

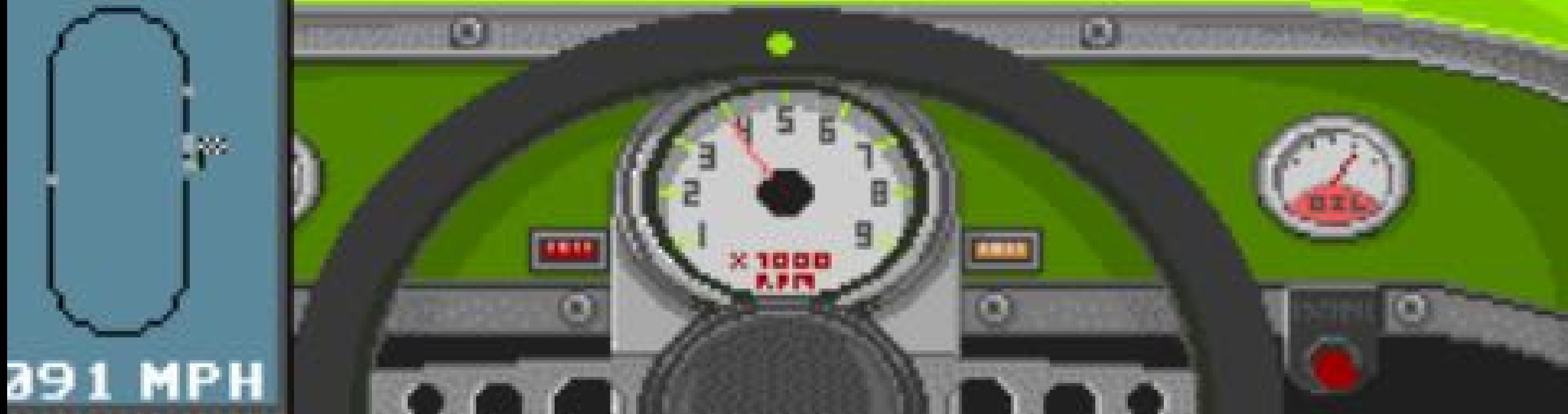


Explore 3D!

