

Computer Graphics on Mobile Devices

VL SS2010 3.0 ECTS

Peter Rautek



Motivation

- Smartphonemarkt stark wachsend
- Integration von
 - Kamera
 - 3D Grafik
 - Internet
 - PC
 - Telefon
- Anwendungen
 - Spiele
 - Augmented Reality
 - Visualisierung (Google Earth, Maps, etc.)
 - TV



- 3 Einheiten
- Einheit I
 - ◆ Vorbesprechung der Laborübung
 - ◆ Android Basics
- Einheit II
 - ◆ Advanced Android Topics
 - ◆ OpenGL ES
- Einheit III
 - ◆ Plattformübergreifende Entwicklung



- Mobile Plattformen
 - ◆ Android
- Computergraphik
 - ◆ OpenGL ES 1.0 (und 2.0)
 - ◆ Ev. Bildbearbeitung, Modellierung, etc.
- Software Engineering
 - ◆ Version Control
 - ◆ Eclipse oder andere Entwicklungsumgebung
 - ◆ Java



- Ziele
 - ◆ Entwicklung eines gemeinsamen Spieles
 - ◆ Open Source Veröffentlichung
 - ◆ Ev. Veröffentlichung am Android Market
- Gruppenarbeit (in 2er Gruppen)
- Implementieren eines Levels (zu Hause oder im VisLab)
 - ◆ Zwischengespräch
 - ◆ Abgabegespräch
 - ◆ Präsentation
- Optional
 - ◆ FAQ Event (vorraussichtlich 2 Events)



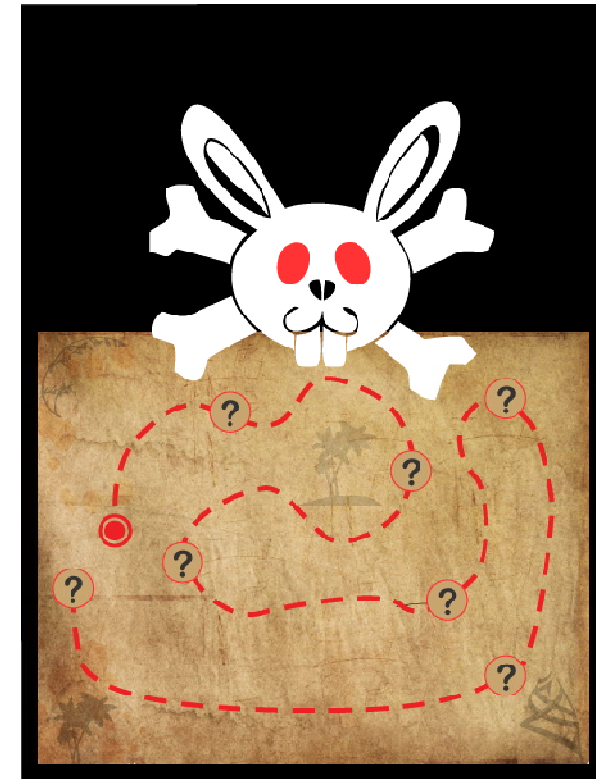
- Informelles Treffen (offen und ohne fixe Struktur)
- Kurzer Vortragsteil über häufige Probleme und Lösungen
- Kurz Probleme und Erfolge besprechen
 - ◆ Level Entwürfe diskutieren
 - ◆ Meinungen einholen
- Coding Session
- Ablauf:
 - ◆ Probleme posten
 - ◆ Code commit
 - ◆ Laptop mitbringen!



