



Game Design and Development

Who am I?

My name is Sebastian Grygorczuk!
I've been developing games for
over 3 years and currently I'm
working on a game called
CLAWArena with a company called
Mocha Chili, where I act as a
gameplay programmer while I
teach introductory game design
classes for the STEM Institute at
CCNY.

My interests outside of gaming are
robotics, hiking, and history.



LibGDX/Java Projects



I began my journey in game development when a friend approached me to create assets for his game. At that time, I already had some experience in crafting pixel art animations.

Encouraged by this initial collaboration, I decided to venture into independent game development. To get started, I chose to work with the same framework my friend had used—LibGDX, which employs Java as its programming language. LibGDX is a versatile collection of programming libraries that enable image processing, sound integration, and code packaging.



Unity/C# Projects

After two years of working with LibGDX, I made the transition to Unity.

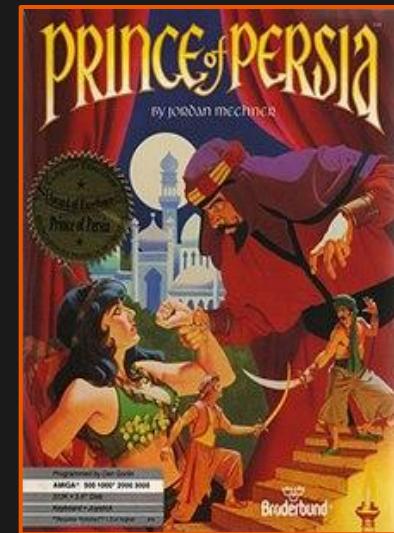
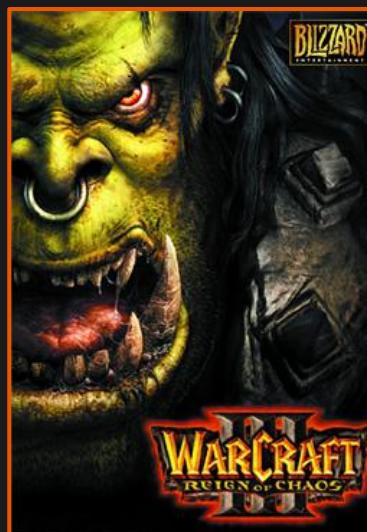
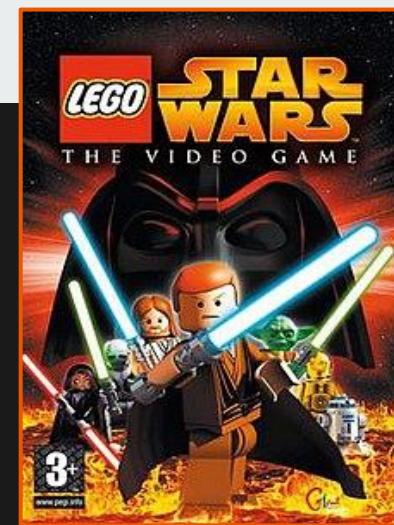
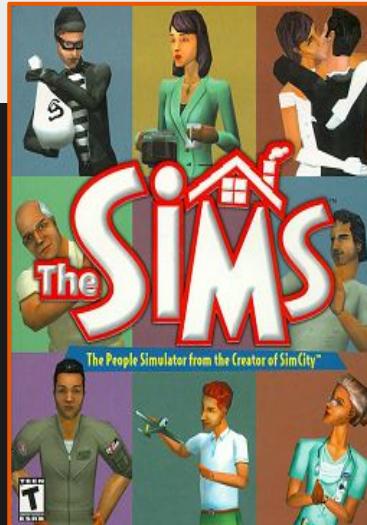
The primary reason for this shift was that LibGDX is not a widely-used engine in the industry. To secure opportunities in the game development field, it's essential to familiarize yourself with the tools that most companies employ.

Unity, in contrast, offers a fully integrated system with a user-friendly UI and external tools readily available. In LibGDX, creating such features would have required building them from scratch.



Why I Make Video Games?

I grew up playing games, whether it was solo, with my family, friends, or even strangers. Video games provided a simple way to connect with people, sharing diverse experiences. With the vast array of game genres available, there's something for everyone. I aspire to facilitate connections through video games





Software

Perks of Unity



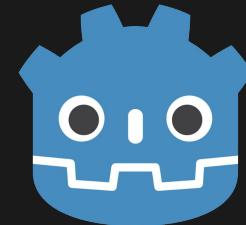
- Unity enables you to port your game to a wide range of platforms, making executable files available for various operating systems, including Windows, Mac, Linux, web browsers, Android, iOS, AR, VR, and popular gaming consoles like Playstation, Xbox, and the Nintendo Switch.
- Unity offers free software for personal use, provided your projects don't generate over \$100,000 in revenue annually.
- Unity is integrated with the Unity Asset Store, which hosts a wealth of both free and purchasable assets.

Other Game Engines

Unreal: an industry-standard 2D/3D game engine that uses C++, is freely available and is currently used for the development of games like Witcher 4



GoDot: a versatile engine for creating 2D and 3D environments and levels, supports programming with C# and C++ as well as visual scripting. It's also free to use.



Construct 3: a web-based 2D game engine with visual scripting, but it requires licensing for all of its tools.



There are dozens of game engines, each with its unique features. However, many of them use standardized tools, making it easier to transfer your skills from one engine to another.

