COMPSCI 715 Advanced Computer Graphics

Unity Basics



Today's Mission



- 1. How does a game engine like Unity work?
- 2. What are game objects and components?
- 3. How do you apply this to your own project?

Schedule 1st Half



Week	Activities	Assignments
1	Tue: course outline, Wed: project topics, Thu: feedforward learning (guest lecture)	Register teams
2	Tue + Thu: exergaming (guest lectures), Wed: academic writing overview, abstracts	Abstract (1.5%)
3	Tue + Thu: Unity, Wed: writing introductions	Introduction (2.5%)
4	Tue: feedforward (guest lecture), Thu: Unity, Wed: writing about related work	Related work (2.5%)
5	Team meetings, Wed: design & implement.	1st prototype (1.5%)
6	Team meetings, Wed + Thu: demos (2.5%)	2nd prototype (1.5 %)

Mid-semester break. So far 12% of individual assignments.

Unity Resources

- Unity Documentation: <u>http://docs.unity3d.com/</u>
- Unity Tutorials: <u>http://unity3d.</u> <u>com/learn/tutorials/modules</u>
- Many video tutorials on YouTube, e.g. <u>https://www.youtube.com/watch?</u> <u>v=9Xr5Rc9Rw6I&list=PLmQnFpk1W81tyuEySbOJ4bG6</u> <u>Z1BrS_0hi</u>
- Thanks to Michael Ivanov for some illuminating figures: <u>http://www.slideshare.net/sasmaster/unity3d-</u> programming-5725801

Unity Introduction

A game engine

- Abstraction: sits on top of OpenGL (ES), DirectX, ...
- Complete: provides all features you need for a game, e.g. graphics, physics, sound, input, networking...
- Inversion of control: the engine runs the game
 - Specific game content/features/behavior are plugged into and managed by the engine
 - Don't call us, we call you: engine calls event handlers

Why Unity?

- Free version available, lots of free resources
- Multi-platform: supports most mobile, desktop & console OSs, browser plugin





Unity Overview



GameObjects

Games consist of them

- Think of visible objects in a game
- But also invisible objects for logic, state etc.
- Can be organized hierarchically in a Scene



Assets Script File Animation (Resource Library) Prefab Material Shader

What a GameObject can do depends on its Components

Camera

Mesh Filter

- Technically Components are themselves objects
- Are just associated with GameObject and can reference it
- Give a GameObject more features by adding components, e.g. visual appearance, physics, dynamic behavior
- Knowing Unity's capabilities means knowing the different components

Unity GUI Overview



Creating & Transforming Objects

Use menu GameObject -> Create (e.g. Create Other -> Cube) or just drag & drop from Assets

- Object appears in current Scene
- Combine objects by dragging into other object in Hierarchy
- Name it, enable/disable it
- Put object on Layer to organize groups of objects
- Tag objects to retrieve them more easily

Transform Component

- Every GameObject has it
- Defines public properties: Position, Rotation, Scale
- Grid cells usually used to mark 1 meter

O Inspector 🔒 →=				
👕 🗹 Cube	🗌 Static 🔻			
Tag Untagged	‡ Layer Default \$)		
🔻 🙏 🛛 Transform	[2] \$,	,		
Position	X 0 Y 0 Z 1.95314			
Rotation	X 0 Y 0 Z 0	1		
Scale	X 1 Y 0.32499 Z 1			

Visual Appearance

MeshFilter

- Selects a mesh for the object from Assets
- Can import new assets from asset packages or files

MeshRenderer

- Select Material for object, e.g. colors, textures, reflective properties...
- Select Shader to use, e.g. Diffuse, Specular...



Camera

- Camera Component makes an object a camera
- Typical camera properties:
 - Perspective/Orthographic Projection
 - Field of View: wide or narrow
 - Clipping planes: near/far visibility
 - Viewport aspect ratio
 - Culling mask: what to draw
 - Clear flags: sky color
- Note: Cogwheel -> Reset

Inspector						2	-=
👕 🗹 Main Ca	mera		(Stat	tic	-
Tag MainCam	nera 🛊	Lay	er Def	aul	t		ŧ
🔻 🙏 🛛 Transfor	rm				(2	\$,
Position X -0	0.0465	Y 0	.18389	z	0		
Rotation X 0		Y 0		z	0		
Scale X 1		Υ 1		z	1		
🔻 💼 🗹 Camera					(2	\$,
Clear Flags	S	Skyb	ox				ŧ
Background							Þ
Culling Mask	E	Very	thing				ŧ
Projection	F	ers	pective	_	_	_	ŧ
Field of View	_	_		_	67	_	=
Clipping Planes	: N	ear	0.3	-		_	-
	Fá	ar	1000				
Viewport Rect							
X 0		Y	0				
W1.5		Н	1				
Depth	-	1		_	_	_	_
Rendering Path		Jsel	Player Se	etti	ngs	-	ŧ
		-					=
			Deferre require:	d li s U	ghti nity	ng Pr	о.
Target Texture	N	one	(Rende	er T	Гех	tu	0
Occlusion Cullir	ng 🔽	1					
HDR]					

Other Visible Objects

Light Component

- GameObject -> Create Other -> Directional / Point / Spot Light ...
- Define light properties, e.g. color, intensity

SkyBox Component

- Textured environment around your scene
- Edit -> Render Settings -> Skybox Material (or add SkyBox component to camera)
- Get Skybox material from Assets -> Import Package -> Skyboxes

Inspector	<u></u>	
🍟 🗹 Point light	🗌 Static	-
Tag Untagged	‡ Layer Default	ŧ
📕 🙏 🛛 Transform	۱ <u>آ</u>	\$,
Position X -0.2	225 Y 1.93771 Z 2.062	230
Rotation X 0	Y 0 Z 0	
Scale X 1	Y 1 Z 1	
🛚 💡 🗹 Light	[]	\$,
Туре	Point	ŧ
Range	10	
Color		Þ
Intensity	-0 1.44	
Cookie	None (Texture)	0
Shadow Type	No Shadows	ŧ
Draw Halo		
Flare	None (Flare)	0
Render Mode	Auto	ŧ
Culling Mask	Everything	ŧ
Lightmapping	Auto	ŧ



💽 **\$**,

Fog	
Fog Color	P
Fog Mode	Exp2 \$
Fog Density	0.01
Linear Fog Start	0
Linear Fog End	300
Ambient Light	P
Skybox Material	Sunny3 Skybox ⊙

Physics

RigidBody Component

- Makes object subject to physical forces, e.g. gravity or impact
- Define physical properties, e.g. mass & drag
- Define physical constraints, e.g. object can only move vertically
- Test by clicking play to start game engine

Collider Components

- E.g. BoxCollider, SphereCollider etc.
- Detects collision between objects
- Physics only works if all involved objects have colliders

🔻 🙏 🛛 Rigidbody	🛐 \$,
Mass	1
Drag	0
Angular Drag	0.05
Use Gravity	\checkmark
Is Kinematic	
Interpolate	None +
Collision Detection	Discrete +
Constraints	
Freeze Position	□x □y □z
Freeze Rotation	□x □y □z

🔻 🤪 🗹 Вож Collider		\$,
Is Trigger		
Material	None (Physic Material)	0
Center	X 0 Y 0 Z 0	
Size	X 0.48461 Y 0.65496 Z 0.484	61