- Interne Präsentation der (vorläufigen) Ergebnisse
- 15-30 Minuten pro Gruppe
- Statusbestimmung
- Feedback zu
 - ◆ Status
 - ◆ Gameplay



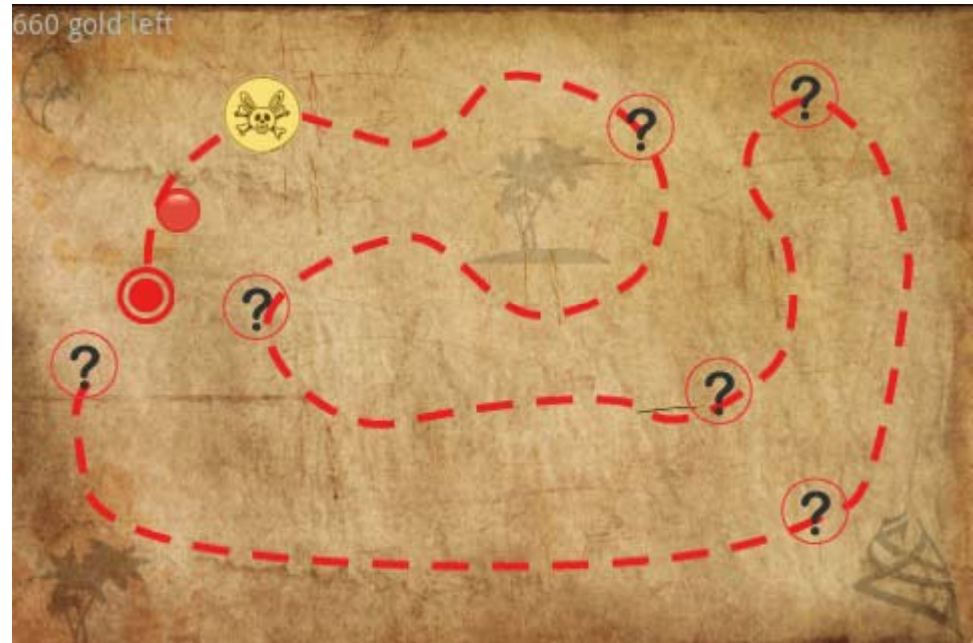
- Inverse Schatzsuche
- Herausforderung: Gold
 - ◆ Verlieren
 - ◆ Verstecken
 - ◆ Ausgeben
- Individuelle Levels
 - ◆ Originelle Umsetzung
 - ◆ Game Play
 - ◆ Ansprechende Graphik



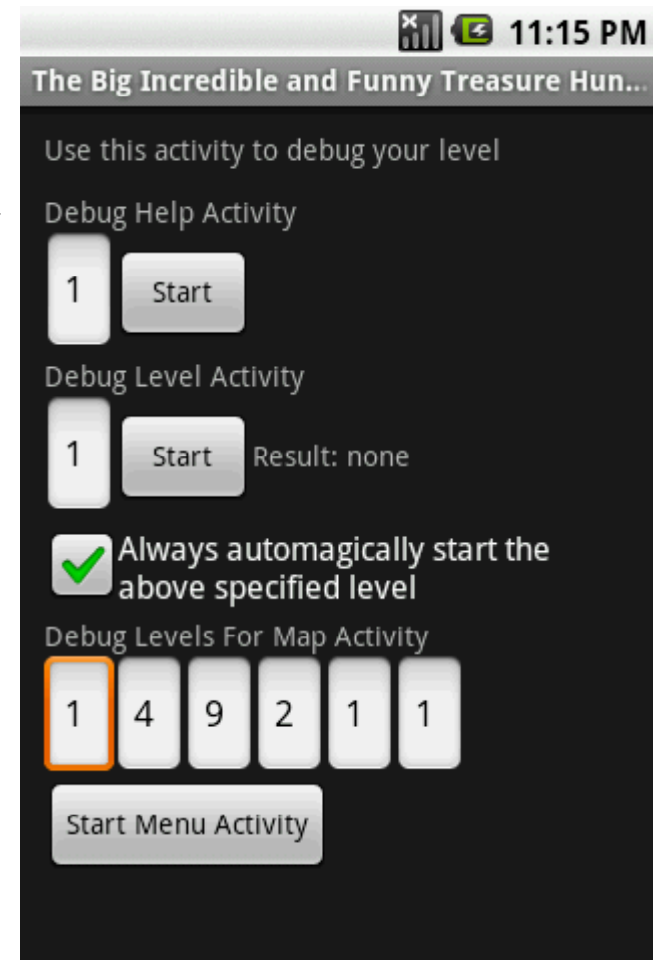
- Menü
- Launcher für Spiel
- Help
- About
- Credits
- Enable/Disable Debugging



