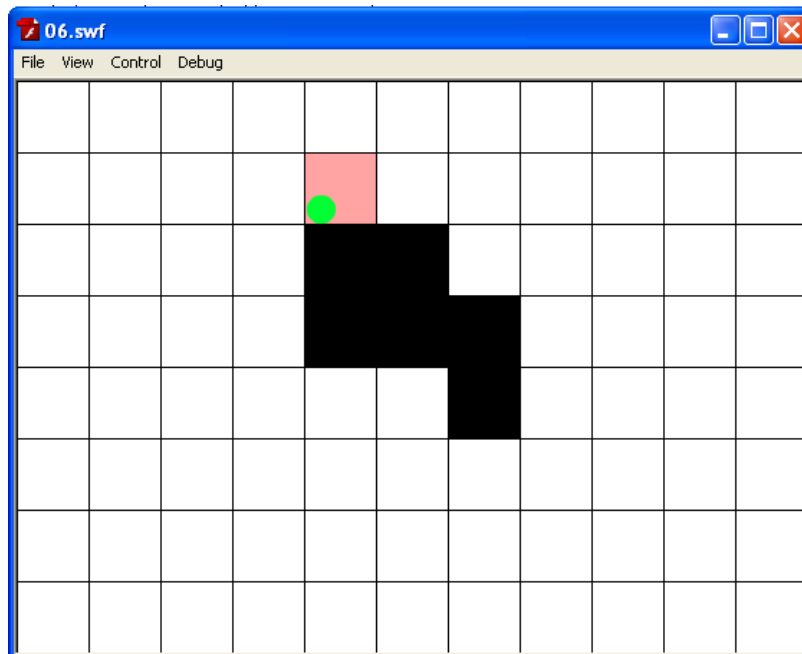


Übungsblatt 4

Ziele

- ✓ Positionserkennung für Objekte in 2D#
- ✓ Kollisionserkennung auf Grid-Basis
- ✓ Pathfinding

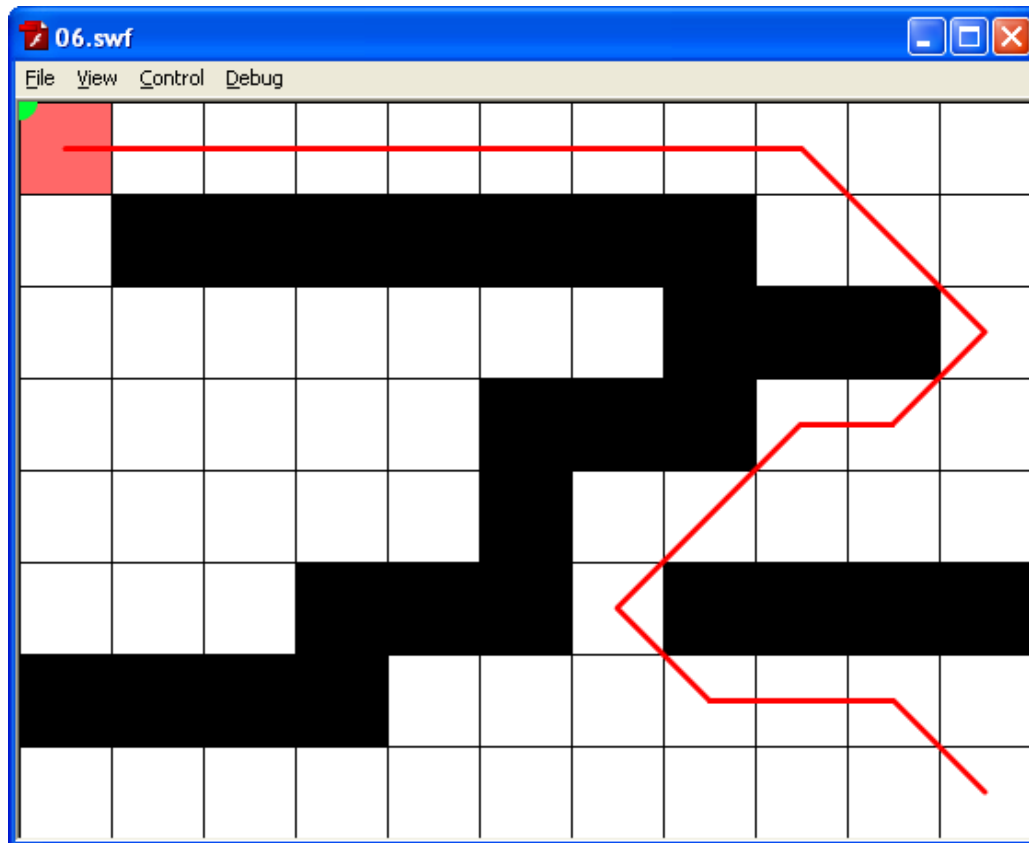


Aufgaben

Aufgabe 4-1: Positionserkennung von Einheiten auf einem Levelraster

Beim Bau von zweidimensionalen Leveln für Spiele wird zumeist ein gedachtes Raster bestimmter Größe für eine grobe Kollisionserkennung verwendet. Nachfolgend sollen Sie eine kleine Anwendung schreiben, die solche Kollisionen zwischen einem Objekt und dem darunterliegenden Raster erkennt.

- Erstellen Sie eine Klasse die ein Raster vorgegebener Größe automatisch auf den Bildschirm zeichnet. Für den Test soll eine Kachel 50x50 Pixel groß sein.
- Erstellen Sie einen MovieClip der als Testobjekt dienen soll. Setzen Sie den MovieClip so, dass der Ursprung des Symbols in der Mitte des Clips liegt.
- Platzieren Sie eine Instanz des Clips auf der Bühne und ermöglichen Sie, dass er mit den Pfeiltasten bewegt werden kann.
- Fügen Sie eine Positionsanzeige hinzu, die die Kacheln farblich hervorhebt auf denen das Objekt gerade liegt.
- Ermöglichen Sie nun Bestimmte Kacheln durch einen Klick schwarz zu färben und diese als „Obstacles“ in einer Liste zu führen.
- Nun soll der Spieler sich nicht mehr auf Obstacles bewegen können.



Aufgabe 4-2: Pathfinding

Sind auf einer so gekachelten Karte Hindernisse platziert ist in den meisten Spielen das finden eines optimalen Weges von Kachel A zu Kachel B wichtig. Überlegen Sie sich einen effektiven Pathfinding-Algorithmus und implementieren Sie diesen.

- Machen Sie sich Gedanken über verschiedene mögliche Pathfinding-Algorithmen. Wie schaut die einfachste mögliche Lösung aus.
- Finden Sie einen Kompromiss zwischen Pfadqualität und Performance des Algorithmus und implementieren Sie den gewählten Algorithmus so dass der kürzeste Pfad zwischen der oberen linken und der unteren rechten Kachel stets dargestellt wird.
- Was ist eine Heuristik beim Pathfinding und wieso optimiert diese die Wegfindung?
- Testen Sie den Algorithmus mit mehr Kacheln? Was fällt auf? Welche Optimierung am Algorithmus sind weiterhin möglich?