



Welcome to Game Design and Development

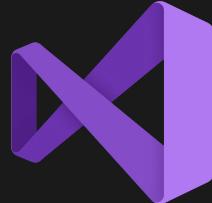
What Software Are We Using



Unity: Our Game Engine.



Visual Studio: The IDE we will be using



GitHub Desktop: The way we'll connect to the use of Version Control System.

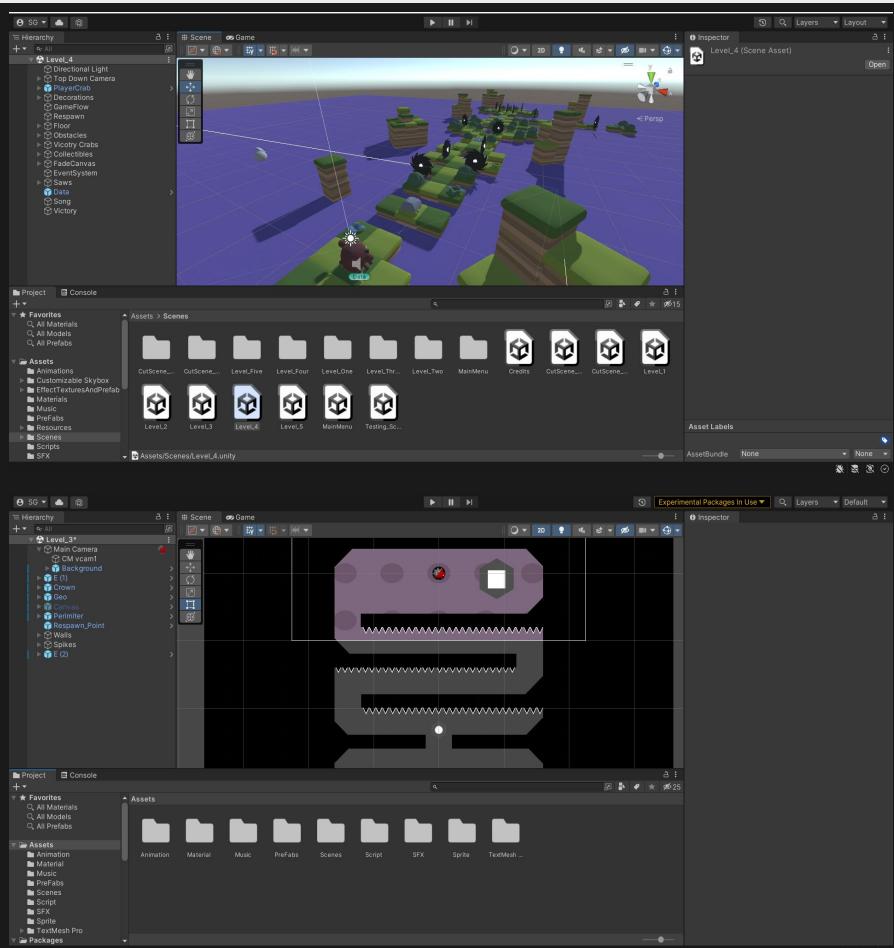


Unity

What is a Game Engine?

- It's a software that already has built in library and features that help you create games.

Things like physics, image processing, sound are already set in place and you have to make the connections between the features to create the game rather than having to program everything from scratch.



Perks of Unity



- Port it to any platform; Unity allows you to create executable files for your game to pretty much any platform that's currently available. This includes: Windows, Mac, Linux, Web Browser, Android, iOS, AR, VR, Playstation, Xbox, and Switch consoles.
- Is free software for personal use as long as your projects don't exceed \$100,000 revenue in a year.
- Is connected to the Unity Asset Store which host a treasure trove of free and purchasable assets.

Games Created With Unity

Unity is a multi-tool that allows whoever wields it to create anything:

Of course primarily we think of games when it comes to Unity and there are plenty of great examples such as "Cuphead", "Hearthstone", "Rust", "City Skylines", and many more.

All of these games cover wildly different styles of gameplay and complexity.

If you're interested in checking the comprehensive list of games made with Unity.



Animation Created With Unity

Unity holds a incredibly powerful animation system which we will discuss during our third week of classes where we will animate our own cutscene.

Beyond just in-game cutscene Unity can create full on animated shows and movies such as “Mr. Carton” by Michaël Bolufer. Furthermore allowing companies such as Disney to film in VR environments using Unity creating movies like “The Lion King [2019]”.



If you're interested in checking out some other [animations](#) made with Unity. ["Lion King" Article.](#)

Non Game Related Software With Unity

With the Unity engine being so versatile other industries have taken it for their own uses.

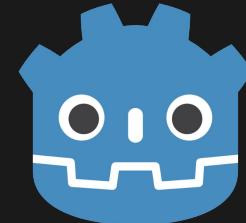
Main two uses are for visualization and simulations, allowing engineerings, architects, medical professionals see what it is they are constructing or working on while simulating the physics of their prototype.



If you're interested in checking out software made for [automobiles](#), [engineering](#) or [aerospace](#) using Unity.

Other Game Engines

Unreal: Industry Standard uses 2D/3D game engine using C++ is free to use. Witcher 4 is being developed in it.



GoDot: Develing into creating 2D and 3D Environments and Levels, allows you to Programming with C# and C++ and Visual Scripting. Free to Use.

Construct 3: Web based 2D game engine with Visual Scripting, you have to licence it for all tools.



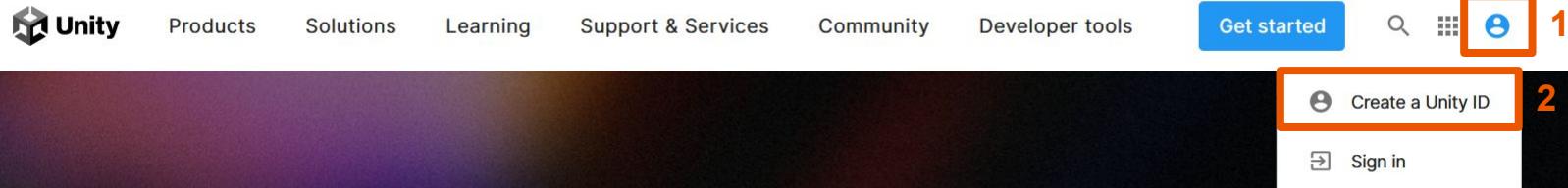
There dozens of game engines all different from each other however, many use similar standardized tools so that once you are familiar in one you can transfer your skills to another easily.

Additional Resources: [Most used Engines](#) [The Best Game Engines of 2021](#)

Making A Unity Account

Before we install anything we will first create an account with Unity. This account will allow you to access the Unity Asset Store from which you'll be able to download 3D and 2D assets to create your projects throughout this class and possibly for your final project.

Head to <https://unity.com/> and click the avatar in top right [1] and then click Create a Unity ID [2]



Making A Unity Account

Fill out the information or use an existing account to connect to Unity.

Create a Unity ID

If you already have a Unity ID, please [sign in here](#).

Email

Password

Username

Full Name

I have read and agree to the [Unity Terms of Service](#)(required).

I acknowledge the [Unity Privacy Policy](#) [Republic of Korea Residents agree to the [Unity Collection and Use of Personal Information](#)] (required).

I agree to have [Marketing Activities](#) directed to me by and receive marketing and promotional information from Unity, including via email and social media(optional).

I'm not a robot 
reCAPTCHA
Privacy - Terms

[Create a Unity ID](#) [Already have a Unity ID?](#)

OR

Installing Unity

To access Unity you will first have to download the Unity Hub.

Unity Hub is center for you to manage your projects and versions of Unity. As time goes on Unity releases newer versions that hold new or improved features.

To start your download head over to
<https://unity.com/download#how-get-started>.

If you scroll down a bit you will see a box that allows you to download Unity Hub for all the different platforms.

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](#)

Unity Hub

Once you've gotten the Unity Hub installed you should be greeted with this window.

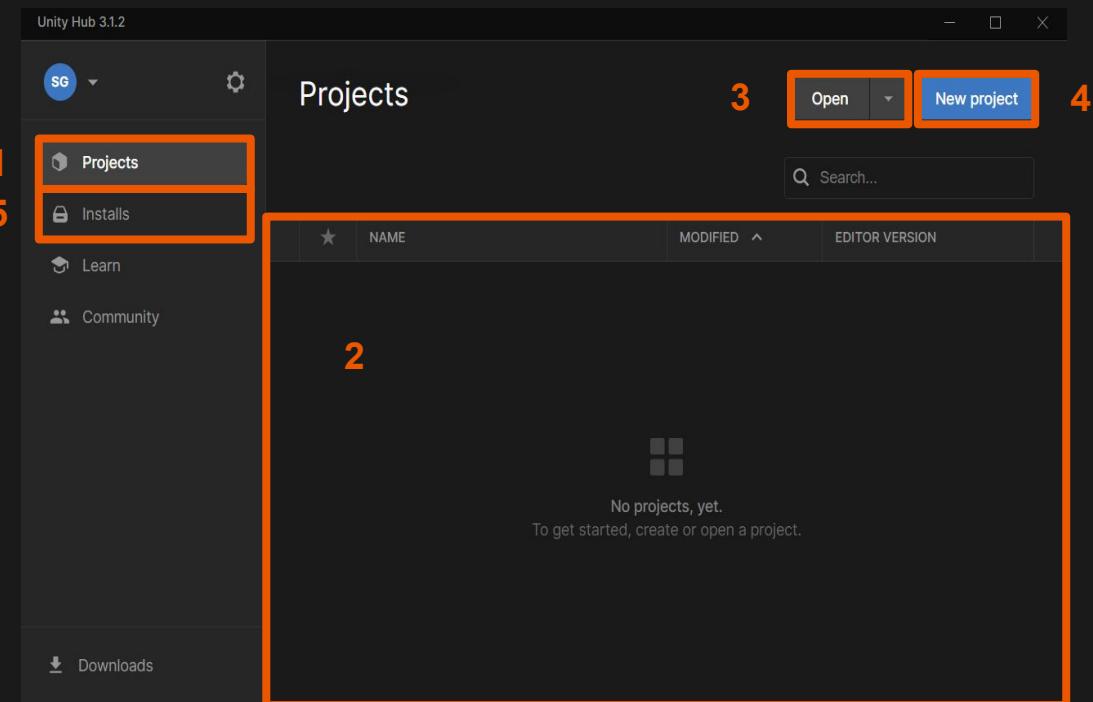
[1] The Projects Tab allows you to view all of the project you've recently worked on.