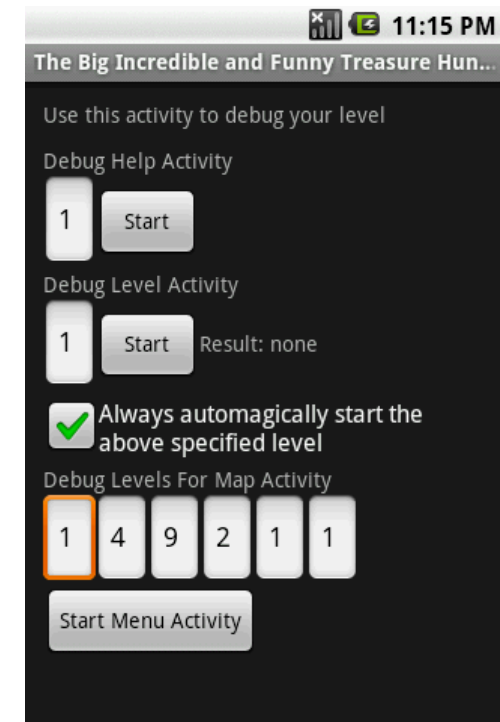
- Map
- Zufällige Zuordnung von Levels
- Launcher für Levels
- Spielerfortschritt
- Levelicons



- Debug
 - ◆ Individuelle Help Activity
 - ◆ Individuelle Level Activity
- Checkbox
 - ◆ Automatischer Start des spezifizierten Levels
- Explizite Zuordnung der Levels



Demo



- Von Anfang an beachten
 - ◆ Spielgeschichte
 - ◆ Game play
 - ◆ Abbildung von Game play auf Hardware
 - ◆ Input/Output
- Einschränkungen
 - ◆ Emulator ist nicht ident mit Hardware
 - ◆ Inhalte
 - ◆ Performance



- Creative Commons
- 2D Graphik
- 3D Modelle
- Level editor
- Anleitung/Hilfe
- Tutorial
- Sound



		Can someone use it commercially?	Can someone create new versions of it?
Attribution			
Share Alike			Yup, AND they must license the new work under a Share Alike license.
No Derivatives			
Non-Commercial			Yup, AND the new work must be non-commercial, but it can be under any non-commercial license.
Non-Commercial Share Alike			Yup, AND they must license the new work under a Non-Commercial Share Alike license.
Non-Commercial No Derivatives			



- 3 Phasen
- Phase I – Alles ausprobieren und Level spezifizieren
 - ◆ Aufsetzen der Entwicklungsumgebung
 - ◆ Android „Hello World“
 - ◆ VL Framework ausprobieren
 - ◆ Eigenes Level Entwerfen
 - ◆ Level Beschreibung (Hilfe) integrieren
 - ◆ Level einchecken (=Anmeldung zur VL)
- Phase II – Implementieren der Funktionalität
- Phase III – Testen und Verschönern