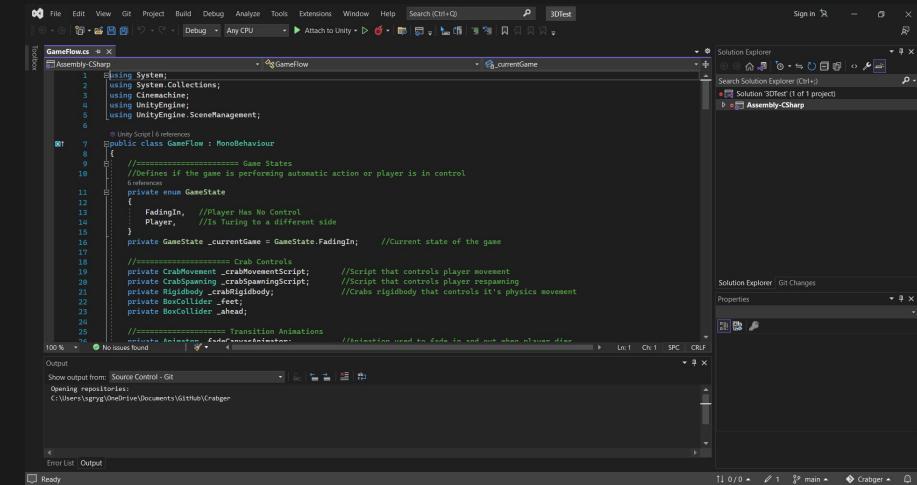
Additional Resources: [Most used Engines](#) [The Best Game Engines of 2021](#)

Visual Studio

An Integrated Development Environment (IDE) is a software application that provides a comprehensive and unified environment for software developers to write, edit, and debug code. It typically includes features such as a code editor, a debugger, a compiler or interpreter, and tools for managing projects and dependencies.

In its simplest form, an IDE can be thought of as a supercharged text editor, as it goes beyond basic text editing by offering various features that streamline the software development process, making it more efficient and productive for programmers.

Visual Studio is a well-known IDE that allows developers to write C# scripts, especially for game development, and it provides the tools necessary to create and manage game objects and their interactions.



```
GameFlow.cs  Assembly-CSharp
1 //using System;
2 //using System.Collections;
3 //using System.Linq;
4 //using UnityEngine;
5 //using UnityEngine.SceneManagement;
6
7 // Unity Script References
8 public class GameFlow : MonoBehaviour
9 {
10     ////////////////////////////////////////////////////////////////// Game States
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         Playing,           //Is Turning to a different side
16         FadingOut
17     }
18
19     private GameState _currentGameState = GameState.FadingIn; //Current state of the game
20
21     ////////////////////////////////////////////////////////////////// Crab Controls
22     private CrabMovement _crabMovement; //Script that controls player movement
23     private CrabSpawning _crabSpawningScript; //Script that controls player respawning
24     private Rigidbody _crabRigidbody; //Crabs rigidbody that controls its physics movement
25     private BoxCollider _feet; //Crabs feet
26     private BoxCollider _head; //Crabs head
27
28     ////////////////////////////////////////////////////////////////// Transition Animations
29     //Animations used to fade in and out when player dies
30 }
```

IDE vs Text Editor

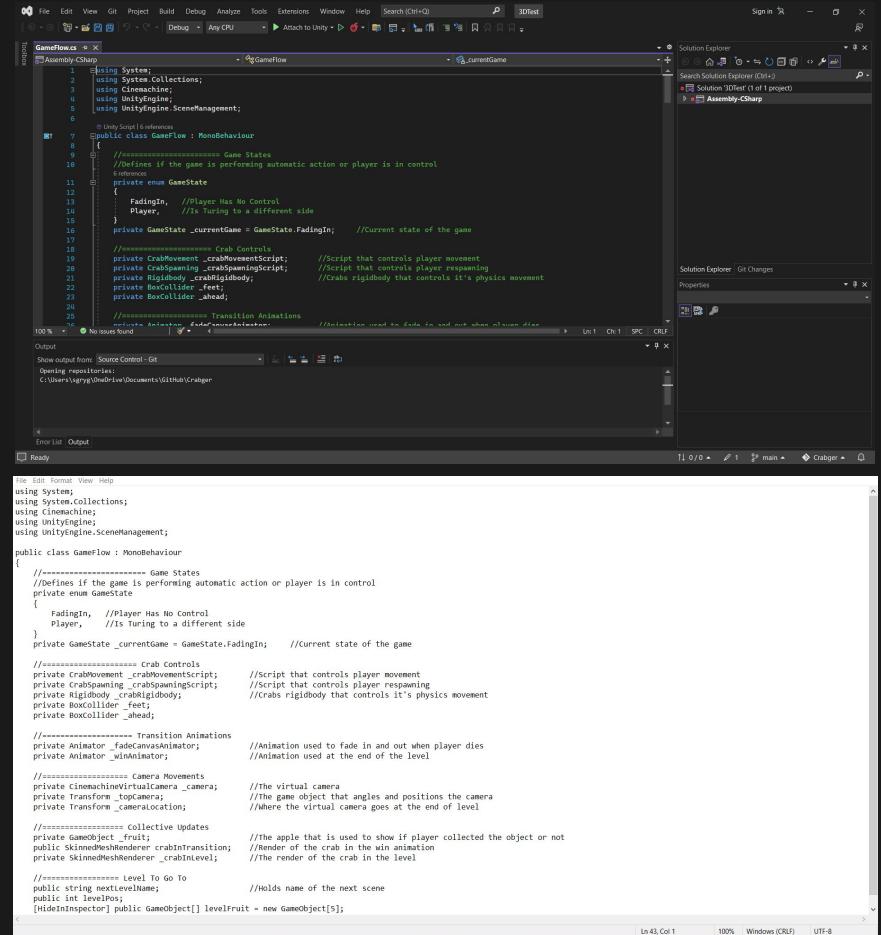
When comparing code written in Visual Studio with that in Windows Notepad, both tools serve the fundamental purpose of allowing you to write code. However, Visual Studio offers several advantages:

First and foremost, it employs syntax highlighting, which color-codes your code based on the meaning of each word. This feature enhances code readability and helps you spot errors more easily.

Secondly, Visual Studio also provides a line counter, an essential tool for debugging your program as it helps pinpoint issues in your code.

In addition, one of the most practical features of Visual Studio is auto-completion, which suggests and completes words or methods as you type. This feature significantly speeds up coding and reduces the chance of syntax errors.

Finally , Visual Studio is an ideal choice for working with C# and Unity because it seamlessly integrates with these technologies. Different Integrated Development Environments (IDEs) are often optimized for specific programming languages, making Visual Studio an excellent choice for C# development.



