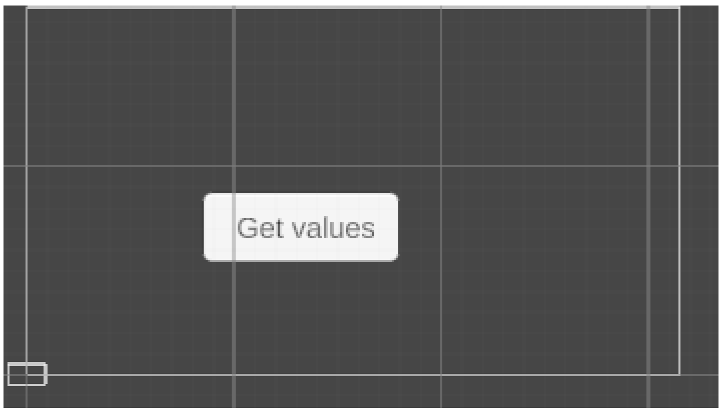
How does it work?

User clicks the button and 2 random values are generated. User gives the sum using an input component. When user clicks the button Check, sum is checked and the result is shown on a text field.

So, we need 2 buttons, 2 text fields and 1 input field.

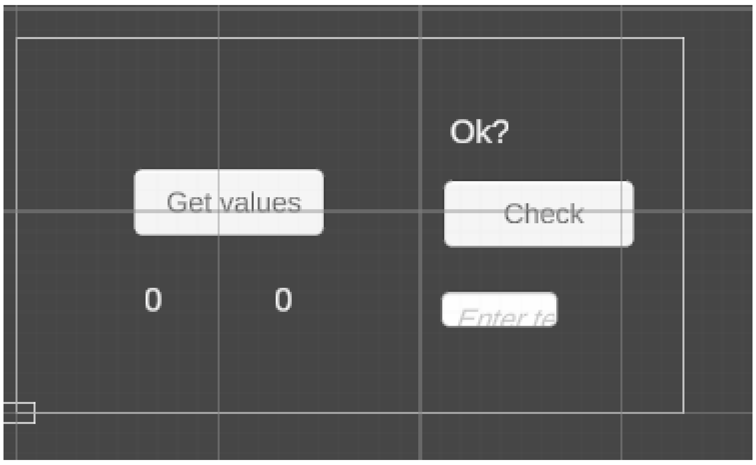
When we add the first button, canvas is added to the Hierarchy: it is the platform form different components.

Button has 2 parts: button itself and a text field. Now text can be ”Generate values” or something like that...You can adjust the font size and button picture so that it is ok. You can manipulate them also later...



Let’s add more components

Version 1



Nw we have write the code. Inside the code we have to get access to those components.

Button ”Get values” generates 2 random values that may be between 30 and 90. Values are added to textfields.

User gives the sum of those values and when Check button is clicked, answer is checked.

Code

We create a new script for those activities above. Script contains functions that then are assigned to buttons..

Script

```
    TMP_Text text1, text2, text3;
    TMP_InputField kk;
    void Start()
    {
```

```
    text1 = GameObject.Find("val1").GetComponent<TMP_Text>();
    text2 = GameObject.Find("val2").GetComponent<TMP_Text>();
    text3 = GameObject.Find("result").GetComponent<TMP_Text>();

    kk = GameObject.Find("answer").GetComponent<TMP_InputField>();
}
int c;
public void readText()
{
    c = Convert.ToInt32(kk.text);
    if (c == d)
    {
        amount++;
        text3.text = "OK: " + amount;
        if (amount == 2)
        {
            // if ok we can add a sound here
        }
    }
}

System.Random gen = new System.Random();

int amount = 0;
int d;
public void generateNew()
{
    int a = gen.Next(30, 90);

    int b = gen.Next(30, 90);

    d = a + b;
    text1.text = "" + a;
    text2.text = "" + b;
    }
```

Ohjelmassa otetaan ensin yhteys konkreettisiin komponentteihin:

```
        text1 = GameObject.Find("val1").GetComponent<TMP_Text>();
        text2 = GameObject.Find("val2").GetComponent<TMP_Text>();

        text3 = GameObject.Find("result").GetComponent<TMP_Text>();

         kk = GameObject.Find("answer").GetComponent<TMP_InputField>();
```

Sitten tehdään metodit, joilla generoidaan luvut ja sitten tarkistetaan annettu vastaus:

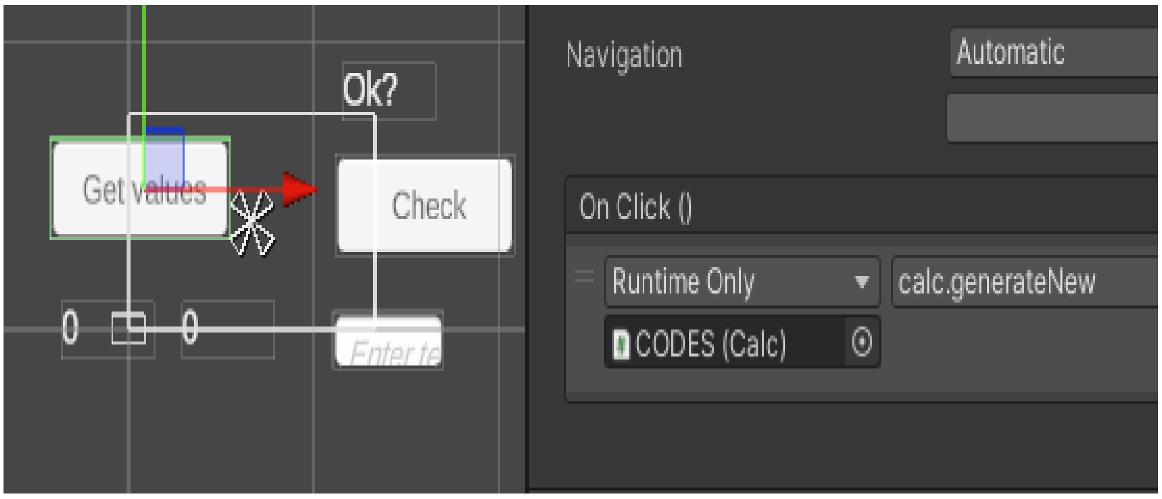
```
System.Random gen = new System.Random();

    int amount = 0;
    int d;
    public void generateNew()
    {
        int a = gen.Next(30, 90);

        int b = gen.Next(30, 90);

        d = a + b;
        text1.text = "" + a;
        text2.text = "" + b;

    }
```

JA

```
    int c;
    public void readText()
    {
        c = Convert.ToInt32(kk.text);
        if (c == d)
        {
            amount++;
            text3.text = "OK: " + amount;
            if (amount == 2)
            {

            }
        }
    }
```

Methods that are of public type are connected to buttons…

Example here:



Navigation: Automatic

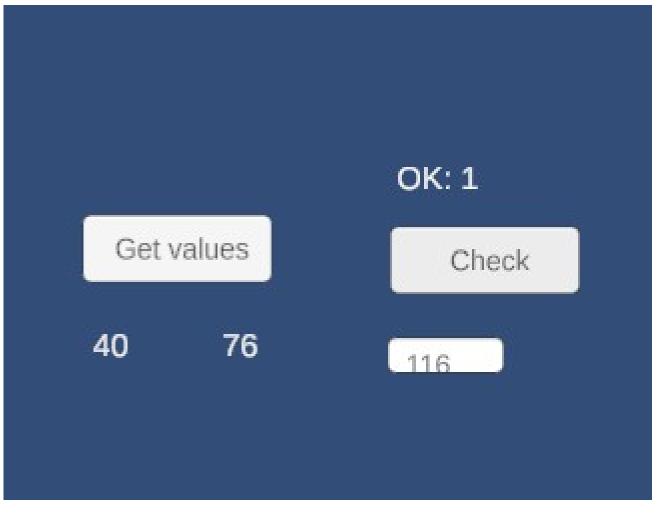
On Click ()

Runtime Only ▾ calc.generateNew

CODES (Calc)

When button ”Get values” is clicked, method calc() is executed...

Test run



Try to make UI better! Change colors, sizes, fonts …

Task

Try to add sounds: one that is played if answer is ok and another that is used when answer is not ok.

You can also count points…

Add images

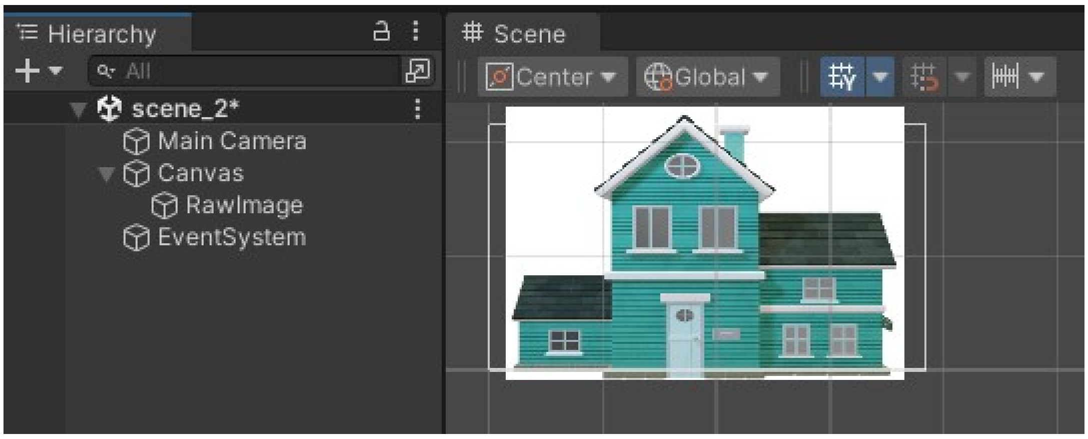
There are many ways.

In UI group there is an object named Image and another called RawImage.

RawImage is ok as a static background image or other parts to make view richer: it is not interactive.

Example here:

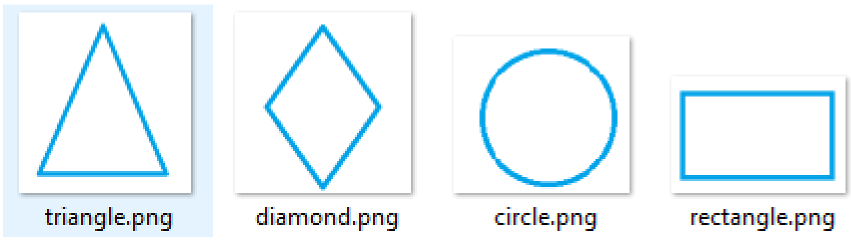
Tässä on kokeeksi laitettu RawImage-kuva:



scene_2*
Main Camera
Canvas
RawImage
EventSystem

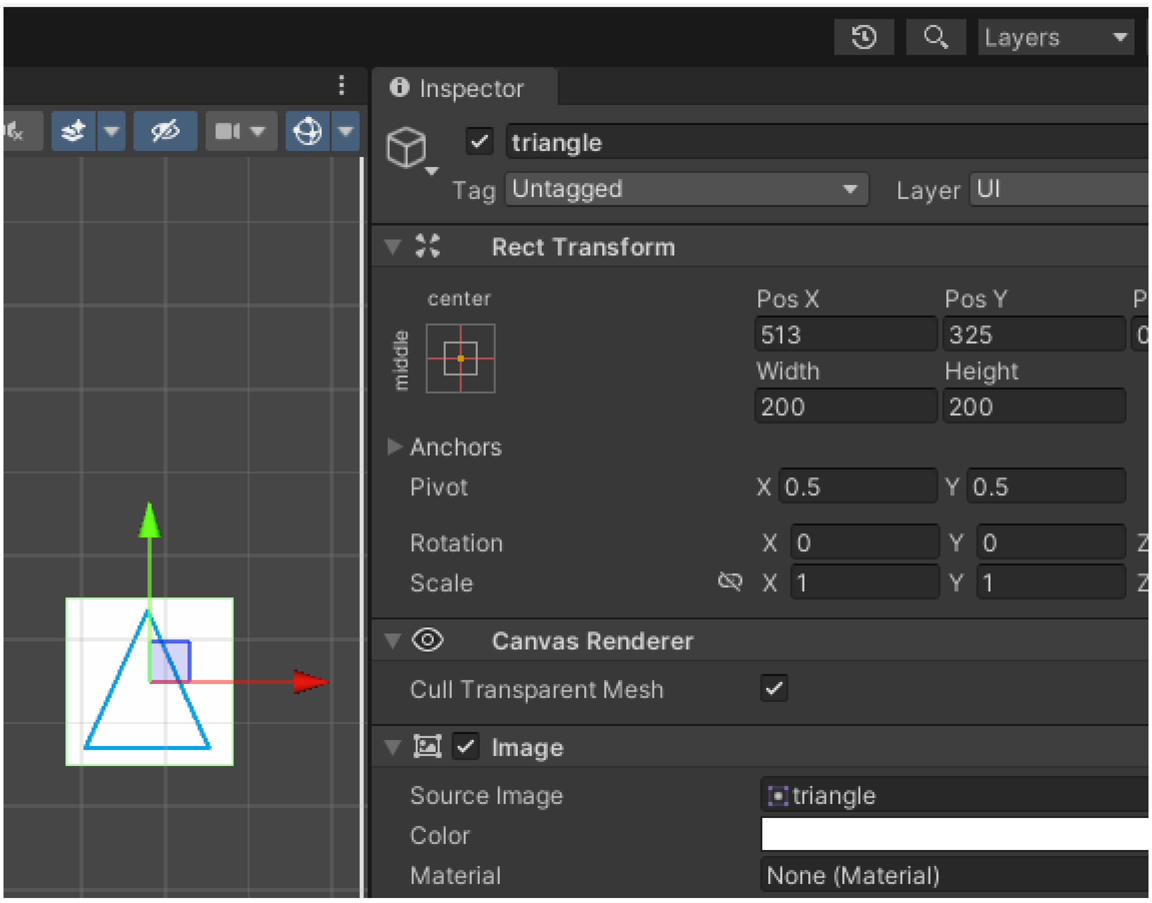
Now we can try Image component: our example tells what does the image present when it is clicked…

Let’s take some shapes:



Let’s take those4 shapes to our project.

Here is a triangle in Image component.



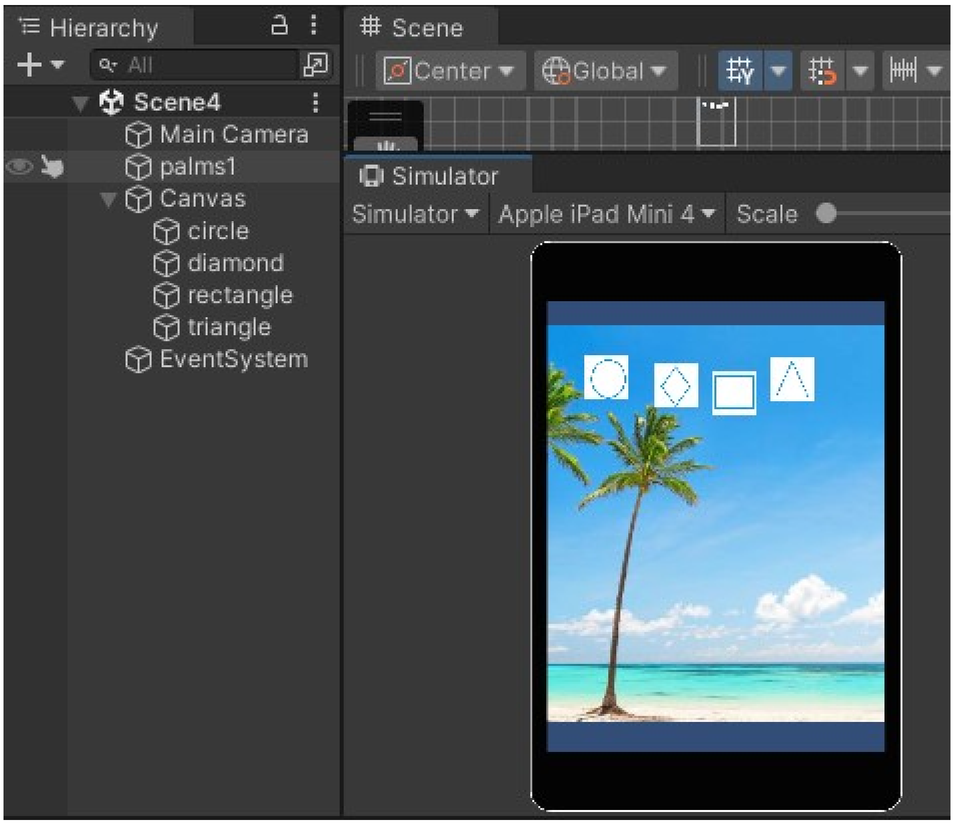
triangle
Tag Untagged Layer UI

Pos X 513 Pos Y 325
Width 200 Height 200
Anchors
Pivot X 0.5 Y 0.5
Rotation X 0 Y 0
Scale X 1 Y 1

Cull Transparent Mesh ✓

Source Image triangle
Color
Material None (Material)

Here are 4 images added to project.

