CoGrammar

Getting Started with Game Development

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

Tech Talk – Housekeeping

- The use of disrespectful language is prohibited in the questions. This is a supportive learning environment for all. Please engage accordingly.
  - Fundamental British Value: **Mutual Respect**
  - Please review Code of Conduct (in Student Undertaking Agreement) if unsure
- No question is daft or silly – **ask them!**
- There are Q/A sessions midway and at the end of the session, should you wish to ask any follow-up questions.
- Should you have any questions after the lecture, please post them to https://forms.gle/G4wZytpMYYn9viuY7
- For all non-academic questions, please submit a query: www.hyperiondev.com/support
- Report a safeguarding incident: http://hyperiondev.com/safeguardreporting
- We would love your feedback on lectures: https://hyperionde.wufoo.com/forms/zsgv4m40ui4i0g/

Objectives

1. Help those interested in game development get a clear understanding of what to continue learning next.

2. Understand the nuances of game development.

3. Understand the different game engine options available.

Question:

What is the difference between Game Development and Game Design?

Game Dev VS Game Design

Game Dev:

- ★ Technical execution that turns game design into a playable game.
- ★ Develops concepts provided.
- ★ Code game mechanics
- ★ Requires writing code

Game Design:

- ★ Focused on artistic and creative aspects.
- ★ Responsible for overall vision and how it plays.
- ★ Involves storytelling and world-building
- ★ Requires creative writing

Question:

How would my skills from my bootcamp be applicable?

Available Roles

Software Engineer:

- ★ Game Programmer.
- ★ Engine Programmer.
- ★ Tools Programmer.
- ★ Network Programmer.
- ★ Graphics Programmer.
- ★ AI Programmer.

Data Scientist:

- ★ Game Analytics.
- ★ Machine Learning Engineer.
- ★ User Experience Researcher.
- ★ Economy Designer
- ★ Performance Analyst
- ★ AI Developer

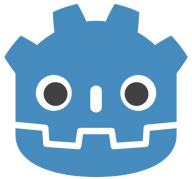
Question:

Where do you get started?

Available Options

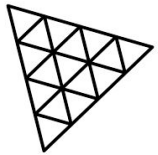
Game Engines:




Frameworks:





Build your own:

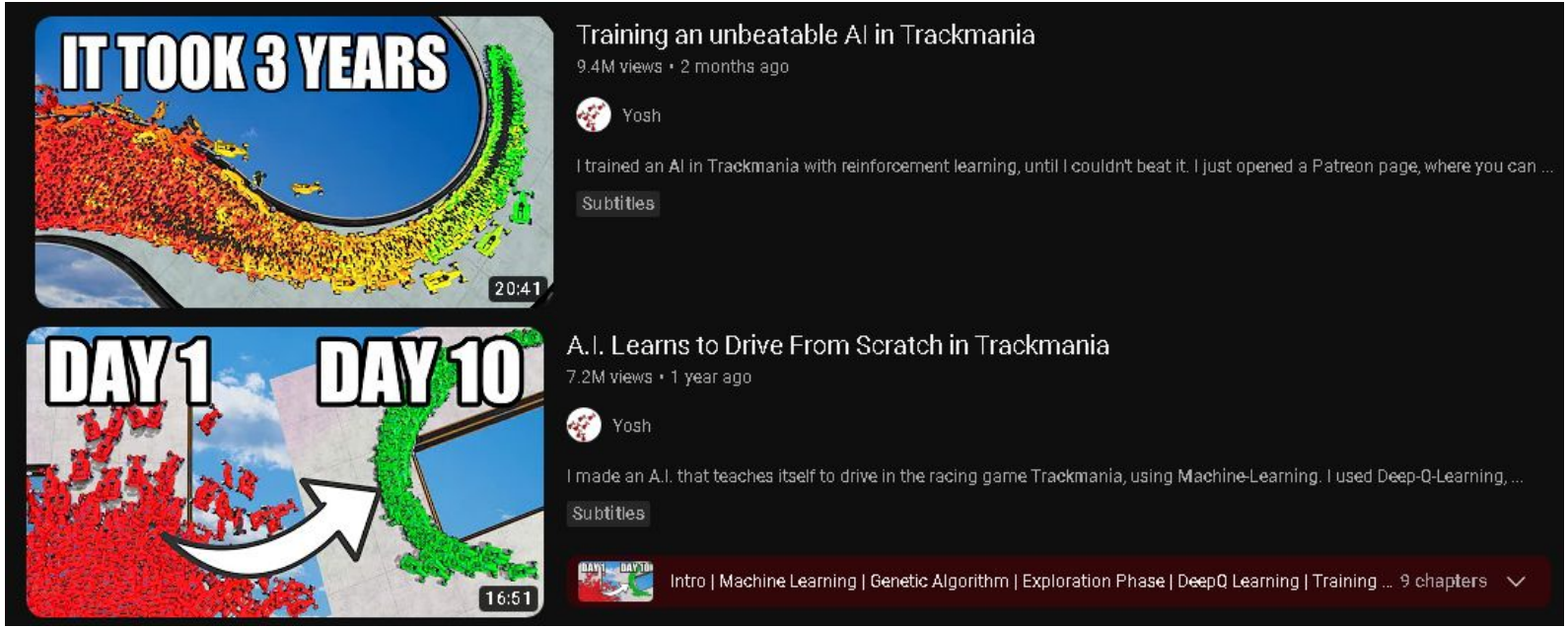
Three JS Portfolio Example:

