Game Development
with Unity
for iOS and Android devices

Johan Sanneblad, HiQ
This presentation can be found online, at http://www.sanneblad.com/
Johan Sanneblad, Ph.D
Business Developer
HiQ Göteborg
2001 - 2005: GapiDraw

Game development tool, created by me for Palm, N-Gage, Symbian, Windows Mobile

Some of the >200 games developed with GapiDraw
2008: AirFight for iOS

Multiplayer dogfight simulator for iOS
Project Leader & Lead Developer
Most recent project:
Project Leader for a large-scale game production for Android & iOS

Launch date: 2012
Agenda

1. Our recent Unity game project
2. Engine choice - cost and startup time
3. Creating a game with Unity in 9 steps
The game project
3D game

**Large levels** split into numerous segments
Detailed characters animated in **Maya**
Look equal to or **better** than current top titles
Our Team Setup 1/2

Project Leader (me)
Responsible for delivering a finished game on time
Cuts features to make the deadline and budget
Coordinates developer tasks

Creative Director
The game designer
Game design documents, copy texts
Ensure that the game is fun to play, and balanced

Art Director
Concept and wireframe sketches
Interface designs
Character drawings for the 3D modelers
Two experienced game developers
Both have significant experience from the games industry

Two experienced 3D environment modellers
Concept art + complete realization of 3D environments

An experienced 3D character modeler + animator
Has worked as animator on titles such as Assassins Creed

A musician that can create STIM free music
A requirement for many clients is that they own the music

Total team: 9 people
Not including people involved at our customer
Engine Choice
#1 Unreal Development Kit
#2 Unity
“Half Life 2 Meets the Companion Cube”
(random YouTube comment)
Rochard - made with Unity
To be launched for **PS3** (PSN) in 2011
Developed by Finnish Recoil Games
Definitely looks **good enough**
UDK versus Unity 1/5

Development start: January 3

Narrowed choice to 2 tools
UDK - Unreal Development Kit
Unity - Game Development Tool

Target devices
UDK: Windows, iOS, consoles
Unity: Mac, Win, Android, iOS, Web, consoles

Host environment
UDK: Windows only
Unity: Mac/Windows (Mac for iOS render & remote)

Development language
UDK: UnrealScript ("the look and feel of C/C++/Java code")
Unity: Boo, C# or Javascript
UDK versus Unity 2/5

Learning time a factor
Unity iPhone 1.0.1: December 2008
UDK for iOS: December 2010

Game Center Integration
Yes, both platforms

Facebook & Twitter
UDK: No, native code not supported
Unity: Yes, requires native code integration (or plugins)

Custom splash screen
UDK: No
Unity: Yes

Support for 2D apps and games
UDK: No (Does not support orthogonal cameras)
Unity: Yes, through plugins such as Sprite Manager 2
UDK versus Unity 3/5

UDK: Visual Scripting with Kismet
UDK versus Unity 3/5

Unity: Third party beta plugins Vizio or uScript
UDK versus Unity 4/5

UDK: Advanced **Material editor**

![UDK Advanced Material Editor](image-url)
UDK versus Unity 4/5

Unity: Third party beta Strumpy Shader Editor
UDK versus Unity 5/5

UDK: Scaleform GFx UI Support
UDK versus Unity 5/5

Unity: Third party plugin EZ GUI
Unity: Be prepared to use (lots of) plugins
## Similar Features

<table>
<thead>
<tr>
<th>Features</th>
<th>UDK</th>
<th>Unity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object import</td>
<td>FBX + ActorX exporter</td>
<td>FBX (Maya, Max) + native</td>
</tr>
<tr>
<td>NVidia PhysX + APEX</td>
<td>“Unreal Lightmass”</td>
<td>NVidia PhysX</td>
</tr>
<tr>
<td>Content browser</td>
<td>With previews</td>
<td>Search for assets</td>
</tr>
<tr>
<td>Animset editor</td>
<td>Morph, sequence editing</td>
<td>Manual setup</td>
</tr>
<tr>
<td>Price</td>
<td>$99 license</td>
<td>Fixed price per developer</td>
</tr>
<tr>
<td>Free &lt; $50,000 income</td>
<td></td>
<td>Basic and Pro editions</td>
</tr>
<tr>
<td>25% of income &gt; $50,000</td>
<td>No royalty</td>
<td>No royalty fees</td>
</tr>
</tbody>
</table>
Key decision points for us:

