Using M3G

Mark Callow
Chief Architect
Agenda

• Game Development Process
• Asset Creation
• Program Development
• MIDlet Structure
• A MIDlet Example
• Challenges in Mobile Game Development
• Publishing Your Content
M3G Game Demo

EXTREME AIR
SNOWBOARDING™

Tail Grab

Copyright 2005, Digital Chocolate Inc.
Game Development Process

- Traditional Java Game

Game logic → Compile → Java MIDlet

Package → JAR file

Assets:
- Images
- Sounds
- Music
- Other

Game Platform:
- 2D Graphics
- Sound
- Network
- Proprietary
- Other

Diagram courtesy of Sean Ellis, Superscape.
Screen Image: Boulder Dash® M.E.™
M3G Development Process

- How M3G Fits

Game logic ➔ Compile ➔ Java MIDlet ➔ Package ➔ JAR file

Assets:
- Images
- Sounds
- Music
- Other

Game Platform:
- 2D Graphics
- Sound
- Network
- Proprietary
- Other

Diagram courtesy of Sean Ellis, Superscape.
Screen Image: Boulder Dash®-M.E.™
Asset Creation

- Textures & Backgrounds
  - Expanded game logic
  - Images
    - Image Editor with PNG output. E. g:
      - Macromedia Fireworks
      - Adobe Photoshop
  - Sounds
  - Music
  - Compile
  - 3D Graphics
  - Game Platform
    - 3D Graphics
    - Sound
    - 2D Graphics
    - Network
    - Proprietary
Asset Creation

- Audio Tools

Expanded game logic

Compile

Assets

Images

Music

3D World

Distribute

Audio Production Tool; e.g.

• Sony Sound Forge®

Commonly Used Formats:

• Wave, AU, MP3, SMAF

3D Graphics
Asset Creation

- Music Tools

Expanded game logic

Compile

Java

Assets

Images

Sounds

3D

Music

MIDI Sequencer; e.g.
- Steinberg Cubase

Formats:
- SMAF, MIDI, cMIDI, MFi

Proprietary

3D Graphics
Asset Creation

- 3D Models

Expanded game logic

Compile

Java MIDlet

Assets

Images
Voices
Music

3D World

3d Modeler with M3G plug-in; e.g.

- Lightwave
- Maya
- 3d studio max
- Softimage|XSI

Game Platforms

2D Graphics

Sound
Network
Proprietary

3D Graphics
Demo: Export 3d Model to M3G
Demo: M3G File Check
Demo: On a Real Phone
**Tips for Designers 1**

- **TIP: Don’t use GIF files**
  - *The specification does not require their support*

- **TIP: Create the best possible quality audio & music**
  - *It’s much easier to reduce the quality later than increase it*

- **TIP: Polygon reduction tools & polygon counters are your friends**
  - *Use the minimum number of polygons that conveys your vision satisfactorily*
• **TIP: Use light maps for lighting effects**
  – Usually faster than per-vertex lighting
  – Use luminance textures, not RGB
  – Multitexturing is your friend

• **TIP: Try LINEAR interpolation for Quaternions**
  – *Faster than SLERP*
  – *But less smooth*
Tips for Designers 3

- **TIP: Use background images**
  - Can be scaled, tiled and scrolled very flexibly
  - Generally much faster than sky boxes or similar

- **TIP: Use sprites as impostors & labels**
  - Generally faster than textured quads
  - Unscaled mode is (much) faster than scaled

- **LIMITATION: Sprites are not useful for particle systems**
Agenda

• Game Development Process
• Asset Creation
• Program Development
• MIDlet Structure
• A MIDlet Example
• Challenges in Mobile Game Development
• Publishing Your Content
Program Development

- Edit, Compile, Package

Expanded game logic

Compile

Java MIDlet

Package

JAR file

Traditional
  - Wtk, shell, editor, make, javac

Integrated Development Environment
  - Eclipse
  - Borland JBuilder
  - Sun Java Studio
Program Development