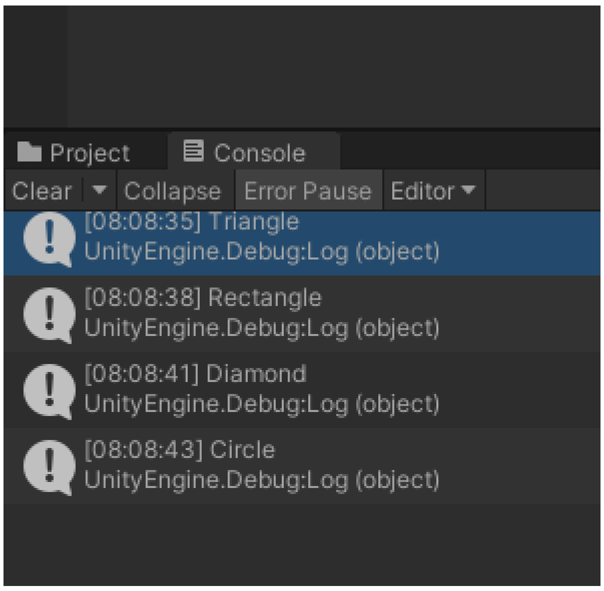


Let’s create the code that tells when the image is clicked. For testing purpose we first add text output only the console.

```
public class clickImage1 : MonoBehaviour, IPointerUpHandler, IPointerDownHandler
{
    public void OnPointerDown(PointerEventData eventData)
    {
        if (this.gameObject.name == "circle")
        {
            Debug.Log("Circle");
        }
        else if (this.gameObject.name == "triangle")
        {
            Debug.Log("Triangle");
        }
        else if (this.gameObject.name == "rectangle")
        {
            Debug.Log("Rectangle");
        }

        else if (this.gameObject.name == "diamond")
        {
            Debug.Log("Diamond");
        }
        else
            Debug.Log(this.gameObject.name);

    }
```

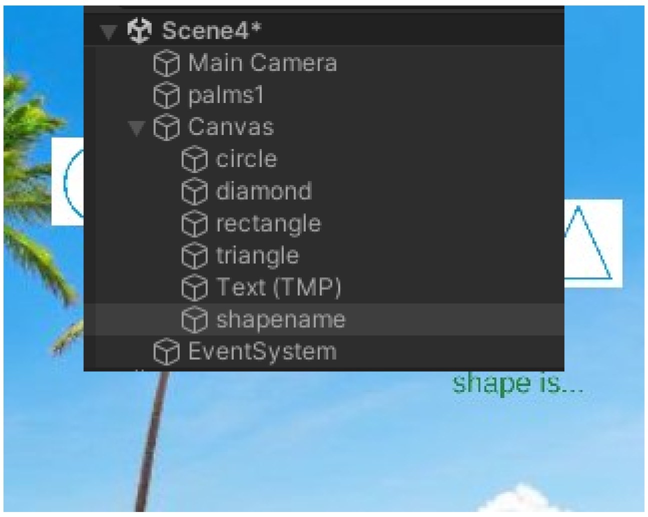
Now we add the script to image components and try it.

This should be printed:



```
[08:08:35] Triangle
UnityEngine.Debug:Log (object)
[08:08:38] Rectangle
UnityEngine.Debug:Log (object)
[08:08:41] Diamond
UnityEngine.Debug:Log (object)
[08:08:43] Circle
UnityEngine.Debug:Log (object)
```

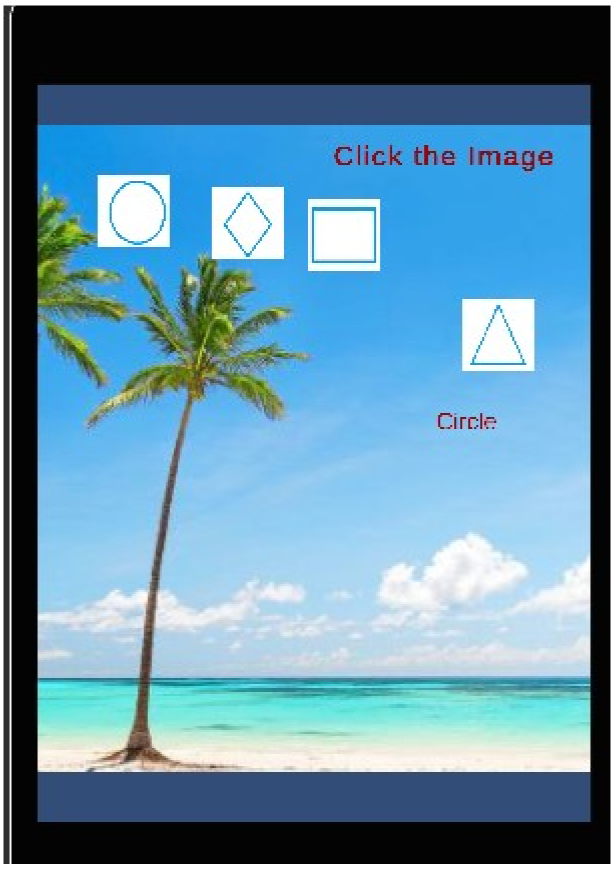
Let's add now a new text component, that shows the message...and another text component that shows the title.


Scene4*
Main Camera
palms1
Canvas
circle
diamond
rectangle
triangle
Text (TMP)
shapename
EventSystem

New code version:

```
 TMP_Text text1;
    void Start()
    {
        text1 = GameObject.Find("shapename").GetComponent<TMP_Text>();
    }
    public void OnPointerDown(PointerEventData eventData)
    {
        if (this.gameObject.name == "circle")
        {
            Debug.Log("Circle");
            text1.text = "Circle";
        }
        else if (this.gameObject.name == "triangle")
        {
            Debug.Log("Triangle");

            text1.text = "Triangle";
        }
        else if (this.gameObject.name == "rectangle")
        {
            Debug.Log("Rectangle");
            text1.text = "Rectangle";
        }

        else if (this.gameObject.name == "diamond")
        {
            Debug.Log("Diamond");

            text1.text = "Diamond";
        }

    }
```

Test run:



Task

Add sounds, too!!
May be you create a serious game that allows player to write the name of the item and that answer is then checked….

Images with buttons

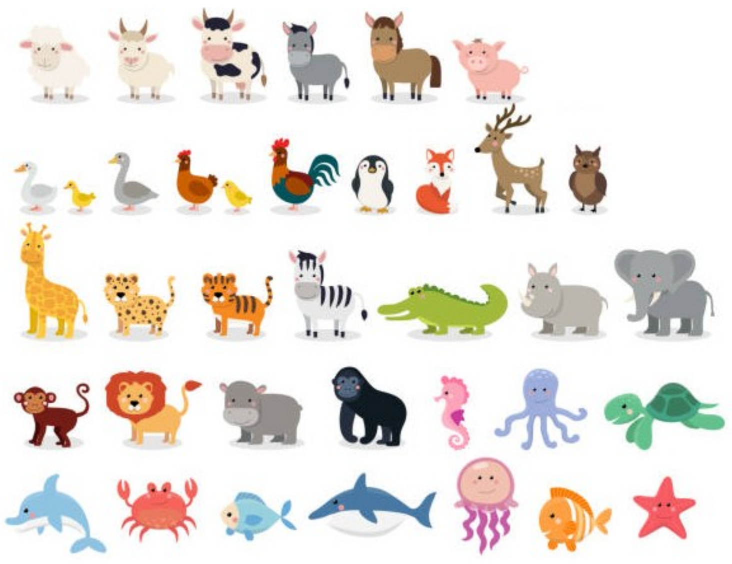
To buttons you can add a smaller image to replace text.

We add buttons first – then we add images to buttons.

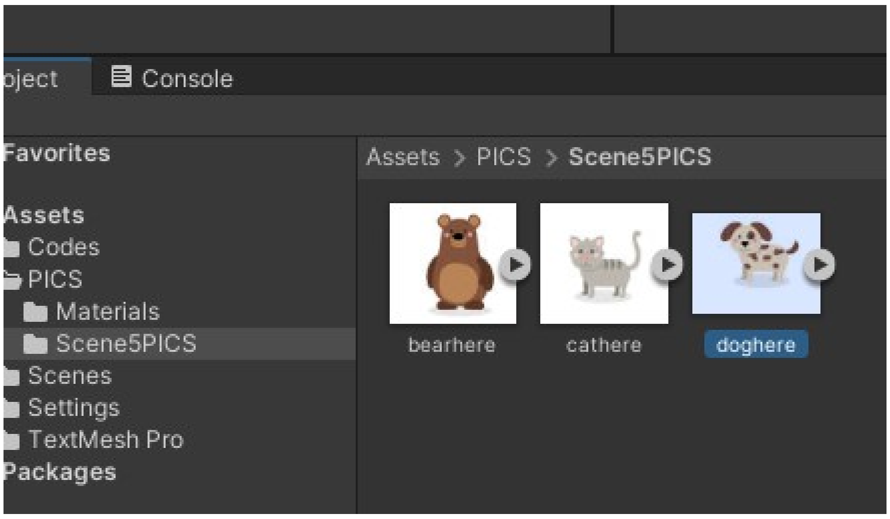
We create code that has a method that is executed when button is clicked.

In the first version we use 3 different methods: one for each button.

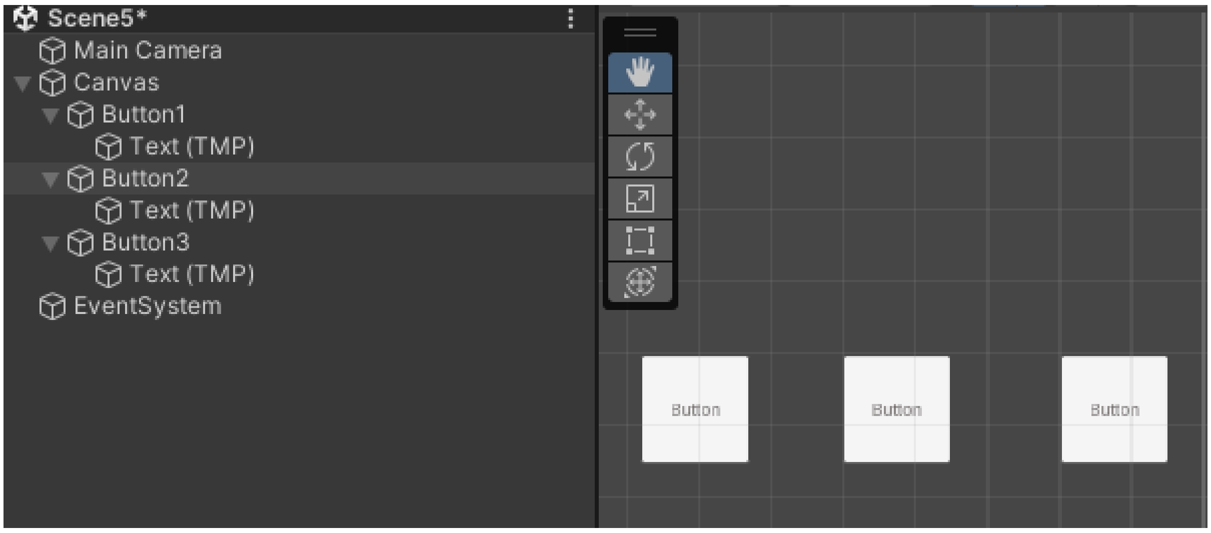
I have a set of animal pictures here:

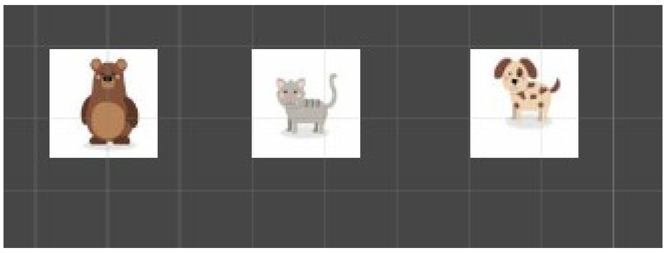


I choose a bear, a cat and a dog now. Then I import pictures to Unity project.



Now I add 3 buttons…





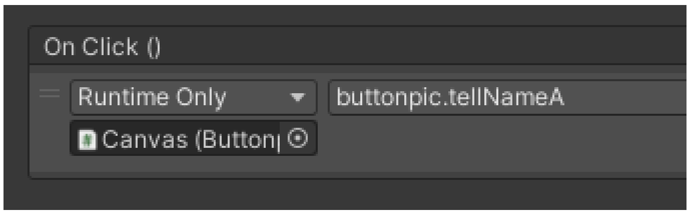
Code 1

```
 TMP_Text text1;
    void Start()
    {
        text1 = GameObject.Find("text1").GetComponent<TMP_Text>();
    }
    public void tellNameA()
    {
        text1.text = "Bear";
    }
    public void tellNameB()
    {
        text1.text = "Cat";
    }
    public void tellNameC()
    {
            text1.text = "Dog";
    }
```

We can now add script to Canvas.

We have to also add ’click eventhandling’ to each button…

Example here:

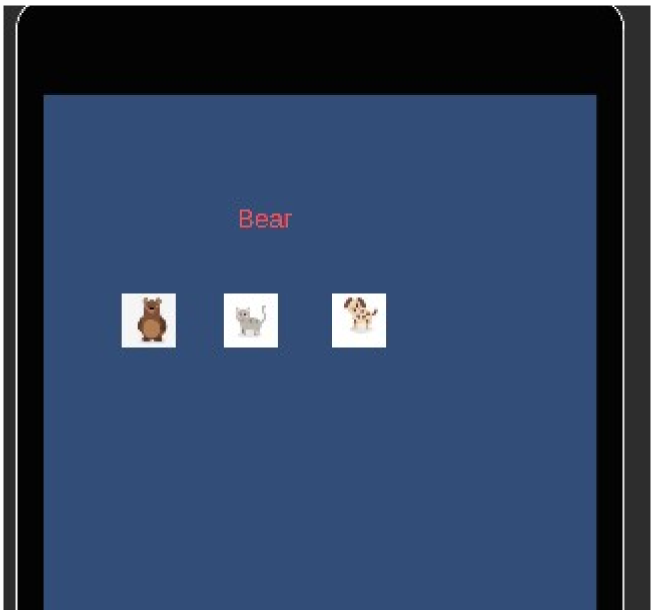


On Click ()

Runtime Only | buttonpic.tellNameA

Canvas (Button)

Test run



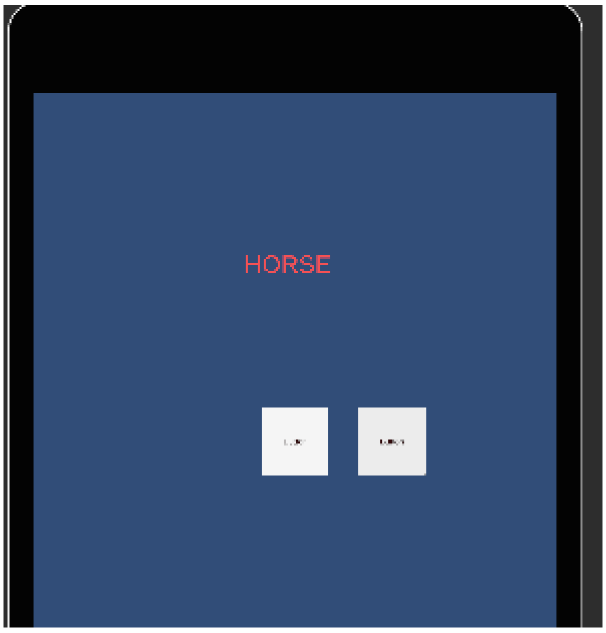
We used own method to each button above. But now we create only one method that identifies the component that was clicked.

Here is the main idea:
EventSystem.current.currentSelectedGameObject.name

Code

```
TMP_Text text1;

    void Start()

    {

        text1 = GameObject.Find("text1").GetComponent<TMP_Text>();

    }

    public void tellName()

    {

        if (EventSystem.current.currentSelectedGameObject.name == "Cow")


            text1.text = "COW";


        if (EventSystem.current.currentSelectedGameObject.name == "Horse")


            text1.text = "HORSE";
    }
```

