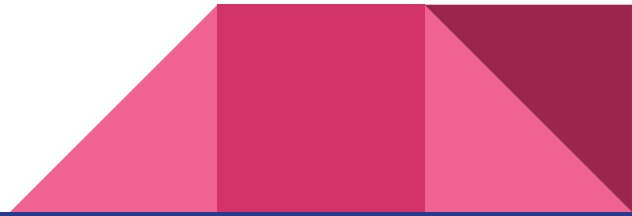
Game Engines

Background

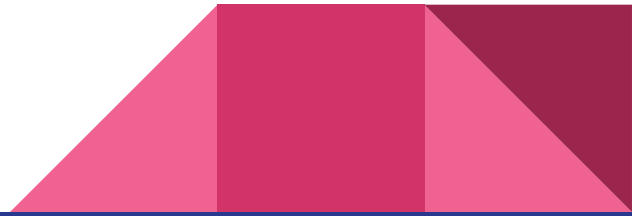
History of Game Engines

- First time concept appears in early 1990’s with Doom
- *Doom* featured a well organized base code
- Separated game code from core components
- Other studios ended up creating new games by modifying existing ones
- By the end of 1990’s *Unreal* and *Quake* were designed for modding
- *Id Software* made money on licensing the Quake Engine (CoD)



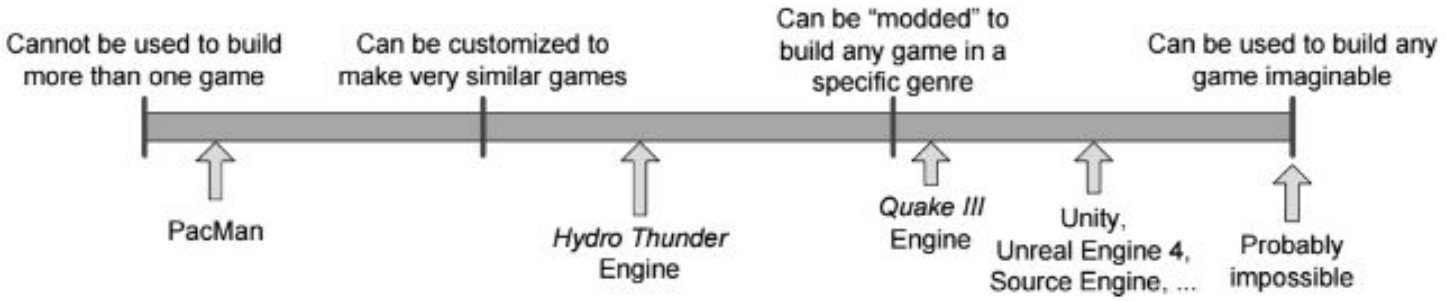
What is a Game Engine ?

- A real time simulator that uses *data driven architecture*
- It lacks hard coded game logic
- It is extensible
- It can be used as the foundation to build many games
- The separation between engine and game code is often blurry
- The perfect scenario does not exist!
- Best case scenarios are *Unity* and *Unreal*



It is a trade-off

- A more generic Engine means less optimized
- The tradeoff is between generality and optimality
- CryEngine is well suited for FPSs
- Unreal has been the king of 3rd person action games
- Unity is the engine of choice for mobile and browser



2003 Context

Source Engine

Source Engine from Valve

- Started from 1998's Half-Life
- Technology based on Quake source code
- Valve focused on mod community
- That successful strategy created:
  - Counter-Strike
  - Team Fortress
- In 2003 Valve announce Source engine:
  - Half-Life 2, Team Fortress 2, Left4Dead
- In 2015 Valve announced Source 2

CryEngine 5 and Lumberyard

Wait Amazon… really?

CryEngine from Crytek

- Started as a game: 2004’s Far Cry
- Supports Win/Xbox 360/Xbox One/PS3/PS4
- Interest came from the high quality graphics
- Not many games licensed
- Successful Crysis series pushed the technology
- Amazon licensed full CryEngine 4 and released Lumberyard for free
- Currently focused on VR

Unreal Engine 4 (not only for video games anymore)

Unreal Engine from Epic Games

- Started from 1998’s Unreal by Tim Sweeney
- Idea was to compete with Id Software
- Supports pretty much all platforms
- Interest came from the high quality graphics
- Good architecture led to many licensing deals
- Today’s engine of choice for AAA games
- Heading research in Real Time Graphics for many areas beyond games

There is more? Welcome Unreal Engine 5 - Full demo here

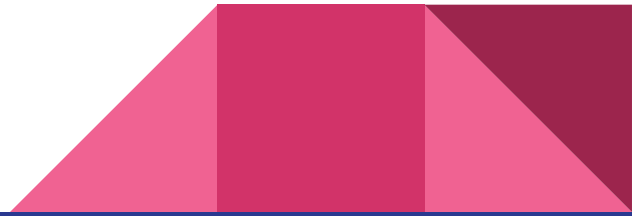
Unity

Unity by Unity Technologies

- Started for OSX in 2005
- Largest base of platforms: 21
- Nintendo’s technology of choice
- Goal to “democratize game development”
- Engine of choice for Indie Developers
- … and most mobile game development
- Unity 2018 pushing for AAA development

Homebrew Game Engines out there

- Most big publishers create their own technology:
  - UbiSoft internal engine *Anvil*
  - Dice's *Frostbite* for Battlefield series
  - Sony's *PhyreEngine*
  - Naugthy Dog's internal engine
  - Rockstar's *RAGE* Engine
  - Avalanche's own engine for Just Cause series
  - Infinity Award's engine for Call of Duty
  - Codemaster's *EGO* engine
  - Square Enix Luminus engine



And open source engines!

- Low price policy of Unity is killing open source engines, still:
  - Godot engine
  - Directus3D
  - Panda3D
  - OGRE
  - Yake based in OGRE
  - Torque engine
  - Irrlicht engine
  - Crystal Space
  - And many more

Homework

- Integrate MathGeoLib (you have the library files ready on Virtual Campus)
  - It should feature common vector, quaternion and matrix math (easy)
  - Test that it can handle intersections between *(do not draw anything yet)*:
    - **Spheres**, Cylinders, **Capsules**
    - **AABB**, **OBB**, Frustums, **Planes**, Segments, **Ray**
    - Convex Hull, Mesh, **Triangles**
    - Careful with it's configuration, could be very slow (read & edit MathBuildConfig.h)
  - You are free to try to use OpenGL Mathematics
- Find a good random number generator
  - (MathGeoLib already has one)

Homework: Code clean

Remove:

- Glut (we will be using Glew)
- Rename the solution and project name to your own engine
- p2Point/p2List/p2DynArray -> use STL
- ModulePlayer anything else related to game code
- Keep glmath.h for now but we will remove it after assignment 1
- Bullet and SDL_Mixer / SDL_Image (including the dlls)
- Do not remove the camera… yet