- Test & Debug

Assets
- Images
- Sounds
- Music
- 3D World

Game Platform
- 2D Graphics
- Sound
- Network
- Proprietary
- 3D Graphics

Carrier/Maker supplied SDK
- Emulator
- Real device

Screen Image: Sega/Wow Entertainment RealTennis.™
Agenda

• Game Development Process
• Asset Creation
• Program Development
• MIDlet Structure
• A MIDlet Example
• Challenges in Mobile Game Development
• Publishing Your Content
The Simplest MIDlet

- Derived from MIDlet,
- Overrides three methods

- `MIDlet.StartApp()`
  - [initialize]
  - [request redraw]

- `MIDlet.destroyApp()`
  - [shut down]

- `Canvas.paint()` performs rendering using Graphics3D object.

- Create canvas; load world.

- Tidy up; exit MIDlet.

- And that’s it.
A More Interesting MIDlet

MIDlet.StartApp()
Create canvas; load world, start update thread

Get any user input via Canvas.commandListener

Game logic, animate, align if necessary

Canvas.paint()
performs rendering using Graphics3D object

Wait to ensure consistent frame rate

MIDlet.destroyApp()
Tidy up; exit MIDlet

Exit request

(initialize)

user input

scene update

request redraw

Runnable.run()
Read user input, update scene

draw

wait

Update loop.

Flow-chart courtesy of Sean Ellis, Superscape
MIDlet Phases

- Initialize
- Update
- Draw
- Shutdown
Initialize

- Load assets: world, other 3D objects, sounds, etc.
- Find any objects that are frequently used
- Perform game logic initialization
- Initialize display
- Initialize timers to drive main update loop
Update

• Usually a thread driven by timer events
• Get user input
• Get current time
• Run game logic based on user input
• Game logic updates world objects if necessary
• Animate
• Request redraw
Update Tips

• TIP: Don’t create or release objects if possible
• TIP: Call system.gc() regularly to avoid long pauses
• TIP: cache any value that may not change every frame; compute only what is absolutely necessary
Draw

• Usually on overridden paint method
• Bind Graphics3D to screen
• Render 3D world or objects
• Release Graphics3D
  – …whatever happens!
• Perform any other drawing (UI, score, etc)
• Request next timed update
Draw Tips

- TIP: Don’t do 2D drawing while Graphics3D is bound
Shutdown

- Tidy up all unused objects
- Ensure once again that Graphics3D is released
- Exit cleanly
- Graphics3D should also be released during pauseApp
MIDlet Review

Main loop.

Don’t create/destroy objects if possible

Throttle to consistent frame rate

Keep paint() as simple as possible

Be careful with threads

Diagram courtesy of Sean Ellis, Superscape
Agenda

- Game Development Process
- Asset Creation
- Program Development
- MIDlet Structure
- A MIDlet Example
- Challenges in Mobile Game Development
- Publishing Your Content
Demo: Using M3G MIDlet
UsingM3G MIDlet

- Display Mesh, MorphingMesh and SkinnedMesh
- Meshes loaded from .m3g files
- View can be changed with arrow keys
- Animation of individual meshes can be stopped and started.
- Animation can be stopped and started with a button push.
- Displays frames per second.
import java.io.IOException;
import javax.microedition.lcdui.*;
import javax.microedition.midlet.*;

public class Cans extends MIDlet implements CommandListener {
    Command cmdExit = new Command("Exit", Command.SCREEN, 1);
    Command cmdPlayPause = new Command("Ctrl", Command.SCREEN, 1);
    private TargetCanvas tcanvas = null;
    Thread renderingT = null;
    private String Filename = "/coffee.m3g";

    public void startApp() {
        if (tcanvas == null)
            init();

        renderingT = new Thread(tcanvas);
        renderingT.start();
        tcanvas.startPlay();
    }
}
public void pauseApp() {
    if (tcanvas.isPlaying)
        tcanvas.pausePlay();
    renderingT.yield();
    renderingT = null;
}

public void destroyApp(boolean u) {
    pauseApp()
    tcanvas = null;
}
synchronized public void commandAction(Command c,
    Displayable d)
{
   if (c==cmdExit) {
       notifyDestroyed();
       return;
   } else if (c==cmdPlayPause) {
       if (tcanvas.isPlaying)
           tcanvas.pausePlay();
       else
           tcanvas.startPlay();
   }
}
Using M3G Initialization