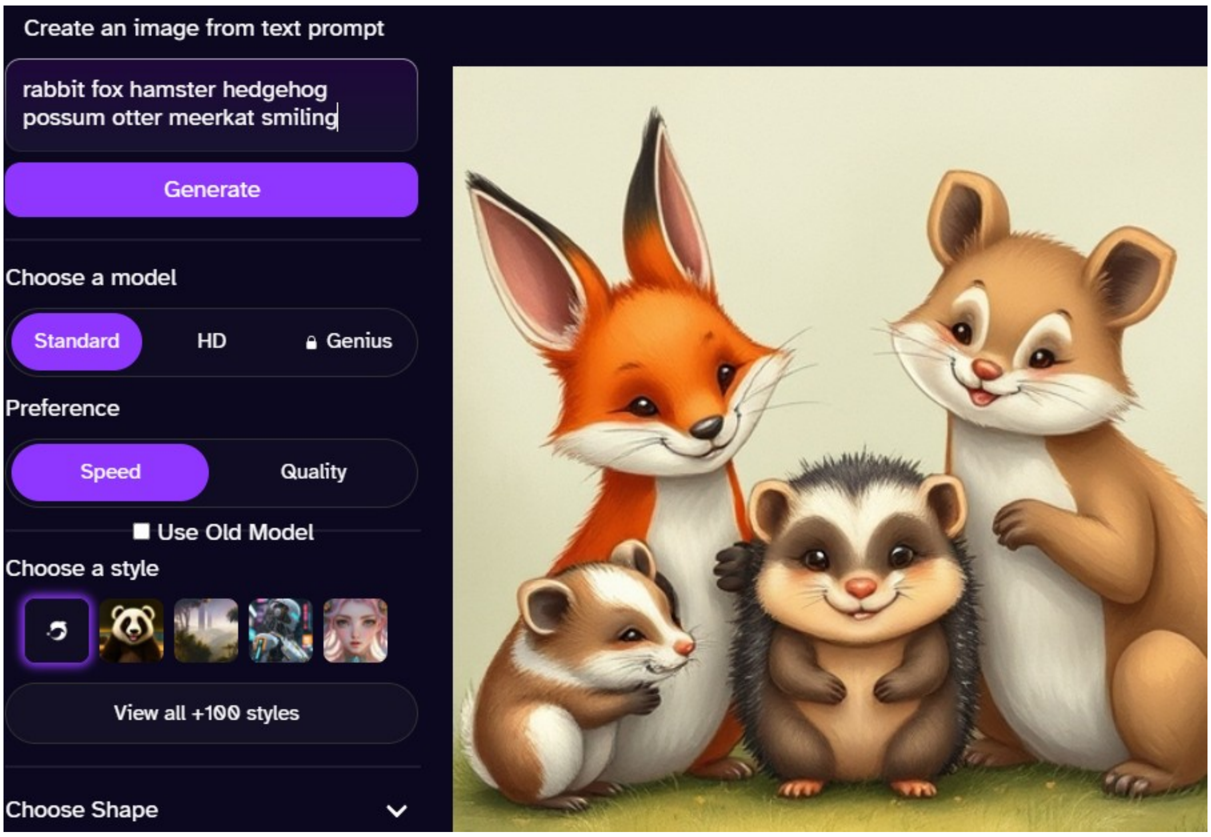
Test run



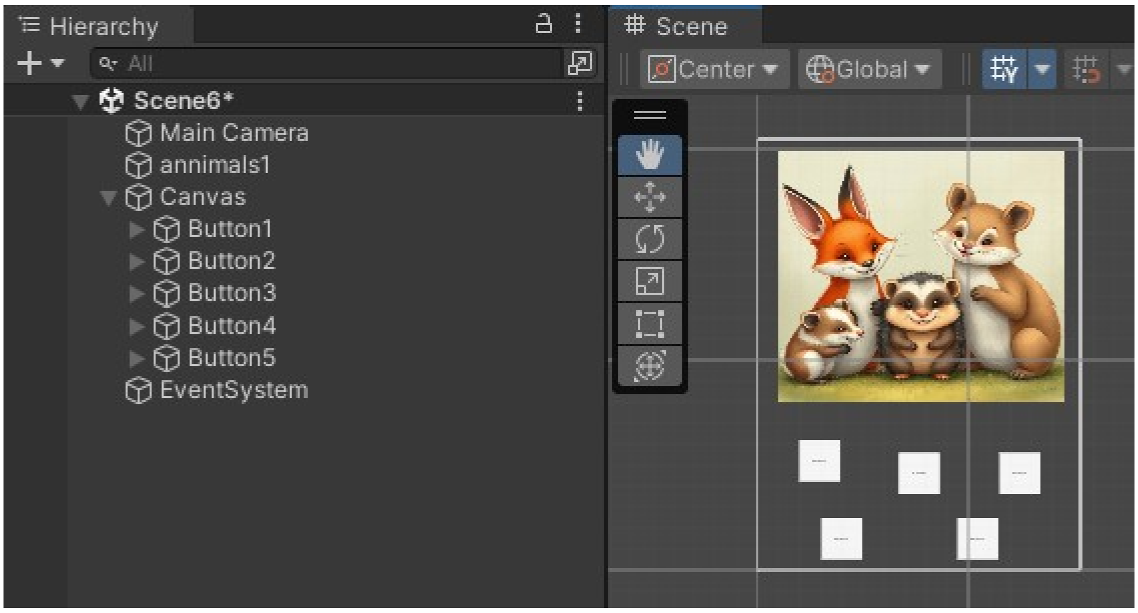
Click parts of a big image

Now we use a bigger image ant want to get information about some parts of the image.

So one way is to use transparent buttons that are added above target parts. Let’s let AI create an image that contains 5 different animals: we user clicks the animal, information is shows on a textbox. Here we generate animas image:

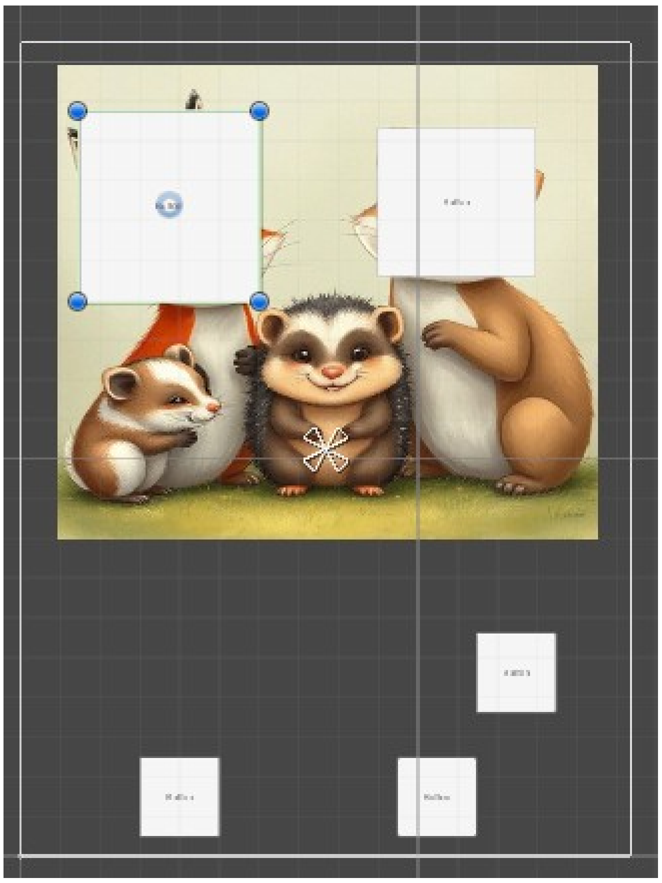


Here are components and image added to the project:



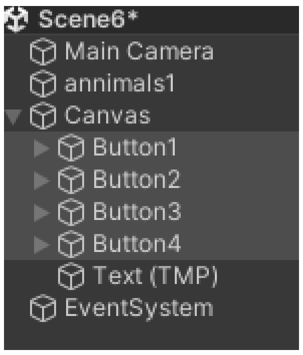
Now we put buttons on the animals and hide them.

Here is the beginning: we have added 2 buttons and later we are going to hide them!

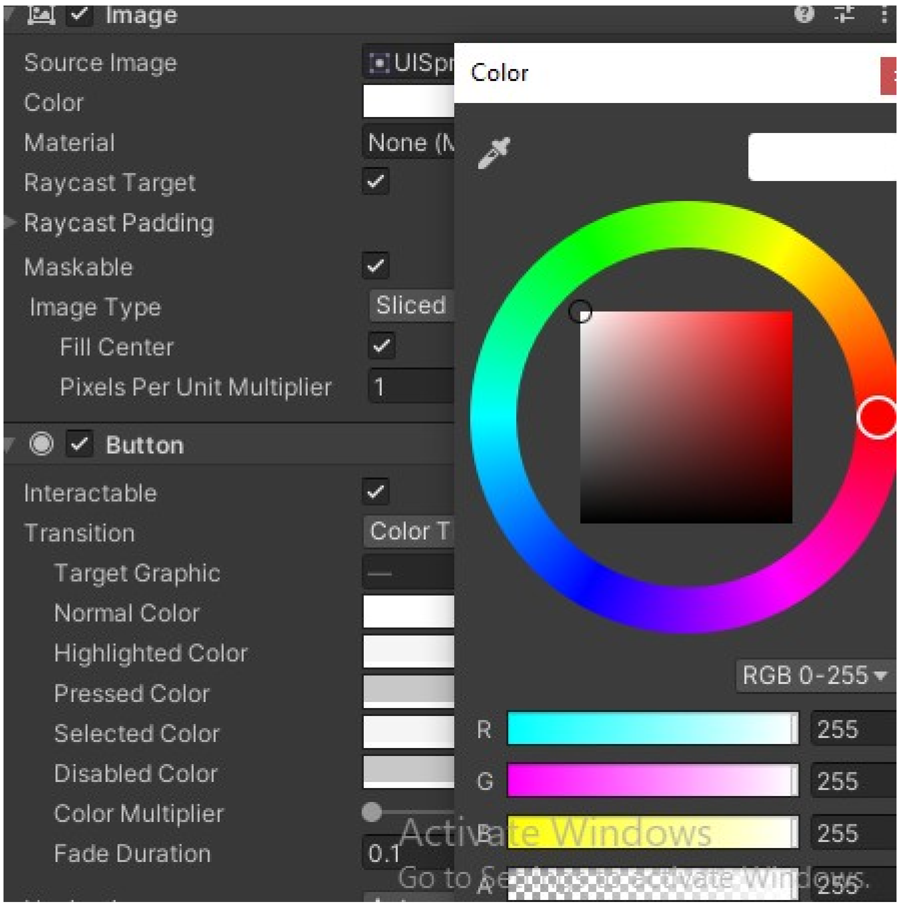


Making buttons transparent:

Select buttons



Adjust color



Image

Source Image UISpr
Color
Material None (M
Raycast Target ✓
Raycast Padding
Maskable ✓
Image Type Sliced
Fill Center ✓
Pixels Per Unit Multiplier 1

Button

Interactable ✓
Transition Color T
Target Graphic
Normal Color
Highlighted Color
Pressed Color
Selected Color
Disabled Color
Color Multiplier
Fade Duration 0.1

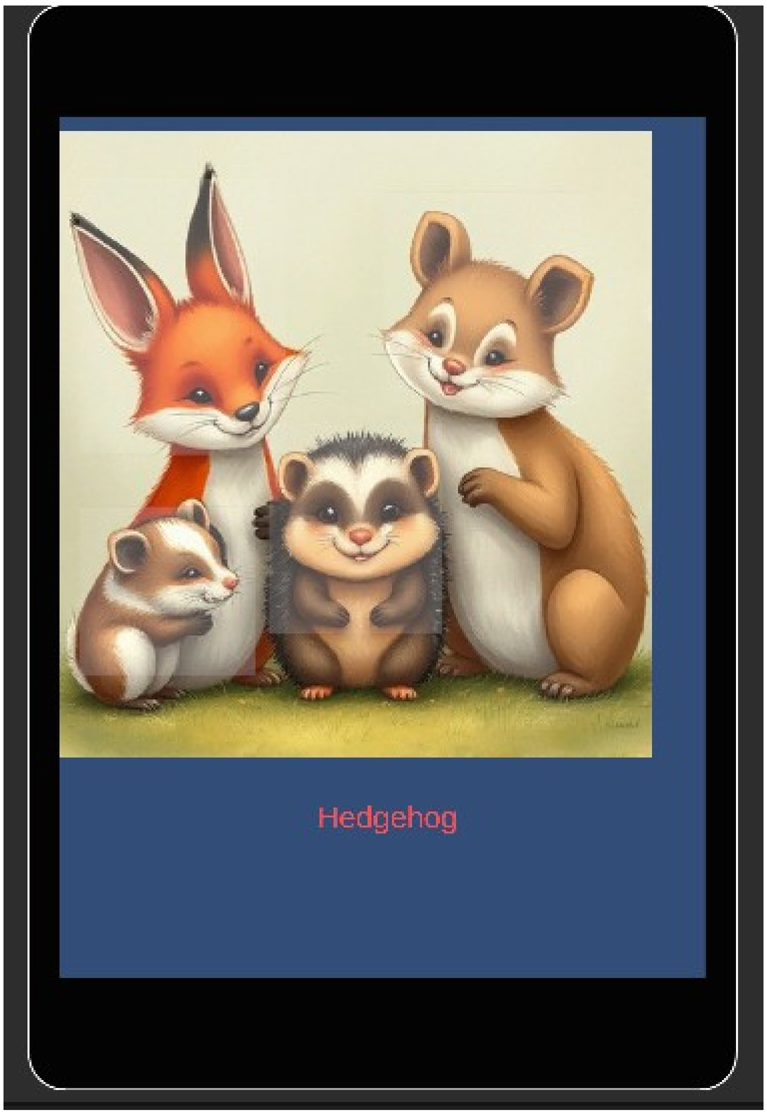
Color

RGB 0-255
R 255
G 255
B 255
A 255

Code

```
TMP_Text text1;
void Start()
{
    text1 = GameObject.Find("text1").GetComponent<TMP_Text>();
}
public void tellName()
{
    if (EventSystem.current.currentSelectedGameObject.name == "Button1")
        text1.text = "Fox";
    if (EventSystem.current.currentSelectedGameObject.name == "Button2")
        text1.text = "Hamster";
    if (EventSystem.current.currentSelectedGameObject.name == "Button3")
        text1.text = "Hedgehog";
}
```

Test run



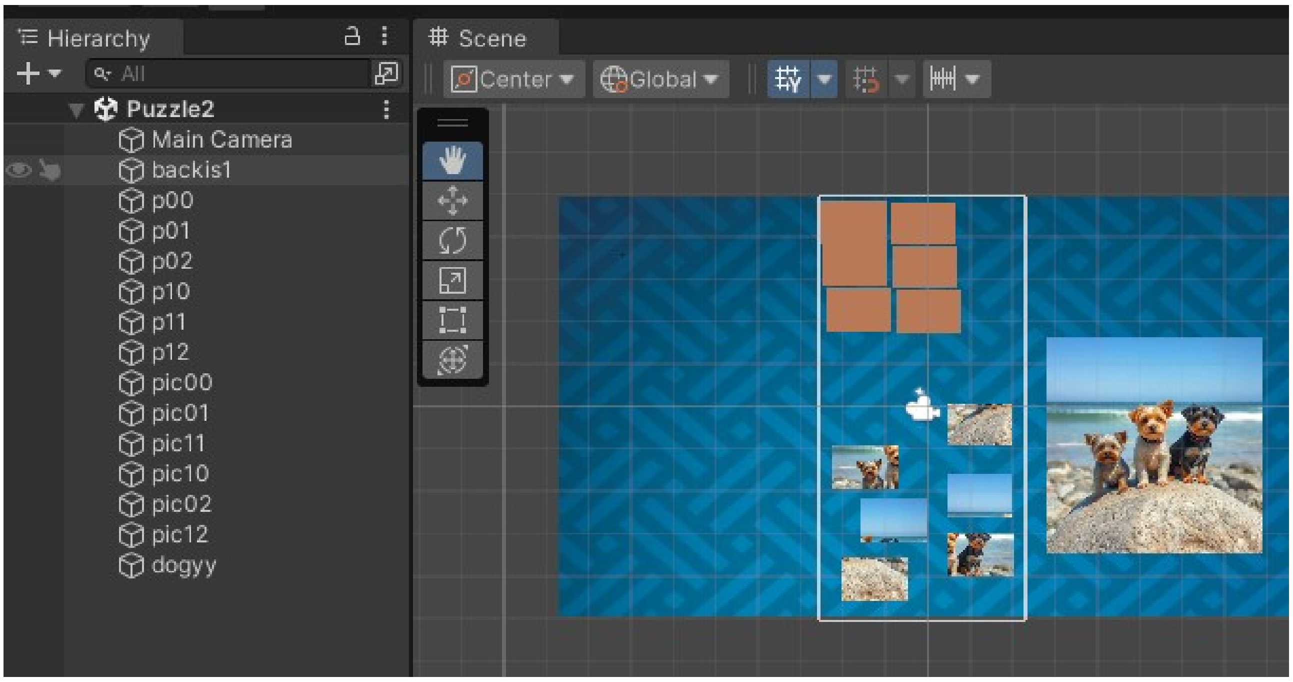
Another example image that can be used: several persons in a photo. By clicking the face you get more information about current person.

Example here

Thank you!

Images: Mobile Puzzle

Result looks like this



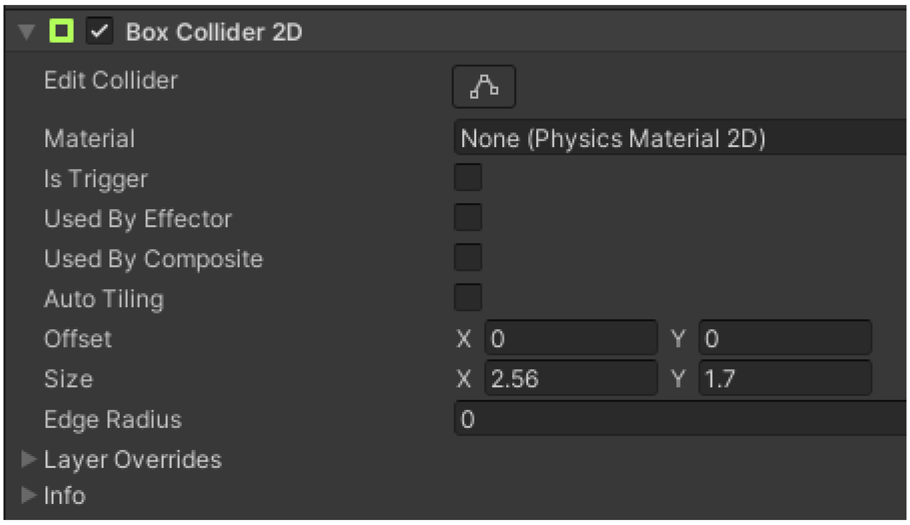
Objects p00 – p12 are places where image parts are to be moved.

Then splitted image is in objects pic00 – pic12.

There is an extra image now that shows the original image…

In the backround there is also an image, object’s name is backis1.

To pictures we add colliders:



Box Collider 2D

Edit Collider	
Material	None (Physics Material 2D)
Is Trigger	☐
Used By Effector	☐
Used By Composite	☐
Auto Tiling	☐
Offset	X 0 Y 0
Size	X 2.56 Y 1.7
Edge Radius	0
▶ Layer Overrides	
▶ Info	

Here is the beginning of the code

```
    private Transform place00;
    private Vector2 initialPosition;
    private float deltaX, deltaY;
    public static bool locked;

    void Start()
    {
        place00 = GameObject.Find("p00").transform;
        initialPosition = transform.position;
    }
```

Then here is the update() method:

```
void Update()
    {
        if (Input.touchCount > 0 && !locked)
         {
             Touch touch = Input.GetTouch(0);
```

```
            Vector2 touchPos =
Camera.main.ScreenToWorldPoint(touch.position);


        switch (touch.phase)
            {
                case TouchPhase.Began:
                    if (GetComponent<Collider2D>() ==
                    Physics2D.OverlapPoint(touchPos))
                    {
                        deltaX = touchPos.x - transform.position.x;
                        deltaY = touchPos.y - transform.position.y;
                    }
                    break;


                case TouchPhase.Moved:
                    if (GetComponent<Collider2D>() ==
                        Physics2D.OverlapPoint(touchPos))
                        transform.position = new Vector2(touchPos.x -
                       deltaX, touchPos.y - deltaY);
                    break;
                case TouchPhase.Ended:
                    if (Mathf.Abs(transform.position.x -
                    place00.position.x) <= 1.5f &&
                        Mathf.Abs(transform.position.y -
                      place00.position.y) <= 1.5f)
                    {
                        transform.position = new
                     Vector2(place00.position.x, place00.position.y);
                        locked = true;
                    }
                    else
                    {
                        transform.position = new
                     Vector2(initialPosition.x, initialPosition.y);
                    }
                    break;
                }
            }
```

To the codes I got help from Internet discussion boards and Alexander Zotov’s videos!

