



CMER

Centre for Mobile Education and Research

Platforms and Engines

Week II



Overview

- **Platforms and Engines**
- **Tools and SDKs**
- **Netbeans Game Development Walkthrough**



Platforms and Engines

- **Nokia S60**
- **Windows Mobile**
- **PalmOS**
- **Android**
- **Blackberry**



Nokia S60

- The platform is implemented in a diverse range of devices
- Provides developers with a consistent set of technologies
- More than 100 million S60 devices have been shipped since April 2007
- Expected more than 250 million in 2008



Others

- Development can be done using
 - C++
 - Using native Symbian OS APIs and a subset of POSIX libraries
 - Java
 - Using MIDP 2.0 and extensive range of JSRs
 - Flash Lite
 - Python



Tools and SDKs

- Tools exist for Java and C++
- IDEs
 - NetBeans with Mobility Pack
 - Eclipse IDE
- Platform SDKs
 - S60 Platform SDKs
 - Series 80 Platform SDKs
 - Nokia Prototype SDKs



Tools and SDKs (Cont.)

- **Device SDKs**
 - Nokia 6131 NFC SDK
 - Early Series 40 Devices SDK
- **SNAP Mobile SDK**
 - For engaging connected multiplayer games



SNAP Mobile Client SDK

- Scalable Network Application Package
- Enables development of connected Java games
- It's the Java counterpart to N-Gage Arena
- Games can be written for devices that support Java ME or SE
- It includes:
 - API libraries
 - Emulation environments
 - Handset and Network text midlets



Windows Mobile 6

- **Gaming API (GAPI) provides solutions for developers who want to write high-performance, real-time games**



GAPI Goals and Solutions

- **Provide Fast, Animated Graphics**
 - Allows direct access to the display's video frame buffer memory. GAPI defines each different device
- **Use hardware keys for game controls**
 - Allows applications to control hardware keys and map best key mappings for each device
- **Turn off Sounds**
 - Turn off sounds when the game is in the foreground



GAPI Goals and Solutions (Cont.)

- **Hide the menu bar**
 - Manages hiding the menu bar and restoring it to the proper state when quitting or losing focus
- **Prevent disruption of the device**
 - Manages focus issues.
 - Provides calls like `IsForeground`
- **Use vibrate alarm as rumble pack**
 - Future solution. Under development



Windows Mobile

- Windows Mobile provides .NET Compact Framework
- IDE - Visual Studio .NET
- Games can be written in
 - C
 - C++
 - C#



Palm OS

- Many APIs are offered for developers such as:
 - Network
 - Sound
 - Imaging
 - LCD
 - UI

- Supports Java and MDIP 2.0



Android

- **Google's mobile device operating system**
- **What is it?**
- **It is a software stack for mobile devices that includes**
 - **An operating system**
 - **Middleware**
 - **Key Applications**



Android (Cont.)

- Relies on Linux v2.6 for core system services such as
 - Security
 - Memory management
 - Process management
 - Network
 - Driver Model
- Kernel acts an abstraction layer between hardware and the software stack



Android (Cont.)

- Provides many libraries for developers
 - System C Library
 - Media libraries
 - Video playback and recording libraries for such formats as
 - MPEG4, H.264, MP3, AAC, AMR, JPG, and PNG
 - Surface Manager
 - Manages access to display subsystem and composites 2D and 3D graphic layers from multiple applications



Android (Cont.)

- SGL
 - Underlying 2D graphic engine
- 3D Libraries
 - Based on OpenGL ES 1.0 API (uses hardware 3D acceleration where available otherwise software acceleration)
- FreeType
 - Bitmap and vector font rendering
- SQLite
 - powerful & lightweight relational database engine available to all applications



Blackberry

- RIM has their own set of components and APIs for development for the BlackBerry
- No specific game development APIs
 - Using combinations of components and managers creates desired effect



NetBeans Game Dev

- NetBeans now provides a Game Builder
- Required Software for the following tutorial:
 - JDK
 - Netbean 6 with Mobility Pack



NetBeans Game Developer Quick Walkthrough Example

NETBEANS GAME DEV



NetBeans Game Dev

- Open the sample Game Builder project
- Samples -> Mobile -> MIDP 2.0 Samples -> Simple game created with Game Builder
- Open the Game and Browse through its elements
- A game design can contain many:
 - Scenes
 - Sprites
 - Layered Tiles



GameBuilderTutorial - NetBeans IDE 6.0 Preview (M9)

File Edit View Navigate Source Refactor Build Run Profile Versioning Tools Window Help

Projects Files

- GameBuilderTutorial
 - Source Packages
 - <default package>
 - demo
 - DemoGameCanvas.java
 - GameDesign.java
 - GameMidlet.java
 - SpriteRandomMovement.java
 - Resources
 - Project Configurations