// From class Cans
public void init() {
    Display disp = Display.getDisplay(this);
    tcanvas = new TargetCanvas(Filename);
    if (tcanvas.hasException)
        notifyDestroyed();
    tcanvas.setCommandListener(this);
    tcanvas.addCommand(cmdExit);
    tcanvas.addCommand(cmdPlayPause);
    disp.setCurrent(tcanvas);
}
class TargetCanvas extends Canvas implements Runnable
   ...
public TargetCanvas(String m3gFile)
{
   try
   {
      fileName = m3gFile;
      g3d = Graphics3D.getInstance();
      Load();
      w = getWidth();
      h = getHeight();
      cameraManip = new CameraManip(gWorld);
   }
   catch(IOException e)
   {
      System.out.println("loading fails:"+fileName);
      hasException = true;
   }
}
// class TargetCanvas
void Load() throws IOException {
    loadObjs = Loader.load(fileName);
    if (loadObjs==null)
        throw new RuntimeException("M3g file error");

    /* find the world node */
    for (int i=0; i<loadObjs.length; ++i) {
        if (loadObjs[i] instanceof World) {
            gWorld = (World)loadObjs[i];
            hasWorld = true;
            break;
        }
    }

    if (!hasWorld)
        throw new RuntimeException("World node not found; incorrect m3g file?");
meshController =
    (AnimationController)gWorld.find(meshControllerId);
morphingMeshController =
    (AnimationController)gWorld.find(morphingMeshControllerId);
skinnedMeshController =
    (AnimationController)gWorld.find(skinnedMeshControllerId);

    /* Clean up after the loading process. */
    System.gc();
}
public void run()
{
    for(;;) {
        long start, elapsed;
        start = System.currentTimeMillis();
        handleInput();
        repaint(); // Request paint()
        elapsed = System.currentTimeMillis() - start;
        // if (want to measure true frame rate)
        // Unfriendly to system!!
        //renderTime += (int)elapsed;
        // else {
        renderTime += (elapsed < 50) ? 50 : (int)elapsed;
        try {
            if (elapsed < 50) Thread.sleep(50-elapsed);
        } catch (InterruptedException e) { } //
    }
}
synchronized protected void paint(Graphics g) {
    if(loadObjs == null) return;
    g.setClip(0, 0, w, h);
    try {
        g3d.bindTarget(g);
        g3d.setViewport(0, 0, w, h);
        render();
    } finally { g3d.releaseTarget(); }
    g.setColor(0xffffffff);
    g.drawString("fps: " + fps, 2, 2, g.TOP|g.LEFT);
}
TargetCanvas *render* method

```java
void render()
{
    if (isPlaying) {
        frameCount++;
        fps = (int)((1000*frameCount) / renderTime) ;
        /* update the scene */
        gWorld.animate((int)renderTime);
    }
    g3d.render(gWorld);
}
```
protected void keyPressed(int keyCode) {
    int action;
    switch (keyCode) {
        case KEY_NUM1: animState ^= MESH_ANIM; break;
        case KEY_NUM2: animState ^= SKINM_ANIM; break;
        case KEY_NUM3: animState ^= MORPHM_ANIM; break;
        default: {
            action = getGameAction(keyCode);
            switch (action) {
                case DOWN: keyState |= DOWN_PRESSED; break;
                case LEFT: keyState |= LEFT_PRESSED; break;
                case RIGHT: keyState |= RIGHT_PRESSED; break;
                case UP: keyState |= UP_PRESSED; break;
                default: break;
            }
        }
    }
}
protected void handleInput()
{
    int start = 0, end;
    int keyState = getKeyStates();
    int deltaX = 0, deltaY = 0;