Thank you!

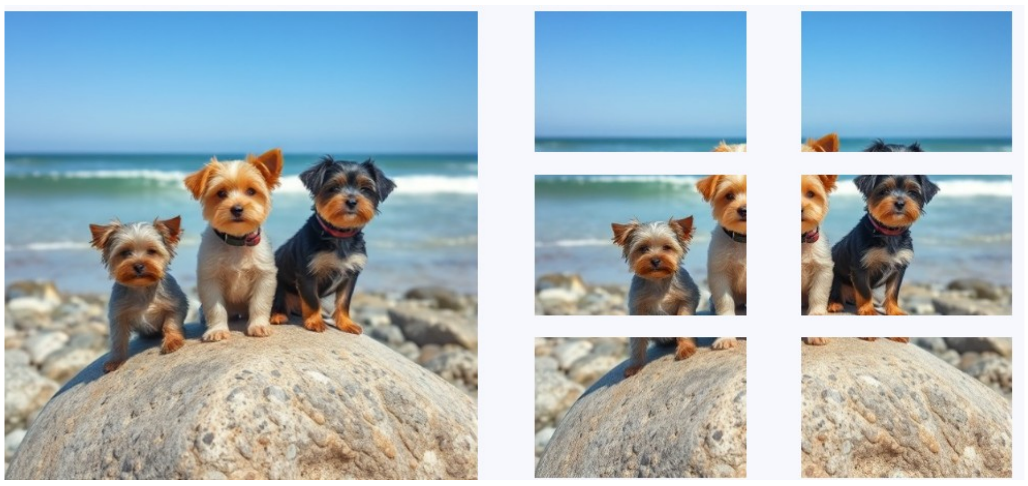
Image has been generated using AI: from text to image.

Place is here:

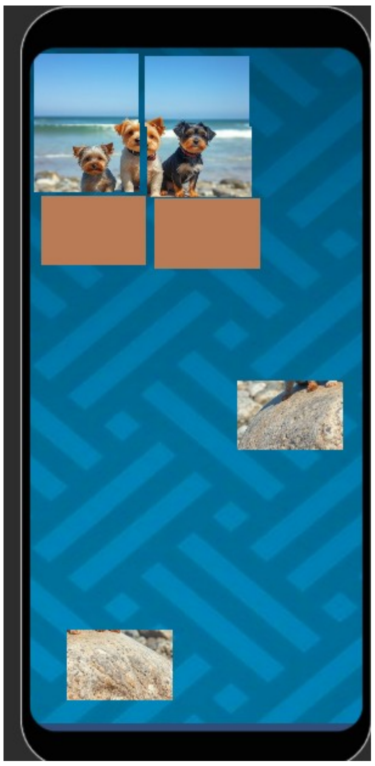
https://deepai.org/machine-learning-model/text2img

As Image splitter has this tool been used:

https://splitter.imageonline.co/#google_vignette



Test run!



And so on!

Add more features!

Character movements

Goal
Player moves around and can also jump.

Can catch items and get points.

There is a route added to a background image.

So we need to use:
animations

images

collisions

sounds

taking time

collecting points

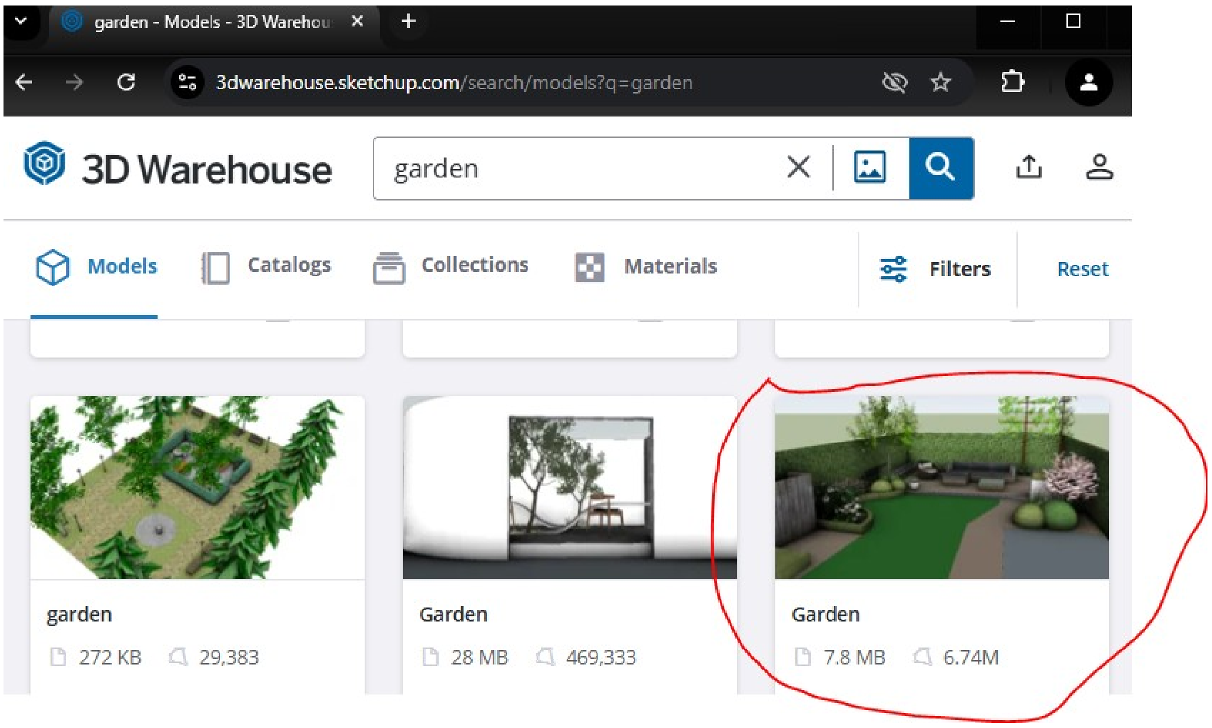
3 scenes

transporting info between scenes

Then we publish the game...

Background image

You can draw it, search from Internet or use directly for example this place

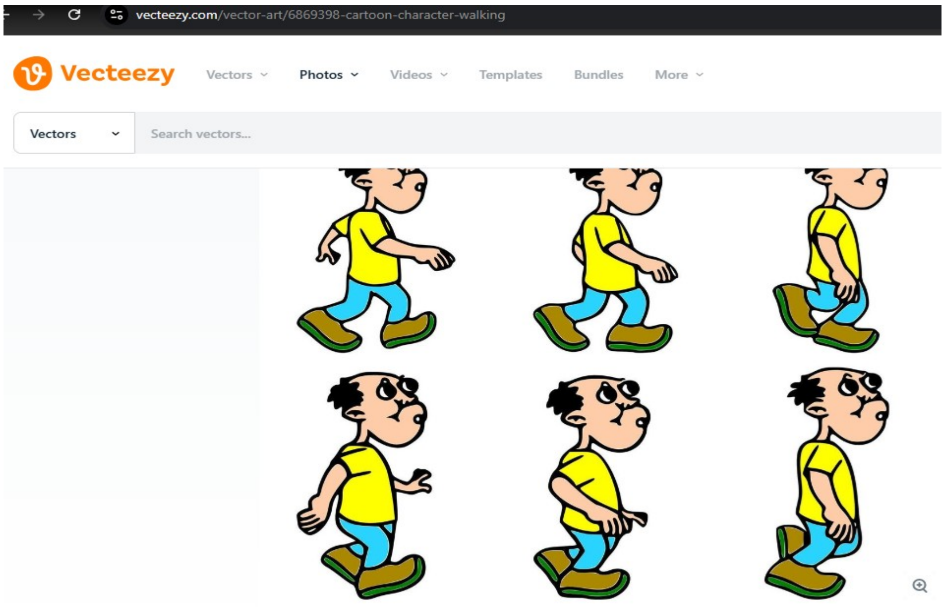


You can also take a photo and convert it to image using pixlr.

Animation

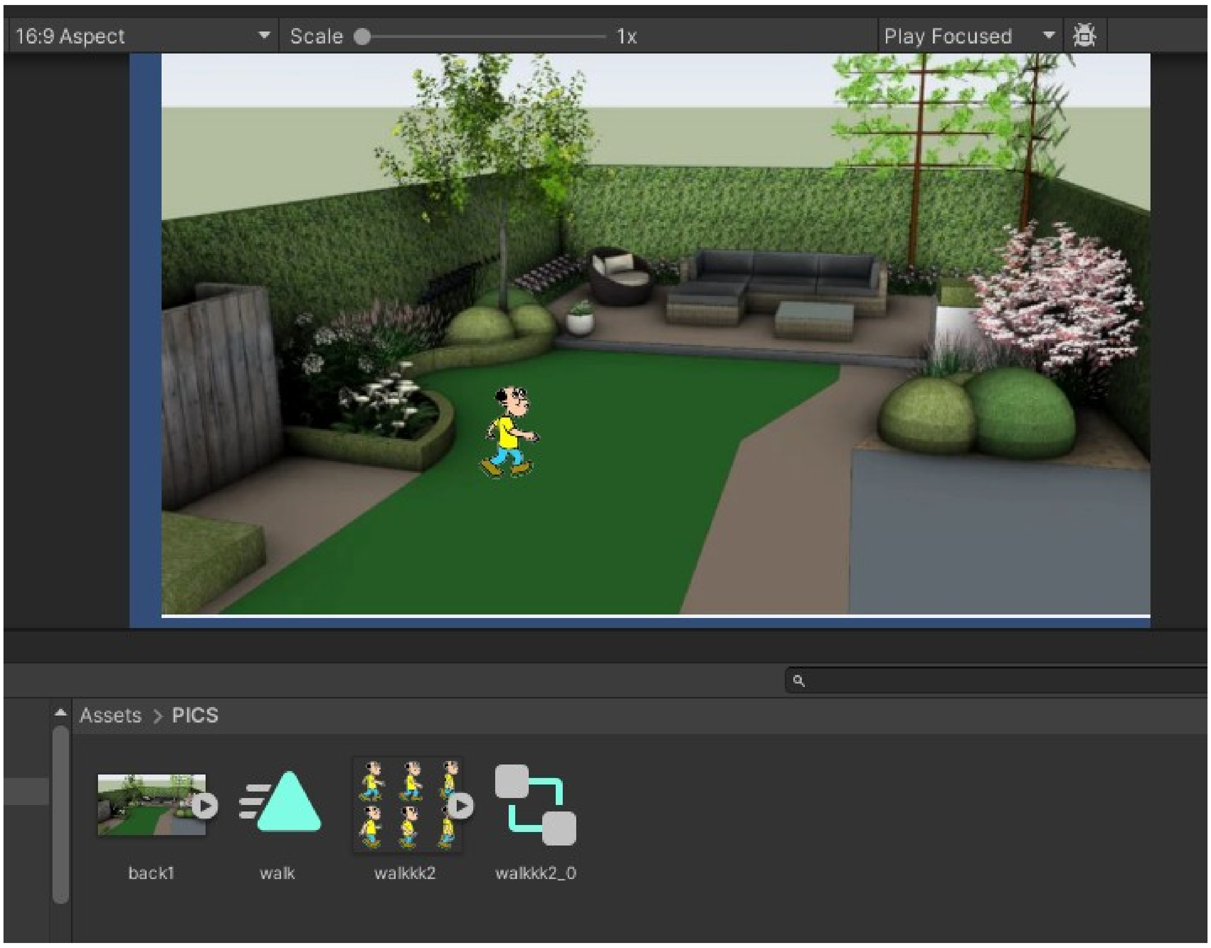
Searching using keywords ”human walking animation free”..

Example



Take it to the project but change first the backgound color transparent...

Here they are



Version 1

Dude walks, jumps and stops...

Add rigidbody2d and collider to dude.

Add square sprite to work as a floor and add collider to the square.

Create this code and add it to dude.

Animation parameters are not used in this version. Later we have to get a better dude!

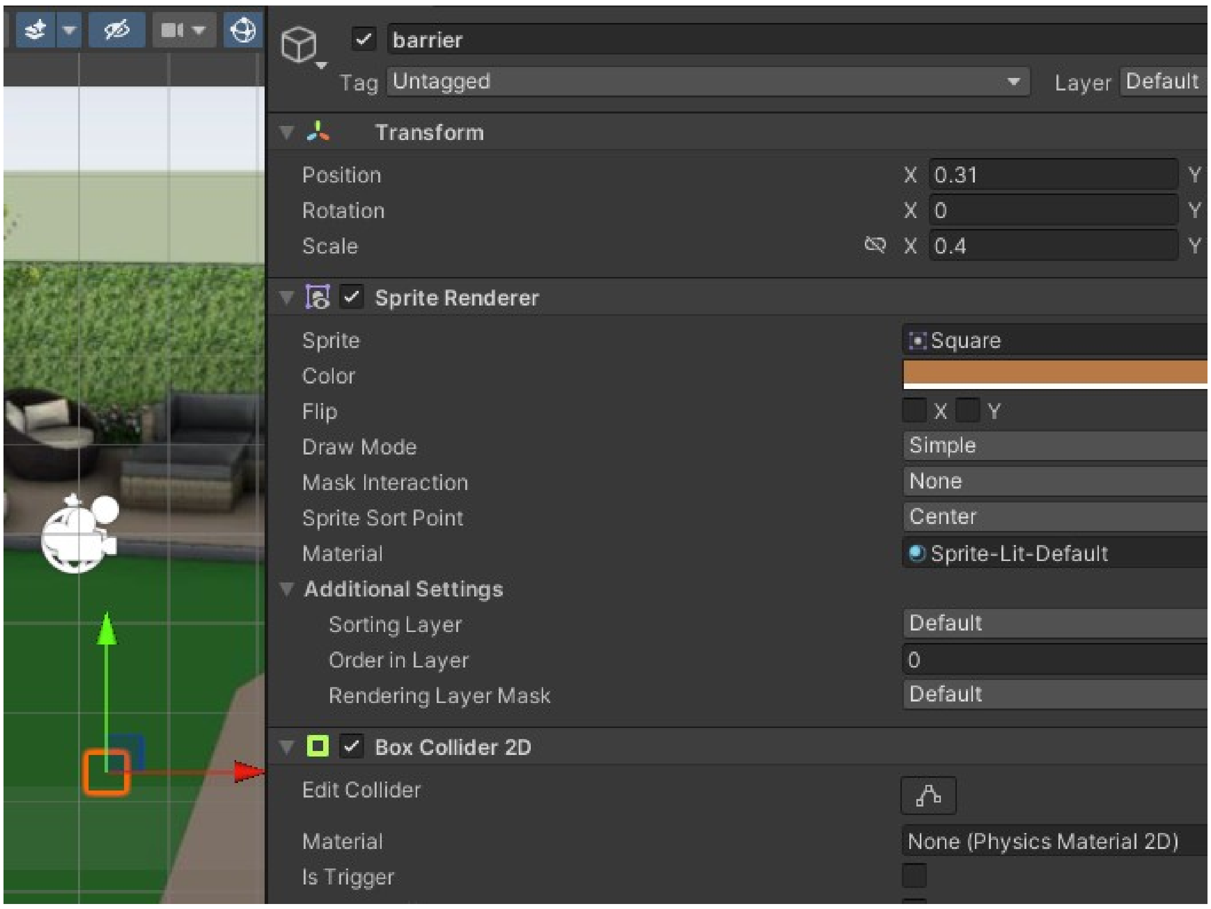
Test

```
GameObject anim1;
    float xx = -5;
    int state = 0;
    void Start()
    {
        anim1 = GameObject.Find("dude_idle_2");
        this.GetComponent<Animator>().SetInteger("mode", 0);
        state = GetComponent<Animator>().GetInteger("mode");
        Debug.Log("" + this.GetComponent<Animator>().GetInteger("mode"));
       // anim1.GetComponent<Transform>().position = new Vector3(-5, 3, 0);
       // xx = anim1.GetComponent<Transform>().position.x;
    }

    private float power = 400f;
    float stepX = 0.01f;
    int dir = 0;

    void Update()
    {
        if (dir == 1)
            this.GetComponent<Transform>().Translate(1f * Time.deltaTime, 0f, 
0f);

        if (Input.GetMouseButtonDown(0))
        {
            this.GetComponent<Animator>().SetInteger("mode", 1);
            this.GetComponent<Transform>().Rotate(0f, -180f, 0f);
            dir = 1;
        }

        if (Input.GetMouseButtonDown(1))
        {
            dir = 1;
            this.GetComponent<Transform>().Rotate(0f, 180f, 0f);
        }


        if (Input.GetKeyDown(KeyCode.Space))
        {
            Jump();
        }

        if (Input.GetKeyDown(KeyCode.S))
        {
           Stand();
        }

    }
    public void Stand()
    {
        this.GetComponent<Animator>().SetInteger("mode", 0);
        this.GetComponent<Transform>().Translate(0f, 0f, 0f);
        dir = 0;
    }
    int n;
    public void Jump()
    {
        n = 0;
        this.GetComponent<Rigidbody2D>().AddForce(new Vector2(0f, 1f) * power * 
1.05f);
    }
```

Version 2

Dude jumps over some hinder.