**Android Support**
Only supported with **Unity**

**Facebook & Twitter integration**
Only supported with **Unity**

**Reliability**
UDK: Only **1 game** launched in December 2010
Unity: **400 000 developers** in December 2010
Platform choice: Unity
Unity Versions

Unity Standard Edition
Free of charge, company may not earn >$100,000

Mobile Platforms for Standard Edition
Android + $400, iOS + $400

Unity Pro, $1,500
+Native code integration (FaceBook, Twitter, Game Center)
+Video playback
+Multiplayer networking and .NET sockets
+Custom splash screen
+Build size stripping
+Static mesh batching (+performance)
+Occlusion culling
+Profiler

Mobile Platforms for Unity Pro
Android Pro + $1,500, iOS Pro + $1,500
Unity Asset Server

Version Control Server for Unity
$500 per developer

Cross-platform: OSX, Windows, Linux
Server is free of charge

Limitations
Scenes are **binary** - not possible to merge changes
No warning when editing changed files on server

Requires additional coordination
We used Skype chats running in the background
Unity Costs

Unity Pro + iOS Pro + Android Pro + Asset Server
Total cost $5,000 (€3,670) per developer

Each license is valid for two machines
“A license of the Software must only be used on one computer at a time. **The Software may be installed on a second computer for sole use by the user of the License** except for discounted Academic versions which may only be installed on a single computer.” (Unity 3.x EULA)

Two developers using one license?
Technically possible, but **not OK** according to the EULA
Unity Third Party Plugins and Content
Unity Asset Store
Unity Asset Store

Antares Universe (VIZIO)

Category: Extensions
Publisher: NeoPax
Avg. Rating: ★★★★☆
Version: 1.02 (beta) (April 03, 2011)
Size: 2.8 MB

Buy $95.00

Description:
"Open Beta period"
Your Dream and your Work from Prototype to Release will come alive in our advanced and unlimited visual environment on any Unity-supported platform. Your previous projects are integrated easily into new environment and will become part of the Universe. There will be nothing impossible for you and you will always be able to see and control everything created by your hands. Forum thread. Support, Tutorial & documentation. Free trial version.

Purchase today at a discount! (Price will be changed after Beta period is over.)

User Reviews
Incredible
Bruno Xavier
If you value speed, flexibility and productivity this tool is a must!

Package Contents
- Editor Default Resources
  - VIZIO Icons Set
    - PNG
      - iconlicence.txt
      - Viziolicon(0).png
      - Viziolicon(10).png
      - Viziolicon(100).png
      - Viziolicon(101).png
      - Viziolicon(102).png
      - Viziolicon(103).png
      - Viziolicon(104).png
      - Viziolicon(105).png

johan@sanneblad.se
Native Plugins

Some plugins are more or less a requirement
Get these when you buy your Unity license

EZ GUI, $199
http://www.anbsoft.com/middleware/ezgui/
The best and simplest way for pixel perfect GUI components

Sprite Manager 2, $149
http://www.anbsoft.com/middleware/sm2/
The best way for animated EZ GUI Buttons and 2D elements

Prime 31: Social Networking, $50
Unity C# interface to Facebook and Twitter integration

Prime 31: Game Center, $75
Unity C# interface to iOS Game Center
Other Things you Need

You need a few other things to get started:

Maya or 3DS Max with subscription
At least one team member should have a license of Maya or 3DS Max to make quick changes to 3D models.
Price: $4,090 (~43,000 Kr)

There are cheaper modeling tools
E.g. the book “Creating 3D Game Art for the iPhone with Unity” uses Modo and Blender for modeling + animation
Luxology Modo: $995
Blender: Free

Modo + Blender might work for single developers
But neither can import FBX files with animation
This means you cannot adjust animations from others

You also need a Mac Pro, MacBook Pro or iMac
Total Cost

Plugin cost (product)
Facebook + Twitter + Game Center + EZ GUI + SM2 = $473

Developer cost
Unity Pro + iOS Pro + Android Pro = $5,000

Typical small team setup with 2 developers
2 x (Unity Pro + iOS Pro + Android Pro) licenses
One license of Maya + subscription
One set of Unity plugins

Total software cost with 2 developers: $14,563
(110 851 kr if Maya is purchased in Sweden)

Add to that the cost of a typical MacBook Pro
Total cost somewhere around ~150 000 kr)
Learning Time

Experienced C++ or C# developer
Approximately 4 weeks full time before full productivity

Most common issues?
1. Game Architecture design
2. GUI design using orthogonal cameras and EZ GUI
3. Handling screen aspect ratios and variations