LAP 1 OF 5  
1:00.4

POS: 6

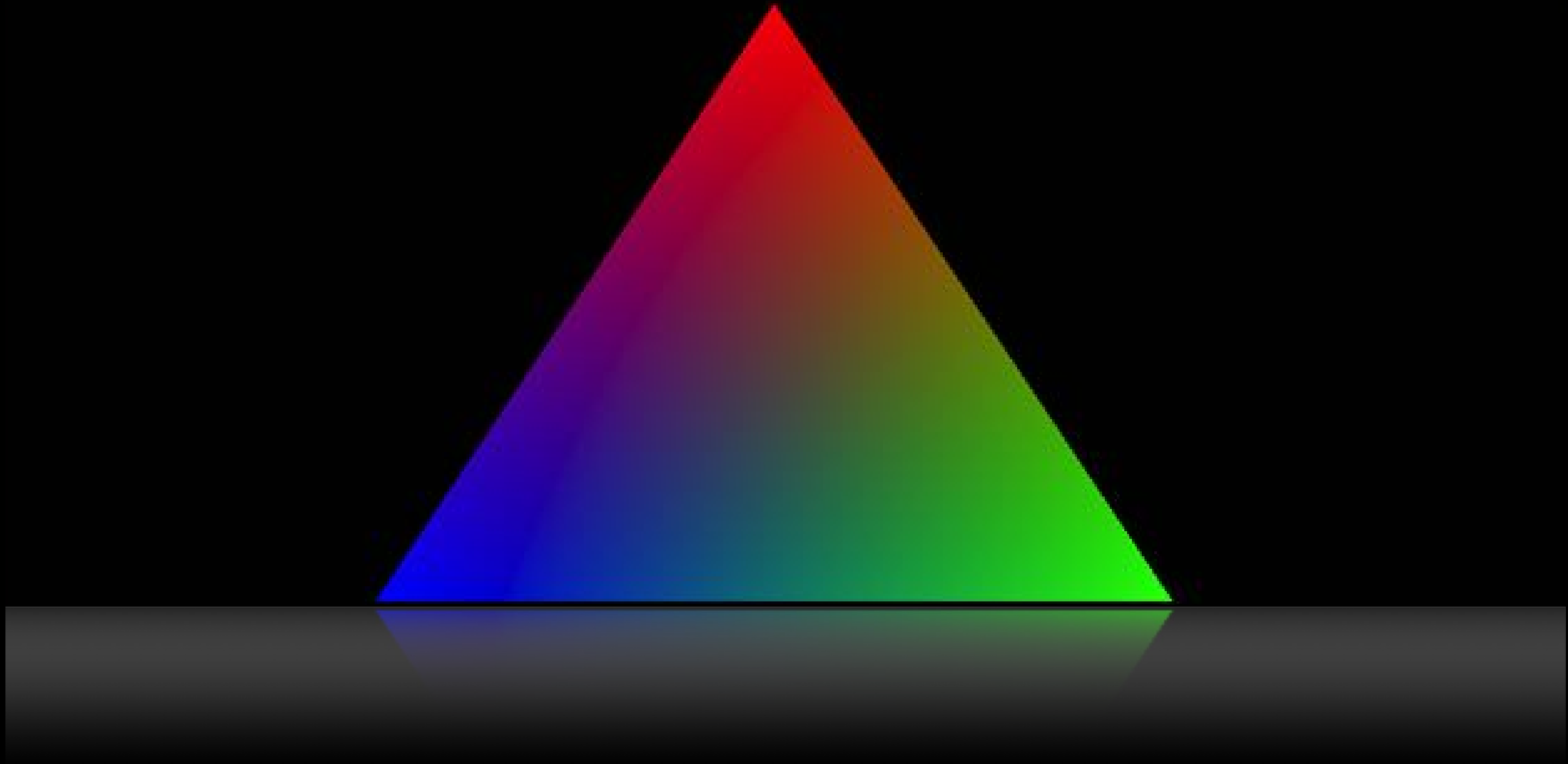
291 MPH





# Once upon a time...

- Bekannte Engines nur per Lizenz
- Mod-Tools von Spielen eingeschränkt
- Code Frameworks
  - Ogre / SDL
  - libcinder / OpenFrameworks
- „klassisch“ OpenGL / DirectX