Add barrier (now square) and boxcollider.



barrier
Tag Untagged   Layer Default

Transform
Position X 0.31 Y
Rotation X 0 Y
Scale X 0.4 Y

Sprite Renderer
Sprite: Square
Color
Flip: X Y
Draw Mode: Simple
Mask Interaction: None
Sprite Sort Point: Center
Material: Sprite-Lit-Default
Additional Settings
Sorting Layer: Default
Order in Layer: 0
Rendering Layer Mask: Default

Box Collider 2D
Edit Collider
Material: None (Physics Material 2D)
Is Trigger

Test.

Now we check collision: if dude can not go over the barrier, we get minus points.

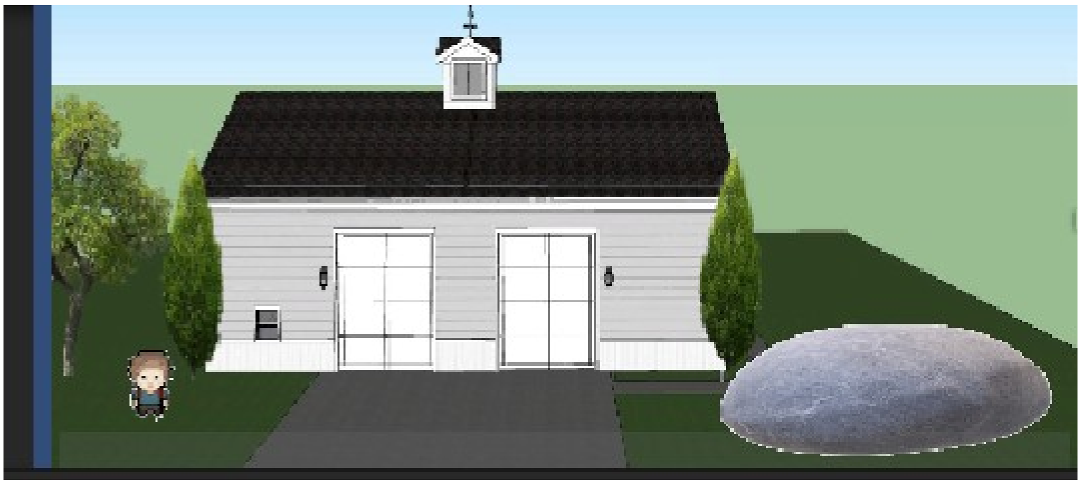
Here is the code:

```
int points = 0;
    private void OnCollisionEnter2D(Collision2D collision)
    {

      if (collision.collider.name  == "barrier")
        {
            points--;
            Debug.Log("" + points);
        }
    }
```

Version 3
Some objects are flying and dude has to catch them.

Now we have a balloon that moves upwards every time dude touches it.

SO we add a balloon and collision detector.



Code

```
GameObject anim1;
    float xx = -5;
    int state = 0;
    void Start()
    {
        anim1 = GameObject.Find("dude_idle_2");
        this.GetComponent<Animator>().SetInteger("mode", 0);
        state = GetComponent<Animator>().GetInteger("mode");
        Debug.Log("" + this.GetComponent<Animator>().GetInteger("mode"));
       // anim1.GetComponent<Transform>().position = new Vector3(-5, 3, 0);
       // xx = anim1.GetComponent<Transform>().position.x;
    }

    private float power = 400f;
    float stepX = 0.01f;
    int dir = 0;

    void Update()
    {
        if (dir == 1)
            this.GetComponent<Transform>().Translate(1f * Time.deltaTime, 0f,
0f);

        if (Input.GetMouseButtonDown(0))
        {
            this.GetComponent<Animator>().SetInteger("mode", 1);
            this.GetComponent<Transform>().Rotate(0f, -180f, 0f);
            dir = 1;
        }

        if (Input.GetMouseButtonDown(1))
        {
            dir = 1;
            this.GetComponent<Transform>().Rotate(0f, 180f, 0f);
        }


        if (Input.GetKeyDown(KeyCode.Space))
        {
            Jump();
        }

        if (Input.GetKeyDown(KeyCode.S))
        {
           Stand();
        }

    }
    public void Stand()
    {
        this.GetComponent<Animator>().SetInteger("mode", 0);
        this.GetComponent<Transform>().Translate(0f, 0f, 0f);
        dir = 0;
    }
    int n;
    public void Jump()
    {
        n = 0;
        this.GetComponent<Rigidbody2D>().AddForce(new Vector2(0f, 1f) * power *
1.05f);
    }
```

Effects

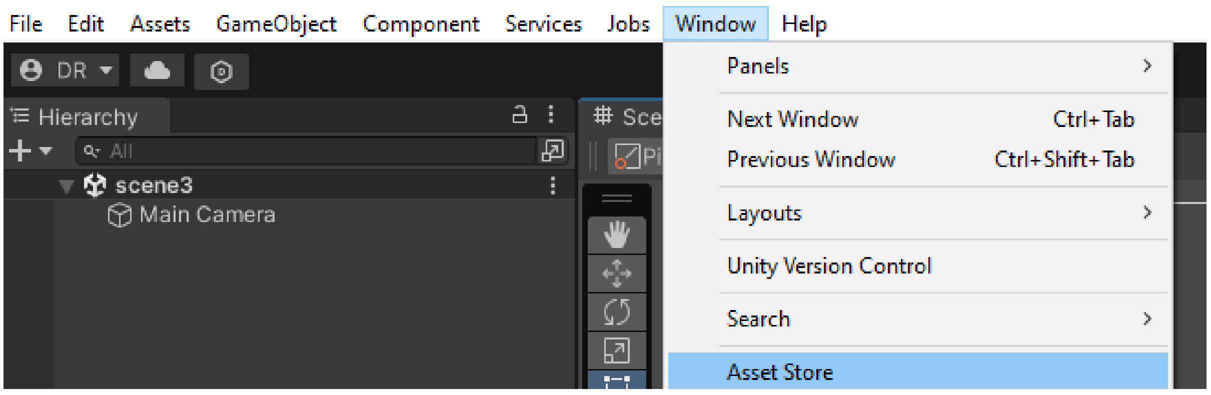
Fire, rain...

Assets Store!

Let's take some items from Assets: e.g trees

Log in first – you can sign in with Google username, too...

Open Window Asset Store:

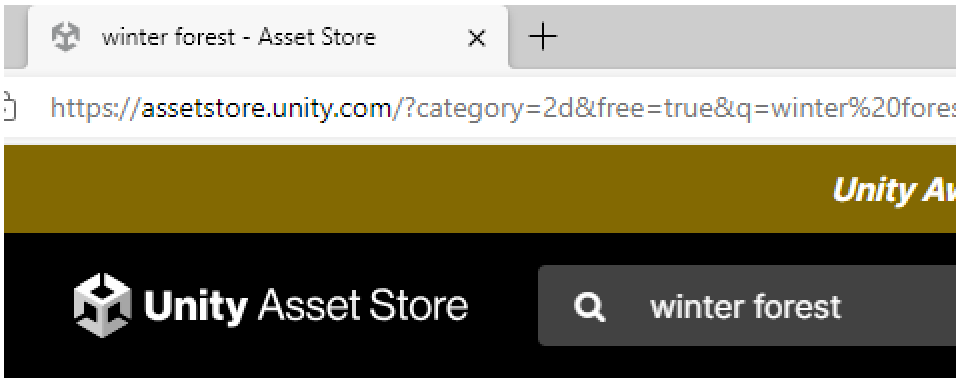


Let's try to take wood items...

Check that contents is free...

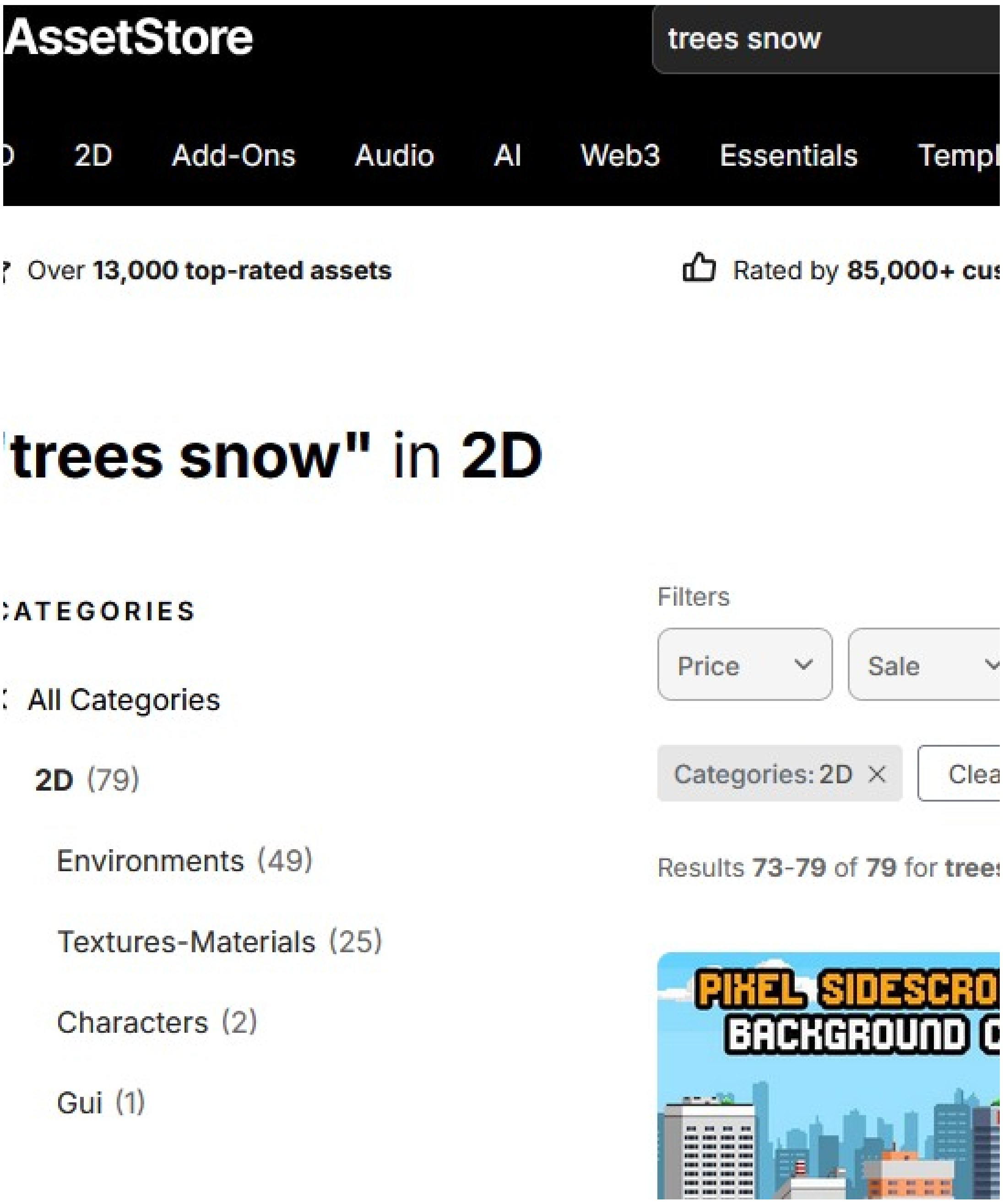
Then download items and import them to you project...

Example 1


winter forest - Asset Store

https://assetstore.unity.com/?category=2d&free=true&q=winter%20fores

New: example 2


AssetStore

trees snow

D 2D Add-Ons Audio AI Web3 Essentials Templ

Over **13,000 top-rated assets**

Rated by **85,000+ cus**

trees snow" in 2D

CATEGORIES

All Categories

2D (79)

Environments (49)

Textures-Materials (25)

Characters (2)

Gui (1)

Filters

Price

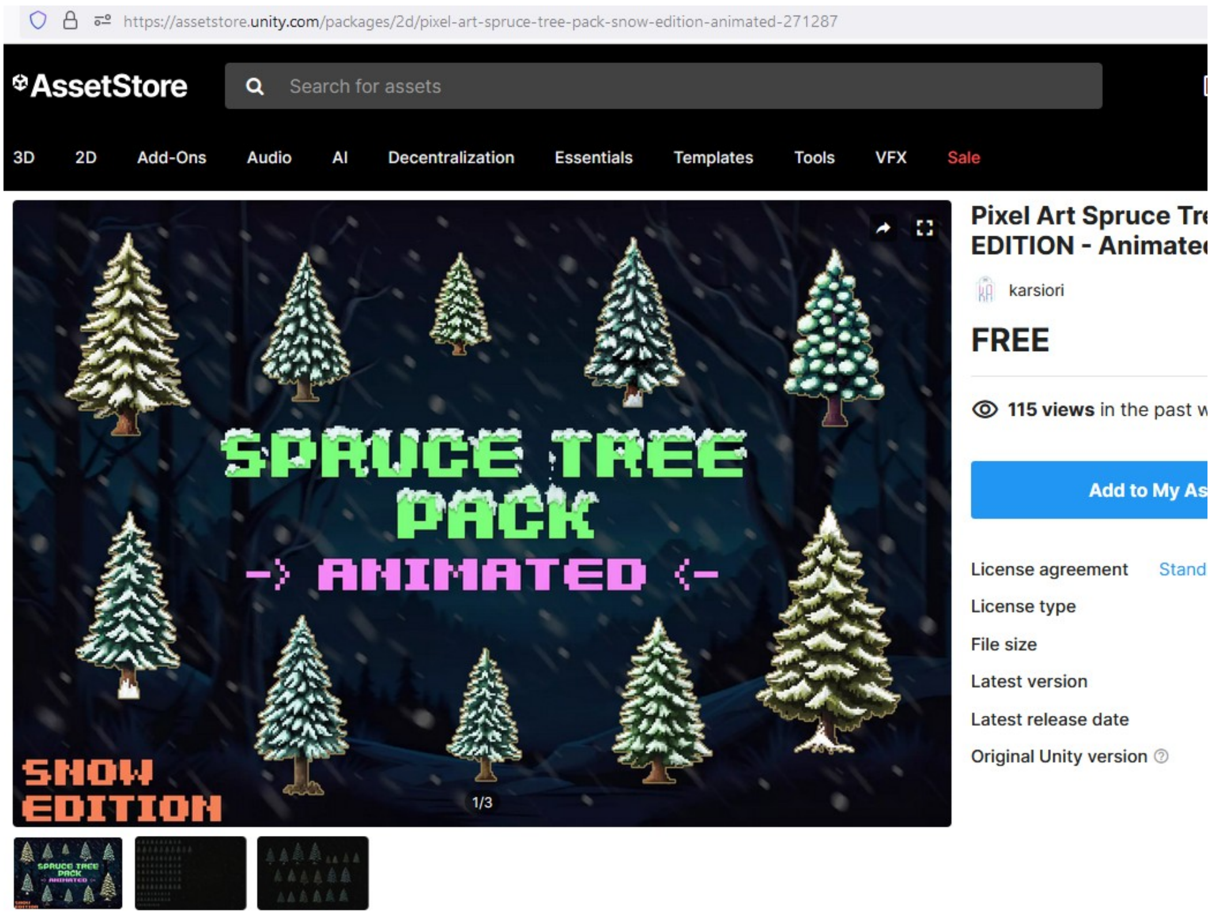
Sale

Categories: 2D

Clea

Results **73-79** of **79** for **trees**

One sample: trees



https://assetstore.unity.com/packages/2d/pixel-art-spruce-tree-pack-snow-edition-animated-271287

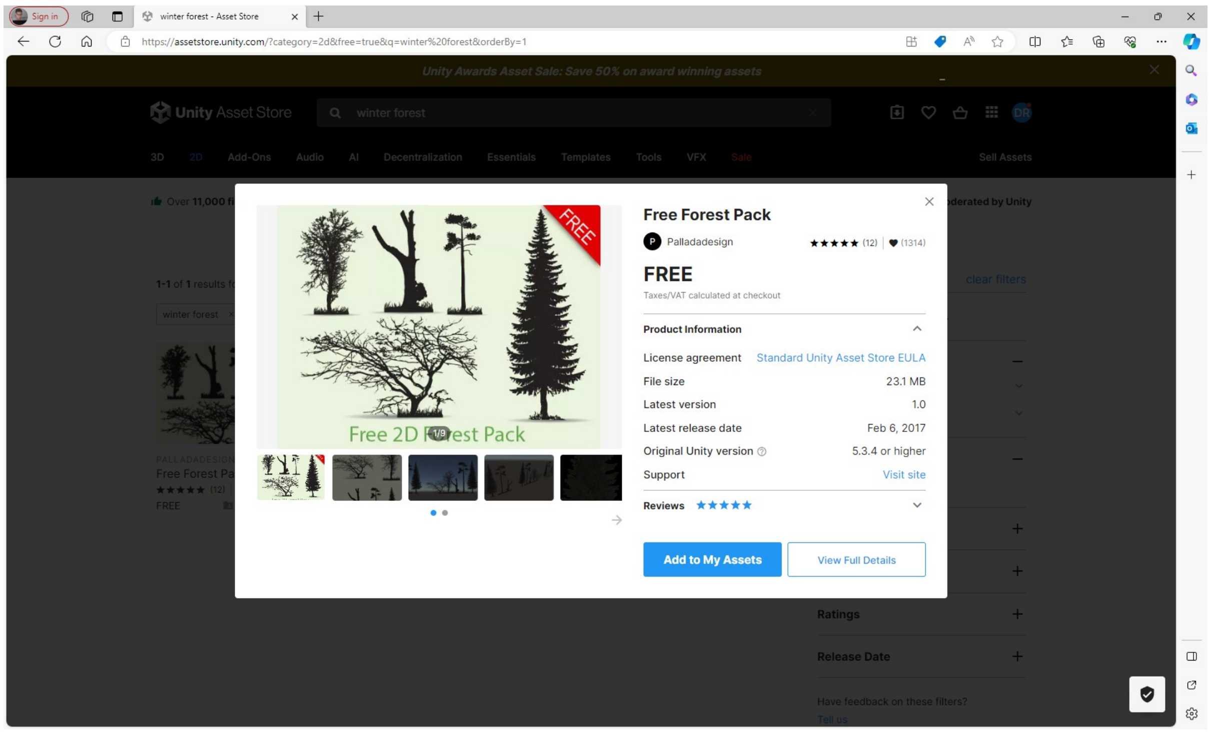
Pixel Art Spruce Tre
EDITION - Animatec

karsiori

FREE

115 views in the past w

License agreement Stand
License type
File size
Latest version
Latest release date
Original Unity version


winter forest - Asset Store
https://assetstore.unity.com/?category=2d&free=true&q=winter%20forest&orderBy=1
Unity Awards Asset Sale: Save 50% on award winning assets
Unity Asset Store
winter forest
3D 2D Add-Ons Audio AI Decentralization Essentials Templates Tools VFX Sale
Sell Assets
Ratings
Release Date
Have feedback on these filters?

