03 Introduction to Unity

Tvorba a dizajn počítačových hier (FMFI)

Návrh a vývoj počítačových hier (FIIT)

Michal Ferko

5. 10. 2023

GDD deadlines postponed!

Idea approval via email: **22. 10. 2023**

GDD submission: **29. 10. 2023**

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
  - GameMaker, CryEngine, Construct, RPG Maker, Frostbite, Source, id Tech…

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

Unity	Unreal Engine	Godot	Game Maker
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 …	Fortnite Sea of Thieves Batman Arkham games PUBG Shenmue III Star Wars Jedi: Fallen Order Tekken 7 The Outer Worlds BioShock Deus Ex Kingdom Hearts III A Way Out Hellblade: Senua’s Sacrifice Mortal Kombat X Borderlands Biomutant Mirror’s Edge …	Dome Keeper Brotato Halls of Torment Cassette Beasts …	Undertale Hotline Miami Hyper Light Drifter Nuclear Throne Spelunky Swords of Ditto Katana Zero Downwell …

Unity (Before Jan 01 2024)

- Free: Forced Unity splash screen
  - cannot earn/raise more than $100k/year
- Pro: Access to Pro Analytics, Team tools… **$2040/seat/year**
- Asset store: full of interesting models/scripts/tools
- Source code access: **EXPENSIVE** (Pro/Enterprise license)
- Scripting in C# (or visual scripting)
- Holds true for games on Unity versions older than 2023 LTS

https://store.unity.com/compare-plans

Unity (After Jan 01 2024)

- Revealed new pricing a few weeks back – **runtime fee**
  - https://blog.unity.com/news/plan-pricing-and-packaging-updates
  - Wanted to charge per-install – VERY BAD IDEA
  - Could bankrupt studios, install bombing, charity/bundles issues etc.
  - Tried doing it retroactively for **already released games**
- Walked back changes after developer outcry
  - https://blog.unity.com/news/open-letter-on-runtime-fee
  - It’s less bad, but trust is lost
  - You report “unique customers”, installs not tracked
  - They say they won’t retroactively change terms
    - But they already tried twice (2019, 2023)

Unity Pricing from 2023 LTS

- Personal: FREE, can use if you earned <$200k in last 12 months
  - **No Unity splash screen needed**
- No game with <$1M revenue in last 12months will have to pay the runtime fee
- Pro: have to use if you earned >$200k in last 12m
  - 2040$/seat/year
- Runtime fee still exists
  - Capped at 2.5% of revenue
- It won’t bankrupt studios, but Unity will still be much more expensive
  - Studios are looking for alternatives

Unity platforms

Unreal

- Free to use
- 5% royalties to Epic Games
  - For every dollar you earn above $1M
- Source code access: **YES** (for most parts of the engine)
  - Not open source!
- 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 and open source
  - Community-driven and community-built
- Absolutely no price and it won’t change
- Not that popular, 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++ – officially supported
  - C# – Godot 3 full support, Godot 4 won’t work on mobile (they’re working on it)

Godot Platforms

Godot 3.5

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

Godot 4.1

- Windows, Mac, Linux
- Android, iOS (not C#)
- HTML5 (experimental)
- Consoles
  - Same issue as with 3.5

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
- Pricing changes have been announced very recently
- Easier to learn than Unreal
- Editor has better performance 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

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