GDV4000
INTRODUCTION TO GAMES INDUSTRY PRACTICE

Adapted by Ian Smith from original work and research by Dr Glenn Jenkins

Games Development Tools and Practices I

INTRODUCTION

Introduction

Problems

- How to get started with Games Development without learning a lot of things?
- How to get started with Unity?
    - Worksheet
- Unity is a big package, what does all this stuff do?
    - Demo/ Videos/ Documentation

INTRODUCTION (CONT.)

Problem Driven

Guided problem solving (preparation for later modules)

‘Think like a programmer’ – Programmers are problem solvers

Consistency

Trying to do this with everything – Developing a mindset that can transferred to different problems

INTRODUCTION (CONT.)

Solutions to Problems

- Most of the solutions will be practical, but not all.
  - Worksheets – guided completion of practical solutions

- Studio / Reading
  - Studios cover a lot of the background theory. We try to move away from didactic delivery as much as we can, but it is important to understand the background to your subject. There may also be required or recommended reading, and videos to watch.

- Contributions
  - You are an active part of your learning – Please suggest books, online tutorials, or any resource you have found helpful.

TODAY'S PROBLEMS

Problem: How to implement our game designs (when we've only just started the course)?

Problem: How do I get started with Unity?

Problem: Unity is a big application, what is all this stuff?

PROBLEM: HOW TO IMPLEMENT OUR GAME DESIGNS (WHEN WE’VE ONLY JUST STARTED THE COURSE)?

Motivation

- Learning about games design
- To design and then implement a game engine from scratch will take years

Solution(s)

- We use an existing game engine.
- Unity – Why?
  - This provides the **game engine**, **graphics engine**, **physics engine**, **sound** etc.

WHY UNITY?

It may not have escaped your attention that Unity caught a fair bit of negative press in the game development world last year over some rather… interesting decisions over runtime fees and licences, which saw them fall rather spectacularly from favour with a lot of their core users (around 500 studios) and fanbase.

They have very recently walked a lot of this back, but it would be fair to say that the Unity community are still wary of what may happen down the road, and many swore off Unity and switched development over to Epic's **Unreal Engine** and the rapidly rising **Godot**.

So why are we using Unity?

When choosing a game engine, we need to consider a few factors…

- What are the needs of our project?
- How big is the team?
- How much development time do we have?

WHY UNITY?

Unity is largely used in Indie-AA titles. It is designed more for solo dev and small studio. It also natively handles 2D very well. As we will be working in 2D for our projects in the first year, and in small groups, it meets this requirement ideally.

Wait, can’t UE5 do 2D too?

Technically yes, but it requires plug-ins to do so (support for the 2D template project in UE5 was discontinued a few updates ago). Due to its massive feature set, Unreal is really at home with larger studios producing AAA titles. It can be used by solo devs on smaller projects and the option to use it for your assessments will certainly be there as you continue through the course.

And Godot?

Yes, Godot can do 2D, but it’s still growing. Unity arguably has more documentation and support available at this stage.

Later in the course you will likely choose which engine to use and be expected to provide a rationale for it.

WHY UNITY?

What can Unity do for us?

- Lots of features
  - Huge list of features for 2D and 3D games
  - C# scripting language – part of the C family, along with C++
- Cross platform
  - Can be used on Windows/Linux/Mac
  - Can deploy to PC, Consoles, Mobile devices
- Component orientated
  - Game Object behaviours via Script Objects
  - Promotes OO thinking and re-use of components.

WHY UNITY?

What can Unity do for us?

- Documentation
  - Very good documentation on Unity's own site, plus a wide user-base of online tutorials, books, community support through forums.
- Cost
  - Free for use by students to download and use off-campus.
  - Academic licence version in the Games Lab and Learning Lab.

- IMPORTANT – If installing off-campus onto your own machine, please make sure you use the same version number as installed in the lab.

WHAT IS A GAME ENGINE?

- A software framework (building blocks) for games creation
- Typically provide
  - 2D/3D graphics
  - Physics Engine
  - Audio
  - Scripting
  - Animation
  - Artificial Intelligence
  - Networking etc.