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

- Solution Explorer:** Shows a single project named "Solution 3DTest" containing an "Assembly-CSharp" file.
- Code Editor:** Displays the "Gameflow.cs" script. The code is as follows:

```
using System;
using System.Collections;
using Cinemachine;
using UnityEngine;
using UnityEngine.SceneManagement;
using UnityEngine.SceneManagementManagement;

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 Turing 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 respawning
    private Rigidbody _rigidbody; //Rigidbody that controls its physics movement
    private BoxCollider _feet;
    private BoxCollider _head;

    //===== Transition Animations
    private Animator _fadedCanvasAnimator; //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 SkinnerMeshRenderer crabInTransition; //Holder of the crab in the win animation
    private SkinnerMeshRenderer _crabInLevel; //The render of the crab in the level

    //===== Level To Go
    public string nextSceneName; //Name of the next scene
    public int levelsPos; //HideInInspector public GameObject[] levelFruit = new GameObject[5];
}
```

The status bar at the bottom indicates "Ln 43, Col 1" and "100% Windows (CR LF) UTF-8".

Possible Alternative - Jetbrains Rider

A worthwhile alternative for development across Windows, Mac, and Linux is JetBrains Rider. It offers a more extensive set of features and boasts a user-friendly interface. I personally use it for game development. However, it's worth noting that Rider is not free and requires licensing.

Unless you plan to engage in game development for an extended period and can justify the expense, I would hesitate to recommend purchasing it

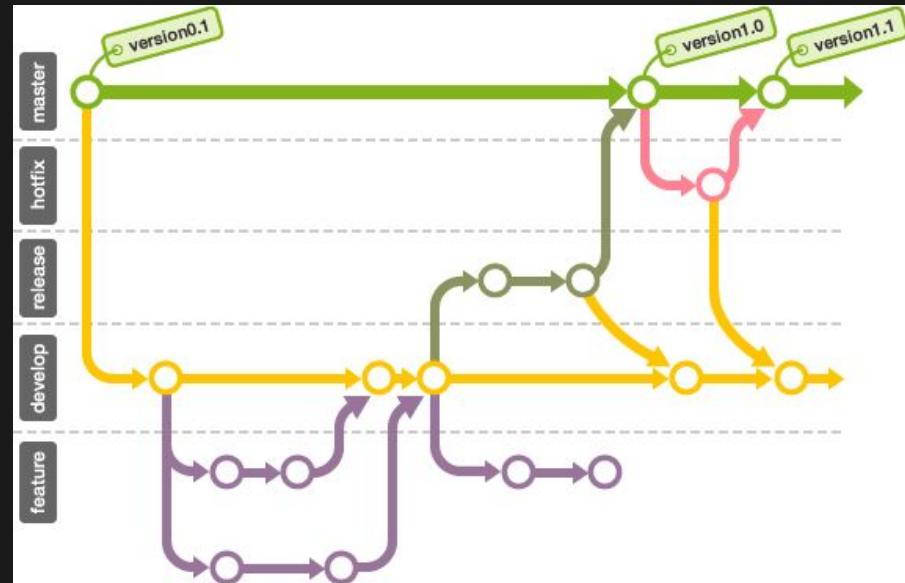
Sebastian Grygorczuk

GitHub - Version Control System

A Version Control System (VCS) is a method for storing and updating project files, whether for individual or group projects. Think of it like a shared Google Doc for your code, where VCS allows you to actively collaborate while keeping a record of the history of changes. It also permits you to roll back to previous points in time.

VCS is especially valuable when multiple people are working on a project. In cases where individuals edit the same parts of the project, VCS identifies conflicts and allows you to resolve them by comparing and choosing the best code changes, line by line.

GitHub, on the other hand, is a hosted Version Control System. It's akin to having all your project files stored in a cloud-based repository, similar to Google Docs but designed specifically for code.



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



Unity Is Everywhere

Games Created With Unity

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

Unity is primarily associated with game development, 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!

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.](#)

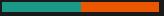
Non Game Related Software With Unity

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 engineers, architects, and medical professionals to see what they are constructing or working on while simulating the physics of their prototype.



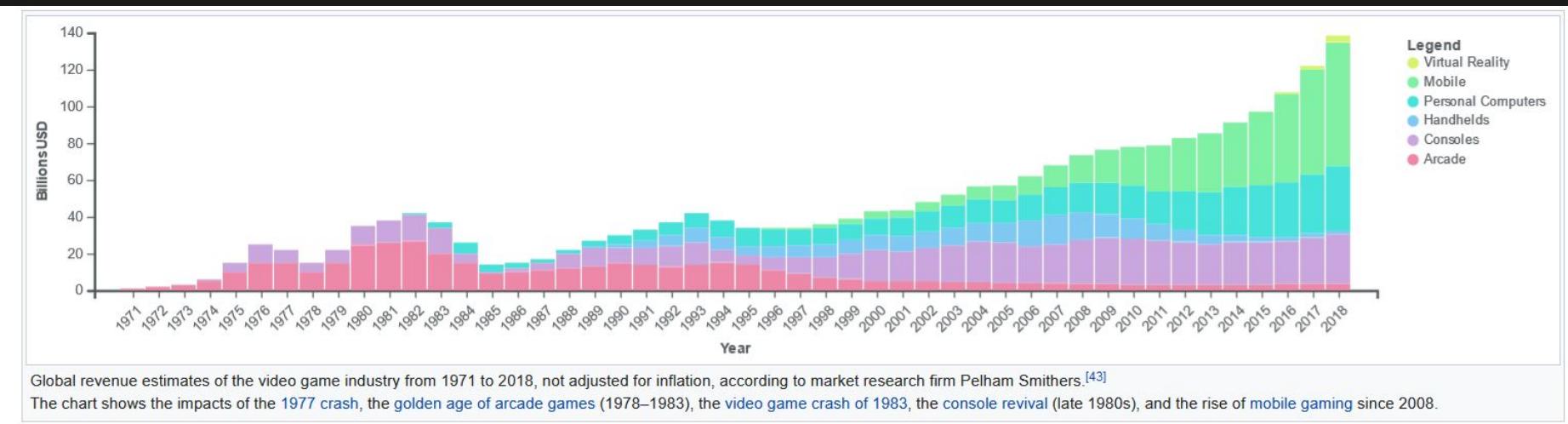
If you're interested in checking out software made for [automobiles](#), [engineering](#) or [aerospace](#) using Unity.