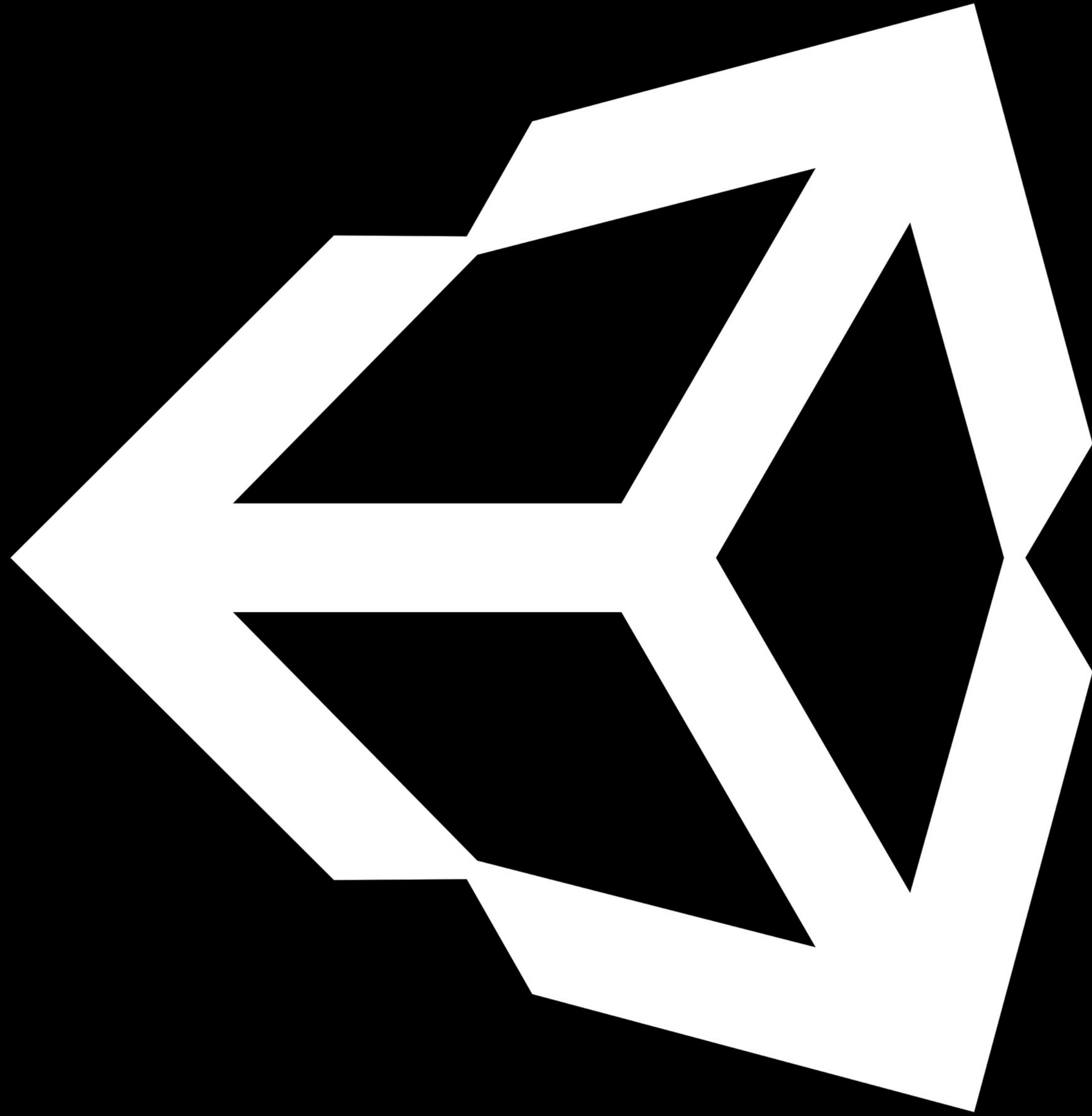






# Okay, let's make a game!

- Programmiersprache
- Frameworks für Dateien parsen etc.
- Vektormathematik, Trigonometrie & Physik
- Beleuchtungsmodelle
- Modellierung und Texturierung
- Audioprogrammierung