[2] This is where all the project will selected from.

[3] This allows you to open pre-existing projects that aren't displayed in [2].

[4] Allows you to create a brand new project.

[5] Install tab allows us to manage the versions of Unity that are currently installed and available for us to use.



Unity LTS

LTS or Long Term Support is a version of Unity released once a year, it is stable, and it will be continuously supported for two years after launch. Each new year has new features so download the 2021.X.XXX which is the one we'll be using for this class.

This will take a fair bit of time and space, so we will comeback to Unity in our next lesson once everyone has it installed.

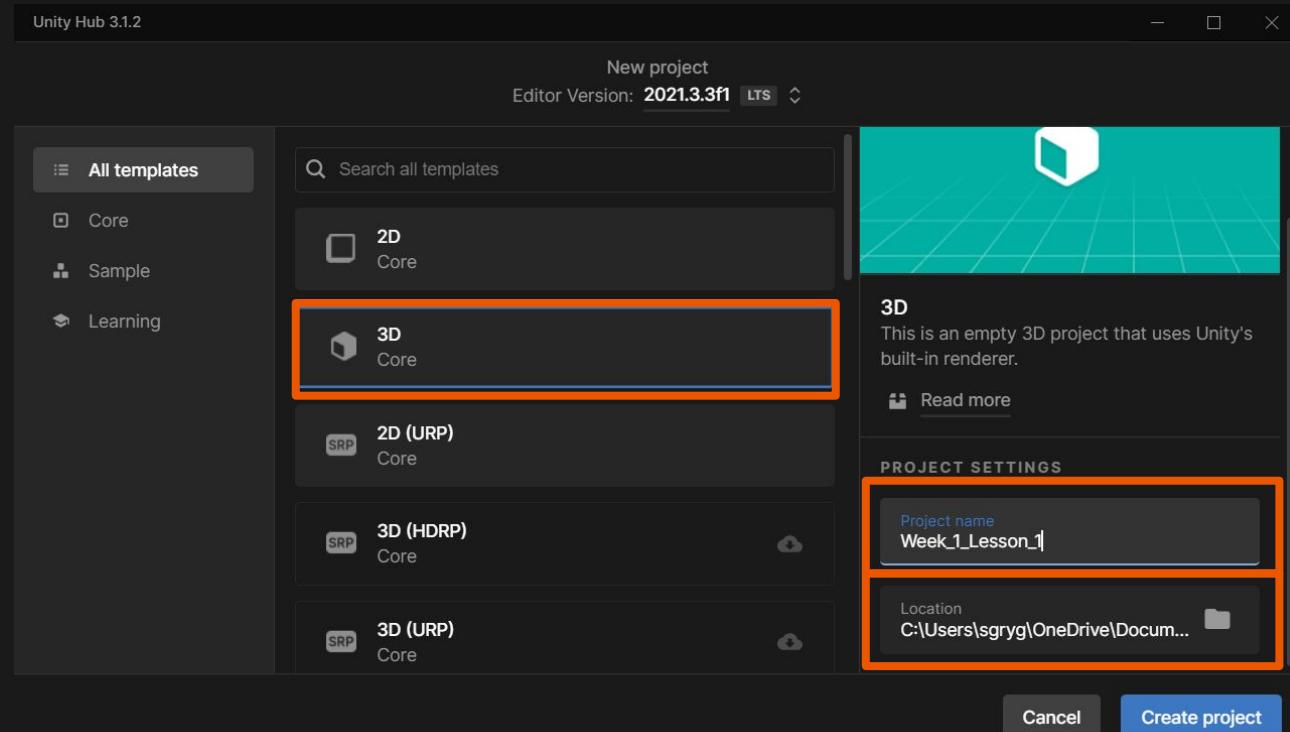
The screenshot shows the Unity Hub application interface. On the left, there's a sidebar with 'Projects', 'Installs' (which is selected and highlighted in blue), 'Learn', and 'Community'. The main area is titled 'Installs' and shows a list of installed versions. The first item is '2021.3.3f1 LTS' with a path 'C:\Program Files\Unity\Hub\Editor\2021.3.3f1\Editor\Unity.exe'. Below this are 'WebGL' and 'Windows' buttons. To the right of the main window is a modal dialog titled 'Install Unity Editor'. It has tabs for 'Official releases', 'Pre-releases', and 'Archive'. Under 'LONG TERM SUPPORT (LTS)', three versions are listed: '2021.3.3f1 LTS' (marked as 'Installed'), '2020.3.34f1 LTS', and '2019.4.39f1 LTS'. Each entry has a blue 'Install' button to its right. A large orange rectangle highlights the '2021.3.3f1 LTS' entry. A small orange number '2' is positioned to the left of the 'Install Unity Editor' dialog. At the top right of the main window, there are 'Locate' and 'Install Editor' buttons, with 'Install Editor' being the active one. A small orange number '1' is at the top right corner of the main window.

Opening up a new Project

When LTS is downloaded you will be able to head back to the Projects Tab and click, 'New Project' and should be met with a similar window to this.

Keep the template to 3D and make sure you choose a name and location of the project.

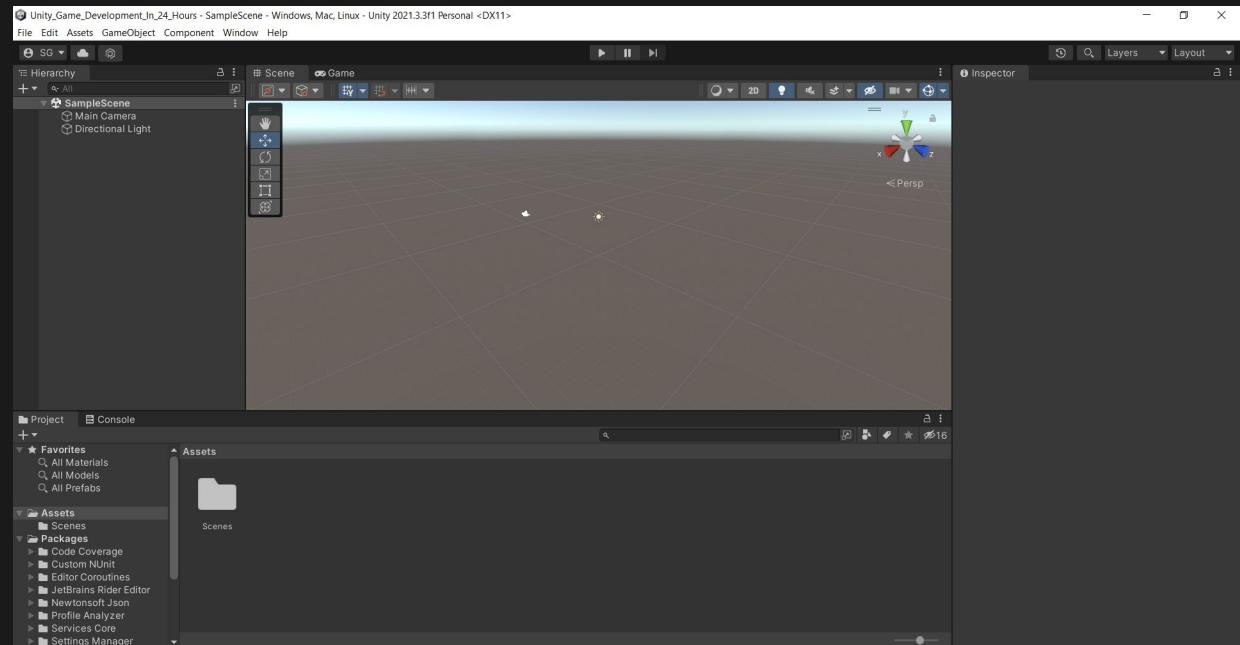
Once all of that is set, feel free to click Create Project.



Unity Once Loaded Up

Creating a new 3D
Project will greet you
with the Unity Game
Engine.

We will go in depth on
what you're seeing in our
next lesson.



