

# Leistungsnachweis im Fach Programmierung 1

Maximilian von Hohenbüchel  
Fabian Cieslik

2020-12-07

## Contents

1	Projektvoraussetzungen	III
2	Idee	IV
3	Beschreibung	V
4	Projektablauf	VI
5	Klassendiagramm	VII

# 1 Projektvoraussetzungen

**Beschreibung:** Die Projektaufgabe besteht darin, ein einfaches Spiel zu implementieren. Die Wahl des Spiels bleibt Ihnen überlassen, beachten Sie jedoch, dass sich im Rahmen von 36 Stunden Arbeitszeit nur sehr begrenzte Spielideen auch umsetzen lassen.

**Details:** Sie programmieren ein Spiel für ein sehr eingeschränktes Display. Dieses enthält nur  $24 \times 48$  Bildpunkte (Pixel), d.h. 24 Reihen mit jeweils 48 Spalten. Jeder Bildpunkt kann 16 Millionen Farben annehmen, wobei die Rot, Grün und Blau-Komponente mit jeweils einem Byte angesprochen wird. Als Steuermöglichkeit stehen Ihnen vier Tasten zur Verfügung, die wie im Cursorblock üblich angeordnet sind. Es gibt nur einen Spieler. Die Zeit für eine Spielrunde sollte bei 20-30 Sekunden liegen.

Zur Ein- und Ausgabe erhalten Sie eine Klasse mit zwei Methoden:

- *public int getKeyboard()*  
Liefert die vier Cursortasten der Tastatur folgende Werte zurück:
  - 0 -> "hoch"
  - 1 -> "runter"
  - 2 -> "links"
  - 3 -> "rechts"
  - -1 -> keine Taste
- *public void showImage(short[] image)*  
Zeigt ein komplettes Bild auf dem Display an, wobei der erste Wert des Arrays die Rot-Komponente des linken oben Bildpunkts ist und der letzte Wert die Blau-Komponente von 0 bis 255 des rechten unteren Bildpunktes. Das übergebene Array muss exakt  $24 * 48 * 3$  Elemente haben für die 24 Zeilen, 48 Spalten und 3 Farbkomponenten pro Pixel. Das Display wird zeilenweise durchlaufen.

## Spielumfang:

- Eine *interaktive Spielerfigur*
- Eine *automatisch gesteuerte Spielerfigur*
- Einen Hintergrund
- Ein *Score-System*
- Ein *Highscore-System*
- Implementierungsvorgaben:
  - Eine generische Klasse
  - Drei davon abgeleitete Klassen (Spieler, Hintergrund, Gegner/NPC)

## 2 Idee

**Name:**

MP - Mari proelium

**Spiel:**

Der Spieler steuert ein Schiff und probiert so lang zu überleben wie möglich. Es existieren Gegner, die sich zufällig bewegen.

Das Spiel wird in einer Vogelperspektive gespielt, man hat dadurch jederzeit den Überblick der gesamten Karte.

Das Spielprinzip der Runden wird in dem Sinne implementiert, dass alle 30 Sekunden neue Gegner auftauchen und man für jede überlebte Runde zusätzliche Punkte bekommt.

Das Punktesystem wird von der Zeit, die man am Leben ist und der Anzahl der besiegten feindlichen Schiffe beeinflusst.

Die Steuerung wird auf die vier verfügbaren Tasten aufgeteilt, sodass man ohne Probleme sein Schiff steuern kann und zugleich auch schießen kann. Die Kollisionsinteraktionen mit feindlichen Schiffen und eventuellen Häfen wird vom System übernommen.

### 3 Beschreibung

#### Karte:

- **Aussehen**

Die Karte hat einen blauen Hintergrund, der den Ozean darstellt. Auf dem Ozean kann es vorkommen, dass es verschiedene Inseln geben kann. Inseln werden als braune oder grüne Pixel dargestellt. Die Platzierung der Inseln wird zufällig am Anfang des Spiels festgelegt und wird nur bei einem kompletten Neustart verändert. Auf den Inseln können Häfen generiert werden, die einem einen Bonus geben, falls man sie erreichen sollte.

#### Spieler:

- **Aussehen**

Ein zwei bzw. drei Pixel langes Schiff.

- **Fähigkeiten**

- Links: 45 Grad Drehung gegen den Uhrzeigersinn
- Rechts: 45 Grad Drehung in den Uhrzeigersinn
- Oben: Vorwärts Bewegung nach vorne
- Unten: Benutzen der Schiffsinternen Kanone

#### Gegner:

- **Aussehen**

Ein zwei bzw. drei Pixel langes Schiff.

- **Fähigkeiten**

- Zufälliges Bewegen auf der Karte
- Bei Spielersicht wird Geschossen

#### Punkte:

- **Punktequellen**

- Beim treffen eines Gegners
- Beim besiegen eines Gegners
- Besiegen aller gerade lebender Geger
- Fürs überleben einer Runde

- **Highscore**

Punkte werden in der Konsole als Highscore nach jedem Tod des Spielers ausgegeben

## 4 Projektablauf

- **Vorbereitung**

Es werden alle Spielnotwendigen Variablen deklariert und initialisiert. In einer *Do-While* Schleife wird daraufhin gestartet um mehrere Spiele hintereinander spielen zu Können. Am Start der Schleife wird die Karte, der Spieler und die Gegner erstellt und gezeichnet und auf eine Eingabe des Benutzers gewartet. Bei Eingabe wird der Spielablauf gestartet. Nach dem Tod des Spielers wird der Punktestand ausgegeben und die Möglichkeit geboten ein neues Spiel zu Starten.

- **Spielablauf**

Zuerst wird der Spieler bewegt und auf Kollisionen überprüft, danach die Gegner. Anschließend wird überprüft ob man eine Runde überlebt hat.

## 5 Klassendiagramm