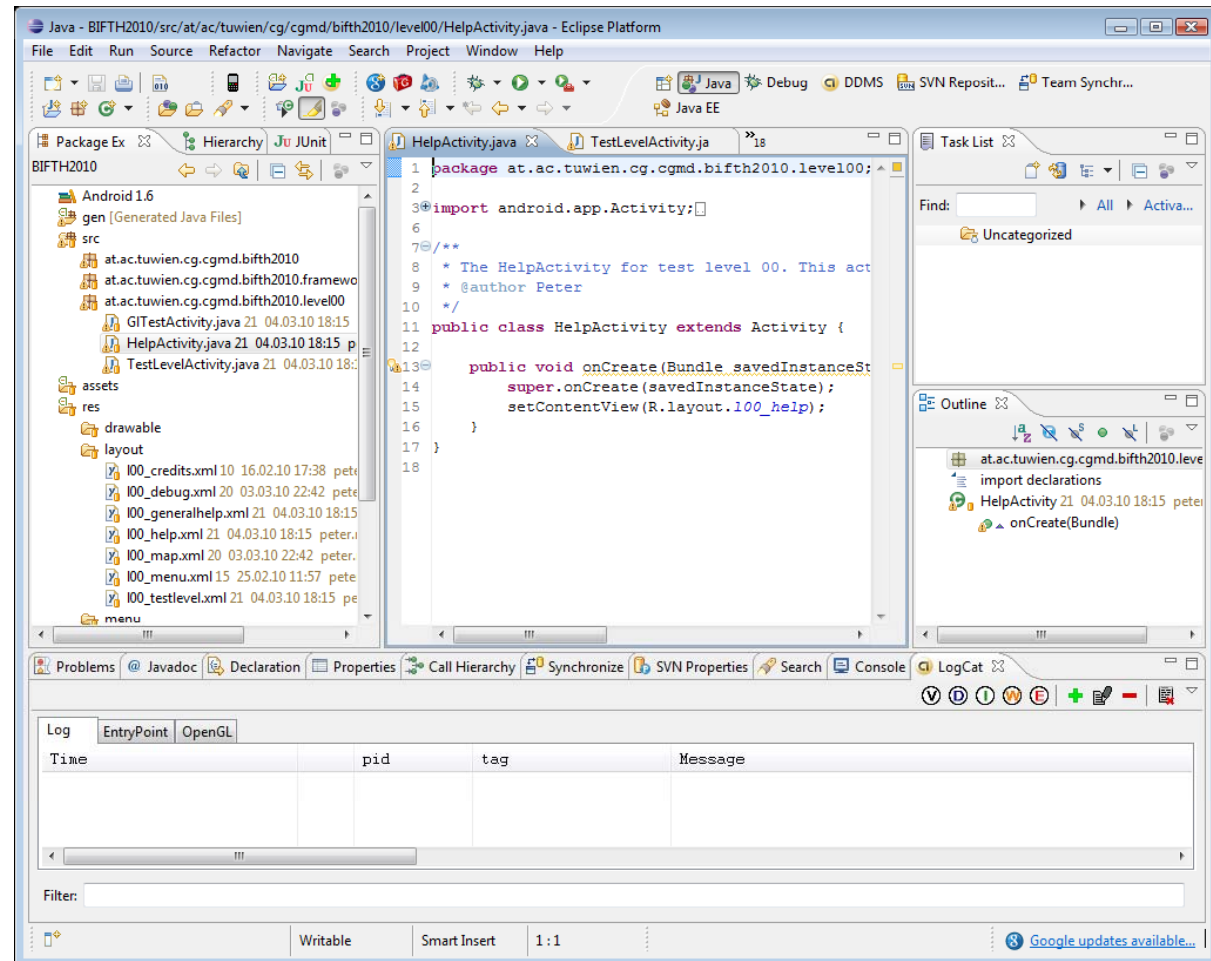
Zeitplan

09.03.2010 15:15	1. Vorlesungseinheit (Vorbesprechung des Übungsteils) Topics: Ablauf der VL, organisatorisches, Thema des Übungsteils, Einführung in Android, Übungs Framework, Aufgabenstellung (Vortragender: Peter Rautek)
16.03.2010 15:15	2. Vorlesungseinheit (Anmeldung zur VL mit der ersten Abgabe) Topics: Android, OpenGL ES (Vortragender: Peter Rautek)
23.03.2010 15:15	3. Vorlesungseinheit Topics: Cross-platform Entwicklung von mobilen Anwendungen (Vortragender: Anton Pirker)
13.04.2010 15:15	FAQ (optional)
20.04.2010 nachmittags	Zwischengespräch: Feedback zur zweiten Abgabe
01.06.2010 15:15	FAQ (optional)
15.06.2010 nachmittags	Abgabegespräch: Beurteilung und Feedback zur dritten Abgabe
22.06.2010 15:15	Präsentationen