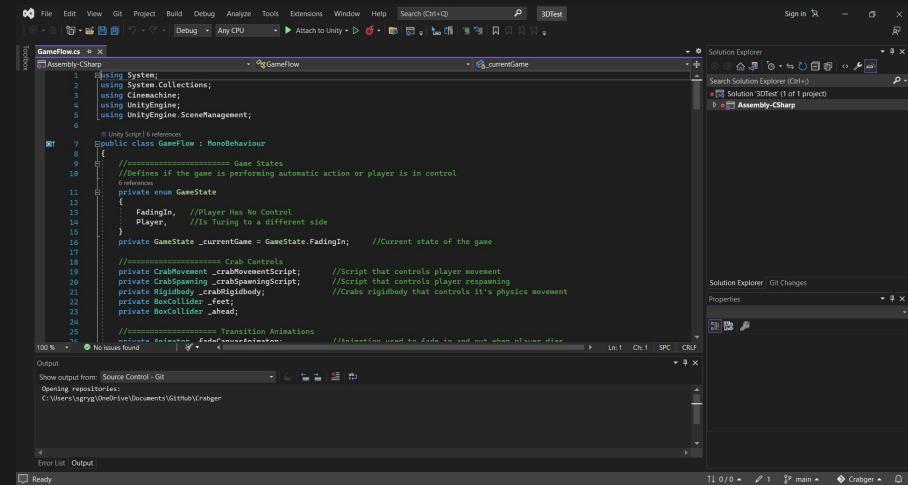
Visual Studio

What is an Integrated Development Environment (IDE)?

- Is a software that allows programmers to write, edit and debug code. In simplest form it's a supercharged Text Editor.

With Visual Studio you will be able to write C# scripts that bring life to all of the Game Objects you will create.

Visual Studio Code is developed by Microsoft.



The screenshot shows the Visual Studio IDE interface with the following details:

- Title Bar:** File, Edit, View, Git, Project, Build, Debug, Analyze, Tools, Extensions, Window, Help, Search (Ctrl+Q), 3D Test, Sign in.
- Solution Explorer:** Shows a solution named "3DTest" containing one project named "Assembly-CSharp".
- Code Editor:** The file "Gameflows.cs" is open, displaying C# code for a Unity script. The code defines a class GameFlow that interacts with various Unity components like Transform, System.Collections, and UnityEngine. It includes enums for Game States (FadingIn, Player, FadingOut) and variables for a player, crab movement, and rigidbody.
- Output Window:** Shows the path to the file: C:\Users\kgryg\OneDrive\Documents\GitHub\Crabger.
- Status Bar:** Displays file navigation icons and the current file name "main.cs".

IDE vs Text Editor



Here is the code shown in Visual Studio versus Windows Notepad. They more or less fulfill the same purpose of allowing you to write code.

However you can immediately tell that Visual Studio Code, color code your work by what the word means. Give you a line counter which is necessary when debugging your program.

Beyond that Visual Studio has useful features such as autocompleting the word or method you are currently writing.

We are using Visual Studio because it works directly with C# and is easily integrated into Unity, different IDEs will be predisposed to working with different programming languages.

```
File Edit View Git Project Build Debug Analyze Tools Extensions Window Help Search (Ctrl+Q) P IDTest
Assembly-CSharp
using System;
2 using System.Collections;
3 using Cinemachine;
4 using UnityEngine;
5 using UnityEngine.SceneManagement;
6
7 @UnityScript (References)
8 public class GameFlow : MonoBehaviour
9 {
10     //=====
11     //Defines if the game is performing automatic action or player is in control
12     private enum GameState
13     {
14         FadingIn, //Player Has No Control
15         Player, //Is Turning to a different side
16     }
17     private GameState _currentGame = GameState.FadingIn; //Current state of the game
18
19     //=====
20     //Crab Controls
21     private CrabMovement _crabMovementScript; //Script that controls player movement
22     private CrabRespawning _crabRespawningScript; //Script that controls player respawning
23     private Rigidbody _rigidbody; //Rigidbody that controls its physics movement
24     private BoxCollider _feet;
25     private BoxCollider _ahead;
26
27     //=====
28     //Transition Animations
29     private Animator _fadeCanvasAnimator; //Animation used to fade in and out when player dies
30     private Animator _winAnimator; //Animation used at the end of the level
31
32     //=====
33     //Camera Movements
34     private CinemachineVirtualCamera _camera; //The virtual camera
35     private Transform _topCamera; //The game object that angles and positions the camera
36     private Transform _cameraLocation; //Where the virtual camera goes at the end of level
37
38     //=====
39     //Collective Updates
40     private GameObject _fruit; //The apple that is used to show if player collected the object or not
41     public SkinnedMeshRenderer crabInTransition; //Holds the crab in the win animation
42     private SkinnedMeshRenderer _crabInLevel; //The render of the crab in the level
43
44     //=====
45     //Level To Go
46     public string nextSceneName; //Name of the next scene
47     public int levels; //Number of the crab in the win animation
48     [HideInInspector] public GameObject[] levelFruit = new GameObject[5];
49
50 }
```

Output
Show output from: Source Control - Git
Opening repositories:
C:\Users\sgryg\OneDrive\Documents\GitHub\Crabger

Error List Output

Ready

File Edit Format View Help
using System;
using System.Collections;
using Cinemachine;
using UnityEngine;
using UnityEngine.SceneManagement;

public class GameFlow : MonoBehaviour

[
===== Game States
//Defines if the game is performing automatic action or player is in control
private enum GameState
{
 FadingIn, //Player Has No Control
 Player, //Is Turning to a different side
}
private GameState _currentGame = GameState.FadingIn; //Current state of the game
===== Crab Controls
private CrabMovement _crabMovementScript; //Script that controls player movement
private CrabRespawning _crabRespawningScript; //Script that controls player respawning
private Rigidbody _rigidbody; //Rigidbody that controls its physics movement
private BoxCollider _feet;
private BoxCollider _ahead;
===== Transition Animations
private Animator _fadeCanvasAnimator; //Animation used to fade in and out when player dies
private Animator _winAnimator; //Animation used at the end of the level
===== Camera Movements
private CinemachineVirtualCamera _camera; //The virtual camera
private Transform _topCamera; //The game object that angles and positions the camera
private Transform _cameraLocation; //Where the virtual camera goes at the end of level
===== Collective Updates
private GameObject _fruit; //The apple that is used to show if player collected the object or not
public SkinnedMeshRenderer crabInTransition; //Holds the crab in the win animation
private SkinnedMeshRenderer _crabInLevel; //The render of the crab in the level
===== Level To Go
public string nextSceneName; //Name of the next scene
public int levels; //Number of the crab in the win animation
[HideInInspector] public GameObject[] levelFruit = new GameObject[5];

Ln 43 Col 1 100% Windows (CR LF) UFT-8

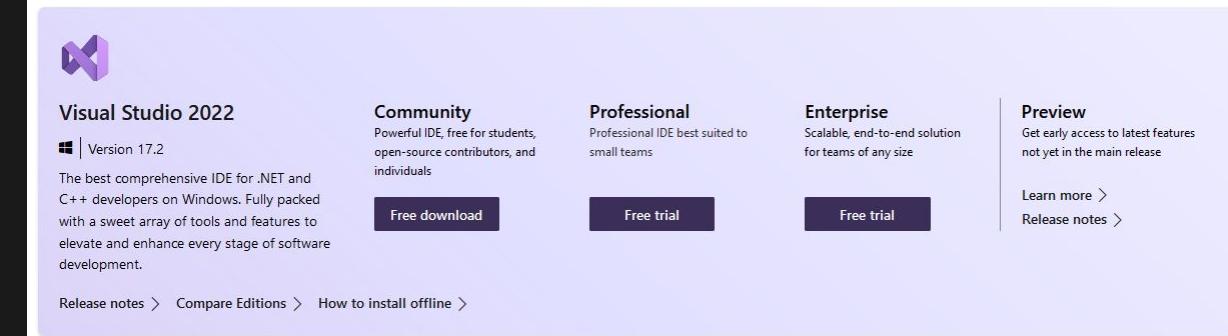
Downloading Visual Studio

Downloads

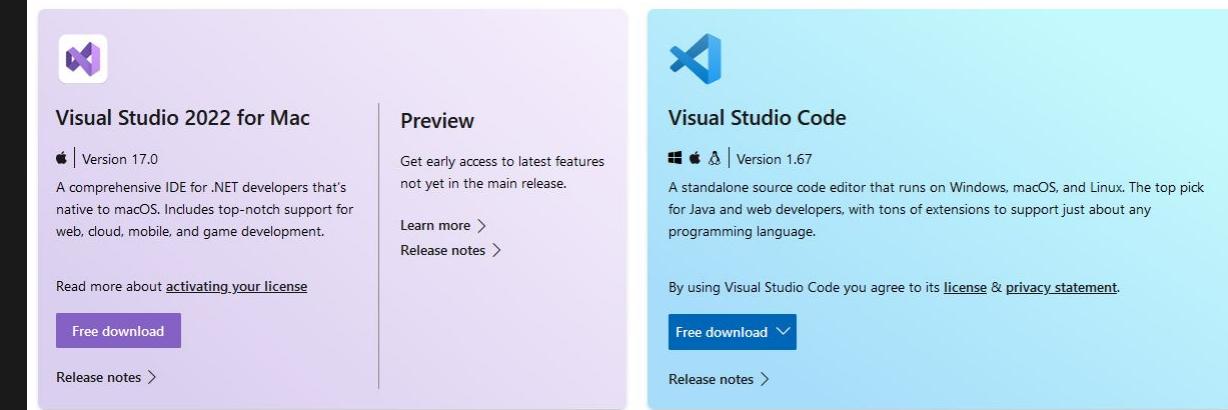
Head over to download:
<https://visualstudio.microsoft.com/downloads/>

From here select the
Community download for
Windows and Free Download
for Mac.