Free Forest Pack

Palladadesign ★★★★★ (12) | ♥ (1314)

FREE

Taxes/VAT calculated at checkout

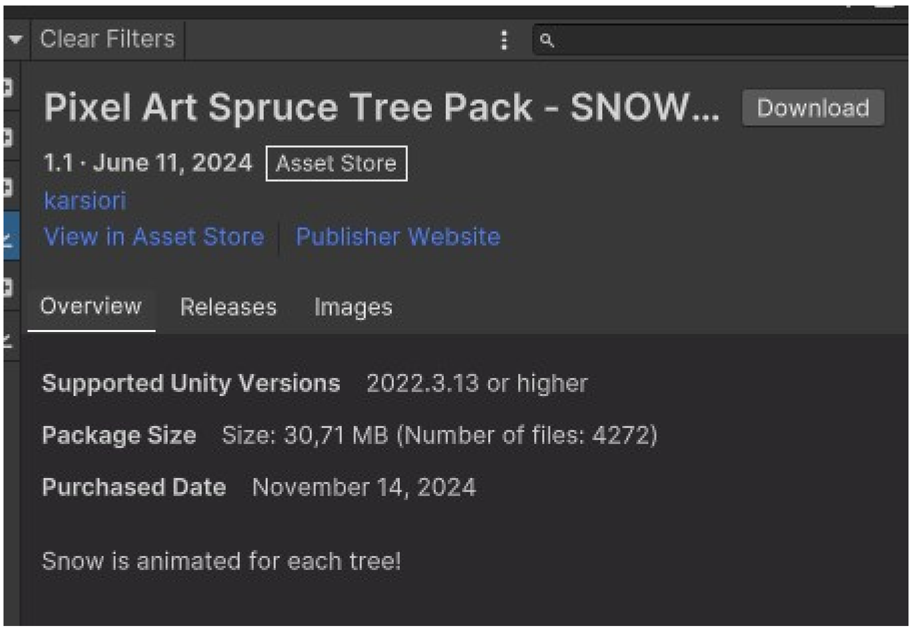
Product Information

License agreement	Standard Unity Asset Store EULA
File size	23.1 MB
Latest version	1.0
Latest release date	Feb 6, 2017
Original Unity version	5.3.4 or higher
Support	Visit site

Reviews ★★★★★

Add to My Assets | View Full Details

AND here is one free item set, too



Pixel Art Spruce Tree Pack - SNOW...

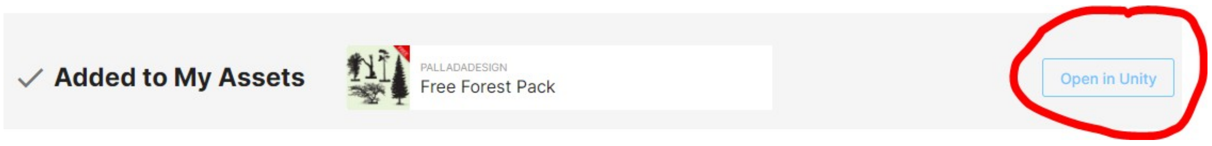
1.1 · June 11, 2024

karsiori

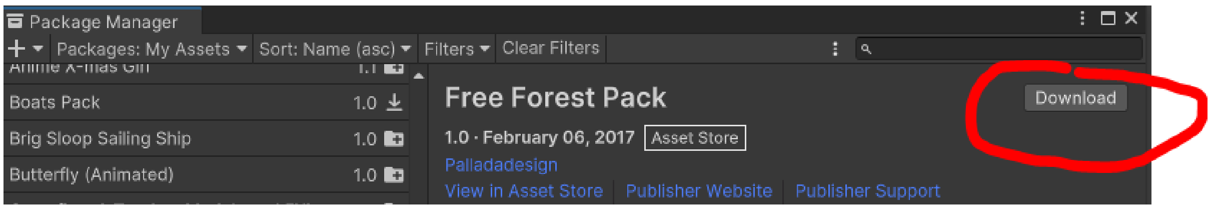
Supported Unity Versions	2022.3.13 or higher
Package Size	Size: 30,71 MB (Number of files: 4272)
Purchased Date	November 14, 2024

Snow is animated for each tree!

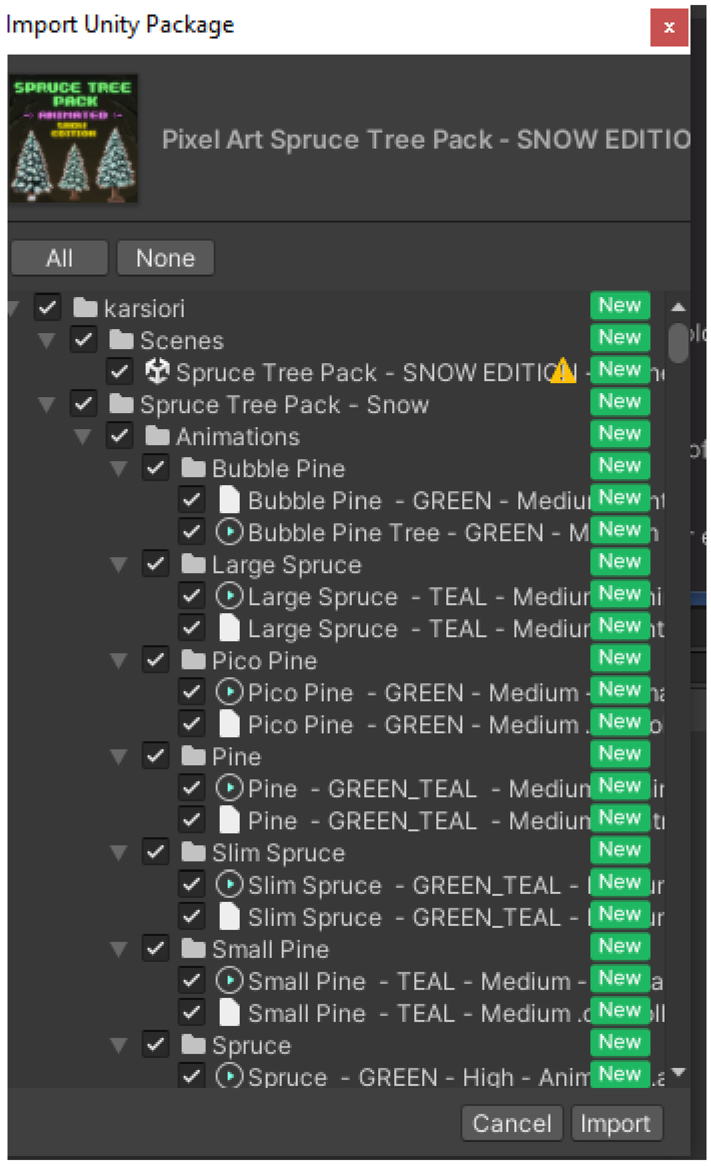
Take items to your project


Added to My Assets
PALLADADESIGN
Free Forest Pack
Open in Unity

and

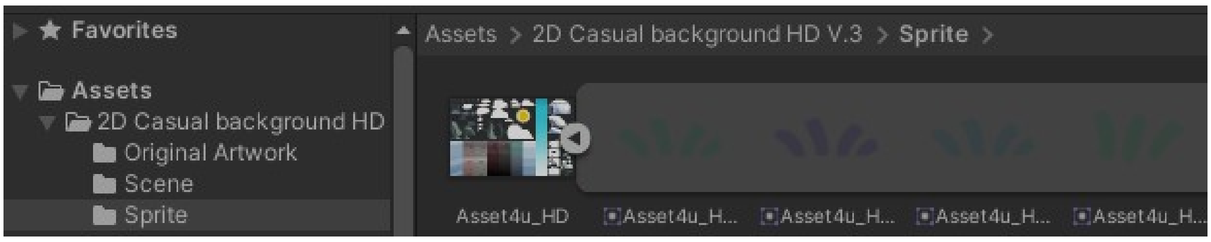

Package Manager
Packages: My Assets
Sort: Name (asc)
Filters
Clear Filters
Anime X-mas Girl 1.1
Boats Pack 1.0
Brig Sloop Sailing Ship 1.0
Butterfly (Animated) 1.0
Free Forest Pack
1.0 · February 06, 2017
Asset Store
Palladadesign
View in Asset Store
Publisher Website
Publisher Support
Download

AND import



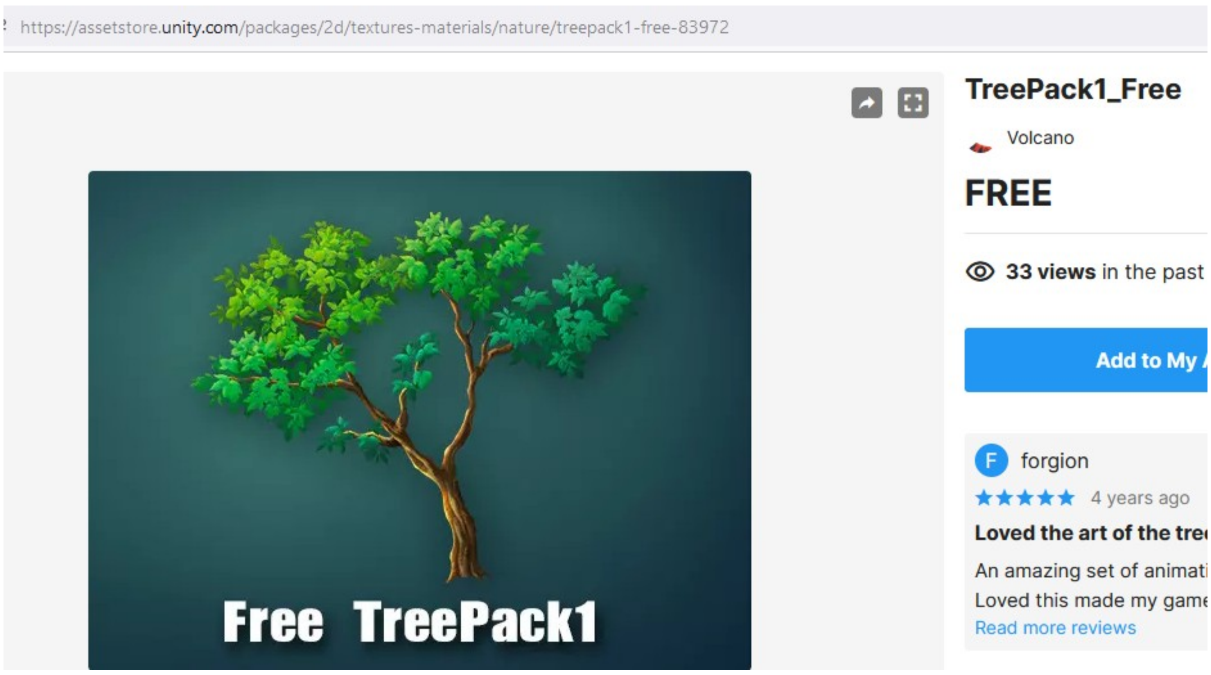
```
Import Unity Package
Pixel Art Spruce Tree Pack - SNOW EDITIO
All  None
karsiori                                          New
  Scenes                                          New
    Spruce Tree Pack - SNOW EDITIO                New
  Spruce Tree Pack - Snow                         New
    Animations                                    New
      Bubble Pine                                 New
        Bubble Pine  - GREEN - Mediu              New
        Bubble Pine Tree - GREEN - M              New
      Large Spruce                                New
        Large Spruce  - TEAL - Mediur             New
        Large Spruce  - TEAL - Mediur             New
      Pico Pine                                   New
        Pico Pine  - GREEN - Medium -             New
        Pico Pine  - GREEN - Medium .             New
      Pine                                        New
        Pine  - GREEN_TEAL  - Mediun              New
        Pine  - GREEN_TEAL  - Mediun              New
      Slim Spruce                                 New
        Slim Spruce  - GREEN_TEAL -               New
        Slim Spruce  - GREEN_TEAL -               New
      Small Pine                                  New
        Small Pine  - TEAL - Medium -             New
        Small Pine  - TEAL - Medium .             New
      Spruce                                      New
        Spruce  - GREEN - High - Anim             New
Cancel  Import
```

It was not good so I took another ;)



Of course we could have been able to use a background image but now it is easier to edit the forest...

For summer images there are e.g these free sets



https://assetstore.unity.com/packages/2d/textures-materials/nature/treepack1-free-83972

TreePack1_Free

Volcano

FREE

33 views in the past

Add to My

forgion

★★★★★ 4 years ago

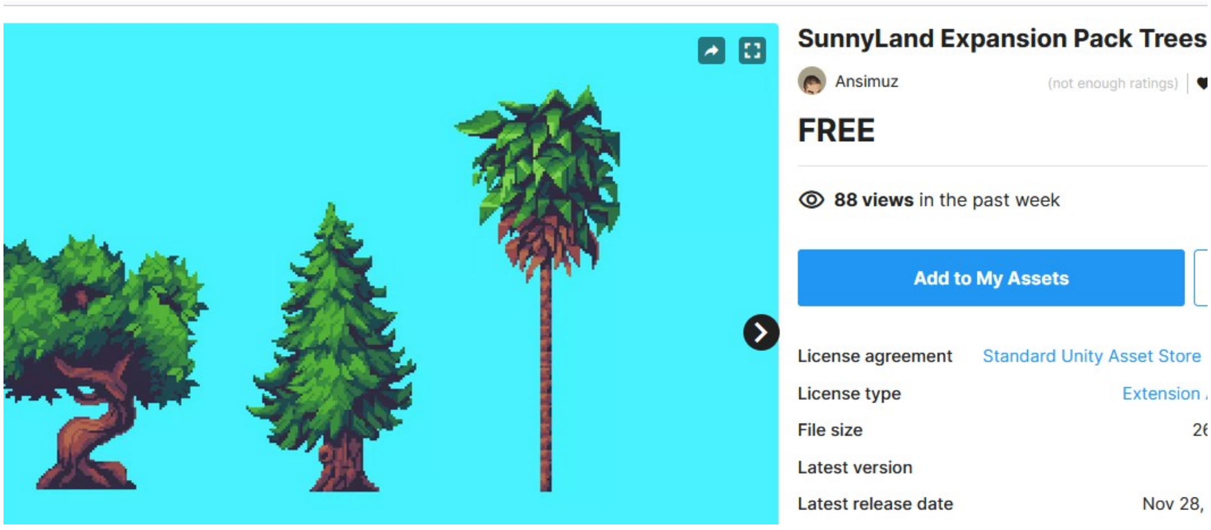
Loved the art of the tre

An amazing set of animati

Loved this made my game

Read more reviews

AND



SunnyLand Expansion Pack Trees

Ansimuz (not enough ratings)

FREE

88 views in the past week

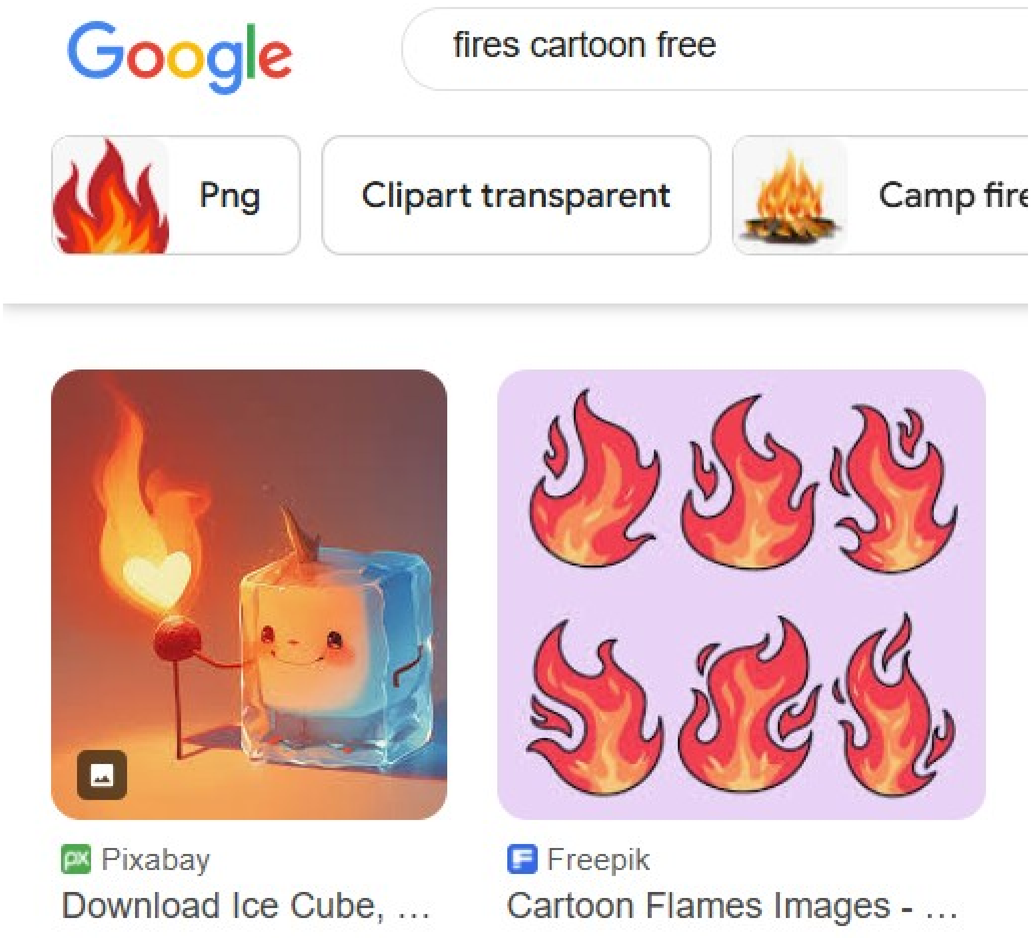
Add to My Assets

License agreement	Standard Unity Asset Store
License type	Extension
File size	26
Latest version	
Latest release date	Nov 28,