Phase I

- Demo
- Voraussetzung
 - ◆ Eclipse
 - ◆ SVN
 - ◆ Android
- Eclipse
- Level Integration
- AndroidManifest
- HelpActivity
- Emulator



- Zwei Möglichkeiten
 - ◆ Direkt im “trunk” arbeiten (empfohlen)
 - ◆ Eigenes Setup – Level später integrieren
- Auf jeden Fall
 - ◆ Namenskonflikte vermeiden
 - ◆ Conventions (lxx_...)
 - Layout
 - Strings
 - View Ids (User-interface)
 - Music, Sounds, Textures, etc.



- Einleitung zur VL

- Fragen ?

- Android 



■ Software stack for mobile devices



- SDK Java
- NDK C and C++
- No native applications
- Emulator / Hardware



- Activity
 - ◆ A screen the user can see and interact with
- ContentProvider
 - ◆ Exchanging content
- Permissions
 - ◆ User will have to grant permissions at install time
- Service
 - ◆ A background process
- Broadcast
 - ◆ Events that might be of interest
- Intent
 - ◆ Late binding mechanism



- Home
- Activity Manager
- Content Provider
- Notification Manager
- Location Manager
- Package Manager
- Camera
- Intents
- Market
- Permissions



- Basic concept for user interaction
- Views & Layouts
- Started with intent
- Return result to calling activity
- Runs in its own thread
 - ◆ Must be kept reactive
 - ◆ Activity not responding (ANR) dialog