    /* Stop & start animation of individual objects by setting
     * active interval on AnimationControllers.
     */
    if (meshController != null) {
        end = (animState & MESH_ANIM) != 0 ? 0 : 1;
        meshController.setActiveInterval(start, end);
    }
    if (skinnedMeshController != null) {
        end = (animState & SKINM_ANIM) != 0 ? 0 : 1;
        skinnedMeshController.setActiveInterval(start, end);
    }
    if (morphingMeshController != null) {
        end = (animState & MORPHM_ANIM) != 0 ? 0 : 1;
        morphingMeshController.setActiveInterval(start, end);
    }
}
if ((keyState & DOWN_PRESSED) != 0) {
    deltaY -= DELTA;
}
if ((keyState & LEFT_PRESSED) != 0) {
    deltaX += DELTA;
}
if ((keyState & RIGHT_PRESSED) != 0) {
    deltaX -= DELTA;
}
if ((keyState & UP_PRESSED) != 0) {
    deltaY += DELTA;
}
if (deltaX != 0 || deltaY != 0)
    cameraManip.rotate( deltaY, deltaX, 0 );
Agenda

- Game Development Process
- Asset Creation
- Program Development
- MIDlet Structure
- A MIDlet Example
- Challenges in Mobile Game Development
- Publishing Your Content
Why Mobile Game Development is Difficult

- Application size severely limited
  - Download size limits
  - Small Heap memory
- Small screen
- Poor input devices
- Poor quality sound
- Slow system bus and memory system
Why Mobile Game Development is Difficult

- No floating point hardware
- No integer divide hardware
- Many tasks other than application itself
  - Incoming calls or mail
  - Other applications
- Short development period
- Tight budget, typically $100k – 250k
Memory

- Problems
  ① Small application/download size
  ② Small heap memory size

- Solutions
  - Compress data ①
  - Use single large file ①
  - Use separately downloadable levels ①
  - Limit contents ②
  - Get makers to increase memory ②
Performance

• Problems
  ① Slow system bus & memory
  ② No integer divide hardware

• Solutions
  – Use smaller textures ①
  – Use mipmapping ①
  – Use byte or short coordinates and key values ①
  – Use shifts ②
  – Let the compiler do it ②
User-Friendly Operation

• Problems
  – Button layouts differ
  – Diagonal input may be impossible
  – Multiple simultaneous button presses not recognized

• Solutions
  – Plan carefully
  – Different difficulty levels
  – Same features on multiple buttons
  – Key customize feature
Many Other Tasks

• Problem
  – Incoming calls or mail
  – Other applications

• Solution
  – Create library for each handset terminal
Agenda

- Game Development Process
- Asset Creation
- Program Development
- MIDlet Structure
- A MIDlet Example
- Challenges in Mobile Game Development
- Publishing Your Content
Publishing Your Content

- Can try setting up own site but
  - it will be difficult for customers to find you
  - impossible to get paid
  - may be impossible to install MIDlets from own site
- Must use a carrier approved publisher
- Publishers often run own download sites but always with link from carrier’s game menu.
- As with books, publishers help with distribution and marketing
Publishing Your Content

• Typical end-user cost is $2 - $5.

• Sometimes a subscription model is used.

• Carrier provides billing services
  – Carriers in Japan take around 6%
  – Carriers in Europe have been known to demand as much as 40%! They drive away content providers.

