2024

Projektdokumentation: Mini-Game programmieren

MINI-GAME MIT SPIELER, GEGNER UND BEHLONUNG PROGRAMMIEREN

ENSAR MEMETI

EINLEITUNG

PROJEKTBESCHREIBUNG

In diesem Projekt habe Ich meine **ersten Schritte** in die **Welt des Programmierens** gemacht. Dieses Projekt habe wir zusammen mit der Ausbildungsklasse programmiert. In diesem Projekt habe Ich die **ersten Grundlagen** wie auch die **ersten Kenntnisse** des **Programmierens** erlernt.

In diesem Projekt geht es darum ein **Mini-Game zu programmieren**. Und in diesem Mini-Game geht es um eine **Benutzerschnittstelle** oder einen **Spieler**, den man erstellt, und den man anhand von der **Tastatur bewegen** und lenken kann. Es wird ein Raster des **Spielfeldes erstellt**, und in diesem Spielfeld darf sich der **Spieler bewegen**. Zu dem gibt es eine **Belohnung**, die man einsammeln kann, während man den Spieler durch das Spielfeld führt.

Doch das ist noch nicht alles! Denn es gibt noch einen **Gegner**, dem man aus dem Weg gehen muss. Dieser **Gegner bewegt sich beliebig** durch das Spielfeld. Wenn man vom **Gegner getroffen** wird, dann heisst es: **GAME OVER!**

PROJEKTZIEL

Unser Ziel bei diesem Projekt war es, die **Grundlagen des Programmierens** zu verstehen und zu vertiefen.

Es war mir auch wichtig das Ich das **Konzept** und das **Allgemeine Programmieren** verstehe und auch umsetzen kann.

Ziel war es auch viel **Spass** am **Projekt** und am **Programmieren** zu haben!

TECHNOLOGIEN ZUR PROGRAMMIERUNG DES MINI-GAME

Für das Programmieren des Mini-Game haben wir das **Programm Processing** benutzt, was eine etwas abgespeckter Variante von **Java** ist, und mit Processing haben wir auch die **ersten Schritte ins Programmieren** begonnen.

Programmiersprache: *Processing (Java)*

Entwicklungsumgebung: *Processing*

TECHNISCHE UMSETZUNG

CODE-AUFBAU (PROCESSING)

- **Erstellung des Live -Demo Fensters**

```
final int ANZAHL_FELDER = 20;  // Globale Variabel
final int FELD_GROESSE = 30;   // Globale Variabel

int spielerX, spielerY;      // Spieler X/Y Koordinaten
int goodieX, goodieY;        // Goodie X/Y Koordinaten
int enemyX, enemyY;          // Enemy X/Y Koordinaten

int score = 0;               // Punktestand

boolean isGameOver = false;  // Status ob Spiel vorbei ist

void setup() {
  size(800, 650);          // Grösse Bildschirm
  frameRate(5);            // Geschwindigkeit wie schnell draw() aufgerufen werden soll

  spielerX = 7;            // Startwert X Koordinate Spieler
  spielerY = 3;            // Startwert Y Koordinate Spieler

  resetGoodie();           // Startwert X/Y Koordinate Goodie

  enemyX = 19;              // Startwert X Koordinate Enemy
  enemyY = 17;              // Startwert Y Koordinate Enemy

  key = 'd';                // setze die erste Taste auf Rechts, damit der Player sich zu Beginn bewegt
}
```

- **Erstellung des Hintergrundes, Spielers, Gegners und Belohnung**

```
void draw() {
  // Fülle den Hintergrund mit weiss
  background(255);

  if (isGameOver) {
    displayGameOver();
  } else {

    // Zeichne alle ELemente
    drawBoard();
    drawPlayer(0);
    drawGoodie();
    drawEnemy();

    // Player Bewegung und RandCheck
    movePlayer();
    handlePlayerCrossingEdge();

    // Enemy Bewegung und RandCheck
    moveEnemy();
    handleEnemyCrossingEdge();

    // Wenn die X & Y Koordinaten von Spieler und Goodie gleich sind
    if (spielerX == goodieX && spielerY == goodieY) {
      // Vergieb neue Random Zahlen 0 - 30 für das Goodie (X/Y)
      resetGoodie();

      // Erhöhe den Score um 100 Punkte
      score += 100;
    }
    // Wenn die X & Y Koordinaten von Spieler und Enemy gleich sind
    if (spielerX == enemyX && spielerY == enemyY) {
      // setze isGameOver auf true
      isGameOver = true;
    }

    displayScore(0, 24);
  }
}
```