Working In Games Industry

Sebastian Grygorczuk

Video Game Industry



The video game industry is a multi-billion-dollar powerhouse that rivals traditional media industries like film, music, and books. It boasts a large number of publishers and game development studios, which constantly releasing new games daily. With the industry's continuous growth, there's a place for aspiring professionals who genuinely want to pursue a career in it.

Additional Resources: [Chart](#)

Sebastian Grygorczuk

What is a Game Developer?

Game development, unlike most software development, doesn't exclusively refer to programming. Games comprise a multitude of elements, including code, art, animations, sound, and more.

A game developer encompasses various roles responsible for creating these assets. These roles include game designers, programmers, writers, 2D and 3D artists, animators, sound engineers, and testers.

Each of these disciplines can further specialize, and throughout this presentation, You will experience each of these roles. The size of the development team can vary, and individuals may take on one or multiple roles, depending on the project's scope. So, be prepared to wear many hats in the world of game development.

Additional Resource: [The Big List of: Video Game Development Team](#)



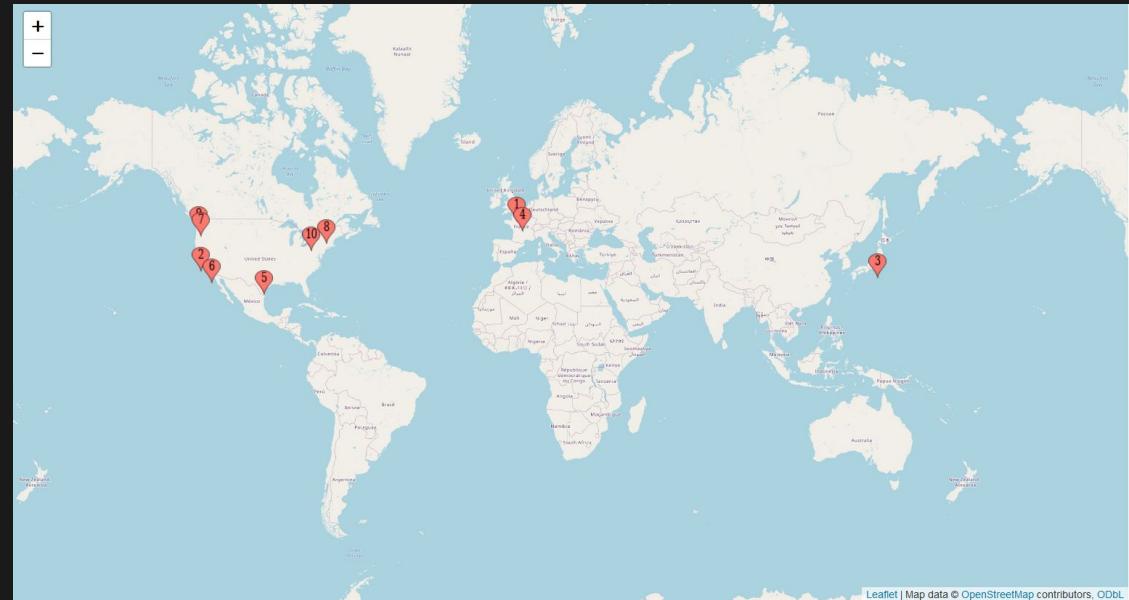
Where Game Studios Exist

Game Development Studios are abound in many place however, New York City is not one of them.

If you'd like work at a large studio you will have to move to one of these cities.

[Sources] Remote work is opening up as a possibility but on location is the preference for larger companies.

1. London
2. San Francisco
3. Tokyo
4. Paris
5. Austin
6. Los Angeles
7. Seattle
8. Montreal
9. Vancouver
10. Toronto



Leaflet | Map data © OpenStreetMap contributors. ODbL

Game Designer

The game designer serves as the creative lead of the project, responsible for establishing the framework of how and why the game will function. As a project expands, the role of the game designer may be divided into two distinct roles: the level designer and the system designer.

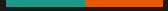
Level designers take the assets created by programmers, artists, and sound engineers, and use them to construct the actual game levels that players engage with.

System designers, on the other hand, work on a broader scale, creating the systems that govern various aspects of the game, such as enemy behavior and questing.

Game designers are often the most recognizable figures in game development, with notable figures like Shigeru Miyamoto leading the way.



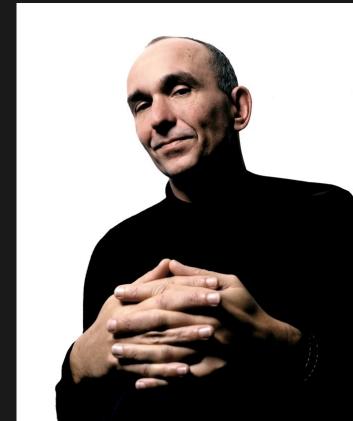
Examples of Game Designers


Shigeru Miyamoto, a legendary designer, is the mastermind behind beloved franchises like the original Zelda, Donkey Kong, and Super Mario Bros. He played a pivotal role in revitalizing the video game industry. Today, he holds the position of General Manager at Nintendo.

Sid Meier is renowned as the creator of the Civilization game series, which successfully translated the appeal of board games to computer screens. He currently serves as the Director of Creative Development at Firaxis Games.

Hideo Kojima, the mastermind behind the Metal Gear Solid series, is credited with popularizing the stealth genre in gaming. He now serves as the CEO of Kojima Productions.

Peter Molyneux is the visionary behind games like Populous, Dungeon Keeper, Black & White, and the Fable RPG series, pioneering the genre of god games. He presently holds the position of CEO at 22cans.



