Generating Virtual Worlds – II
Animation, Physics, Scripting

Derek Juba
Amitabh Varshney
Scripting

Unity Scripting Tutorials
http://unity3d.com/learn/tutorials/topics/scripting
Scripting

Supported languages

- C# (like Java)
- JavaScript (not like Java)
- Boo (like Python)
Scripting

- Scripts are attached to objects

- Scripts can access components on their object and on other objects
  - For example, access Rigid Body component and change physics properties
  - See GetComponent

- Scripts are Event Driven
  - Called when an event is triggered, run, and then return control to Unity
Scripting Events

- **Awake**
  - Called when object is created

- **Start**
  - Called when object is enabled
Scripting Events

- **Update**
  - Called every frame
  - `Time.deltaTime` gives time since last update

- **FixedUpdate**
  - Called at fixed time intervals of length
    `Time.fixedDeltaTime`
Scripting Events

- **OnCollisionEnter / OnTriggerEnter**
  - Object enters a collision/trigger region
- **OnCollisionStay / OnTriggerStay**
  - Object is within a collision/trigger region
- **OnCollisionLeave / OnTriggerLeave**
  - Object leaves a collision/trigger region
Scripting Events

- OnMouseDown / OnMouseUp / OnMouseEtc...
  - Triggered when a mouse button is clicked, released, etc. over an object
  - Useful for GUIs

- Many others
using UnityEngine;
using System.Collections;

public class CoroutinesExample : MonoBehaviour
{
    public float smoothing = 1f;
    public Transform target;

    void Start ()
    {
        StartCoroutine(MyCoroutine(target));
    }

    IEnumerator MyCoroutine (Transform target)
    {
        while(Vector3.Distance(transform.position, target.position) > 0.05f)
        {
            transform.position = Vector3.Lerp(transform.position, target.position, smoothing * Time.deltaTime);
            yield return null;
        }
    }
}
Physics

Unity Physics Tutorials
http://unity3d.com/learn/tutorials/topics/topics/physics
Physics RigidBody

- Component for Unity physics
- Contain physical properties
  - Mass, Drag, Use Gravity, etc...
- Can be combined with Joints
  - Spring, Fixed, Hinge
  - Can be set to break
Physics Colliders

- React to other objects that have a Collider if one has a RigidBody
  - Normally cannot intersect
  - If Collider is a Trigger, objects pass through it

- Simple shapes
  - Sphere, Capsule, Cube
  - Can combine to form more complex shapes

- Generally have simpler geometry than meshes

- Have Material properties
  - Friction, Bounciness, etc.
Physics Optimizations

• Physics can be expensive
• RigidBodies can be set to Kinematic
  – Motion is controlled directly, not by physics
  – Collisions still checked
  – Saves physics computations
• Objects can be set to Static
  – Do not move
  – Saves collision checks between Static objects
Physics Forces

• Scripts can apply forces to object
  – Can apply accelerations (gradual) or impulses (instantaneous)
  – For VR, may want to minimize accelerations of player...

• Forces can be linear or rotational (torque)
Physics Raycasting

Physics engine can collide a ray against objects
- Check line of sight
- Check what is hit by bullet
- Etc.
Other Physics

• Non-rigid-body physics
  - Hair
  - Cloth
  - Fluids (water, smoke...)

• Unity support
  - Built-in
  - External assets
Animation

Unity Animation Tutorials
http://unity3d.com/learn/tutorials/topics/animation
Animation

- Animations are assets, just like Models, Sounds, or Textures
- Can be time-consuming to create
- Recommend using existing, if possible
Animation

- Unity has a built-in animation program, the Animation View

- Create key-frames and Unity will transition between them

- Contains tools to create animation loops
Animation

• Colliders can be adjusted to match object shape during animation

• Scripts can get the current point in an animation
  – Play a sound

• Scripts can also alter the animation
  – Respond to user input or game events
Animation

• Different animations can be automatically blended
  – Idle -> Walk -> Run -> Jump
  – Mix in heavy breathing

• Blend Trees can blend multiple animations
  – different proportions based on various parameters