Introduction to Unity

Salman Cheema
Lecture #3
Monday, January 13th, 2014

What is Unity3D

- Game Development Tool
  - www.unity3d.com

- Useful Features
  - Imports 3D models easily*
  - Terrain Modeling Tool
  - Integrated Physics Engine (NVidia PhysX)
  - Audio
  - Networking
  - Highly scriptable (C#, Boo and Javascript)
  - Very easy to prototype games

- Free and Pro versions
  - Pro version has more features, e.g. ability to play videos
Why use Unity?

<table>
<thead>
<tr>
<th>Unity</th>
<th>OpenGL, DirectX, XNA</th>
</tr>
</thead>
<tbody>
<tr>
<td>• High Level</td>
<td>• Low level</td>
</tr>
<tr>
<td>• Little programming</td>
<td>• Lots of Programming</td>
</tr>
<tr>
<td>• Components already available</td>
<td>• Start from scratch</td>
</tr>
<tr>
<td>• Highly visual</td>
<td>• Result isn’t immediately visible</td>
</tr>
</tbody>
</table>

Goals for Today

• Unity’s Editor
• Prefabs
• Scripting
• Creating a small FPS
  • Camera Setup
  • Terrain Modeling and Lighting
  • Importing/using Assets (Models, Textures, Sounds, particles)
  • Basic Player State Management
  • Collision Detection
  • Simple Enemy AI
• Using the Kinect with Unity3D
The Unity Editor

- **Scene View**
  - Position/manipulate objects in the scene

- **Inspector**
  - Alter properties of game objects
  - Can also be used at runtime

- **Project**
  - Shows everything in current project
  - Directory Structure

- **Hierarchy**
  - Contains objects in currently loaded scene

Components of a Unity Project

- **Scenes**
  - Initial Menu, Game Level(s), High Scores, ...

- **Game Objects**
  - Player, Geometry, Particles, Camera(s), ...

- **Scripts**
  - Behavior for Player, NPCs, Weapons, Collisions, ...

- **Other Resources**
  - Sounds, Fonts, Images, Prefabs ...
Scripting

- Most important aspect of a game
  - Change behavior of Game objects
  - Only form of required programming
  - Can be written in C#, JavaScript, or Boo

- Some Important Methods (for overriding)
  - Start: Called when a script is instantiated
  - Update: Called once every frame
  - FixedUpdate: Physics update
  - OnGUI: Used to display GUI (score, health, ...)
  - OnCollisionEnter: Collision Detection
  - OnTriggerEnter: Collision with a Trigger

- Online Manual

Scripting: Important Classes

- General Purpose
  - GameObject, Input, Application, ...

- Mathematics
  - Vector3, Quaternion, Mathf, Ray, ...

- Audio Related
  - AudioClip, audio, ...

- Physics Related
  - Rigidbody, Collider, Physics, ...

- GUI Related
  - Texture2D, GUI, ...
Example: Small First Person Game

- Create a project
- Model terrain
- First person controller
- Mess around with Lights & Flare
- Set up a health meter

Building Stuff in your Game

- Everything in a scene - 'GameObject'
  - Examples: Characters, Power Ups, Explosions, ...

- GameObject is a Container
  - Can be empty
  - Can have custom components
    - Scripts, Colliders, RigidBodies
  - Can be arranged in hierarchy (Parenting)
  - Useful for constructing complicated objects

- Example: Create a Health Powerup
  - Add 3D model for Heart
  - Position point lights to light the model
  - Animate Powerup
Prefabs

- Prefabs = “Templates”

- What if you want a bunch of powerups
  - Solution: Create a template (prefab) and clone it

- Changes to template applied to all instances

- Example: Create a prefab for health powerups

Using the Integrated Physics Engine

- Rigidbody Component
  - Forces, velocity, ...

- Collider Component
  - Box, Sphere, Capsule, Mesh

- Trigger
  - Ignored by the physics engine
  - Can be used to trigger game events, cut scenes, etc

- Example: Write code to collect health powerup
Scripting Examples

- Show Health Bar (done)
- Animate Power Up (done)
- Script to Pick Powerup
- Enemy AI script
- Combat

Scripting : Fine Print

- A script can be applied to multiple game objects
  - Each game object gets own copy
  - Public variables visible in Inspector
    - modifiable at runtime
    - Can drag and drop

- Be careful with parenting and tags
  - Components referenced in script may be within children

- Make use of Debug.Log for debugging

- *Be Cautious: Build incrementally*
Helpful Unity Links

- Models
  - Google Sketchup warehouse
- User Manual
- Components Manual
- Script Reference

Using the Kinect with Unity3D

- Use the Kinect-Unity Interface Plugin
- Pre-requisites
  - Microsoft Kinect SDK
- Tested Configuration: Unity 3.5.7f6 with Microsoft Kinect SDK 1.7
Interfacing Kinect with Unity

- Instructions to get dll (download from GitHub)

- Write unity script to interface with DLL

- Key Components
  - KUInterface.dll
  - Put in “Assets/Plugins” folder in your project

- Example

Questions?