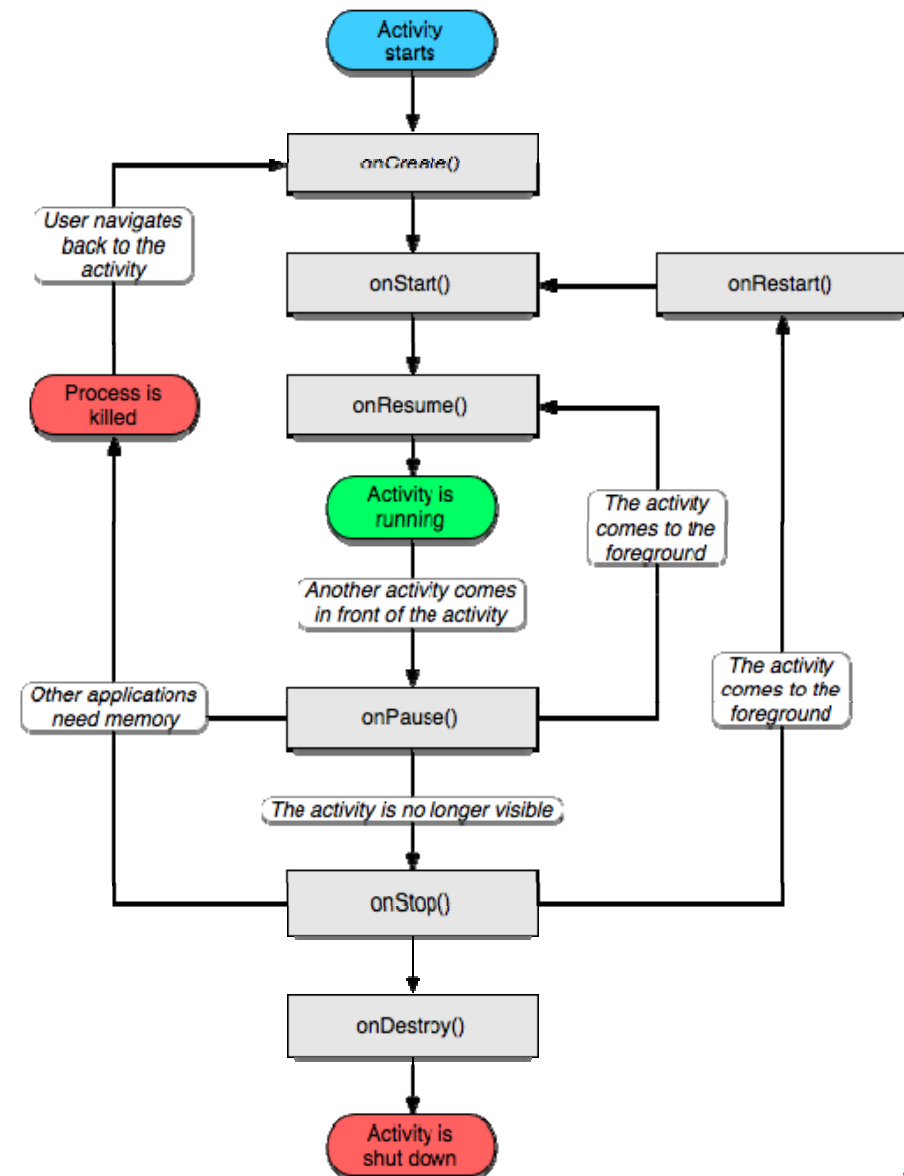


■ State changes

- ◆ Create
- ◆ Start
- ◆ Resume
- ◆ Pause
- ◆ Stop
- ◆ Destroy

■ Multitasking

■ Maintain continuous user interaction



■ Example

- ◆ User plays a game
- ◆ Answers a phone call
- ◆ Enters info into calendar
- ◆ Service syncs calendar, email, etc.
- ◆ User continues the conversation
- ◆ Ends the phone call
- ◆ Goes to toilet - phone enters sleep mode
- ◆ User continues playing game



- Activity might be
 - ◆ Paused:
 - Pause all running threads
 - Store the state
 - ◆ Resumed:
 - Restore state
 - Resume all threads
 - ◆ Destroyed:
 - Stop all threads
 - Store persistent objects



- Xml File
 - ◆ Name, value pairs
 - ◆ E.g., store text, store user progress
- InstanceState
 - ◆ E.g., store cursor position, music track, etc.
 - ◆ onSaveInstanceState(Bundle out)
 - ◆ onRestoreInstanceState(Bundle in)
 - ◆ onCreate(Bundle in)
- Database
- Plain text files



- Resource (./res folder)
 - ◆ Drawables
 - ◆ Layouts
 - ◆ Values
 - ◆ Raw (sound, video, etc.)
- Automatically generated R file (./gen folder)
 - ◆ Easy to reference
 - ◆ Eclipse auto-completion
- Configuration dependent
 - ◆ Changed automatically