Game Programmer

Game programmers are the backbone of game development, responsible for making the game function and giving life to the assets created by artists. However, the role of a game programmer can vary widely depending on the project's scope. Programmers can specialize in various sub-roles, including:

Programming roles can break down into many sub roles:

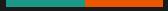
- [Gameplay Programmer role](#)
- [Backend / Server / Online Programmer role](#)
- [AI Programmer role](#) [Graphics Programmer role](#)
- [UI Programmer role](#)
- [Animation Programmer role](#)
- [Physics Programmer role](#)
- [VFX Programmer role](#)
- [Audio Programmer role](#)
- [Tools Programmer role](#)

These sub-roles handle specific aspects of game development. Game programmers, while often less recognized than designers, have made significant contributions to the industry. You may still recognize names like [John Carmack](#) among these talented individuals.



Sebastian Grygorczuk

Examples of Game Programmers


[John Carmack](#) is credited with shaping the current state of the first-person shooter (FPS) genre. He programmed iconic titles like Wolfenstein 3D, DOOM, and Quake, which laid the foundations for modern FPS shooters like Call of Duty and Overwatch. He currently holds the position of Chief Technology Officer at Oculus VR.



[Tim Schafer](#) is a programmer who played a pivotal role in leading point-and-click adventure games through their golden age. His work on franchises like Secrets of Monkey Island and Grim Fandango is highly regarded. Schafer now serves as the CEO of Double Fine.



[Tim Sweeney](#) is an engine developer responsible for creating the original Unreal Engine, a game engine that played a significant role in transitioning FPS games from 2D to full 3D environments. He is currently the CEO of Epic Games.



[Yuji Naka](#), known for his work as a programmer on the original Sonic Trilogy, helped SEGA compete with Nintendo's Mario franchise. He now holds the position of CEO at Prope.

Game Artist

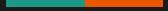
Game Artist can be broken down into some many roles; concept artist, sprite artists, texture artist, 3D modelers, riggers, animators, lighting. The specific roles a game artist undertakes depend on the project's required fidelity and complexity. In some cases, a single person may dedicate years to crafting a single item or aspect of the game's visual elements.

An example of that would be the cape in Batman Arkham Asylum where "there was one person working on nothing but the cape for two years, so there are over 700 animations and sound clips attached to the cape alone. That's why it looks so beautifully realistic." [\[Source\]](#)

Despite the crucial role of conveying the game's essence to players, game artists often receive less recognition for their hard work. An example of a recognized artist in the game industry is [Tetsuya Nomura](#).



Examples of Game Artists


[Tetsuya Nomura](#) played a significant role as an artist on most of the modern Final Fantasy games. He is widely recognized for his iconic character designs, including Cloud and Sora from the Kingdom Hearts series. Today, he holds the position of Game Director at Square Enix.

[Josh Scherr](#) served as the Lead Cinematic Animator for the first three Uncharted games. These games introduced cinematic storytelling and blockbuster experiences that were groundbreaking at the time and continue to set a high standard in the industry. Currently, he is a writer at Naughty Dog.[\[Example of Work\]](#)

[Keiji Inoue](#) is a Visual Effects Artist known for contributing to games like Pikmin, Wii Sports, Super Mario Odyssey, and The Legend of Zelda: Breath of the Wild. Currently, he holds the position of Lead Effects Artist at Nintendo.[\[Example of Work\]](#)

[Jamie McNulty](#) served as the Environment Artist for notable games like BioShock, BioShock Infinite, and Gears of War 4 and 5. Presently, McNulty works as an Environment Artist at Deviation Games. [\[Example of Work\]](#)



Sound Designers

Sound designers, much like artists, are often underappreciated in the game development process. Regardless of how engaging the game's visuals may be, without proper sound effects to provide feedback to the player's actions or music to set the tone, the gaming experience may fall short of enjoyment.

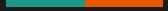
Sound designers also collaborate with voice actors, contributing to the overall immersive experience

Some sounds and songs are synonymous with the games they were made for. One of my personal favorite is Stormwind Theme by Jayson Hayes.



Sebastian Grygorczuk

Examples of Sound Designers


Jason Hayes is a veteran composer creating pieces for *Starcraft*, *Warcraft* and *Diablo* and is continuing his work at Blizzard Entertainment. [[Example of Work](#)]



Kōji Kondō is a veteran composer known for creating music for iconic franchises like Legend of Zelda and Mario. He continues his work at Nintendo as a composer. [[Example of Work](#)]



Masato Nakamura was the sound producer of *Sonic the Hedgehog 1* and *2* soundtracks and is most likely responsible for the famous Sonic Ring Sound Effect. Is a bassist for the band Dream Come True but frequently works on Sound Design in games.



Phillip Kovats held the role of sound director for projects like *God of War*, *The Last of Us*, *Uncharted*, and many other Sony titles. He was responsible for how the games sound and for mixing the voice acting. Currently, he holds the position of Senior Director of Sound at PlayStation Studios. [[Example of Work](#)].

Sebastian Grygorczuk

Publisher and Game Development Studio

Let's get a few more definitions and relationships out of the way:

A Game Development Studio is the company responsible for planning and producing the game. While not all studios need to use publishers, most require financial and commercial backing. The studio comprises various individuals, including those involved in finance, advertising, legal, and more. At its core, developers are the ones primarily responsible for creating the game.

A Publisher is the company that provides funds to the game development studio to create the game. Game development can be an expensive and time-consuming process, involving numerous individuals and financial resources. Publishers front-load the costs of game development, often through agreements that guarantee them a return on their investment. As a result, they often have significant influence over the development and direction of the game.

Welcome to Xbox Game Studios

Our 23 game development studios, now including the studios under Bethesda Softworks, focus on delivering great games for everyone, wherever they play – on console, PC, or mobile devices. We're responsible for developing and publishing some of the biggest franchises in history: Age of Empires, Forza, Gears of War, Halo, Minecraft, Fallout, Microsoft Solitaire, Microsoft Flight Simulator, DOOM, The Elder Scrolls, and many more. We believe that play is the thing that unites everyone, because when everyone plays, we all win.



NINJA THEORY

OBSIDIAN

Playground Games



UNDEAD LABS

Bethesda Softworks



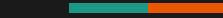
Tango Gameworks



ARKANE



Indie Developers



Let's give honorable mentions to some inspiring **Indie Game Developers** that take all the talents of a Game Development Studio and perform them all by themselves.