Some tasks take longer to master
E.g. material / shader programming

So, to sum it up
License cost for 2 developers is ~110 000 kr
Hardware cost for 2 developers is ~40 000 kr
Learning time for 2 developers is ~320 hours

Make sure you can afford to purchase and learn the software before starting your game journey!
unity android
now shipping
Unity for Android

Support for Android added in Unity 3.0
Unity 3.3 first non-beta Android released on March 4, 2011

Can develop for Android using Mac or Windows

Unity for Android is JIT’ed
Final distribution archive is created within editor
Unity for iOS is AOT (Xcode integration, postprocessing)

Max size: 50 MB
Compared to iOS, where a typical game is hundreds of MB

Unity Remote
Forward screen touches and sensor data to Unity
Works in both Windows and Mac OSX

Currently scene loading is MUCH slower than iOS
Known issue, you want to limit your data size on Android
Creating a simple game in Unity
The Ball Game ™
Create main menu graphics in Photoshop
Create a new document 1024 x 768 for iPad

Draw a gradient background

Create normal and touch down buttons

Save background as PNG

Save each button as PNG

Note: each button is saved as a 1024x768 image

Save image as PSD
Create a new document 1024 x 768 for iPad

Draw a gradient background

Create normal and touch down buttons

Save background as PNG

Save each button as PNG

Note: each button is saved as a 1024x768 image

Save image as PSD
Create game level and colliders in Modo
Zoom to right **unit** scale
Unity uses 1 unit = 1 meter

Create game level **objects**

Set **materials**

Create **colliders**
Prefix with Cn (CX = Box Collider, etc)

**Save** as Modo object

Use Modo2Unity script to save **FBX**
Script removes camera, light, etc
Zoom to right **unit** scale  
Unity uses 1 unit = 1 meter  

Create game level **objects**  
Set **materials**  
Create **colliders**  
Prefix with Cn (CX = Box Collider, etc)  

**Save** as Modo object  

Use Modo2Unity script to save **FBX**  
Script removes camera, light, etc
Set up level, lightning, colliders in **Unity**
Create new Unity project
Create some standard folders
Materials, Scripts, Textures

Import Level1.FBX, Menu graphics, Helper scripts
AssignMaterial.js, BuildColliders.js

Drag Level1.FBX to a new Scene
Save Scene as Level1.unity

Change Camera near / far distance

Add a Directional Light
Create Colliders from meshes prefixed with CX
Create Lightmaps

Add Rigidbody + Sphere Collider to our Ball

Change Project Physics settings (precision)
Create new Unity project

Create some standard folders
   Materials, Scripts, Textures

**Import** Level1.FBX, Menu graphics, Helper scripts
   AssignMaterial.js, BuildColliders.js

Drag Level1.FBX to a new **Scene**

**Save** Scene as Level1.unity

Change **Camera** near / far distance

Add a Directional **Light**

Create **Colliders** from meshes prefixed with CX

Create **Lightmaps**

Add Rigidbody + Sphere Collider to our **Ball**

Change Project **Physics** settings (precision)
Create BallController.cs script

# Get device Tilt values

# Normalize Tilt values

# Add Tilt force to the Ball

Drag BallController script to the Ball in Unity
Create BallController.cs script

# Get device Tilt values

# Normalize Tilt values

# Add Tilt force to the Ball

Drag BallController script to the Ball in Unity
Check if the ball falls down
Open BallController.cs script

# Store original Ball position
# Store original Ball rotation
# If Ball Y position < 1.0 then reset values
Open BallController.cs script

# Store original Ball position
# Store original Ball rotation
# If Ball Y position < 1.0 then reset values
Create main menu with EZ GUI
Create a new Scene: MainMenu1024

Set camera to Orthographic
Size = 384 (half height of 1024 x 768)

Add EZ GUI: UI Manager

Add EZ GUI: Simple Sprite for background

Add EZ GUI: Load Scene Button for Play button
Set it to load Level1.unity

Add a material for the background
Shader = Unlit / Texture

Add a material for all buttons
Material will be created by EZ GUI Build Atlases
Shader = Unlit / Transparent

Size and position the Play Button
Using EZ GUI Size Sprites

Add Level1.unity to the Build settings
So it can be loaded by the Play button
Create a new Scene: **MainMenu1024**

- Set camera to **Orthographic**
  - Size = 384 (half height of 1024 x 768)