• In some cases, only carrier approved games can be downloaded to phones
  – Enforced by handsets that only download applets OTA
  – Developers must have their handsets modified by the carrier
Publishers

- Find a publisher and build a good relationship with them
- **Japan**: Square Enix, Bandai Networks, Sega, Namco, Infocom, etc.
- **America**: Bandai America, Digital Chocolate, Jamdat, MForma, Glu Mobile (formerly Sorrent)
- **Europe**: Digital Chocolate, Superscape, Glu Mobile (formerly Macrospace), Upstart Games
Other 3D Java Mobile APIs

Mascot Capsule Micro3D Family APIs

- Motorola iDEN, Sony Ericsson, Sprint, etc.)
  - com.mascotcapsule.micro3d.v3 (V3)

- Vodafone KK JSCL
  - com.j_phone.amuse.j3d (V2), com.jblend.graphics.j3d (V3)

- Vodafone Global
  - com.vodafone.amuse.j3d (V2)

- NTT Docomo (DoJa)
  - com.nttdocomo.opt.ui.j3d (DoJa 2, DoJa 3) (V2, V3)
  - com.nttdocomo.ui.graphics3D (DoJa 4) (V4)

Mascot Capsule Micro3D Version Number
Mascot Capsule V3 Game Demo

DEEP LABYRINTH®
DELUXE EDITION

Copyright 2005, by Interactive Brains, Co., Ltd.
Summary

• Use standard tools to create assets
• Basic M3G MIDlet is relatively easy
• Programming 3D Games for mobile is hard
• Need good relations with carriers and publishers to get your content distributed
Exporters

3ds max
- Simple built-in exporter since 7.0
- www.digi-element.com/Export184/
- www.mascotcapsule.com/M3G/
- www.m3gexporter.com

Maya
- www.mascotcapsule.com/M3G/
- www.m3gexport.com

Softimage|XSI
- www.mascotcapsule.com/M3G/

Cinema 4D
- www.c4d2m3g.com

Lightwave
- www.mascotcapsule.com/M3G/

Blender
- www.bight.ca
SDKs

- Motorola iDEN J2ME SDK
  - idenphones.motorola.com/iden/developer/developer_tools.jsp
- Nokia Series 40, Series 60 & J2ME
  - www.forum.nokia.com/java
- Sony Ericsson
  - developer.sonyericsson.com/java
- Sprint Wireless Toolkit for Java
  - developer.sprintpcs.com
- Sun Wireless Toolkit
  - java.sun.com/products/j2mewtoolkit/download-2_2.html
SDKs

- VFX SDK (Vodafone Global)
  - http://via.vodafone.com/vodafone/via/Home.do

- VFX & WTKforJSCL (Vodafone KK)
IDE’s for Java Mobile

- Eclipse Open Source IDE
  - www.eclipse.org
- JBuilder 2005 Developer
  - www.borland.com/jbuilder/developer/index.html
- Sun Java Studio Mobility
  - www.sun.com/software/products/jsmobility
- Comparison of IDE’s for J2ME
  - www.microjava.com/articles/J2ME_IDE_Comparison.pdf
Other Tools

- Macromedia Fireworks
- Adobe Photoshop
- Sony SoundForge
- Steinberg Cubase
- Yamaha SMAF Tools
Publishers, Japan

- Square Enix
  - http://www.square-enix.com/jp
- Bandai Networks
  - http://www.bandai-net.com/
- Sega
  - http://www.sega.co.jp/
- Namco
  - http://www.namco.com
- Infocom
  - http://www.infocom.co.jp/
Publishers, America

- Bandai America
  - http://www.bandai.com/

- Digital Chocolate
  - http://www.digitalchocolate.com/

- Jamdat
  - http://www.jamdat.com

- MForma
  - http://www.mforma.com/

- Glu Mobile
  - http://www.glumobile.com/
Publishers, Europe

– Digital Chocolate
  – http://www.digitalchocolate.com/

• Superscape
  – http://www.superscape.com/

• Glu Mobile
  – http://www.glu.com

• Upstart Games
  – http://www.upstartgames.com/
Thanks: HI Mascot Capsule Version 4 Development Team, Koichi Hatakeyama, Sean Ellis, JSR-184 Expert Group