Toby Fox, is a breakout star, serving as a game designer, programmer, artist, and sound designer, creating *Undertale* as a one-man team

Supergiant Games, a small but highly creative team, has crafted hits like *Bastion*, *Transistor*, and *Hades*. They are known for their beautiful art, music, and innovative storytelling approaches.

Scott Cawthon is the creator of the massive franchise *Five Nights at Freddy's*, which has evolved into a multimedia empire.

If you're interested in gaining insights into the world of indie game development, you can watch '[Developing Hell](#)' a documentary that provides a behind-the-scenes look at how Supergiant Games created their latest game, *Hades*.





Organizations

Playcrafting & Play NYC

Playcrafting is a local game development community that serves as a connection between game developers and publishers. They also organize annual indie game development conventions known as Play NYC.

To get involved, consider joining their Discord server where they host various events throughout the year, which they will announce in the community.

You can also attend Play NYC as a guest by purchasing a badge to see what's happening. Alternatively, if you're part of the Discord community, you can volunteer.

This is an excellent opportunity to meet indie game developers who are actively working on their next games.

<https://playcrafting.com/>

<https://play-nyc.com/>



NYU Game Center

The NYU Game Center, part of New York University's renowned Tisch School of the Arts, is the Department of Game Design. They offer a diverse range of courses in game design, development, and game studies.

Various study options are available, including a BFA in Game Design, an MFA in Game Design, an undergraduate minor, and numerous courses open to all NYU students. You can find more information about their academic programs and courses [Click here to learn more about our academic programs and courses.](#)

While the program is costly due to it being a private college, the NYU Game Center remains accessible to the public. They frequently host free lectures featuring industry professionals who share their game development experiences. Additionally, they organize playtesting sessions, allowing individuals to showcase their games or try out others.



Follow their social media and join the mailing list to see upcoming events and play test sessions.

<https://gamecenter.nyu.edu/follow/>

GUMBO



The EGD Collective is a grassroots movement spanning across North America with the mission of aiding every student in completing their education and finding their path within the gaming industry.

This community originally originated at Hunter College and has grown into a cross-college network where students help each other break into the gaming industry.

Earlier this year, they organized a game convention called Waffle Games, featuring speakers from the NYC game industry. The event also provided students and indie developers with the opportunity to showcase their games.



Follow their social media join their mail list or discord to keep track of events they host.
<https://www.egdcollective.org/community>

GUMBO

The Gumbo Collective is a community of local indie game studios that collaborate to share a workspace in Brooklyn. At this shared space, individuals can rent a desk, providing them with a dedicated place to work.

The Gumbo Collective also organizes monthly playtest sessions for both video games and board games. It's an excellent opportunity to connect with other indie game developers and receive valuable feedback on your work.

<https://gumbonyc.org/>



The group uses MeetUp app to post their upcoming events.

<https://www.meetup.com/gumbo-collective/>



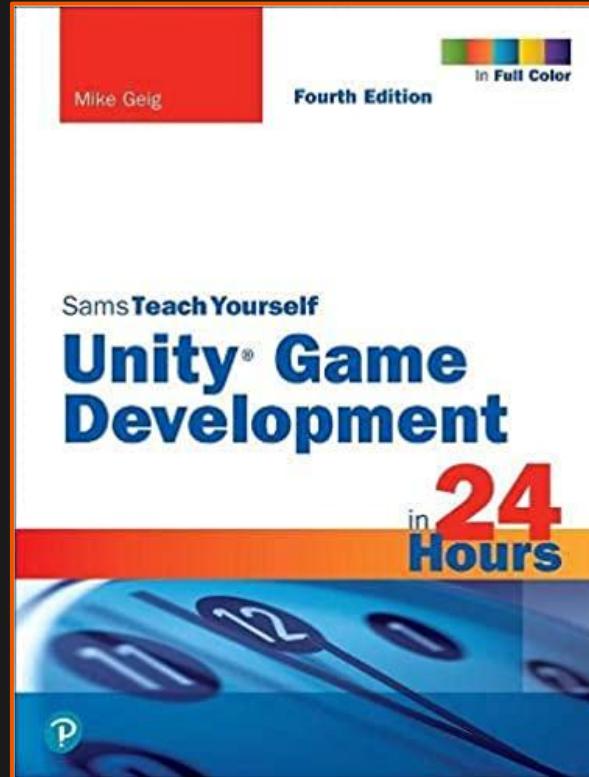
Resources

Unity Game Development in 24 Hours 4th Edition by Mike Geig



Unity Game Development in 24 Hours
4th Edition is an ideal textbook to use to learn game development because it's an introduction to Unity and many of its designer focused tools with programming included, but not programmer focused.

This is the book I used as a basis for my class curriculum when I teach game development. It can be simple and you will need to look elsewhere for deeper looks into subjects presented but it's a great entry point.



Unity Pathways

Unity has its own coursework that's free and it will teach you all of the aspects of Unity.

Studying the Unity Essentials and Junior Programmer will give you more than enough knowledge to start building any game.

<https://learn.unity.com/learn/pathways>



Unity Essentials

2 weeks • Foundational +600 XP

New to Unity? This guided learning Pathway includes everything you need to get started.

[Unity Essentials Pathway ↗](#)



Junior Programmer

12 weeks • Foundational +3000 XP

Ready to code? This guided learning Pathway will take you from zero to job-ready!

[Junior Programmer Pathway ↗](#)



Creative Core

10 weeks • Beginner +3000 XP

Ready for more? Level up your core understanding of Unity with the creative aspects of the engine.

[Creative Core Pathway ↗](#)



VR Development

6 weeks • Beginner +600 XP

Ready to develop for VR? This guided learning Pathway will prepare you for a job in the VR industry!

[VR Development Pathway ↗](#)



Mobile AR Development

8 weeks • Foundational +600 XP

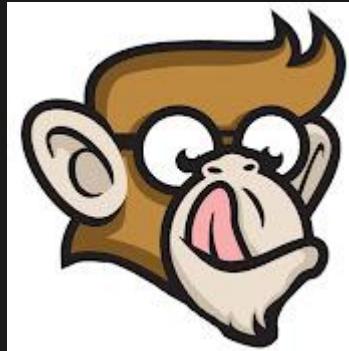
Ready to create AR experiences? In this learning pathway, you'll develop AR apps compatible with iOS and Android devices!

