

03 Introduction to Unity

Tvorba a dizajn počítačových hier (FMFI)

Návrh a vývoj počítačových hier (FIIT)

Michal Ferko

3. 10. 2024

Game Engines

- **Unity** – ~70% of all mobile games, ~50% of all Steam games
 - *“50% of games across devices are made with Unity”* – Unity, 2022
- **Unreal** – ~15% of all Steam games, much higher percentage on consoles
- **Godot** – the biggest/most famous open-source engine

- Lots of other engines or frameworks
 - GameMaker, XNA, CryEngine, Construct, RPG Maker, Frostbite, Source, id Tech...

<https://www.gamedeveloper.com/business/game-engines-on-steam-the-definitive-breakdown>

<https://steamdb.info/tech/>

Unity	Unreal Engine	Godot	Game Maker
<p>Among Us Genshin Impact Twelve Minutes Valheim Hearthstone Cuphead Ori 1 & 2 Monument Valley 1 & 2 Cities: Skylines Inside Hollow Knight Pokemon Go Superhot Gwent Beat Saber Overcooked Untitled Goose Game Fall Guys ...</p>	<p>Fortnite Sea of Thieves PUBG Deep Rock Galactic Black Myth: Wukong Palworld Manor Lords Stray It Takes Two Star Wars Jedi: Fallen Order Tekken 7 The Outer Worlds BioShock Kingdom Hearts III A Way Out Mortal Kombat X Borderlands ...</p>	<p>Dome Keeper Brotato Halls of Torment Cassette Beasts ...</p>	<p>Undertale Hotline Miami Hyper Light Drifter Spelunky Loop Hero Pizza Tower Nuclear Throne Swords of Ditto Katana Zero Downwell Forager ...</p>

Unity

- Free
 - Cannot earn/raise more than \$100k/year (**\$200k/year with Unity 6**)
 - Unity Splash Screen FORCED (**OPTIONAL with Unity 6**)
- Pro €1.877/seat/year (~**€2.040/seat/year from 2025**)
 - Access to Pro Analytics, Team tools...
 - **Needed to deploy to consoles – Xbox/PS/Switch** (platform devkit still required)
- Asset store: full of interesting models/scripts/tools
- Source code access: **EXPENSIVE** (Pro/Enterprise license)
- Scripting in C# (or visual scripting)

Unity platforms

- Editor: Windows/Mac/Linux
- Release platforms:

iOS



macOS

Windows

Meta



visionOS

tvOS



ARCore



Magic Leap

androidtv

Unreal

- Free to use
- 5% royalties to Epic Games
 - For every dollar you earn above \$1M
 - Can reduce this if you release on Epic Games Store
- Source code access: **YES** (for most parts of the engine)
 - Not open source, only source available
- Marketplace
- Scripting: C++, Blueprints (visual scripting)
 - Has bindings for other languages

Unreal platforms

What platforms does UE5 support?

Unreal Engine 5 enables you to deploy projects to Windows PC, PlayStation 5, PlayStation 4, Xbox Series X, Xbox Series S, Xbox One, Nintendo Switch, macOS, iOS, Android, ARKit, ARCore, OpenXR, SteamVR, Oculus, Linux, and SteamDeck. You can run the Unreal Editor on Windows, macOS, and Linux.

PlayStation 5, PlayStation 4, Xbox Series X, Xbox Series S, Xbox One, and Nintendo Switch console tools and code are available at no additional cost to developers who are registered developers for their respective platform(s).

Godot

- Free forever and open source
 - Community-driven and community-built
- Gained a lot of popularity in the last year (especially thanks to Unity's PR disaster)
- Not that feature-complete as Unity or Unreal
- Can buy assets on a few sites, but much smaller support than Unity
 - Also much smaller (but more active) community
- Scripting
 - GDScript – similar to python
 - C/C++ – official support
 - C# – Godot 4.2+ should have all platform support

Godot 4.3 Platforms

- Windows, Mac, Linux
- HTML5
- Android, iOS
- Consoles
 - Cannot be part of Godot directly
 - There is a studio porting to consoles made by Godot authors

Note

In practice, the process is quite similar to Unity and Unreal Engine, except that you need to contact a third-party developer to handle the porting process. In other words, there is no engine that is legally allowed to distribute console export templates without requiring the user to prove that they are a licensed console developer. Doing so would violate the console manufacturer's NDA.