- Add EZ GUI: **UI Manager**
- Add EZ GUI: **Simple Sprite** for background
- Add EZ GUI: **Load Scene Button** for Play button
  - Set it to load Level1.unity

- Add a material for the **background**
  - Shader = Unlit / Texture

- Add a material for all buttons
  - Material will be created by EZ GUI **Build Atlases**
    - Shader = Unlit / Transparent

- Size and position the Play Button
  - Using EZ GUI **Size Sprites**

- Add Level1.unity to the **Build settings**
  - So it can be loaded by the Play button
Create end screen with EZ GUI
Disable **PVRT compression** on textures
Enable *trigger action* for the game end screen
Add a **Trigger** collider to our Goal object

Add **OnTriggerEnter()** to BallController.cs

Load the level “GameEnd1024” once the ball enter

Add the scene **GameEnd1024** to Build settings

Switch **target platform** to iOS

**Play** the game!
Add a **Trigger** collider to our Goal object

Add **OnTriggerEnter()** to BallController.cs

Load the level “GameEnd1024” once the ball enter

Add the scene **GameEnd1024** to Build settings

Switch **target platform** to iOS

**Play** the game!
Tips and tricks
First tip: Unity PlayMaker Add-on
$99 from the Unity Asset Store
You can create a **complete** 2D or 3D game with Unity & PlayMaker - no coding required!
Performance Budget

From **Unite 2010**: “Unity 3 for Mobile”

Available RAM
iPhone 3GS / iPad: ~120 Mb
iPhone 4G: >200 Mb
RAM is also used as Video RAM

iPhone 3G
7-10K vertices (for 30 FPS)
5K skinned vertices
40-100 objects on screen

iPhone 4 & iPad
30-40K vertices (for 30 FPS)
15K skinned vertices
70-200 objects on screen

iPad 2
>150K vertices (for 30 FPS)
Get Your Units Right

“Unity's default unit scale is 1 unit = 1 meter”
Critical for physics calculations to work correctly

“If you are modeling a human make sure he is around 2 meters tall in Unity”

Set your grid accordingly
Modo default = m
Maya default = cm (need to change to m)
From the **Unity Script Reference**:

“An object falls at the **same acceleration** no matter big or small, heavy or light. If your game world has a large scale, objects will still fall at the **same acceleration**.

But since everything is larger, objects **will appear** like they fall slower”
Screen Sizes

Android: High resolution phones
- 800x480 (Nexus S, Galaxy S, ...)
- 854x480 (Sony Ericsson X10, ...)
- 1024x480 (Acer’s upcoming 4.8” smartphone, ...)
- 960x540 (Motorola Droid 3, ...)

Android: Low resolution phones
- 320x240 (X10 Mini, ...)
- 432x240
- 400x320 (Samsung Galaxy 3 i5800, ...)
- 480x320

Android: Tablets
- 1024x600 (Galaxy Tab 7”, ...)
- 1280x800 (Motorola Xoom, Galaxy Tab 8.9 & 10.1, ...)

iOS
- 480x320, 960x640, 1024x768
Screen Sizes: What To Do

For *games* you can crop background images. If a few pixels are lost, it might not matter that much.

Tablets: 2 images
- 1280x800
- 1024x768 (cropped run-time to 1024x600)
All Android & iOS tablets supported

High resolution phones: 2 images
- 960x640 (cropped run-time to 960x540)
- 854x480 (cropped run-time to 800x480)
iPhone 4 & most Android phones supported

Source graphics always created as 2k pixel images
So we can easily move to future iPad retina displays.
We keep to try as much as possible as *vector graphics*.
Miscellaneous Tips

For pixel perfect graphics with EZ GUI
Orthogonal camera size = screen height / 2

Positioning UI elements with EZ GUI
http://forum.anbsoft.com/viewtopic.php?f=9&t=228&p=1150#p1150

Unity Pro Maya Extensions
Exporting morph targets to Unity and more
http://adammechtley.com/

Automatic collider generation from mesh
http://www.anicombo.com/blog/?p=404
To Summarize

If you need Android, Facebook & Twitter support
You have to choose Unity over UDK

UDK is a complex beast, more so than Unity
Learning time is longer with UDK than with Unity

Plan time and budget for necessary purchases
150 000 kr for two developers working with Unity
320 hours for two developers learning Unity

Once you know Unity however, you can work fast
You can actually work really, really fast

Unity is quickly becoming a “de facto standard”
Everyone seems to be using it

If you want to create 2D apps and games
Then you have to choose Unity over UDK
Thank You!

johan.sanneblad@hiq.se