Source: **bruno-simon.com**

Unreal Engine 5 Example:

Source: youtube.com

Data Science Example:



Source: youtube.com

Question:

Should you make your own game engine or framework?

Creating your own engine

Pros:

- ★ **Learning Experience.**
- ★ Control.
- ★ Customisation.
- ★ Optimization.
- ★ Intellectual Property.

Cons:

- ★ Complexity.
- ★ Time-Consuming.
- ★ Resource-Intensive.
- ★ Maintenance.
- ★ Lack of Community Support.

Should only be considered if your goal is to create a game engine and not a specific game!

Why use game frameworks?

Pros:

- ★ Learning and Understanding.
- ★ Lightweight.
- ★ Flexibility.
- ★ Simplified. Development.
- ★ Community Support.

Cons:

- ★ Less Hand-Holding.
- ★ Steep Learning Curve.
- ★ Limited Tools.
- ★ Performance Tuning.
- ★ Less Out-of-the-Box Functionality.

Why use game engines?

Pros:

- ★ Efficiency.
- ★ Ease of Use.
- ★ Rich Feature Set.
- ★ Cross-Platform Development.
- ★ Community and Support.
- ★ Asset store and free assets.*

Cons:

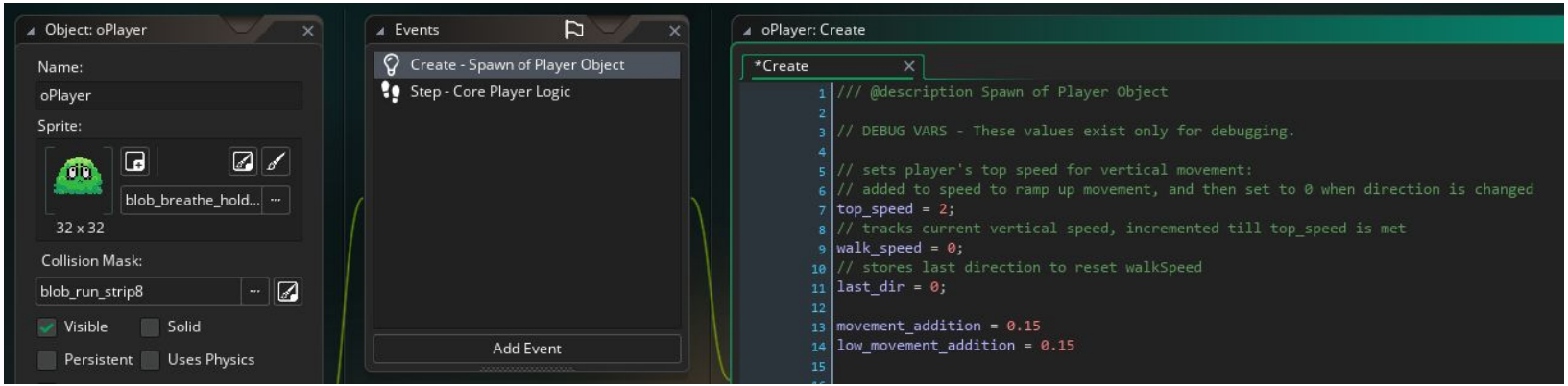
- ★ Cost.
- ★ Learning Curve.
- ★ Overhead.
- ★ Less Control.
- ★ Royalties.*

Question:

What are key overarching principles in game development?