**„To democratize  
game development“**



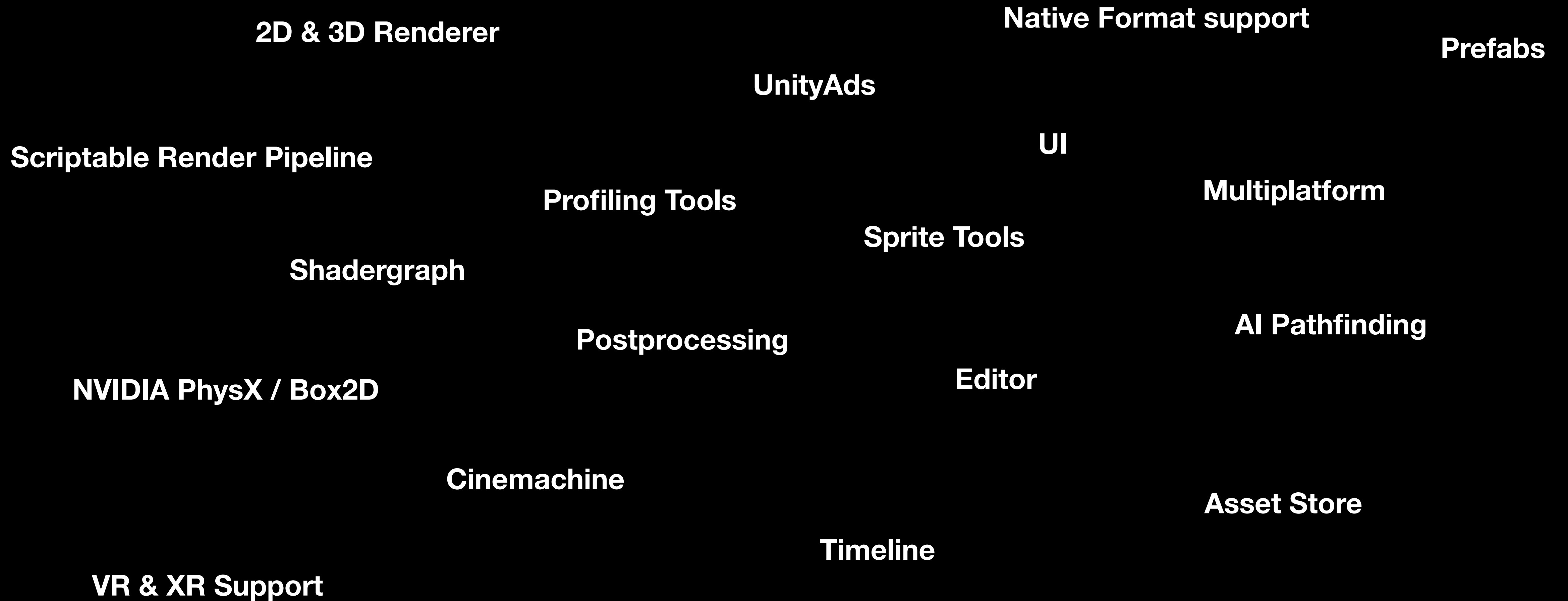


# Why Unity?



© 2015 KISS Catalog, Ltd. Under License to Epic Rights.