WHAT IS A GAME ENGINE?

- Aims:
  - Increase software re-use
    - Multiple titles for the investment in engine code
    - Reduced development time
    - Increase profit

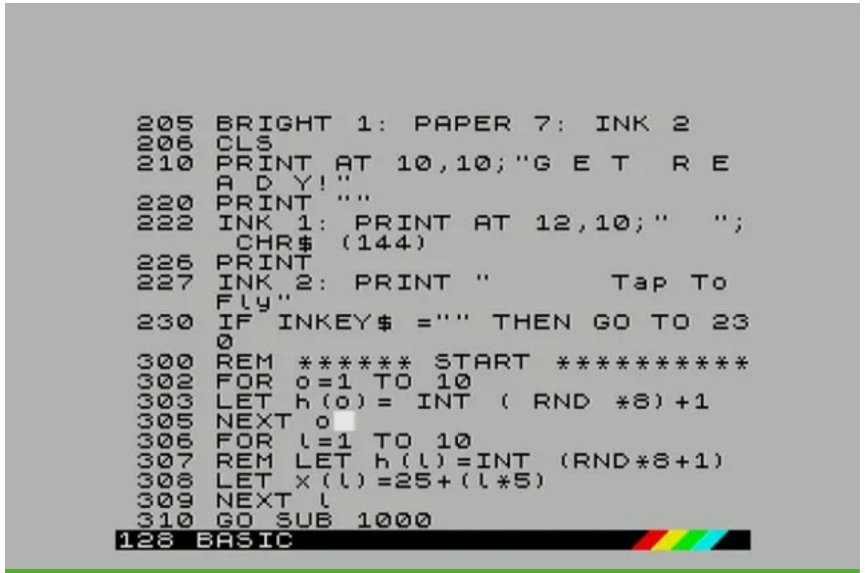
IO Interactive's Glacier Engine has been used to make every *Hitman* game. It was built around the needs of the project, as there was not an off-the-shelf engine that could do what they needed (ioi, 2024).

  - Provide platform independence
    - Lower levels of the engine connect to OS libraries. These are the only parts than need to be changed.
    - Increase profit

WHAT IS A GAME ENGINE?

A little history…

- Early games written as ‘one offs’
  - Very tight coupling to hardware (which was fast-moving) and would need to be entirely rewritten for different computers.
  - Minimal re-use of components in the 80s
  - Doom in the 90s made it easy for gamers to alter the sprites – modding.
  - Late 90s – A definite move to separate the game engine from the game.



```
205 BRIGHT 1: PAPER 7: INK 2
206 CLS
210 PRINT AT 10,10;"G E T  R E
    A D Y!"
220 PRINT ""
222 INK 1: PRINT AT 12,10;"   ";
     CHR$ (144)
226 PRINT
227 INK 2: PRINT "          Tap To
    Fly"
230 IF INKEY$ ="" THEN GO TO 23
    0
300 REM ****** START **********
302 FOR o=1 TO 10
303 LET h(o)= INT ( RND *8)+1
305 NEXT o
306 FOR l=1 TO 10
307 REM LET h(l)=INT (RND*8+1)
308 LET x(l)=25+(l*5)
309 NEXT l
310 GO SUB 1000
128 BASIC
```

In the 1980s, games were written in one huge block of code.

PROBLEM: HOW DO I GET STARTED WITH UNITY?

Motivation

- Two sub-problems:
  - How do I start Unity?
    - Create an account
  - How do I create a new project?
    - Simple, but you need to set some options on creation.

Solution(s)

- Worksheet
  - The only way to learn is to do, make mistakes, and try again.

PROBLEM: UNITY IS A BIG APPLICATION, WHAT DOES ALL THIS STUFF DO?

Motivation

- Unity is a game engine with a lot of features
- The editor provides access to those features, hence the level of complexity

Solution(s)

- Demos
  - We will go through it live in class.
  - Worksheets with simple exercises – Practice breeds familiarity.
  - Don’t be afraid to play around with the editor yourself.