

Prof. Dr. Michael Rohs, Dipl.-Inform. Sven Kratz

Mensch-Maschine-Interaktion 2 (Mobile Interaktion) WS 2011/2012

Übungsblatt 2

Aufgabe 1: Android 2D Grafik, Touch Input [Einzelabgabe]

Schreiben Sie eine Anwendung, mit der der Benutzer mit dem Finger durch ein Labyrinth auf dem Display navigieren kann (siehe Beispiel-Screenshots unten). Die Aufgabe des Benutzers besteht darin, sich mit dem Finger ausgehend vom Start-Quadrat oben links zum Ziel-Quadrat unten rechts durch das Labyrinth zu bewegen, ohne den Finger vom Display abzuheben. Jede Berührung einer Labyrinth-Wand soll als Fehler gezählt und unten links angezeigt werden. Bei erfolgreicher Ausführung (keine Fehler) soll eine entsprechende Mitteilung auf dem Display erscheinen (z.B. als Meldung unten rechts).

Bei der Lösung der Aufgabe dürfen Sie ein Programmgerüst verwenden, das auf der Webseite der Vorlesung zu finden ist. Dieses kann einfach in Eclipse importiert werden. Das Labyrinth ist in einer XML-Datei spezifiziert, die in MainActivity.java eingelesen wird. Die Wände des Labyrinths liegen in einem Array (lines) vor.

Da der Finger den Berührungspunkt verdeckt, soll ein „Offset-Cursor“ verwendet werden. D.h. der Berührungspunkt wird nach oben versetzt, so dass er nicht vom Finger verdeckt ist (siehe Abb. 1, rechts). Diese Idee wurde zuerst beschrieben in: Potter, Weldon, Shneiderman: Improving the accuracy of touch screens: an experimental evaluation of three strategies. CHI 1988.

Die Lösung der Aufgabe soll im Einzelnen wie folgt vorgenommen werden:

- a) Importieren Sie das Programmgerüst in Eclipse.
- b) Implementieren Sie ein Options-Menü (als Bestandteil der MainActivity), mit dem man zwischen 0 Offset und 100 Pixel vertikalem Offset umschalten kann.
- c) Zeichnen Sie das Labyrinth in der Methode onDraw der Klasse LabyrinthView. Verwenden Sie die Linien-Daten aus der XML-Datei (gespeichert in Lines).
- d) Zeichnen Sie die Start- und Zielfelder.
- e) Fügen Sie Berührungspunkte, die über onTouchEvent gemeldet werden zu einer Liste bzw. einem Array hinzu. Zeichnen Sie die eingegebene Spur. Bei erneuter Eingabe (MotionEvent.ACTION_DOWN) soll eine neue Spur gezeichnet werden.
- f) Implementieren Sie die Methode countCollisions, die die Anzahl der Kollisionen der Eingabespur (bzw. der Strecke zwischen den beiden letzten Berührungspunkten) mit den Labyrinth-Wänden berechnet. Beachten Sie die Kommentare im Programmgerüst.
- g) Geben Sie die bisherige Anzahl an Kollisionen links unten auf dem Display aus. Optional: Falls Sie ein Gerät benutzen, geben Sie Vibrations-Feedback, falls eine Kollision erfolgt.
- h) Geben Sie Feedback, falls das Labyrinth erfolgreich durchquert wurde.
- i) Erstellen Sie ein zweites, einfacheres Labyrinth in XML, das ohne Offset-Cursor bedient werden kann. Wie groß sollten die Wege zwischen den Linien dabei gewählt werden (so dass auch ohne Offset-Cursor kaum Kollisionen vorkommen)?

Hinweise: Verwenden Sie keine Klassen des Packages `android.gesture`.
Das Programm muss kompilieren, sonst wird es nicht korrigiert!

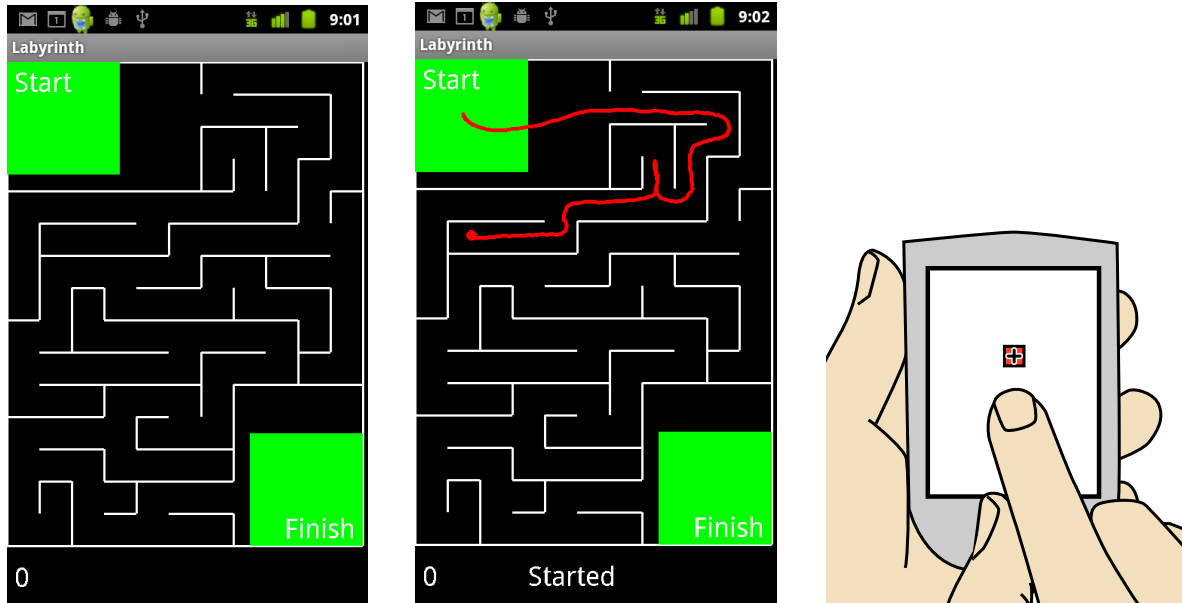


Abbildung 1. Links: Beispiel-Labyrinth. Mitte: Labyrinth mit Eingabe-Spur. Bild rechts aus: Potter, Weldon, Shneiderman: Improving the accuracy of touch screens: an experimental evaluation of three strategies. CHI 1988.

Abgabe

Achtung: Plagiate sind verboten und führen zum Ausschluss aus der Veranstaltung!
Dieses Übungsblatt muss einzeln, nicht in Gruppen, bearbeitet werden. Exportieren Sie Ihr Projekt aus Eclipse (Export → Archive file) und geben Sie es als zip-Datei bis Montag, den 7.11.2011 um 12:00 Uhr im **neuen** UniWorX Portal (<https://uniworx.ifi.lmu.de/>) ab. Sie sollten Ihre Lösung in der Übung vorstellen können.