



## Working with Unity3d

TOMMI HORTTANA, [tommi.horttana@tkk.fi](mailto:tommi.horttana@tkk.fi)  
 PETRI LANKOSKI, [petri.lankoski@taik.fi](mailto:petri.lankoski@taik.fi)

## Unity Basics

- Game Objects
  - Basic building block
  - Hierarchies
- Components
  - Define the behaviour of game objects
  - Transform
  - Meshes
  - Animations
  - Materials
  - Scripts
  - Physics modeling

## Workflow

- Working with Assets
  - Place assets in the Assets folder
  - Unity auto-imports them
  - Only move and delete assets through Unity
  - Regularly backup Assets and Library folders
- Prefabs
  - Sort of blueprints of game objects
  - Should be made for most objects

## Where to do things

- Outside Unity
  - models
  - animations
  - textures
  - sounds
  - GUI graphics
- In Unity
  - levels
  - scripts
  - physics
  - materials/shaders
  - GUI

## Modeling

- Support for Blender (.blend), Maya (.mb, .ma), Cheetah3d (.jas)
- XSI SoftImage & ModTool via FBX
  - colors not supported, use textures
  - plot IK & constraints
  - neutral pose breaks the model
- Test the pipeline from modeling software to Unity

Game Project 2008-2009 5

## Animations

- Imported with the model from the 3D software
- Skeleton
  - Plan for Mixin
    - Consistent Naming
    - Clear Hierarchy
  - Same hierarchy for similar models (i.e. shared skeleton for humanoids)

Game Project 2008-2009 6

## Textures

- Unity can import multilayered PSD & tiff
  - no need to flat them
- Textures for game objects should be  $2^n \times 2^m$  (2, 4, 256, 512, 1024, 2048, ...)
  - older cards does not support textures larger than 1024
- GUI textures
  - disable mip maps
  - use clamp (not repeat)
- Alpha channel better than transparent layer
  - at least in PSD

Game Project 2008-2009

7

## Fonts

- True type fonts (.ttf) supported
- Size definable in import setting
  - duplicate & set size in import setting
- Custom fonts
  - ttf font editor
  - photoshop, etc.
    - fonts are textures in Unity

Game Project 2008-2009

8

## Physics Engine

- Realistic physics engine
  - Ageia PhysX™ physics engine
- Colliders define object shape and size
- Rigid Bodies define physical behaviour
  - gravity
  - persistence of movement
- Physics Materials
  - friction
  - bounciness

Game Project 2008-2009

9

## Scripts

- Needed for game mechanics
  - the physics engine
- Javascript, C#, and Boo available
  - Examples in documentation in Javascript
  - Best support for C# from us
  - Mixing possible in some cases
- Variables can be exposed to the Unity graphical interface
  - good for defining game object specific values

Game Project 2008-2009

10

## Scripts: MonoBehaviour

- Awake()
  - Initializations
- Start()
  - Initializations
  - After all Awake() is executed for all MonoBehaviour instances
- Update()
  - Every called in frame
- LateUpdate()
  - Called after Update()
  - Use for follow camera, etc.
- FixedUpdate()
  - Called every fixed framerate frame
- OnGUI():
  - Called every frame
  - GUI handling goes here

## GUI

- OnGUI()
- Can be scaled to fit different screen sizes
  - GUI.matrix = ...
- Design layout and gui elements for some arbitrary size
  - e.g., 1280x854
- GUI.skin
- GUI & GUILayout methods
  - button, label, box, ...

## Sound

- Supports common audio formats
- Native support for localized 3D sound effects
- Plugin (for example FMOD) needed for more sophisticated soundscapes
  - Sync events with the music beat
  - Modelling echoes caused 3d space
  - Reactive sounds

Game Project 2008-2009

13

## Plugins

- Possibilities to extend functionality of Unity
  - Wiimote support
  - FMOD
  - Audio input
- Plugins are not possible to use with Web Player

Game Project 2008-2009

14

## Optimizing Models

- Aim for 1500–4000 triangles / object
  - combine near objects if they have less than 1500 triangles
- Aim for single material (+ texture) for an object
  - the object will be drawn once for each material; an object with 10 materials is drawn 10 times
  - texture maps
- Characters
  - 2500–5000 triangles
  - 15–60 bones
  - one material

Game Project 2008-2009

15

## Optimizing Scenes

- Pixel lights are expensive
  - object is rendered once for every pixel light
  - use them for dynamic lights
  - light maps for static lights (or draw lights directly to textures)
- Dynamic Shadows
  - soft shadows are more expensive than hard
  - if problems with frame rate: thing using blob shadows and shadow projectors

Game Project 2008-2009

16

## Optimizing Scripts

- Optimize if frame rate drops below 30 in low end target machine
- Avoid heavy operation in Update() and inner loops
- Use MonoBehaviour.enabled to disable/enable scripts
- Java script & Boo: avoid Duck Type

Game Project 2008-2009

17

## Optimizing Scripts

- Light operations: plus, minus, times
- division takes ~20 times more cycles (compared to light operations)
- Heavy operations: square root, sin, cos (~40–50 times more)
- Heavy operations: Normalize, Distance
  - $\text{distance} * \text{distance} > \text{dif.sqrMagnitude}$
- Heavy operations: raycasting, Object.Find, FindObjectOfType, FindGameObjectsWithTag

Game Project 2008-2009

18

## Lies

- Characters:
  - 5000–7000 triangles
  - multiple objects (eyes, eyelids, etc.)
  - toon shader / flat shader
- Location models are rather high poly
  - flat shaders
  - unoptimized in terms of materials
- One directional light
- Postprocessing effects + texture animations
- Playable in PowerBook G4
  - frame rate ~10 (should stay over 30)