[Mobile AR Development Pathway ↗](#)

YouTube

Outside of the traditional ways of learning I highly suggest using YouTube, there's a lot of independent game developers documenting their journeys and sharing tips and tricks they've learned on their paths.

Here are some of the ones I recommend for learning but always feel free to look for more.



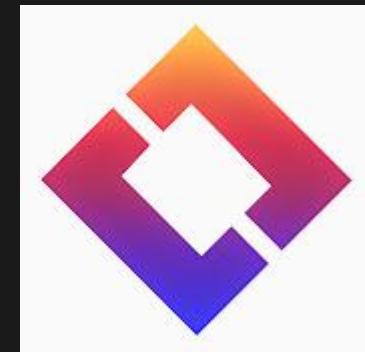
[Code Monkey](#)



[Blackthornprod](#)



[Tarodev](#)



[Brackeys](#)



Get Involved

itch.io & Game Jams

itch.io is a website that hosts independent games, assets, and game jams. You can try out games in the browser and upload your own for the world to see.

Joining game jams is a great way to test your skills and put pressure on yourself to finish a project.

These jams can last from as short as one hour to several months, but I would recommend participating in one that lasts a week or two.

The screenshot shows the itch.io homepage. At the top, there's a navigation bar with links for Browse, Halloween Sale 2023, Developer Logs, Jams, Dashboard, Feed, and Community. A search bar is located on the right side of the header. Below the header, there's a sidebar titled "FILTER RESULTS" with dropdown menus for Platform (Windows, macOS, Linux, Android, iOS, Web), Price (Free, On Sale, Paid, \$5 or less, \$15 or less), and When (Last Day, Last 7 days, Last 30 days). The main content area displays search results for "Games" (834,387 results) sorted by "Popular". The results include five game cards: "THE LOST FEAR" (Rising Wolf Games, Adventure), "TEKE TEKE: Moonlit Dread" (Med, Survival), "The Final Pin" (JordiBoi, Adventure), "Fears to Fathom - Ironbark Lookout" (\$5.99, Rayll, Adventure), and "The Stalked" (GIF). Below these cards, there are more game cards partially visible: "THE SHOPPING" and "BROKEN THROUGH".

<https://itch.io/>

Game Jam Advice

1. Make sure the game is accessible. People may not want to download your game, so make it playable in a browser. Unity allows you to create a WebGL version of the game; use it.
2. For your screenshots, use GIFs. They're an effective way to capture the attention of anyone scrolling by.
3. Interact with others. If you want people to try your game, be willing to try other people's games as well. Comment on and rate their games, and they will likely return the favor.
4. Lastly, enjoy yourself. Try to have a good time. By the end of this, you will have a game, new ideas and skills, and perhaps even some new friends.

Footnotes [View game page →](#)

We're all just footnotes in other's stories

Submitted by [Orczuk \(@_Orczuk_\)](#) — 6 hours, 56 minutes before the deadline

[Edit this submission](#)

[Remove this submission](#)



[Play game](#)

[Footnotes's itch.io page ↗](#)

Results

Criteria	Rank	Score*	Raw Score
Theme	#7	4.156	4.156
Game Design	#31	3.875	3.875
Innovation	#49	3.656	3.656
Overall	#49	3.635	3.635
Audio	#79	3.500	3.500
Graphics	#191	3.438	3.438
Fun	#196	3.188	3.188

Ranked from 32 ratings. Score is adjusted from raw

► Rating Distribution

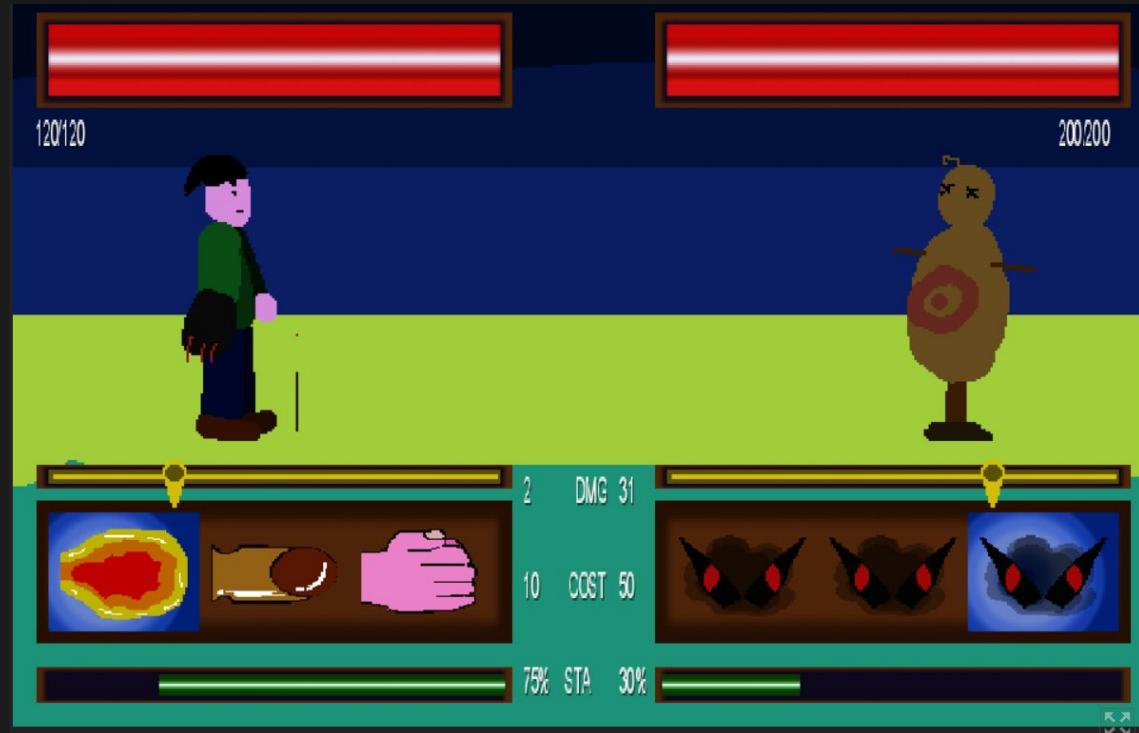
"Your first ten games will suck — so get them out of the way fast."

