

03 Introduction to Unity

Tvorba a dizajn počítačových hier (FMFI)

Návrh a vývoj počítačových hier (FIIT)

Michal Ferko

10. 10. 2019

Game Engines

- Unity
- Unreal
- CryEngine
- GameMaker
- Lots of other players
 - Construct, Godot, RPG Maker
 - Frostbite, Source, id Tech
 - Hero Engine, Leadwerks, Blender
 - ...

Unity	Unreal Engine	CryEngine	Game Maker
<p>Hearthstone Cuphead Ori and the Blind Forest Monument Valley 1 & 2 Cities: Skylines Inside Hollow Knight Alto's Adventure Pokemon Go Superhot Firewatch Gwent Beat Saber Overcooked Untitled Goose Game</p>	<p>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</p>	<p>Far Cry series Crysis series Kingdom Come: Deliverance Prey Evolve</p>	<p>Undertale Hotline Miami Hyper Light Drifter Nuclear Throne Spelunky Swords of Ditto Katana Zero Downwell</p>

Unity

- Free: Unity splash screen, cannot earn more than \$100k
- Plus: Get rid of Unity logo, 35\$/month (at least 12 months)
- Pro: Access to Pro Analytics, Team tools... (75\$/month)
- Asset store: full of interesting models/scripts/tools
- Fully multiplatform (PC, Mac, Linux, Xbox, PS, Android, iOS + tons more)
- Source code access: **EXPENSIVE**
 - Enterprise license
- Scripting in C#

Unreal

- Free to use
- Pay 5% royalties to Epic games
- Source code access: **YES**
- Marketplace
- Fully multiplatform (PC, Mac, Linux, Xbox, PS, Android, iOS)
- Scripting
 - Blueprints (visual), C++

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>