# Features





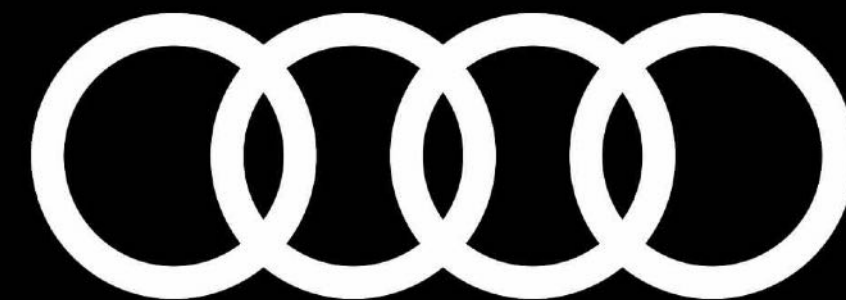
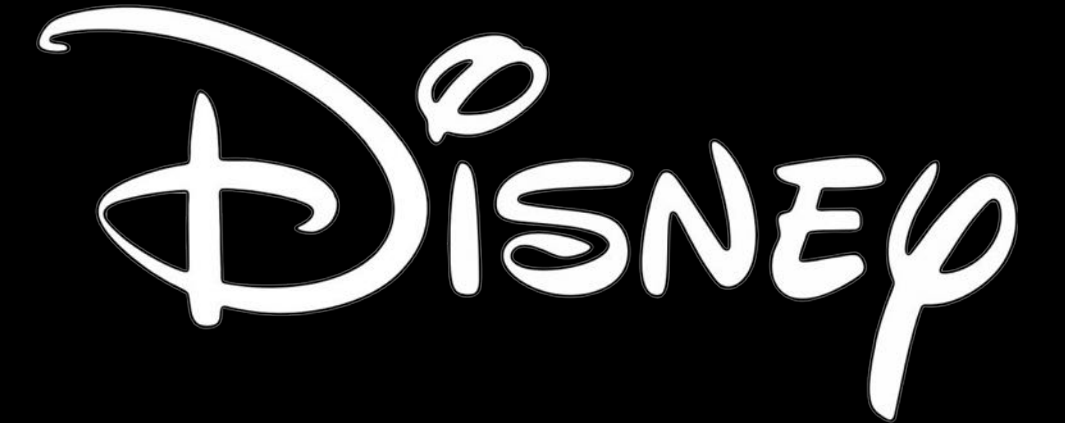
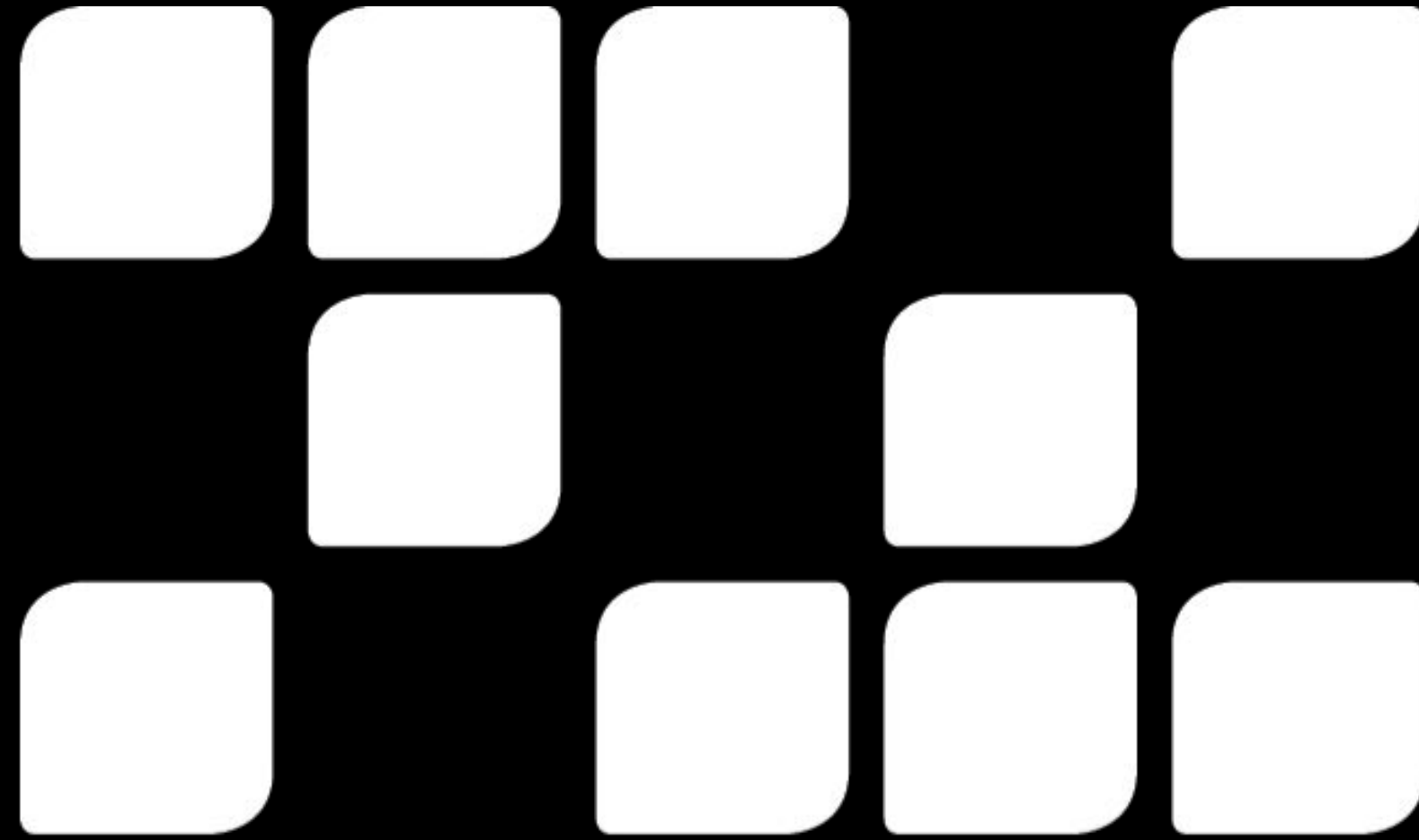
# Platforms