This will install Visual Studio
Installer, a Hub that will allow
you to download libraries for
whatever programming work
you intend to do.



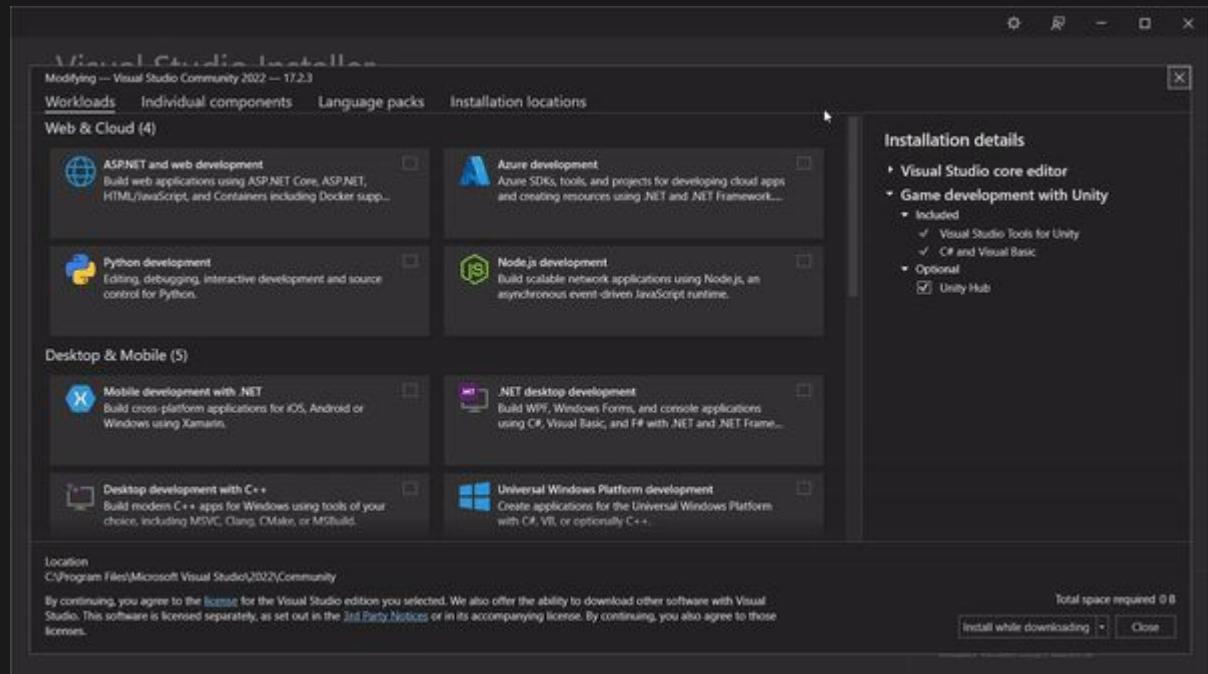
The screenshot shows the Visual Studio Downloads page. At the top, there's a purple header with the Microsoft logo and the text "Visual Studio 2022". Below it, there are four main download options: "Community", "Professional", "Enterprise", and "Preview". Each option has a brief description and a "Free download" or "Free trial" button. Below these options, there are links for "Release notes", "Compare Editions", and "How to install offline".



The screenshot shows the Visual Studio Downloads page with two additional options: "Visual Studio 2022 for Mac" and "Visual Studio Code".
Visual Studio 2022 for Mac: This section includes the Visual Studio 2022 for Mac logo, a brief description, a "Free download" button, and links for "Read more about activating your license", "Release notes", and "Learn more".
Visual Studio Code: This section includes the Visual Studio Code logo, a brief description, a "Free download" button, and links for "Release notes" and "Learn more".

Visual Studio Hub - Workload Installs

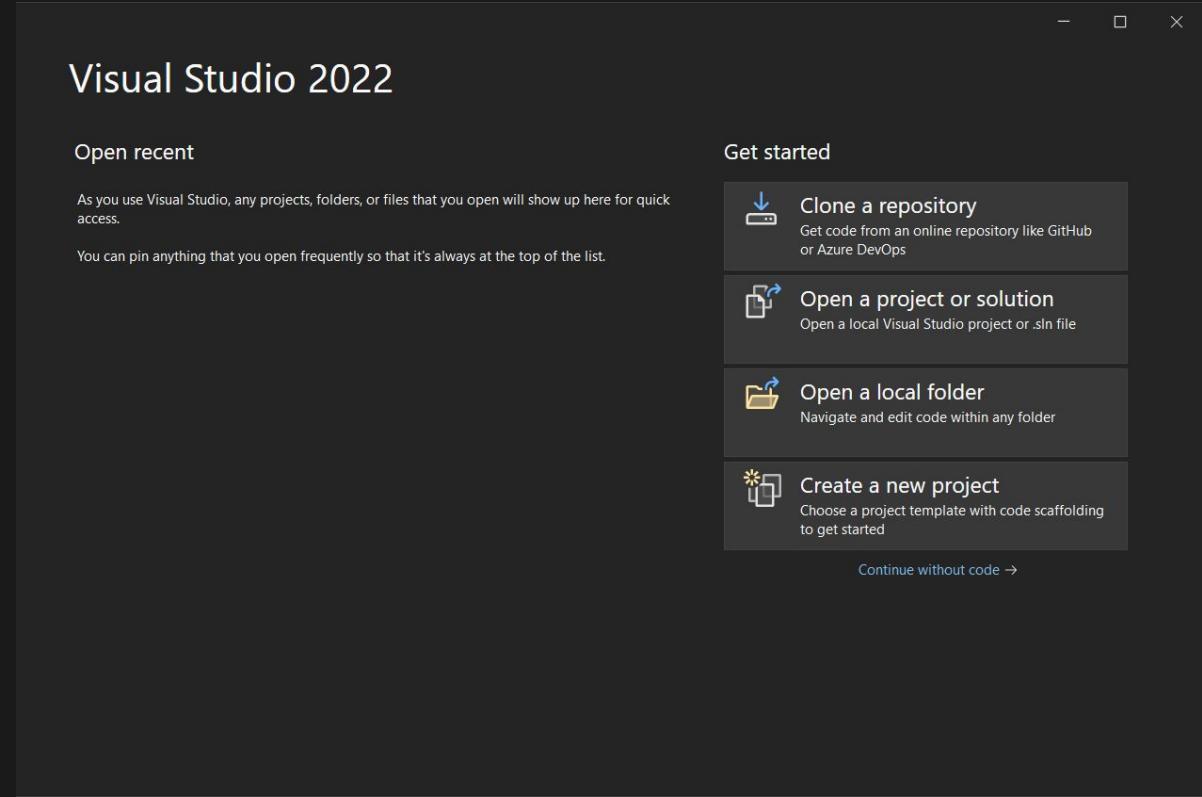
During the installation it will ask you if you want to download any Workloads, scroll down and check the “Game Development with Unity” that’s the only thing you’ll need for this course.



Visual Studio

Once you finish your installation you will be met with this screen allowing you to select a Script.

For now this is where we'll stop as we will become much more familiar with it once we get into Week 2 of the class.

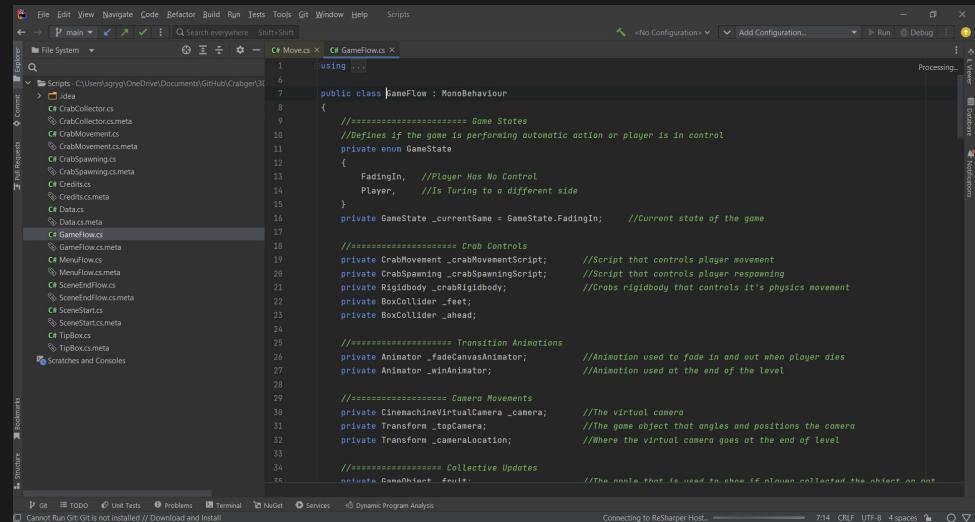


Possible Alternative - Jetbrains Rider

If you ever end up working on Linux, Visual Studio Code will not be your friend. It's heavily scaled back compared to it's Windows and Mac versions becoming more or less a Text Editor with color coded code.

An alternative to Visual Studio Code would work on Windows, Mac and Linux would be Jetbrains Rider, it has more features, has better UI and I personally use it to program games. However, Rider has no free version and you have to licence it out.