Overarching principles

Object-Oriented Programming (OOP) and Game Objects.



Overarching principles

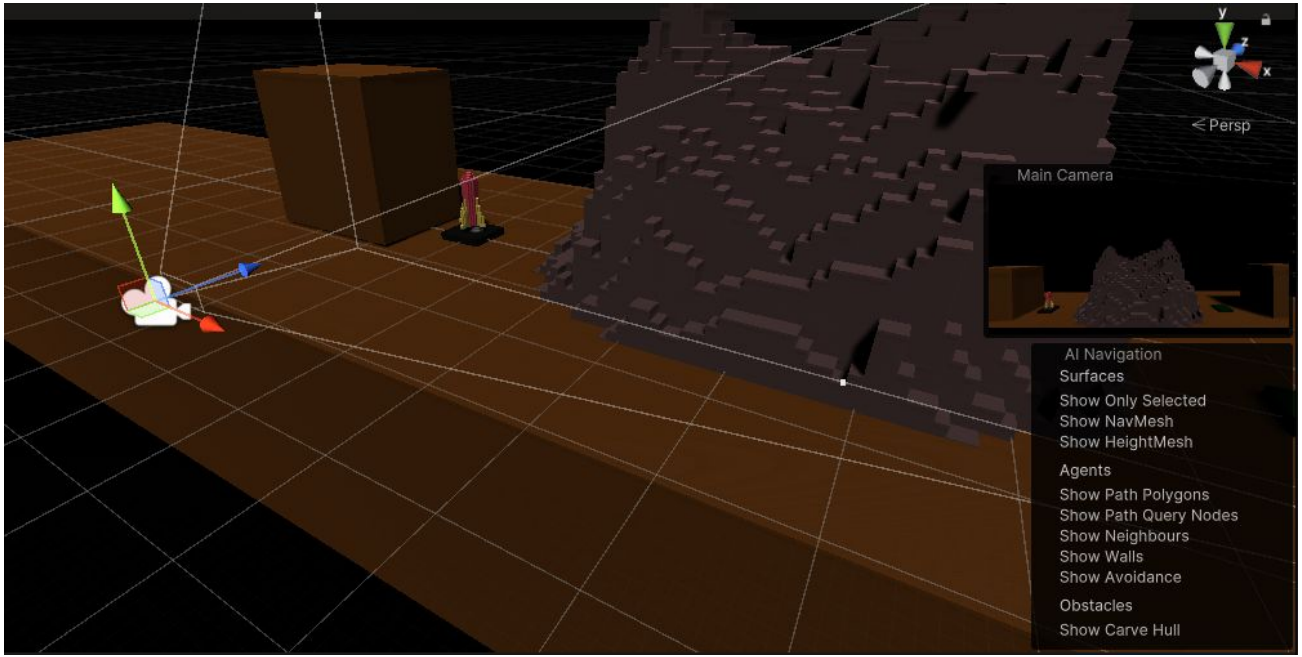
Game Loop.

Assets > Scripts > NewBehaviourScript.cs

```
1   using System.Collections;
2   using System.Collections.Generic;
3   using UnityEngine;
4
5   public class NewBehaviourScript : MonoBehaviour
6   {
7       // Start is called before the first frame update
8       void Start()
9       {
10
11      }
12
13      // Update is called once per frame
14      void Update()
15      {
16
17      }
18  }
```