Don't wait until you know everything.

Don't aim for perfection in your first game.

Just go out there and make your first game.

Once you create one, the next one will be easier, and so on. Keep making games, and each one will be better than the last.

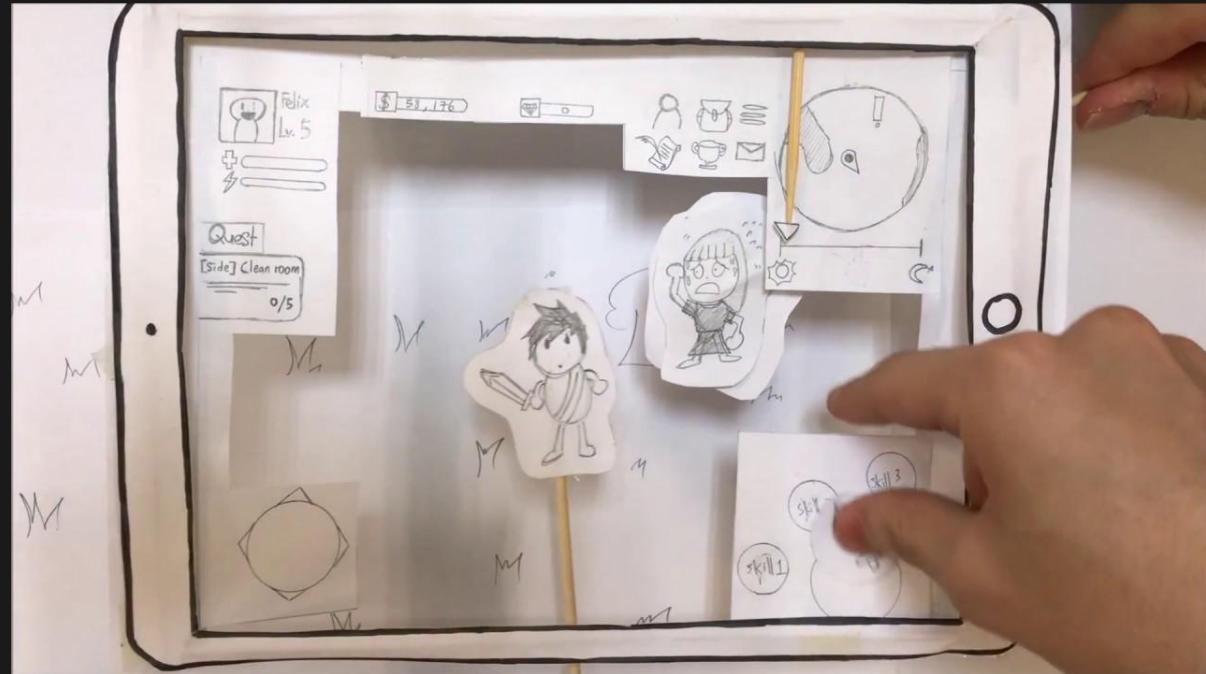


<https://orczuk.itch.io/paws-vs-dummy>

Paper Prototyping

Before you even open up any engine that you plan to work with, it's best practice to sit down with pen and paper and start drawing ideas. Games are complex systems, and attempting to keep everything in your head can be overwhelming.

Begin by sketching basic ideas you want in your game, and then break them down into smaller tasks that you can work on.



Asset Resources



You don't need to be an artist or a musician to create a game. Many generous individuals have provided free resources to help you begin your journey.



Stores:

<https://assetstore.unity.com/>

<https://www.humblebundle.com/bundles>

Visual Assets:

<https://craftpix.net/>

<https://www.kenney.nl/>

Sound Assets:

<https://soundbible.com/>

<https://freesound.org/>

<https://99sounds.org/free-sound-effects/>

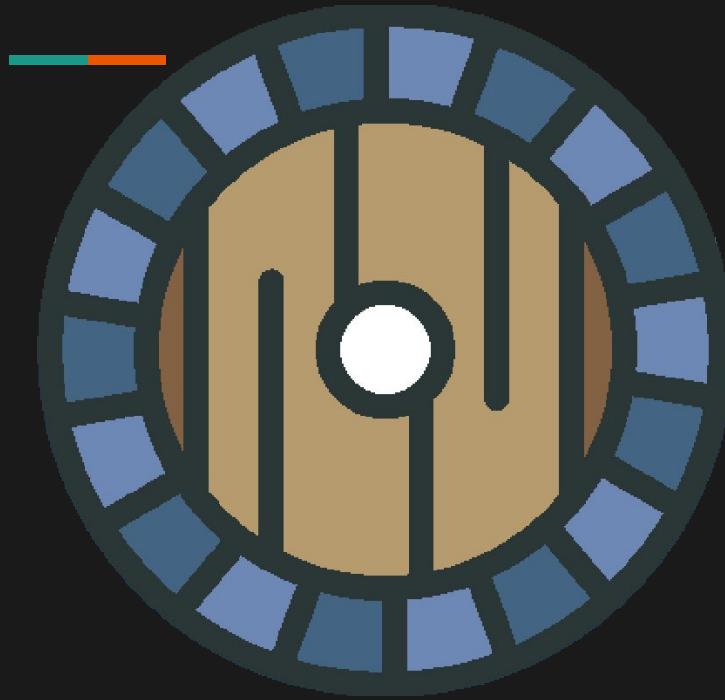
<https://soundimage.org/>

Font Assets:

<https://www.1001fonts.com/>

<https://cooltext.com/>

Thank You



Take a Chance, Make the Game

Sebastian Grygorczuk



<https://www.linkedin.com/in/sgrygorczuk/>



<https://orcduk.com/>