Unless you are actively programming games for an extend period of time I don't recommend you spend money on it.



```
File Edit View Navigate Code Refactor Build Run Tests Tools Git Window Help Scripts
<- P main \v Search everywhere Shift+Shift
File System < C:\Users\sgryg\OneDrive\Documents\GitHub\Crabger3d\1 Move.cs x C:\Gameflow.cs x
> \idea
  \CrabCollector.cs
  \CrabCollector.cs.meta
  \CrabMovement.cs
  \CrabMovement.cs.meta
  \CrabSpawning.cs
  \CrabSpawning.cs.meta
  \Credits.cs
  \Credits.cs.meta
  \Data.cs
  \Data.cs.meta
  \Gameflow.cs
  \Gameflow.cs.meta
  \MenuFlow.cs
  \MenuFlow.cs.meta
  \SceneEndFlow.cs
  \SceneEndFlow.cs.meta
  \SceneStart.cs
  \SceneStart.cs.meta
  \TipDolox.cs
  \TipDolox.cs.meta
  \Scratches and Console
  \Scratches and Console.meta

using ...

public class Gameflow : MonoBehaviour
{
    //=====
    //***** Game States
    //Defines if the game is performing automatic action or player is in control
    private enum GameState
    {
        FadingIn, //Player Has No Control
        Player, //Is Turning to a different side
    }

    private GameState _currentGame = GameState.FadingIn; //Current state of the game

    //=====
    //***** Crab Controls
    private CrabMovement _crabMovementScript; //Script that controls player movement
    private CrabSpawning _crabSpawningScript; //Script that controls player spawning
    private Rigidbody _crabRigidbody; //Crabs rigidbody that controls its physics movement
    private BoxCollider _feet;
    private BoxCollider _ahead;

    //=====
    //***** Transition Animations
    private Animator _fadeCanvasAnimator; //Animation used to fade in and out when player dies
    private Animator _winAnimator; //Animation used at the end of the level

    //=====
    //***** Camera Movements
    private CinemachineVirtualCamera _camera; //The virtual camera
    private Transform _topCamera; //The game object that angles and positions the camera
    private Transform _cameraLocation; //Where the virtual camera goes at the end of the level

    //=====
    //***** Collective Updates
    private GameObject _fruit; //The handle that is used to show if player collected the object or not
}

Connecting to ReSharper Host...
7:14 CR LF UTF-8 4 spaces
Cannot Run Git: Git is not installed // Download and Install
```

Additional Resources: [Jetbrains Rider](#) & [Visual Studio vs JetBrains Rider: A Detailed Comparison](#)

GitHub - Version Control System

What is Version Control System (VCS)?

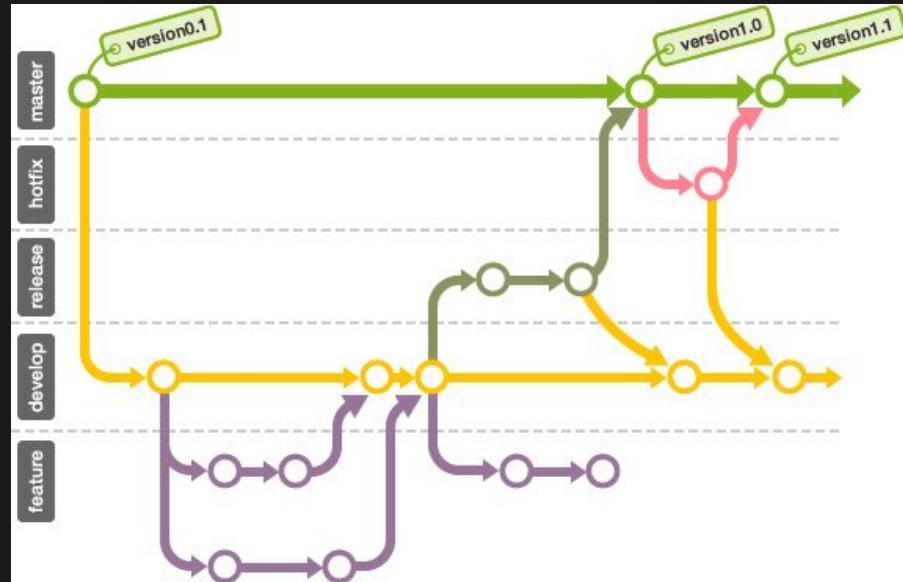
- A way of storing and updating project files for individual or group project.

Think of our game as a Google Doc and VCS will allow us to all make changes to it actively while recording the history of changes and allowing us to rollback to previous moment in time.

It allows two or more people to work on a project and if those individuals edited the same work it will show them as conflicts allowing you to go line by line of code to choose the best of both parts.

What is GitHub? It's the Hosted Version Control, it's all of your files you store on Google.

Additional Resources: [Git vs. GitHub: What's the difference](#)

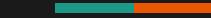


Creating A GitHub Account

Let's start by making an account on GitHub <https://github.com/>, GitHub is a web-based version-control and collaboration platform for software developers and is operated by Microsoft.



Creating A GitHub Account



Create a GitHub account by filling out your information in GitHub's adventure styled form.

```
Welcome to GitHub!
Let's begin the adventure

Enter your email
✓

Create a password
✓

Enter a username
✓

Would you like to receive product updates and announcements via email?
Type "y" for yes or "n" for no
✓
```

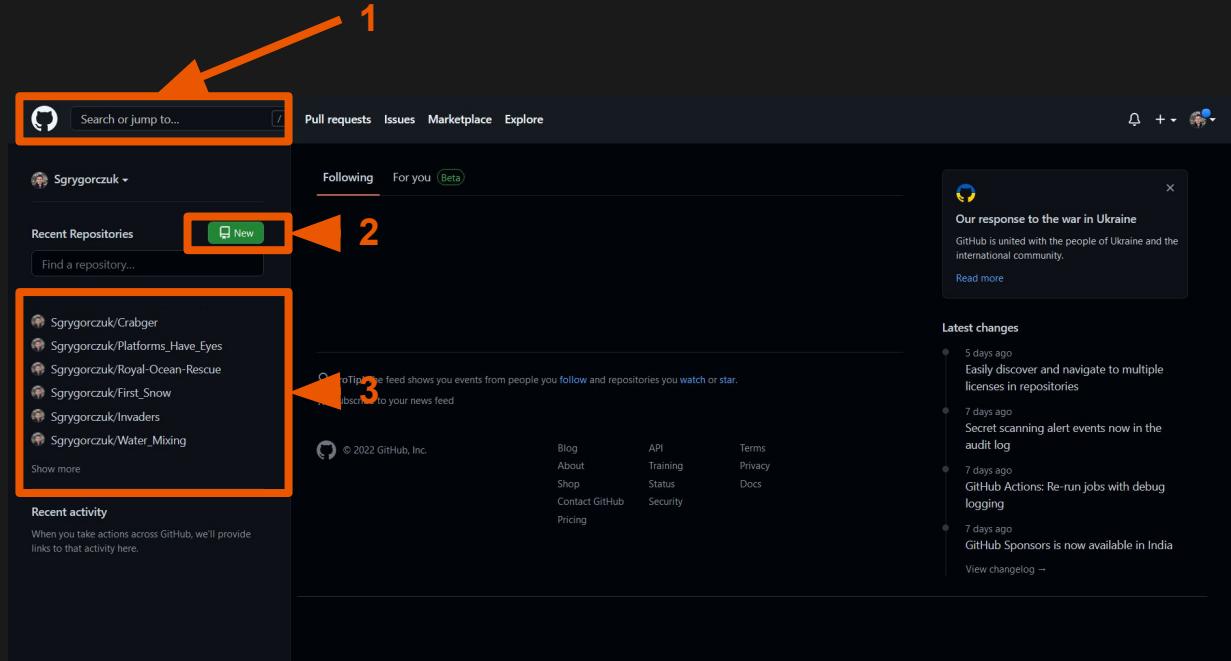
GitHub

Once you're done setting up your details you will be sent to the Landing Page. Let's go over few of it's features

[1] the search bar allow you to browse through everything that is publicly stored on GitHub. All of the code you will find here is called Open-Sourced, means it's available for everyone to access and modify. There may be limitations depending on project, check the ReadMe attached to the project.

[2] Creating a new Repository, repositories are just folders.

[3] History of Recent Work, will allow you for quick access to whatever you're working on.