Fire

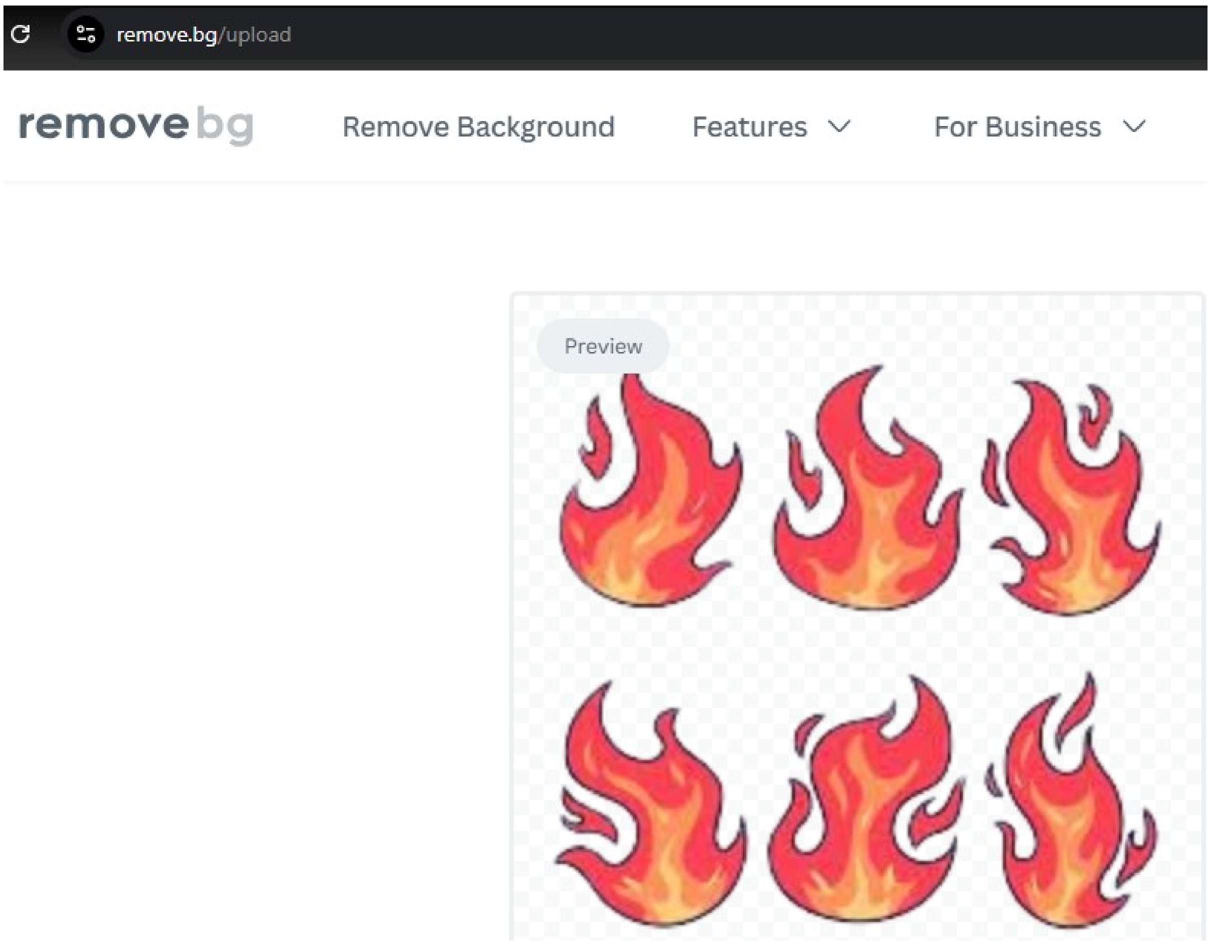
Now we need fire!!!

From Internet we can found e.g this set

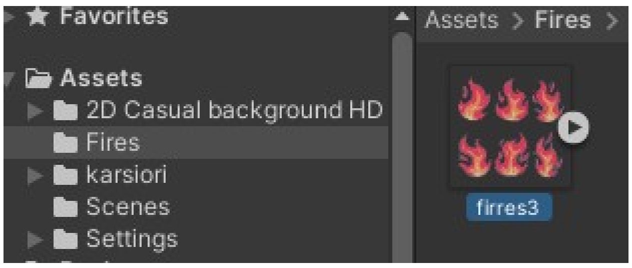


We take the image and remove background using some online tool.

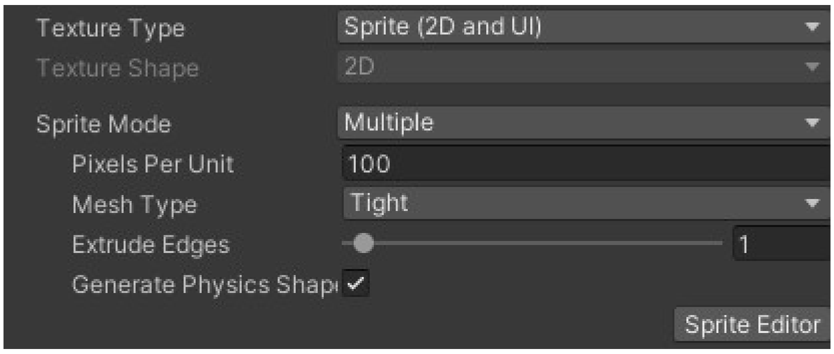
Here is one online tool in use:



Now fire is in our project.

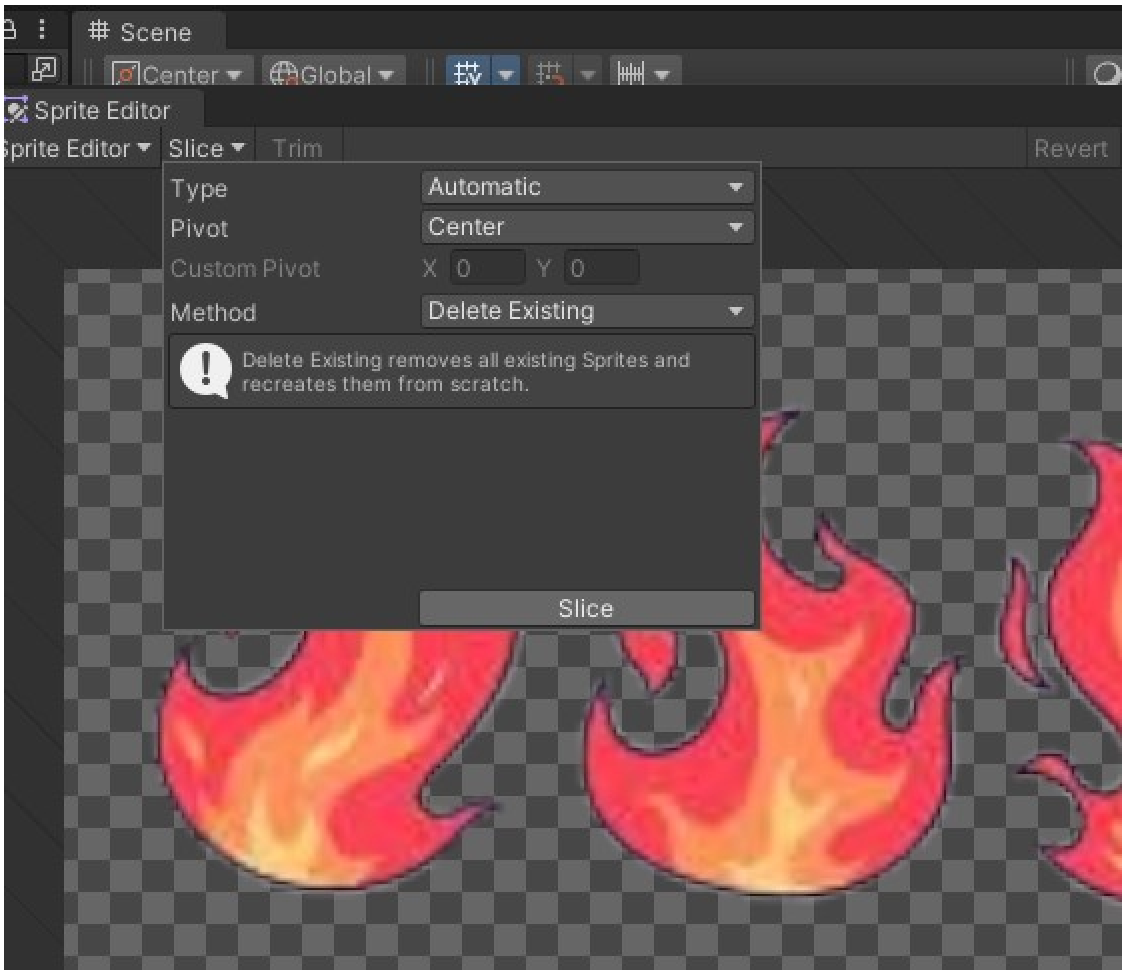


We make an animation



Texture Type	Sprite (2D and UI)
Texture Shape	2D
Sprite Mode	Multiple
Pixels Per Unit	100
Mesh Type	Tight
Extrude Edges	1
Generate Physics Shap	✓

And

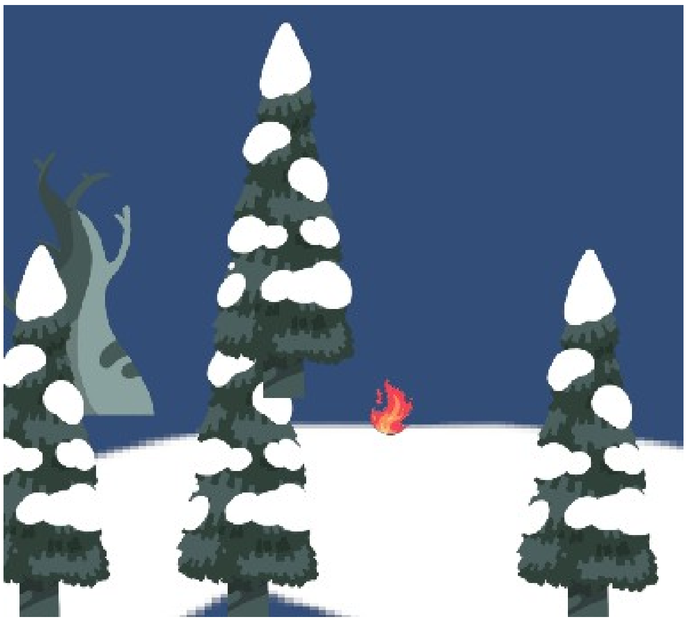


Type: Automatic
Pivot: Center
Custom Pivot: X 0 Y 0
Method: Delete Existing

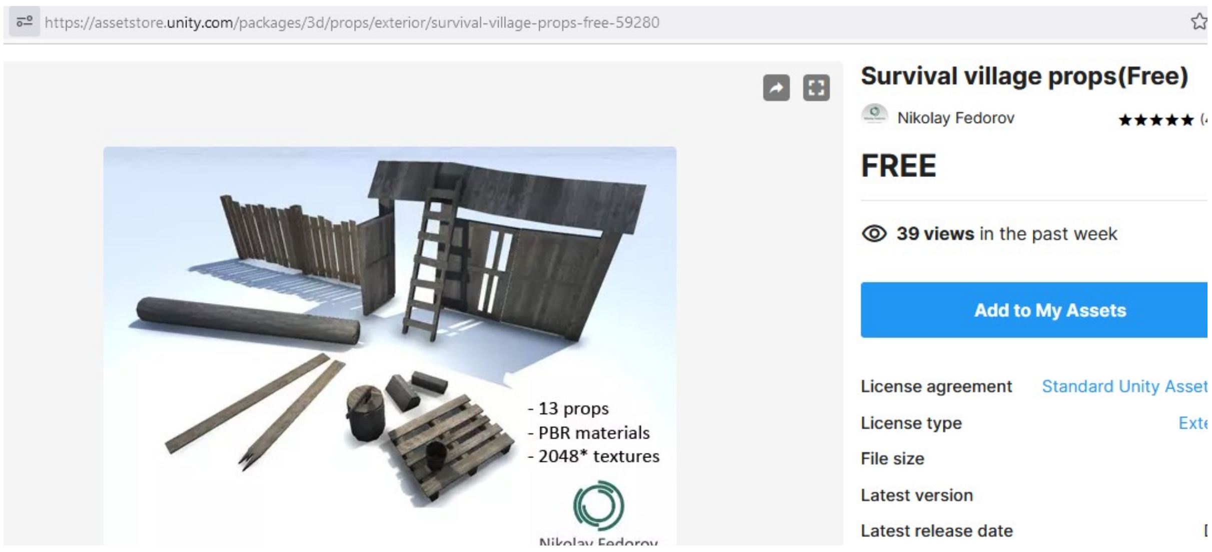
Delete Existing removes all existing Sprites and recreates them from scratch.

And there it is

You can adjust size, place, speed...



Let's take something to burn!



https://assetstore.unity.com/packages/3d/props/exterior/survival-village-props-free-59280

Survival village props(Free)

Nikolay Fedorov ★★★★★ (

FREE

39 views in the past week

Add to My Assets

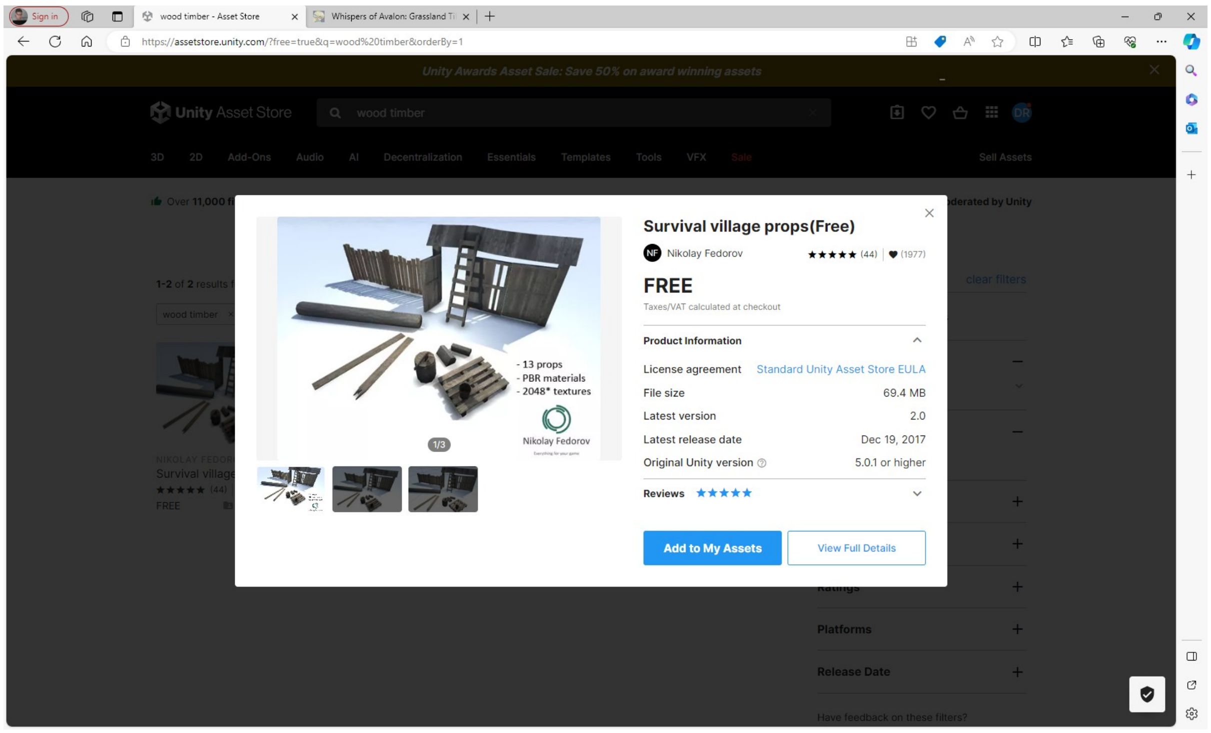
License agreement Standard Unity Asset

License type Ext

File size

Latest version

Latest release date


https://assetstore.unity.com/?free=true&q=wood%20timber&orderBy=1

Unity Awards Asset Sale: Save 50% on award winning assets

Survival village props(Free)

Nikolay Fedorov ★★★★★ (44) | ♥ (1977)