ios  
Oculus  
Nintendo Switch  
Android  
Tizen  
macOS  
Linux  
SteamOS  
WebGL  
Windows  
Facebook  
Xbox One  
Playstation 4

25+ platforms at all

# License

- Personal - free - up to 100k \$ revenue
- Plus - 25\$ month - more features & up to ~200k \$ revenue
- Pro - 125\$ month - full featured
- Enterprise - on request - full source access



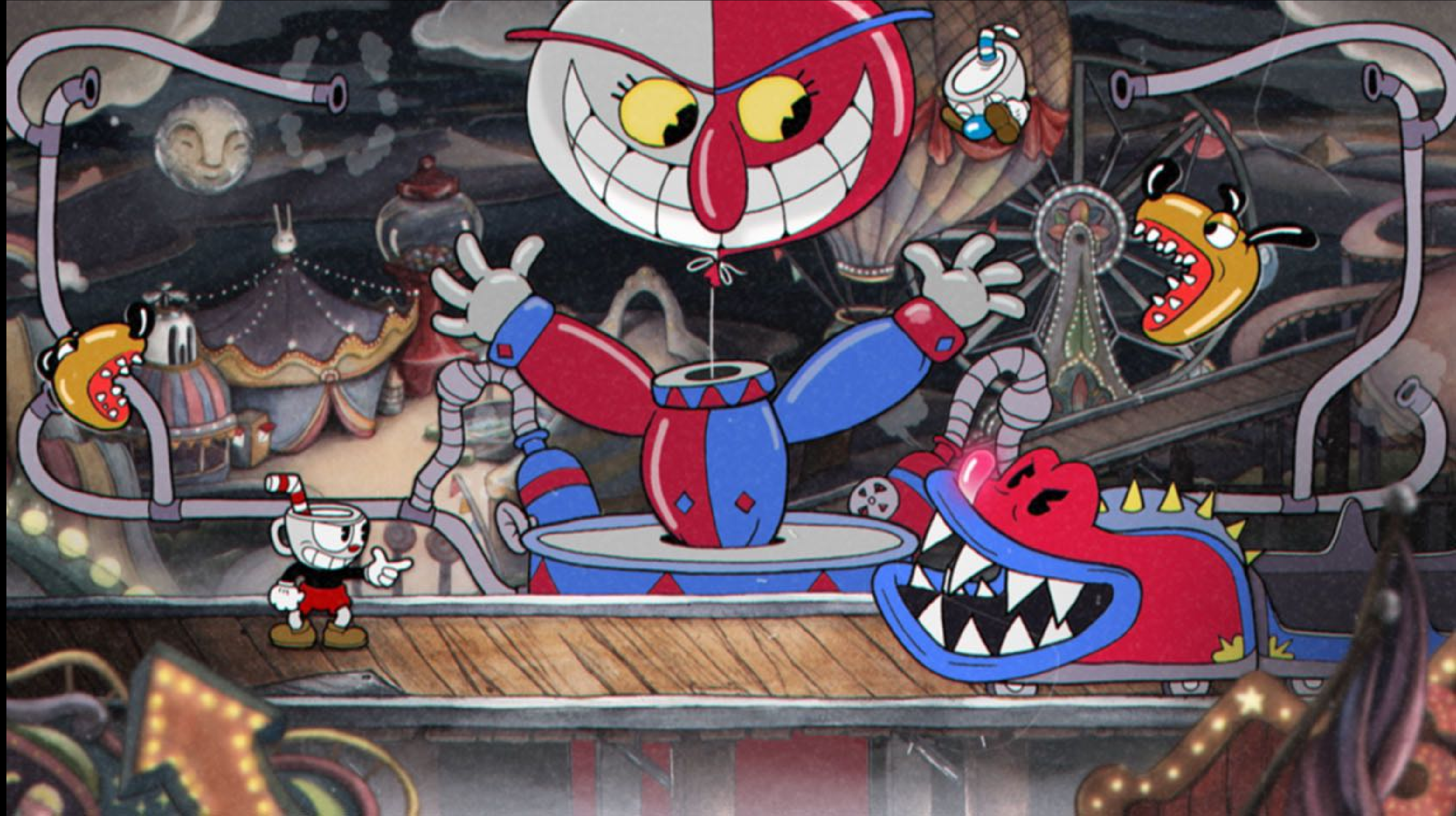






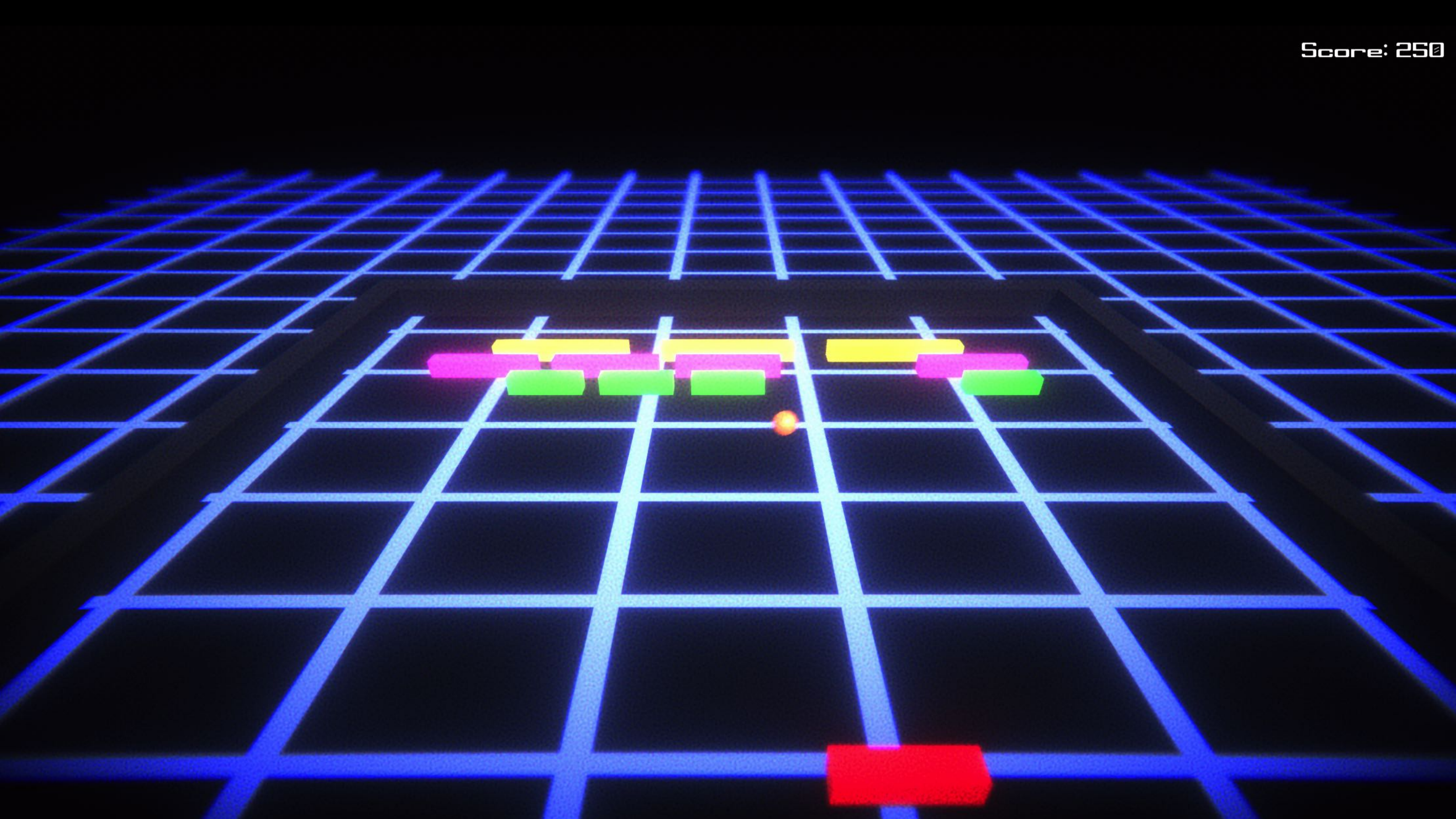








Score: 250





# Skizze

- Objekte in Welt
- Eingabebehandlung
- Treffererkennung
- Physik
- Logik
- Cooler Retro-Look!