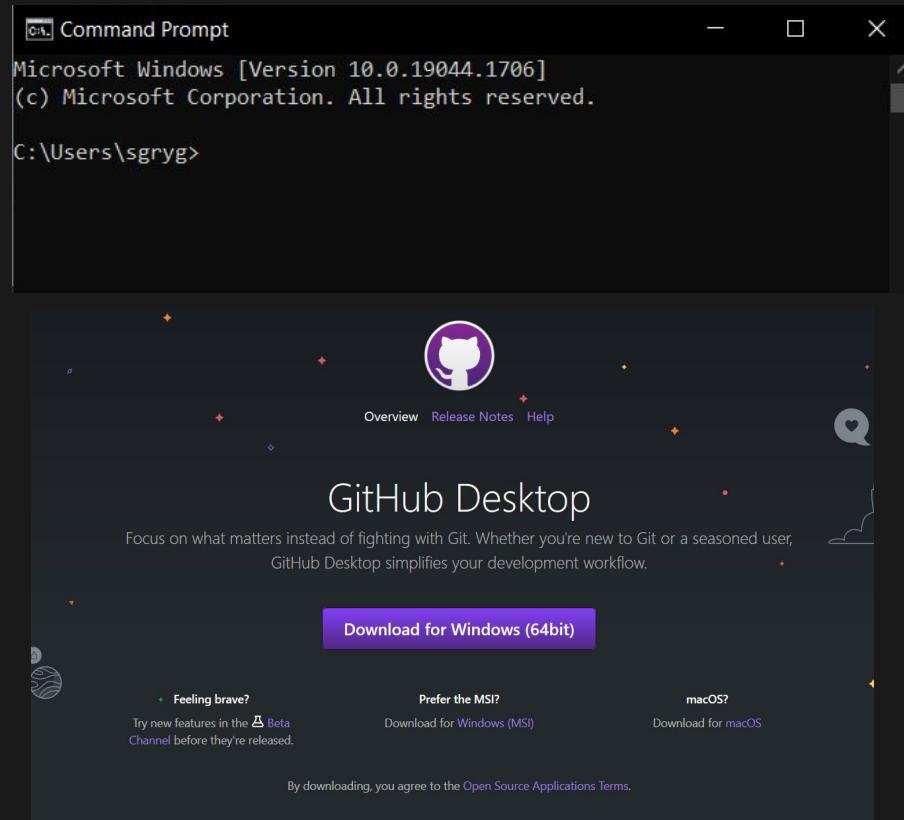
Downloading GitHub Desktop

The way a lot of people work with VCS is by using the Command Line, although giving you greater control over how and what you are doing it can be very overwhelming to have to move folders and uploading project files using command line prompts.

To alleviate that we will download GitHub Desktop which we will connect to the GitHub account we just made to have a visual way to control our files.

Go to <https://desktop.github.com/> select the system you're working on and download the program.

Anyone feeling more daring and wants to use the Command line can use this [Cheat Sheet](#).



GitHub Desktop

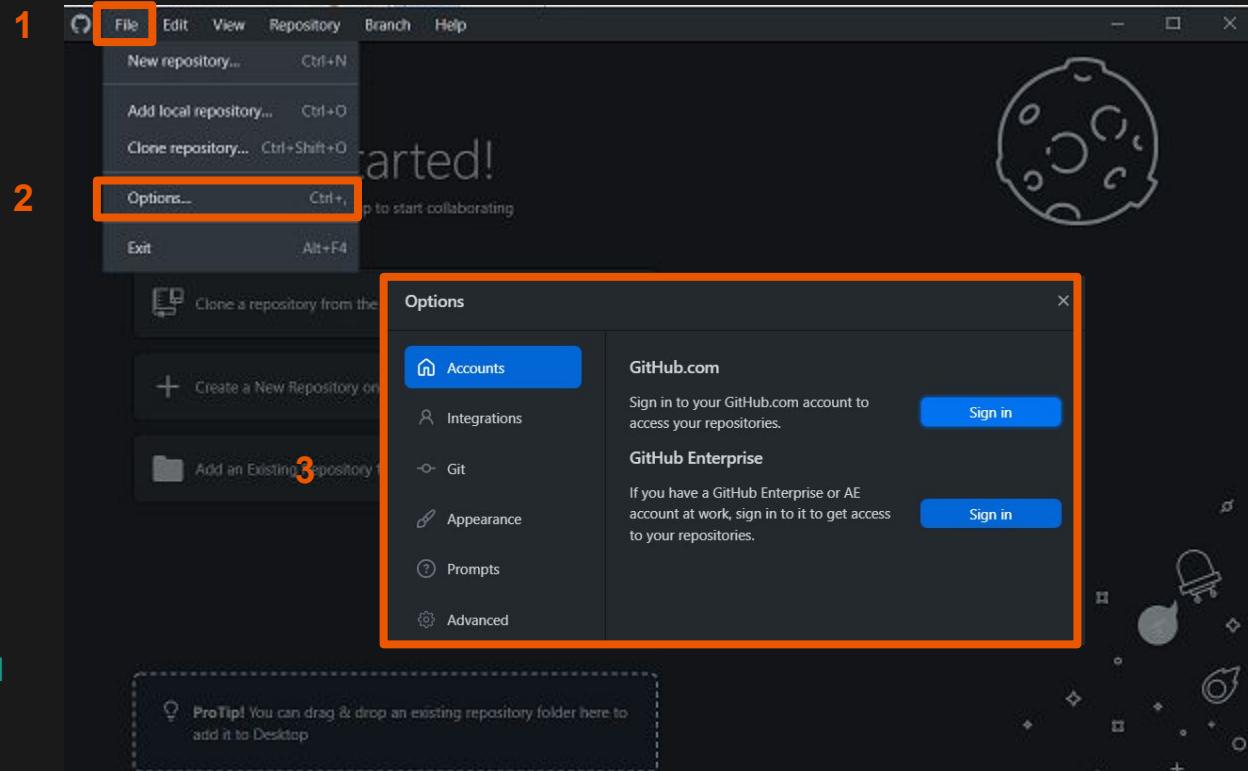
Once GitHub Desktop is installed you will be met with this page.

First thing we're going to do is connect you to the GitHub account we just made.

Click on [1] the File in the Toolbar and then select [2] Options.

Alternatively you can just click **Ctrl + ,**

From there you'll receive the [3] PopUp, click sign in. It's going to send you to your browser and since you've just made your account and should still be logged in it will automatically connect.



Version Control System Commands

Git is full of keywords that we will use inside of GitHub Desktop to access and modify our projects. The ones we'll encounter the most are:

Clone: Copying/Downloading the Repository to make a local copy.

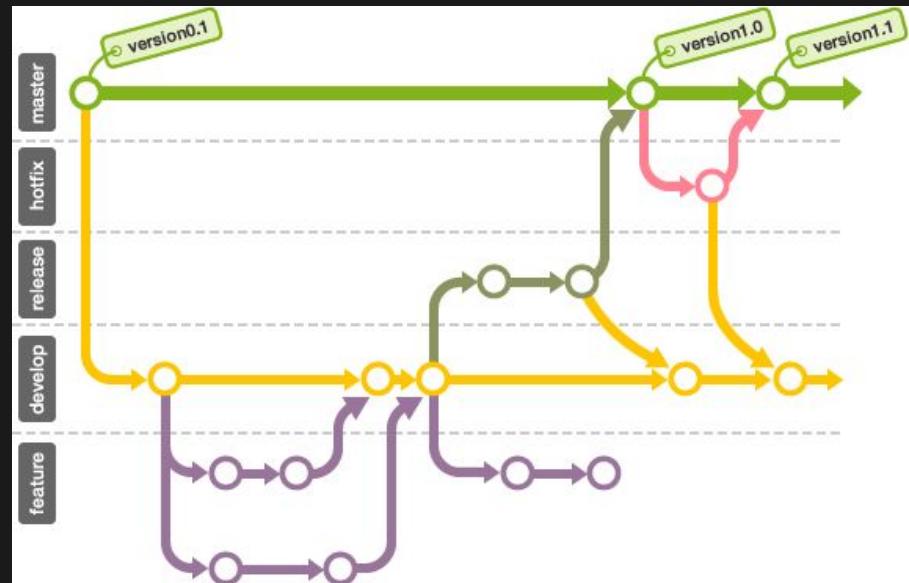
Fork: Separating from the current branch. Allowing you to one day Merge or keep it as your own personal project.

Merge: Connecting your branch to it's parent. Have to resolve any conflict.

Commit: Indicates that you are creating a snapshot of what your repository looks like now.

Push: Allows you to put whatever changes have been saved in the Commit into the cloud version of the project.

Pull: Pulls down any updates that occurred in the project, if you already have the project on your desktop.



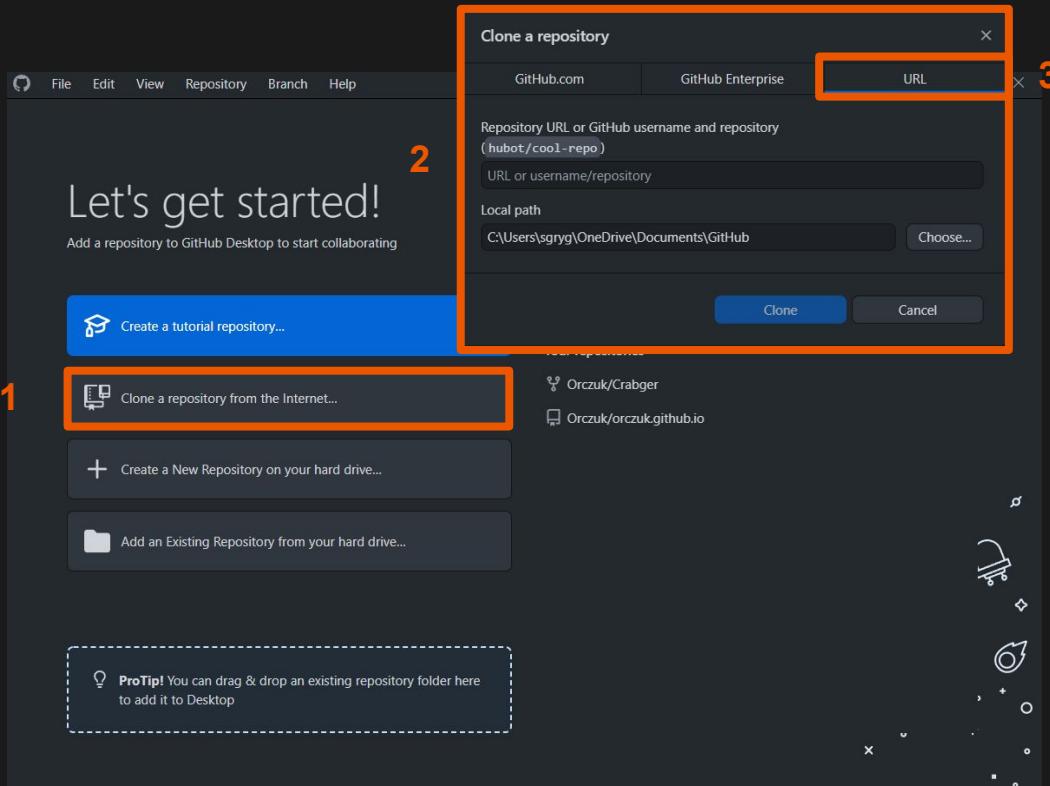
Version Control System Commands In Action

Let's start by cloning the Project Files for Day 2 Lesson. From you landing page click on the [1] Clone a repository from the Internet.