FREE

Taxes/VAT calculated at checkout

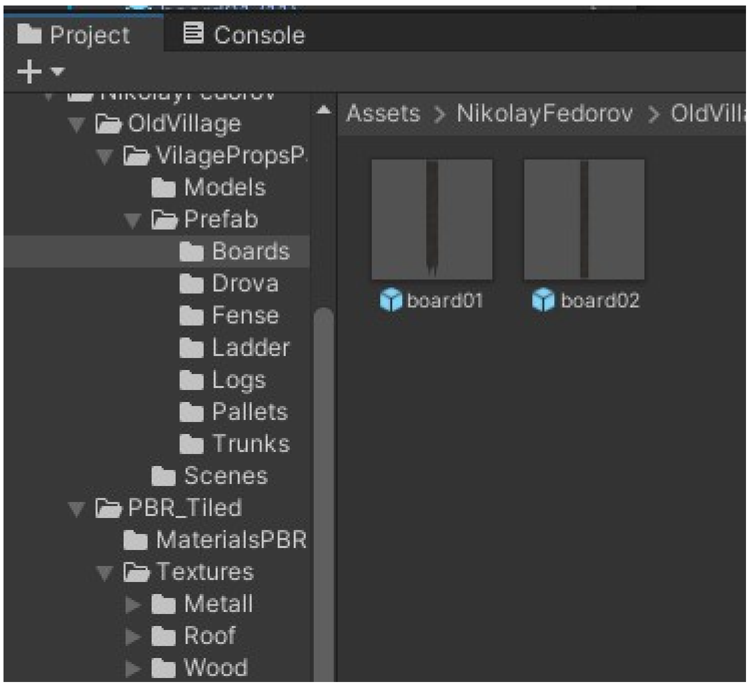
Product Information

License agreement	Standard Unity Asset Store EULA
File size	69.4 MB
Latest version	2.0
Latest release date	Dec 19, 2017
Original Unity version	5.0.1 or higher

Reviews ★★★★★

Add to My Assets | View Full Details

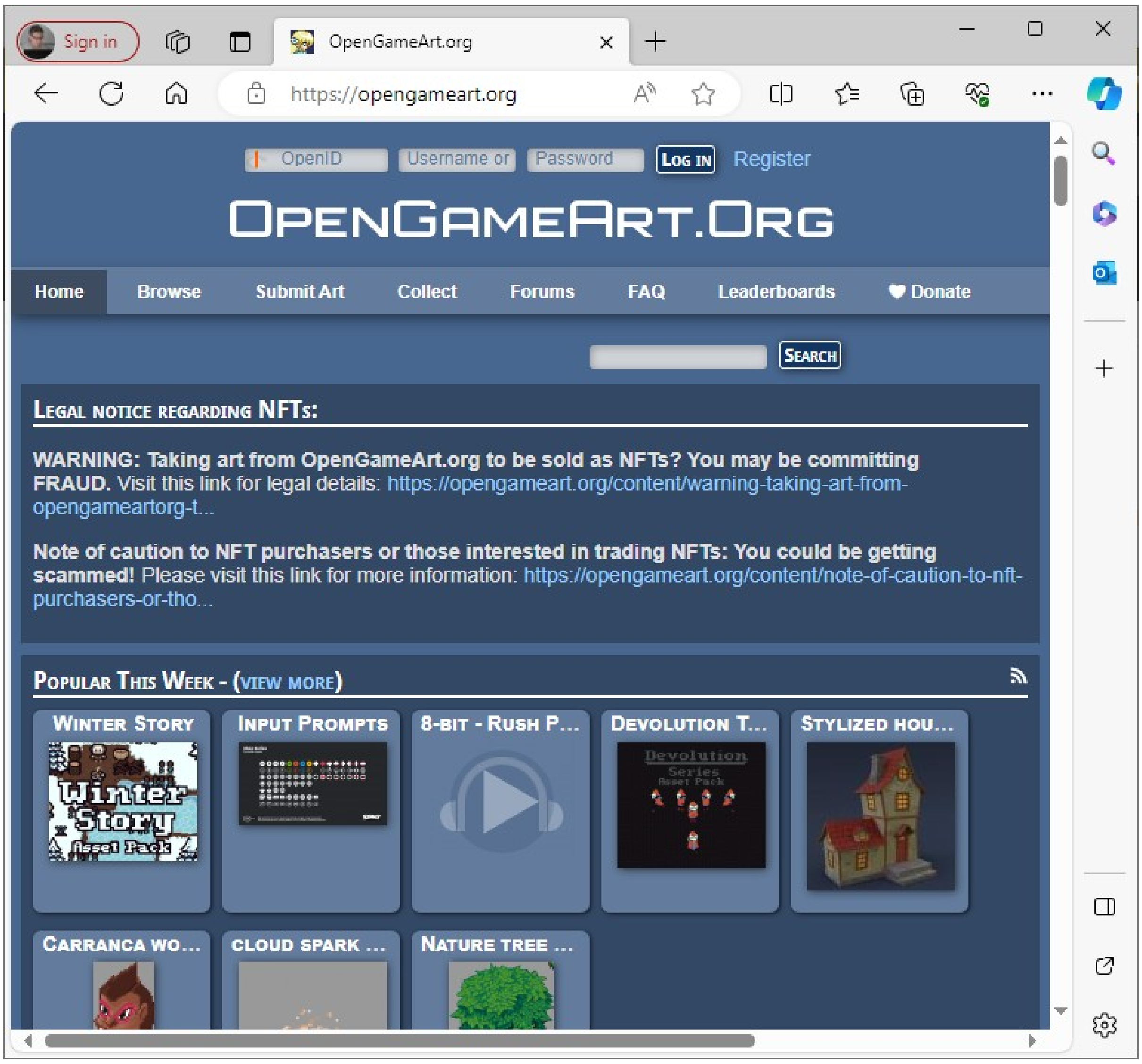
A lot of items...



Example

AND here, too

Check places where you can get items, eg. Here:



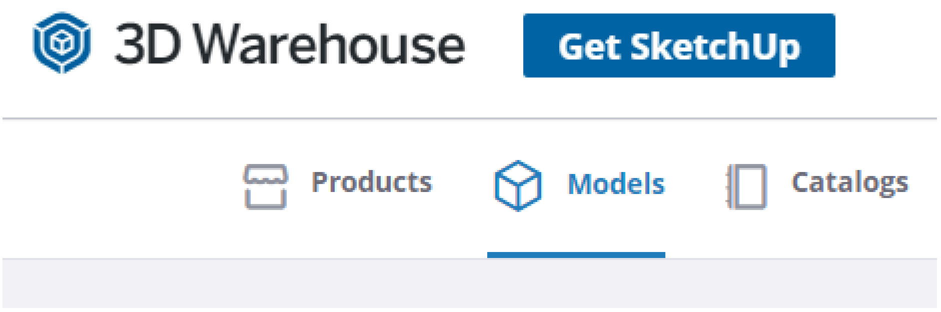
Legal notice regarding NFTs:

WARNING: Taking art from OpenGameArt.org to be sold as NFTs? You may be committing FRAUD. Visit this link for legal details: https://opengameart.org/content/warning-taking-art-from-opengameartorg-t...

Note of caution to NFT purchasers or those interested in trading NFTs: You could be getting scammed! Please visit this link for more information: https://opengameart.org/content/note-of-caution-to-nft-purchasers-or-tho...

Popular This Week - (view more)

Also 3D items can be found



Rain

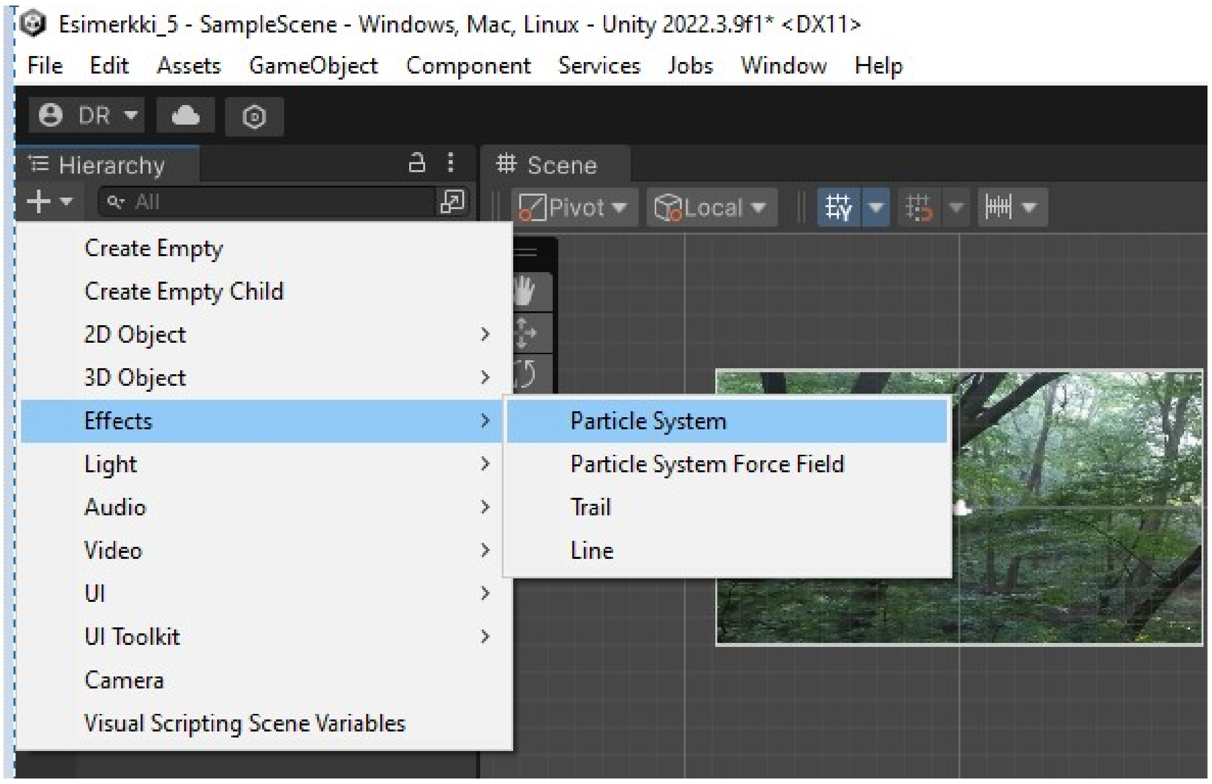
More effects: rain!!!

Take some nature image to the background.

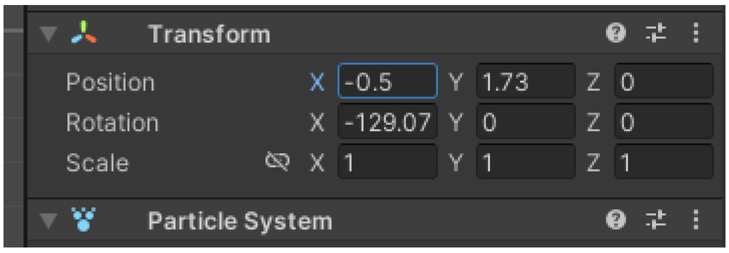
Add image to the Pics folder.

Add image to the project.

Add particles!



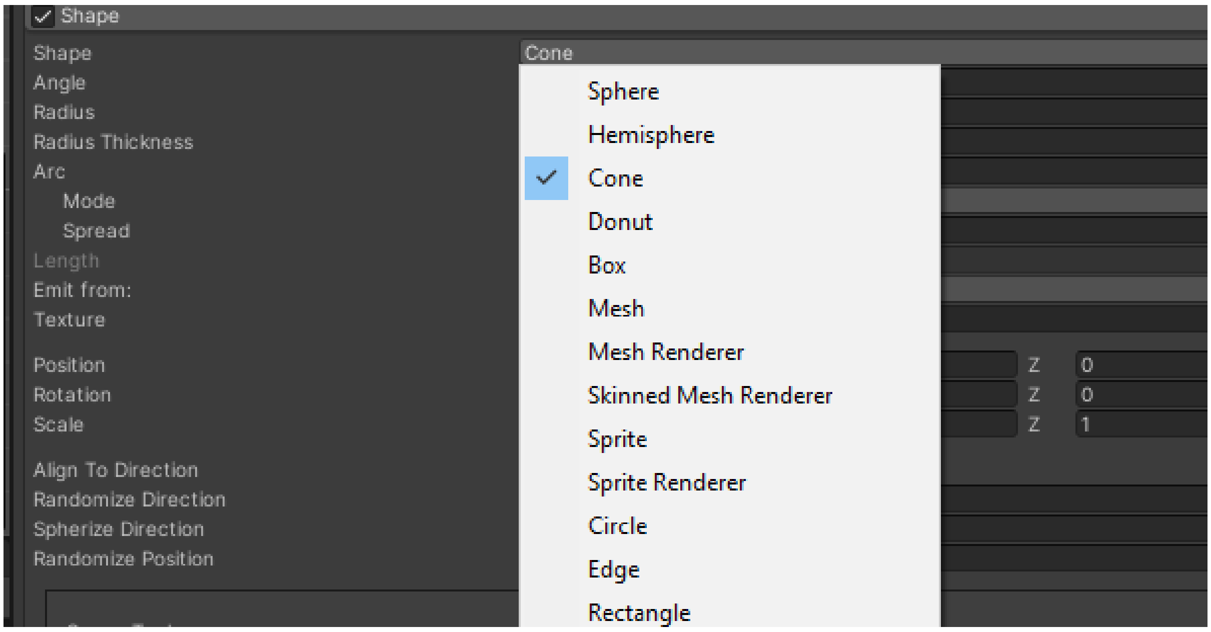
You can move particle sets by using Transform:


Transform

Position	X	-0.5	Y	1.73	Z	0
Rotation	X	-129.07	Y	0	Z	0
Scale	X	1	Y	1	Z	1

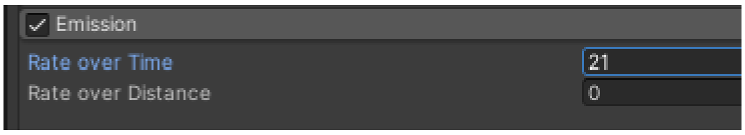
Particle System

You can manipulate particle area and other geometric properties by using Shape fields:

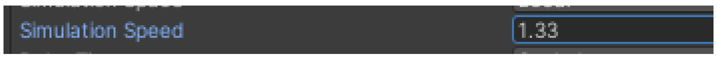


Try!

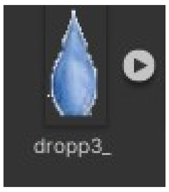
More speed to particles: e.g Emission can be used


Emission
Rate over Time 21
Rate over Distance 0

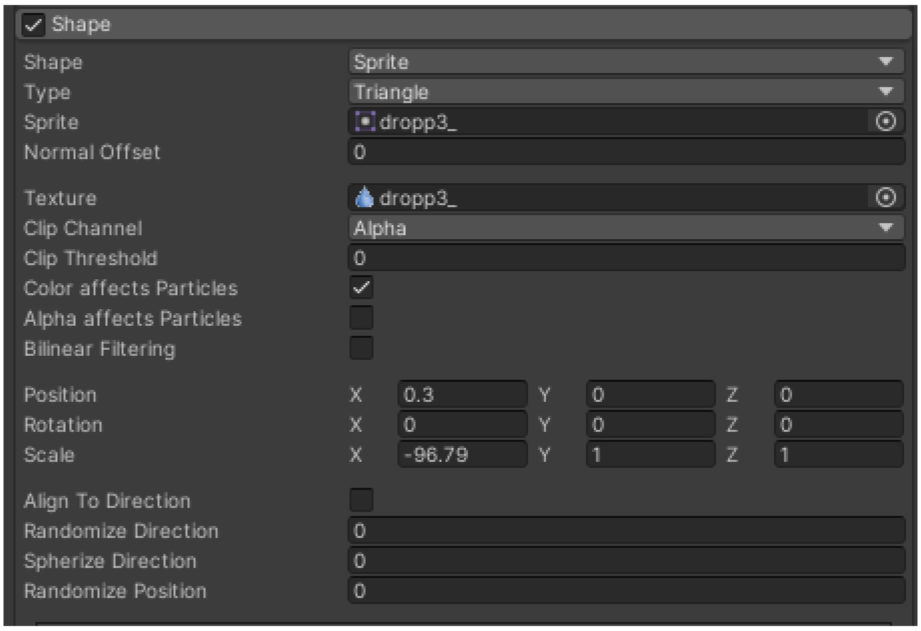
Or simulation speed:


Simulation Speed 1.33

Particles: we can use drop image.



Then changes to shape properties:



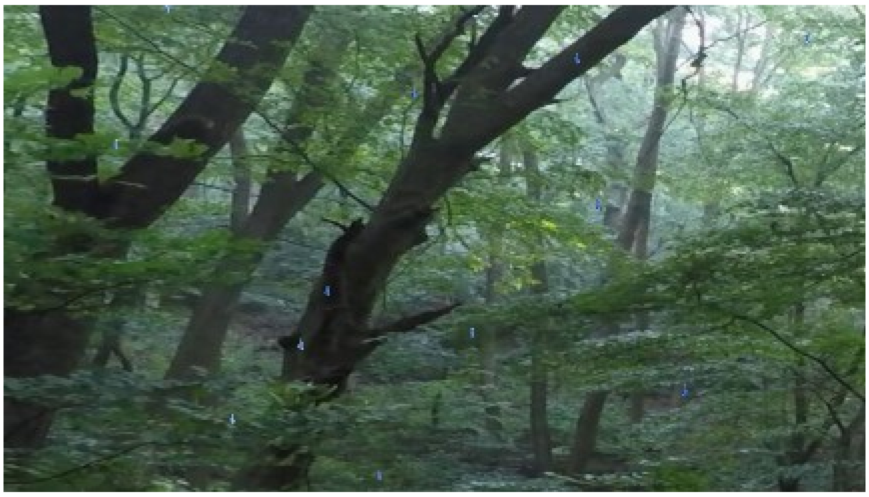
✓ Shape						
Shape	Sprite					
Type	Triangle					
Sprite	dropp3_					
Normal Offset	0					
Texture	dropp3_					
Clip Channel	Alpha					
Clip Threshold	0					
Color affects Particles	✓					
Alpha affects Particles	☐					
Bilinear Filtering	☐					
Position	X	0.3	Y	0	Z	0
Rotation	X	0	Y	0	Z	0
Scale	X	-96.79	Y	1	Z	1
Align To Direction	☐					
Randomize Direction	0					
Spherize Direction	0					
Randomize Position	0					

Shape is now a sprite.

So rerence to the drop image are added to fields Sprite and Texture.

Try different values.

Big drops raining now (remember climate change ;)).



Note

To 3D projects it is possible to import **Standard Assets** package that contains good effects, fires, explosions etc.

Comments are welcome!

Thank you!