Game Builder Navigator

Type	Z	View	Lock	Name	X	Y
	0			Trees	191	3
	1			Karel	36	30
	2			Thomas	87	81
	3			Things	14	101
	4			Water	111	98
	5			Base	0	0

GameDesign.java x

Source Game Builder Forest

0 14 50 100 150 200 254 300 350 400 450

50

101

150

200

250

300

341

350

400

Things, Water, Base



NetBeans Game Dev

- The center is the Scene Editor itself
 - You can drag layer to change their positions I the scene using the mouse.
 - Right-clicking the mouse brings up a number of options available to add, remove and edit layer attributes
- The toolbar on top of the Editor Panel
 - Allows to preview generated code
 - Switches graphical view and more



NetBeans Game Dev

- **Game Builder Navigator (bottom left)**
 - Lists all of the available layers used in the scene as well as attributes
 - Layer type (either Sprite or Tiled Layer)
 - Z order
 - View (layer is visible or not)
 - Lock – can the layer be moved around
 - Layer Name
 - X and Y position of the layer relative to the scene



NetBeans Game Dev

- **Adding a New Level**
 - Create a new scene by clicking Net Scene in the scene editor toolbar.
 - Name the scene and hit OK.
 - This creates a blank scene
- **Add Layers**
 - You can add layers by right clicking inside the scene editor and adding either a sprite or tiled layer



NetBeans Game Dev

The screenshot shows the NetBeans IDE 6.0 Preview (M9) interface. The main window is titled "GameBuilderTutorial - NetBeans IDE 6.0 Preview (M9)". The menu bar includes File, Edit, View, Navigate, Source, Refactor, Build, Run, Profile, Versioning, Tools, Window, and Help. The toolbar contains various icons for file operations, editing, and running. The Projects and Files panels on the left show the project structure, including Source Packages, <default package>, demo, DemoGameCanvas.java, GameDesign.java, GameMidlet.java, SpriteRandomMovement.java, Resources, and Project Configurations. The Game Builder Navigator table is visible, showing the following data:

Type	Z	View	Lock	Name	X	Y
	0			Thomas	100	59
	1			Karel	59	58

The main canvas displays a game scene with a grid. A character named "Karel" is positioned at (59, 58). The canvas also shows a "Desert" background and a "Game Builder" toolbar. The canvas has a coordinate system with X and Y axes ranging from 0 to 450.



NetBeans Game Dev

- **Create a New Tiled Layer**
 - Click the 'Create New TiledLayer' button from the scene editor toolbar
 - Select your tiles
 - Press OK



Create new TiledLayer

TiledLayer name:

Select image:

- /platform_tiles.png
- /topview_tiles.png

Adjust tile size in pixels:

Tile width: 16 px Tile height: 16 px



NetBeans Game Dev

- Next, the Tiled Layer Editor appears
- The editor contains several components:
 - Toolbar (top)
 - Editor Panel (center)
 - Animated Tile List on the right
 - Image Resource Panel (bottom)
 - Allows to select the tile for editing or usage



GameBuilderDemo - NetBeans IDE 6.0 Preview (M9)

File Edit View Navigate Source Refactor Build Run Profile Versioning Tools Window Help

GameDesign.java x DemoGameCanvas.java x

Source Game Builder Sand

Tiled Layer: Sand Rows: 20 Cols: 20

New Animated Tile
AnimWater

Image: /topview_tiles.png Tile Count: 112 Tile Size: 16x16

Index: 1

Output Subversion



NetBeans Game Dev

- Click the Paint Mode button from the toolbar
- select the tile you imported from the resource panel
- Drag the mouse over the editor panel to paint that tile.
- A completed painting example is on the previous slide



NetBeans Game Dev

- **Add A New Tiled Layer to the Scene**
 - Select the saved scene from the combo box in the editor tab
 - Right click inside the scene editor and select
 - AddTiledLayer -> 'Your Selected Layer'
 - Drag the layer to X, Y 0,0
 - Move the layer to the bottom of the sprite list (bottom left) to have all sprites show ON the tile



GameBuilderTutorial - NetBeans IDE 6.0 Preview (M9)

File Edit View Navigate Source Refactor Build Run Profile Versjonning Tools Window Help

Projects Files

- GameBuilderTutorial
 - Source Packages
 - <default package>
 - demo
 - DemoGameCanvas.java
 - GameDesign.java
 - GameMidlet.java
 - SpriteRandomMovement.java
 - Resources
 - Project Configurations

Game Builder Navigator

Type	Z	View	Lock	Name	X	Y
	0			Karel	59	58
	1			Thomas	96	92
	2			Sand	0	0

GameDesign.java x

Source Game Builder Desert



NetBeans Game Dev

- Now that your graphics are complete
- All that remains is to complete the game code!
- Then the game is complete