That should give you the [2] Clone a repository PopUp, select [3] URL and paste in

https://github.com/Sgrygorczuk/Week_1_Lesson_2

This repository holds the project files that we will use tomorrow to explore the Unity Game Engine.

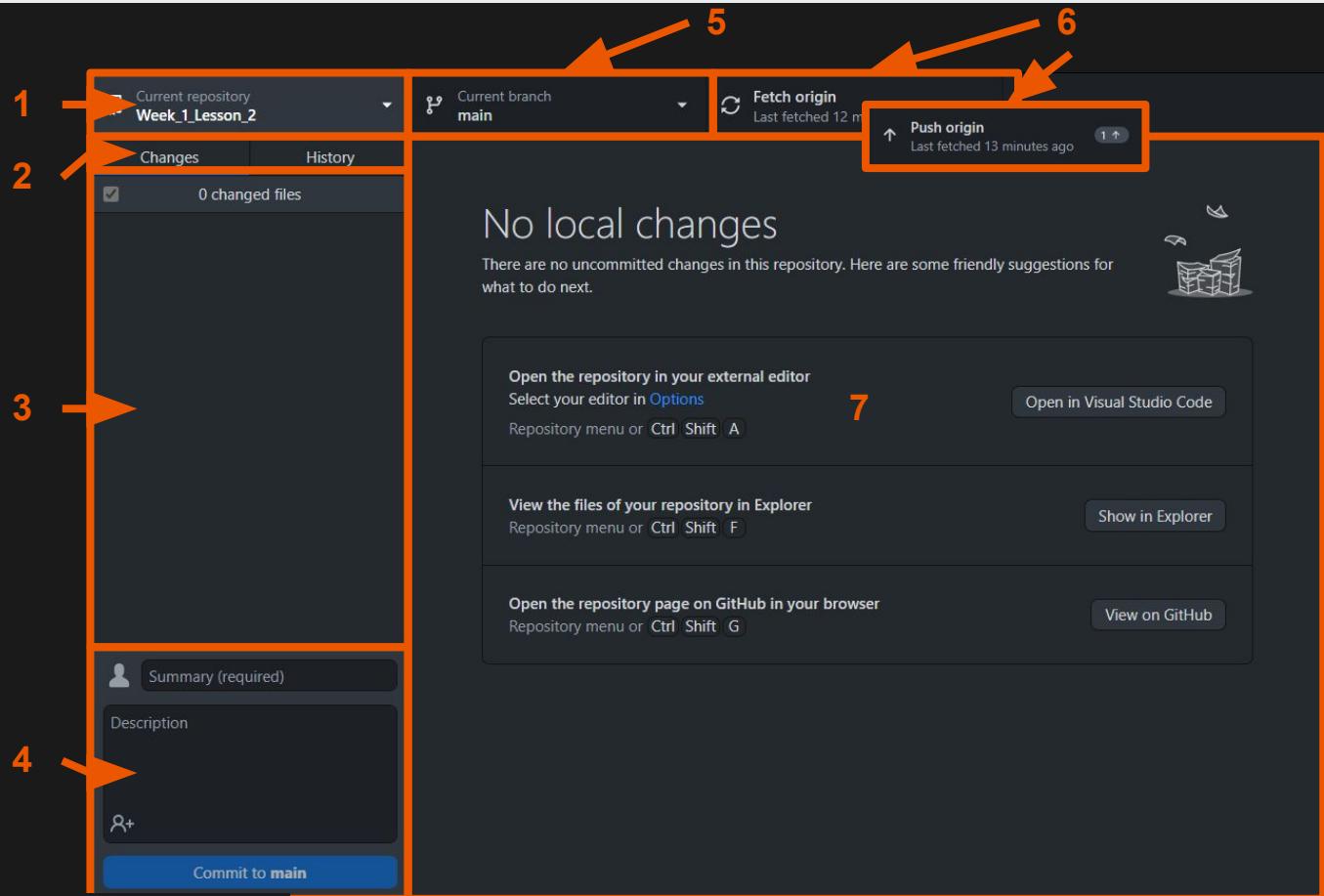


Version Control System Commands In Action

Now the GitHub Desktop will look a lot different. Let's break down the sections.

[1] Project name, clicking on it will allow you to switch to a different project.

[2] Changes and History Tabs, allows to change what's shown in [3] either all the files that are changed or the history of Pushes the Repository has received.



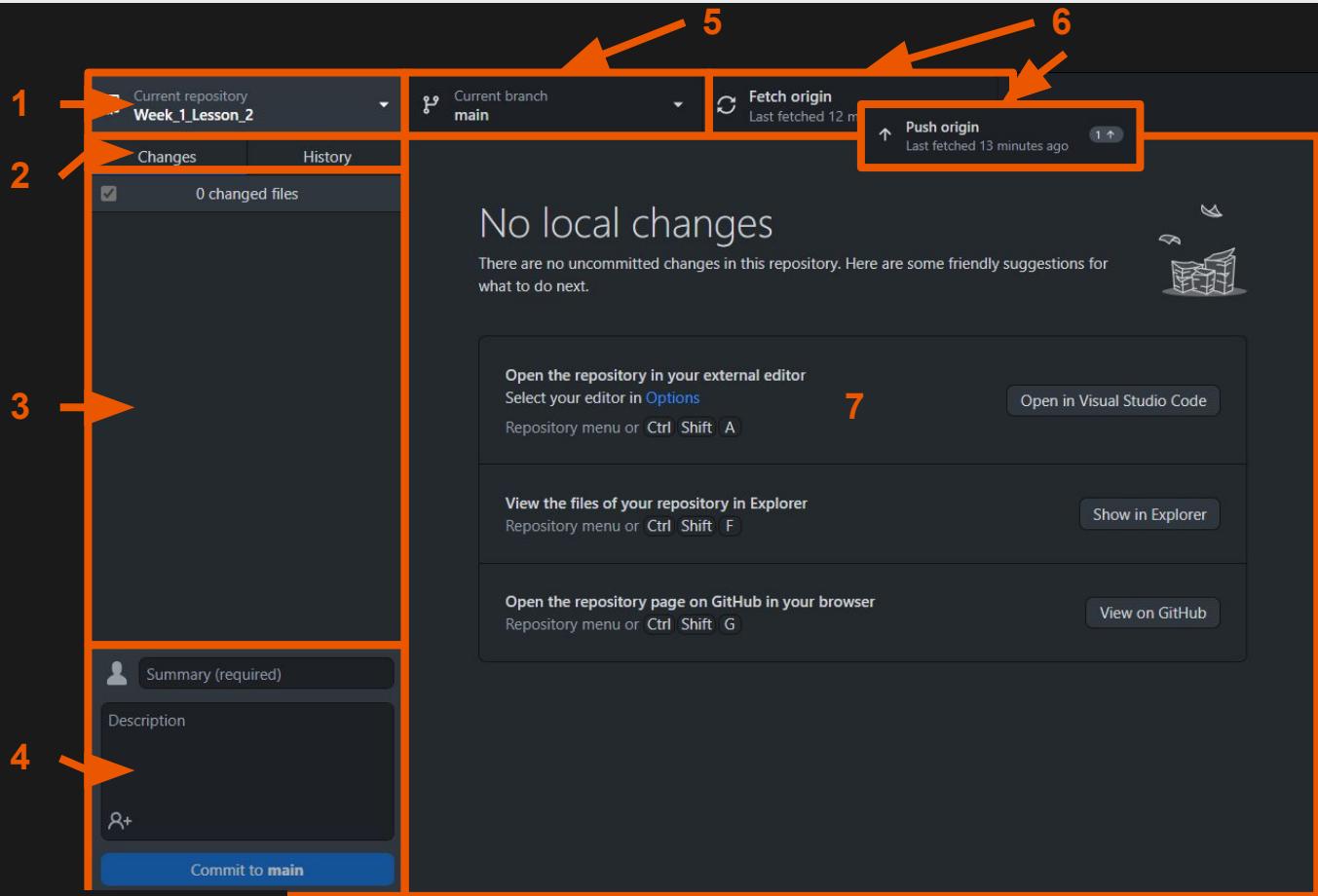
Version Control System Commands In Action

[4] Allows you to Commit or snapshot your changes, you have to provide it with a Title and a Description so that when you look back on history you know what you were doing with the given changes.