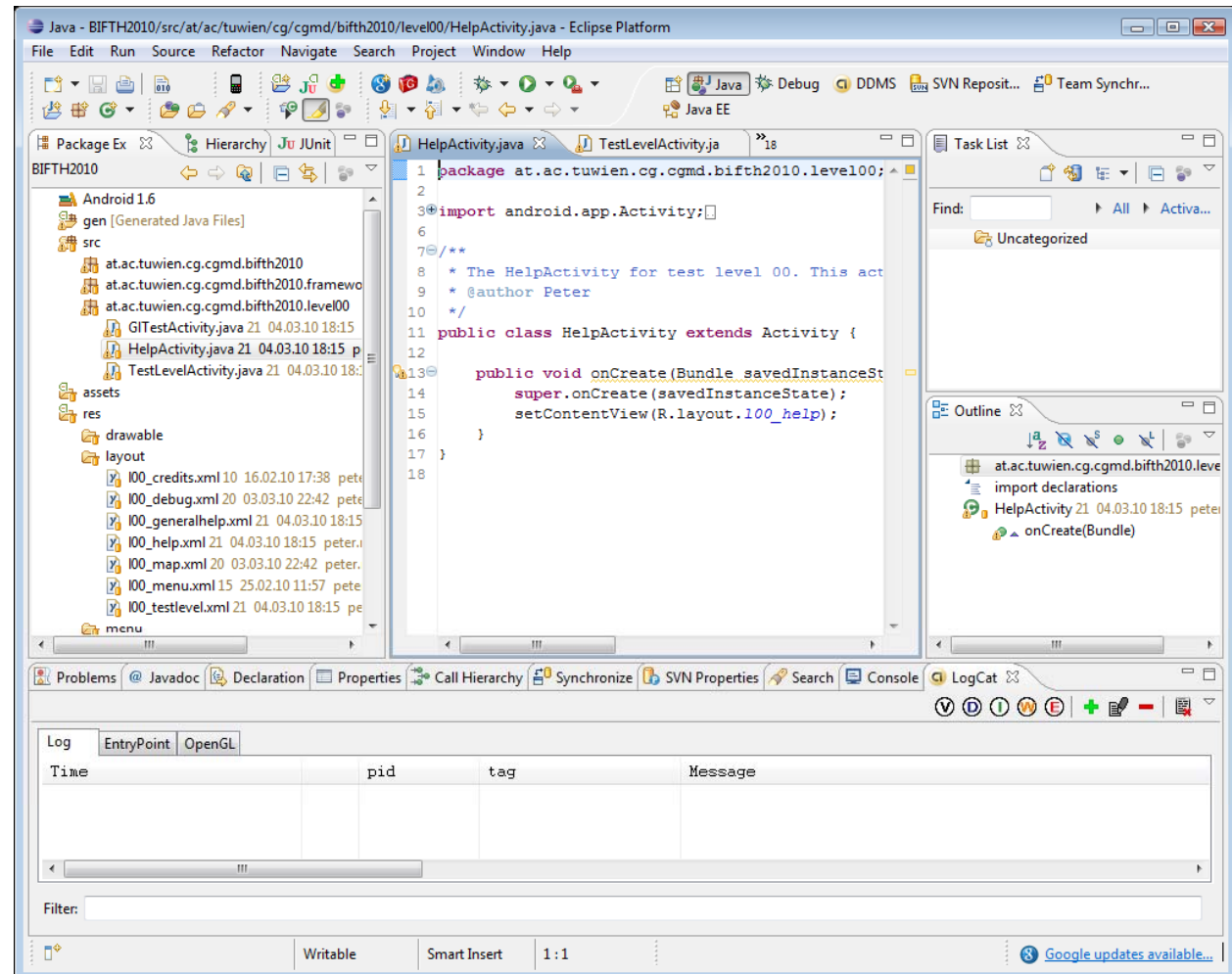


- Android Virtual Device (AVD)
- Configuration
- AVD Manager
- Interaction with Eclipse
- Functionkeys
 - ◆ F1-F12
- F6: Trackball
- Ctrl F12: Landscape – Portrait Mode
- OpenGL ES: no performance estimate



Development with Eclipse

- Breakpoints
- Variables
- Perspectives
- LogCat
 - ◆ Filters
- Callstack
- Subclipse
 - ◆ Team item
 - ◆ Browser
 - ◆ Import



Where to Start? – Links!

- Download these slides and explore the links
- CGMD Homepage
<http://www.cg.tuwien.ac.at/courses/CGMD/>
- Google IO Videos:
 - ◆ Android basics
<http://sites.google.com/site/io/building-an-android-application>
 - ◆ Real time games for Android
<http://code.google.com/events/io/2009/sessions/WritingRealTimeGamesAndroid.html>
- Khronos OpenGL ES
http://www.khronos.org/opengles/1_X/
- Blogs about Android and OpenGL ES
 - ◆ Robert Green <http://www.rbgrn.net/topic/programming/all>
 - ◆ Replica Island <http://replicaisland.blogspot.com/>
 - ◆ Per-Erik Bergman <http://blog.jayway.com/author/pererikbergman/>



Fragen?



Viel Spass beim Coden!