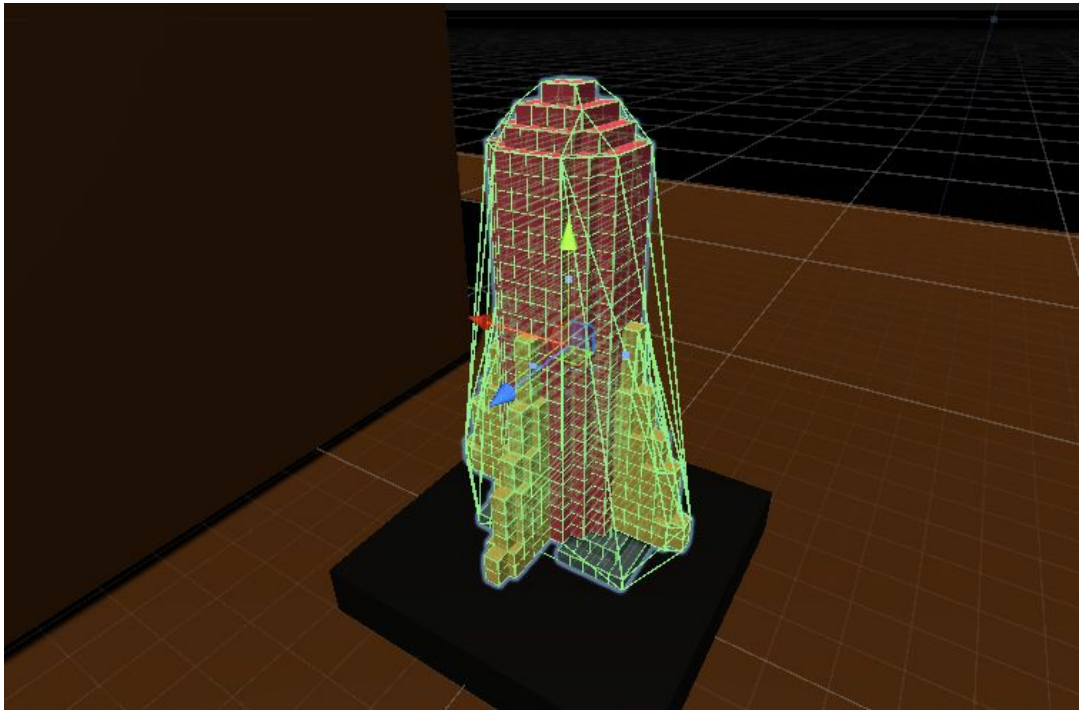
Overarching principles

Camera Controls, Resolution and Viewport.



Overarching principles

Collision and Physics.



Overarching principles

- ★ Graphics and Visuals.
- ★ Sound and Music.
- ★ User Interface (UI).
- ★ Asset and Resource Management.
- ★ Testing and Debugging.
- ★ Performance Optimisation.

Pop Quiz!

What is the correct logic for moving an object in game?

A) - While position is less than required position, add to the player's axis.

B) - If position is less than required position, add to the player's axis.

C) - For each increment the player is away from the required position add to the player's axis.

Question:

Which game engine is best?

It depends on the game you want to create!

Choice examples:

- ★ 2D Pixel Art Game - Gamemaker or potentially Unity
- ★ Voxel Art FPS Game - Unity or potentially Unreal
- ★ Hyper Realistic Nature Sim - Unreal or CryEngine

Why Gamemaker?

What sort of games would you make on this engine?

- ★ Primarily 2D games like 2D platformers, Top-Down Shooters or Visual Novels.
- ★ Lightweight games that could run on practically any hardware.

What sort of games should you not make on this engine?

- ★ 3D games.
- ★ Large scale or more complex projects.

Gamemaker also has a no code option with GML Visual*

Why Unity?

What sort of games would you make on this engine?

- ★ Most 2D or 3D games.
- ★ Relatively lightweight projects, indie titles or projects that need to run on entry level gaming hardware.

What sort of games should you not make on this engine?

- ★ Some more visually intense games with complex details.
- ★ Text-Only games or simple web games.

Why Unreal?

What sort of games would you make on this engine?

- ★ High-End games.
- ★ Enthusiast-level projects that require modern gaming hardware.

What sort of games should you not make on this engine?

- ★ Simple 2D games.
- ★ Text-Based Games.
- ★ Mobile games that require low overhead.

What about Godot?

What's different about this engine?

- ★ Free and open source.
- ★ Community created and similar to Unity.

What you will skip if you pick this engine?

- ★ Asset stores and freebies.
- ★ Monetization models.
- ★ Proprietary technologies.
- ★ Professional support and services
- ★ Corporate backing

Question:

What’s the next step for you?

Factors to consider:

- ★ Project Scope.
- ★ Learning Curve.
- ★ Programming Knowledge.
- ★ Asset Needs.
- ★ Target Platforms.
- ★ Budget.
- ★ Graphics and Performance.
- ★ Community and Support.
- ★ Future Opportunities.

Challenge:

Spec out your game and choose the best engine for your project!

Wrapping Up

Choosing the right game engine

Depends on the overall scope and specs of your game.

All game engines have overarching principles

A good understanding of the game loop and OOP is required for all game engines.

CoGrammar

Questions around game engines and game development

CoGrammar

Thank you for joining