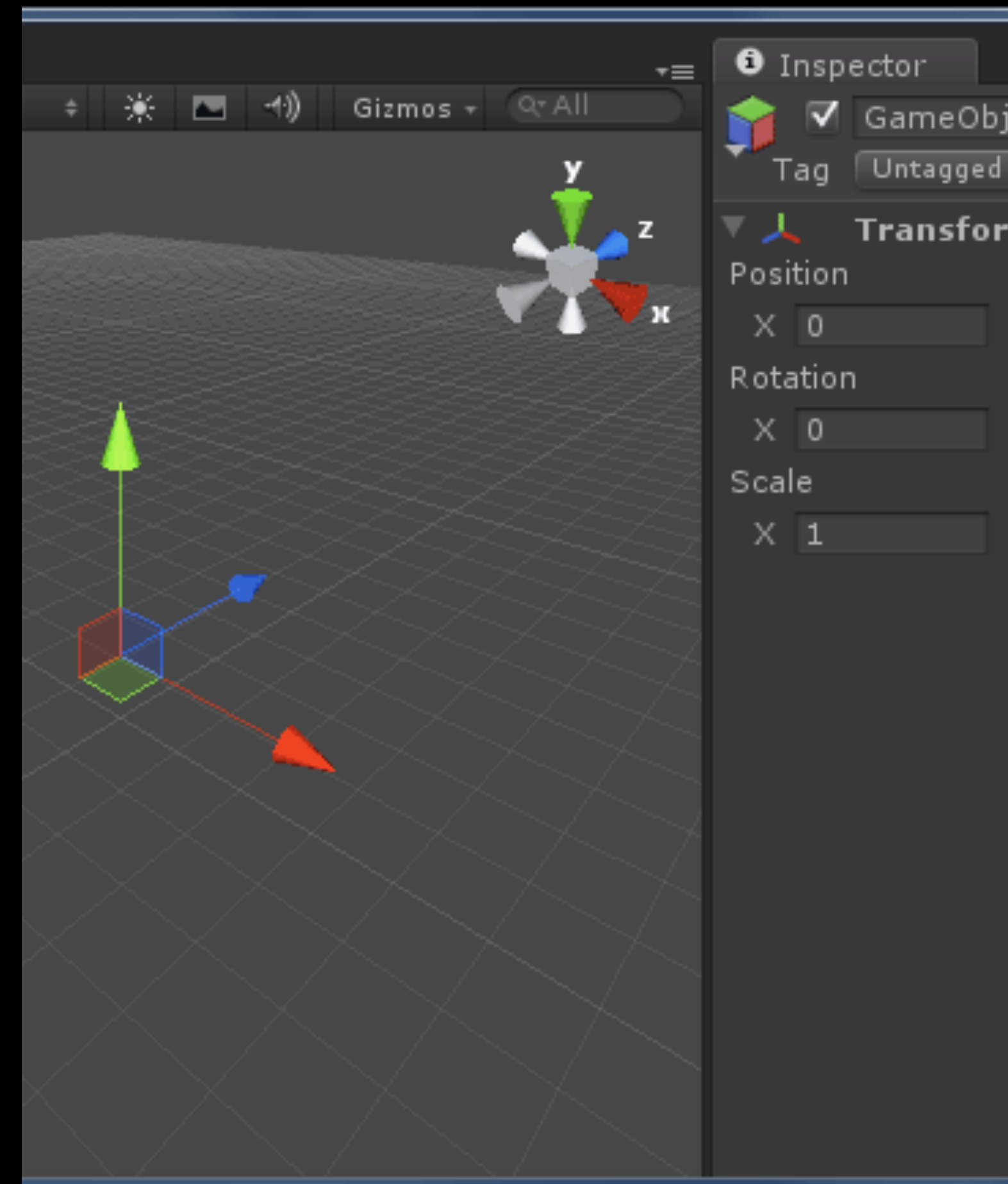


# Game Loop

- Engine-Durchlauf eines gerenderten Bildes
- Alle Objekte in Welt werden aktualisiert
- Reihenfolge der Objekte & Skripte ist nicht definiert
- Ein Durchlauf sollte nie länger als 16 ms dauern [60 FPS]

# GameObject

- Grundlegendes Objekt in Unity
  - Spieler
  - Weltobjekt (Tür,...)
- Beinhaltet Components
  - Transformation
  - Material
  - Skripte
  - ...



# Input System

- Im Update() der GameLoop
- Events von verschiedenen Eingabegeräten
- Zustände werden aktiv abgefragt

# Physics

- NVIDIA PhysX / Box2D
- „Self managed“ von Unity
- Collider & Verhalten von Festkörpern definieren



# Prefabs

- Schablonen für Objekte in Welt
- Instanziierung während der Laufzeit
- Varianten einer Basisschablone

# UI-System

- Vorgefertigte UI-Elemente & Eventbehandlung
- Customizing
- Erweiterung
- Automatische Auflösungsbehandlung

# Post Processing

- Effekte wie DoF, Ambient Occlusion oder Bloom
- Relativ günstig im Gegensatz zur „realen“ Berechnung
- Wirkt sich auf Gesamtbild aus
- Als Shader implementiert
- Unity bringt Toolset mit

Done



<https://github.com/hacku/BreakoutClone>



# Further reading

- <https://unity3d.com>
- <https://docs.unity3d.com/2019.1/Documentation/Manual/>
- <https://www.youtube.com/user/Unity3D>

# Shader

- Mini-Programm auf Grafikkarte
  - Vertex, Geometry & Pixelshader
  - hohe Parallelität
- Programmierung in C-Dialekt
- quasi Un-Debugbar ;-)

# Shader

- Standard ist Uber-Shader
  - Shader hat viele Features
  - Downstriping bei Kompilierung
- PBR Rendering
  - Materialeigenschaften nähern sich physikal. Gegebenheiten an