Picking an engine

- License/pricing – very important
- Features & other support
 - anything except Unity on mobile will be complicated
- Some engines are more suited for some games
 - Unreal – shooters, aspirations to do AA/AAA development
 - It is said to have worse performance on mobile – but I cannot verify this
 - Unity – mobile, indie PC/Console

Why Unity

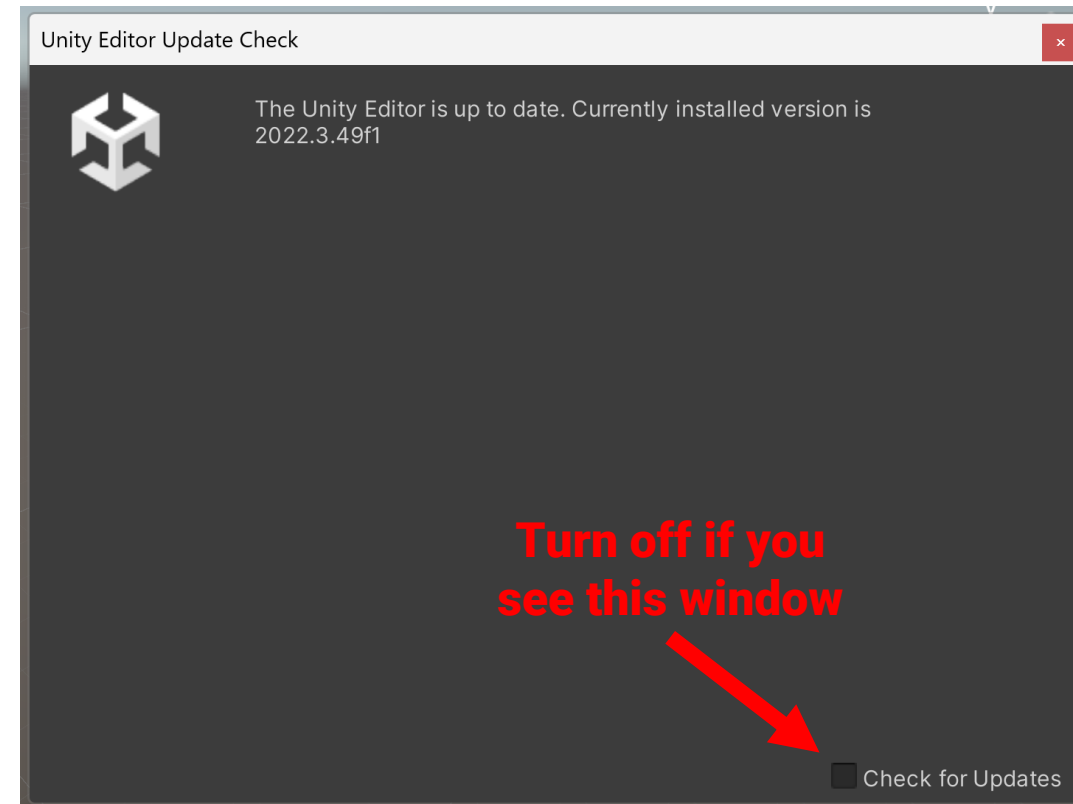
- I have 10+ years experience, can provide better support
- Unity rolled back crazy pricing changes
- Easier to learn than Unreal
- Editor has better performance and lower requirements than Unreal
- More feature-complete compared to Godot
- De-facto standard for indie/mobile development

- Let's hope they don't change their terms again

Unity

Recommendations

- Use Unity Hub to manage Unity installations
- Start new projects with the latest LTS version (2022.3 now)
- Don't just automatically update Unity
- The only reasons to update a project:
 - New bugfixes you need
 - New features you need



Interface overview

- Scene View
- Hierarchy
- Game View
- Project Panel
- Inspector
- Toolbar

Basic Concepts

- Game Objects
- Components
- Prefabs
- Importing assets
- Project Preferences
 - Other project settings
- Build

Scripting

- Adding, naming, renaming, removing C# scripts
- Visual Studio
- GetComponent<Type>() -> LOW PERFORMANCE!
- Input.GetKey()
- Renderer component -> material color
- Private and public variables, linking with editor
 - Assigning prefabs/existing game objects to variables
- Awake() and Start()
- Update() and FixedUpdate()

Scripting (2)

- Vector3 class and utility functions -> magnitude, Dot, Cross, ...
- Component.enabled and checkboxes in editor
- Light component -> enabled
- Adding mesh filters, renderers and materials
 - Example with light
- GameObject.SetActive() and checkboxes in editor
 - GameObject.activeSelf, GameObject.activeInHierarchy
- Visual Studio debugging
- Transform component -> Translate, Scale, Rotate
 - Why not use with colliders
 - And only kinematic rigidbodies

Scripting (3)

- Vector3.forward, Vector3.right, Vector3.up
- Time.deltaTime in Update()
 - Running bot example
- Vector3.Lerp
 - Moving light example with changing color
- GameObject.Destroy(GameObject[, time])
- GameObject.Destroy(Component[, time])
- Input.GetButton...
- Access scripts of other Game Objects
- Instantiate()
- Value vs. Reference types

References

- <http://unity3d.com/learn>