[5] Tells you what branch you're on, most of the time it will be main and you won't need to change it.

[6] Tells you last time the project was updated and if you have a commit ready will allow you to push it to the cloud.

[7] Will display the changes line by line of the file currently selected in [3].

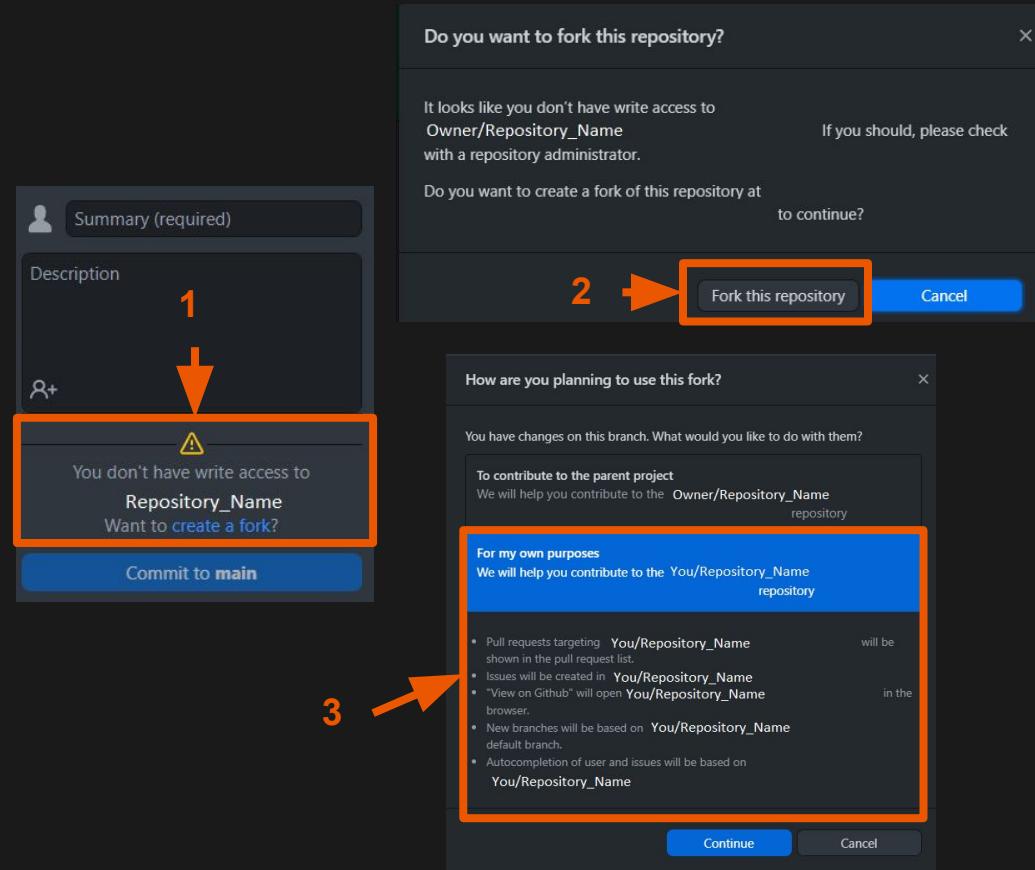


Forking Your Own Copy

Because the Project Files that were downloaded are my files, you won't be able to just Commit your changes to my copy of the project.

To make your own Copy of the project we will Fork it. GitHub Desktop will put a [1] warning on Commit telling you that you don't have access to the Repository and you should Fork.

If so click [1] Create a Fork, then in the PopUp menu select [2] Fork this Repository and select [3] For My Own purposes, that will create a distinct copy that only you will be able to edit while the first option will fork off of mine branch.

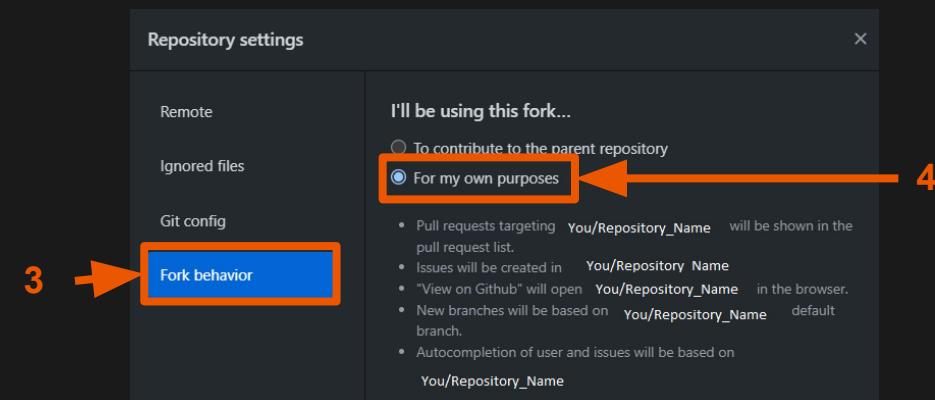
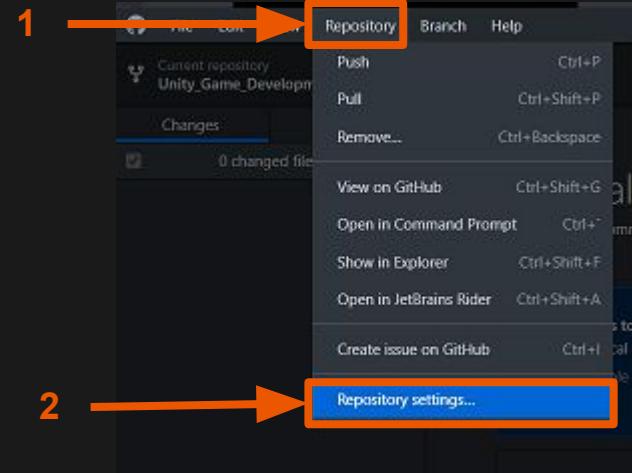


Forking Your Own Copy

Incase that Warning Sign doesn't show up in the Commit Area, you can go to the Toolbar select [1] Respiratory and go to [2] Repostiary Settinging.

This will bring up the PopUp go to [3] Fork Behavior and Select [4] For My Own Purposes. After that click Save.

This will have the same effect as if the Warning PopUp giving you a copy of the Repository to commit and Push with your own changes.



.gitignore File

When creating a repository GitHub it will ask if there is any files that shouldn't be tracked. For Unity you definitely want to do that.

If you don't select .gitignore template: Unity, GitHub will keep track of thousands of files that are used during the execution of the program that don't need to be uploaded.

If you didn't selected it you can alway just add a text file called .gitignore with the code to the right and save it in the project folder.

Add .gitignore

Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: Unity ▾

```
# This .gitignore file should be placed at the root of your Unity project directory
#
# Get latest from https://github.com/github/gitignore/blob/master/Unity.gitignore
#
/Libraries/
/Ittemp/
/Oobj/
/BBuildId/
/BBuilds/
/LLogs/
/MemoryCaptures/

# Asset meta data should only be ignored when the corresponding asset is also ignored
!/[A]sets/**/.meta

# Uncomment this line if you wish to ignore the asset store tools plugin
# /[A]sets/AssetStoreTools*

# Autogenerated Jetbrains Rider plugin
[A]sets/Plugins/Editor/JetBrains*

# Visual Studio cache directory
.vs/

# Gradle cache directory
.gradle/

# Autogenerated VS/MD/Consulo solution and project files
ExportedObj/
.consulo/
*.csproj
*.unityproj
*.sln
*.suo
*.tmp
*.user
*.userprefs
*.pidb
*.booproj
*.svd
*.pdb
*.mdb
*.opendb
*.VC.db

# Unity3D generated meta files
*.pidb.meta
*.pdb.meta
*.mdb.meta

# Unity3D generated file on crash reports
sysinfo.txt

# Builds
*.apk
*.unitypackage

# Crashlytics generated file
crashlytics-build.properties
```

[Browse](#)[Summer Sale 2022](#)[Developer Logs](#)[Jams](#)[Dashboard](#)[Feed](#)[Community](#)Search for games or creators

Orczuk



POPULAR TAGS

Horror games	Multiplayer
Visual novels	HTML5 games
Simulation	macOS games
Roguelike	Linux games
Browse all tags →	

BROWSE

Games	Game assets
Tools	Albums & soundtracks
Physical games	Comics
Books	Randomizer

GAMES BY PRICE

[Download app](#)

Get the most out of itch.io! Install and manage your games with our desktop app

[Baba Is You XTREME \(Playthrough\) Slip, ...](#)

Baba Is You XTREME

Meet Baba in this new adrenaline-filled puzzler!

**FREE**[Get the eldritch artifact →](#)[See more on itch.io's YouTube](#)

Itch.io is a website where developers of all skill and talent converge to post their creations and participate in game jams, competitions that last in most cases a weekend or a week which is a similar to what the final project will look like. Many famous indie games get their start here as game jam entries.

This is a great place to see what your final project could look like.

Titles To Try



[Paddle Panic](#)



[Dark Snow](#)



[Growing Up](#)



[Project Catalyst](#)



[Mismatch Patch](#)



[Snakebird NES](#)

Challenge: Itch.io Games



Get into groups and discuss the games you've tried out and discuss:

What games did you find while looking through Itch.io?

Was there anything that looked like something that fits the game you want to create?

What did you like about the games you played?

What did you dislike about the games you played? Think about how you would have fixed it if you were working on the project.