TECHNISCHE UMSETZUNG

CODE-AUFBAU (PROCESSING)

- **Zeichnung des Spielfeldes, des Spielers und der Belohnung**

```
 */
void drawBoard() {
  // Da der Hintergrund weiss ist, haben wir unseren Raster verloren
  // Deshalb müssen wir den mit jeder Spielerbewegung wieder zeichnen
  fill(255);
  stroke(0);

  for (int y = 0; y < ANZAHL_FELDER; y++) {
    for (int x = 0; x < ANZAHL_FELDER; x++) {
      rect(x * FELD_GROESSE, y * FELD_GROESSE, FELD_GROESSE, FELD_GROESSE);
    }
  }
}

/**
 * Zeichne den Player
 * Farbe des Spielers kann bestimmt werden in Graustufen von Weiss bis Schwarz
 */
void drawPlayer(int playerColor) {
  // Zeichne das nächste Element mit einer Füllfarbe: Schwarz
  fill(playerColor);
  // Zeichne das SpielerRechteck an der neuen Position (X/Y)
  rect(spielerX * FELD_GROESSE, spielerY * FELD_GROESSE, FELD_GROESSE, FELD_GROESSE);
}

/**
 * Zeichne den Goodie
 */
void drawGoodie() {
  // Zeichne das nächste Element mit einer Füllfarbe: Violett
  fill(255, 0, 255);
  // Zeichne das GoodieRechteck an der Random Position (X/Y) definiert in der Setup()
  rect(goodieX * FELD_GROESSE, goodieY * FELD_GROESSE, FELD_GROESSE, FELD_GROESSE);
}
```

- **Zeichnung des Gegners und Punktestand anzeige**

```
 * Zeichne den Gegener
 */
void drawEnemy() {
  // Zeichne das nächste Element mit einer Füllfarbe: Violett
  fill(255, 0, 0);
  // Zeichne das GoodieRechteck an der Random Position (X/Y) definiert in der Setup()
  rect(enemyX * FELD_GROESSE, enemyY * FELD_GROESSE, FELD_GROESSE, FELD_GROESSE);
}


/**
 * Punktestand anzeigen
 */
void displayScore(int textColor, int textSize) {
  // Füllfarbe schwarz für das nächste Element
  fill(textColor);
  // Schriftgrösse 24
  textSize(textSize);
  // Schreibe einen Text auf den Bildschirm an der Position X=610 y=30
  text("Score: " + score, 610, 30);
}
```

TECHNISCHE UMSETZUNG

CODE-AUFBAU (PROCESSING)

Bewegen des Spielers

```
 * Bewege den Player
 */
void movePlayer() {
  // Prüfe welche Taste zuletzt gedrückt wurde
  // Zähle die X oder Y Koordinate rauf oder runter basierend auf dem letzten KeyStroke
  // Zähle so lange weiter bis eine andere Richtung gedrückt wird
  if (key == 'w') {
    spielerY--;
  } else if (key == 's') {
    spielerY++;
  } else if (key == 'a') {
    spielerX--;
  } else if (key == 'd') {
    spielerX++;
  }

  // Selbe Steuerung aber mit den Pfeiltasten
  // Achtung nicht mit key vergleichen sondern mit keyCode
  if (keyCode == UP) {
    spielerY--;
  } else if (keyCode == DOWN) {
    spielerY++;
  } else if (keyCode == LEFT) {
    spielerX--;
  } else if (keyCode == RIGHT) {
    spielerX++;
  }
}
```

Spielerposition aktualisieren

```
 * Aktualisiere Position des Spielers beim Verlassen des Boards
 */
void handlePlayerCrossingEdge() {
  // Wenn die X-Position des Spielers grösser als das Feld ist (rechts), setzte den Spieler zurück auf 0
  if (spielerX > ANZAHL_FELDER-1) {
    spielerX = 0;
  }
  // Wenn die X-Position des Spielers kleiner als das Feld ist (links), setzte den Spieler zurück auf Anzahl_Felder-1
  if (spielerX < 0) {
    spielerX = ANZAHL_FELDER -1;
  }
  // Wenn die X-Position des Spielers grösser als das Feld ist (unten), setzte den Spieler zurück auf 0
  if (spielerY > ANZAHL_FELDER -1) {
    spielerY = 0;
  }
  // Wenn die Y-Position des Spielers kleiner als das Feld ist (oben), setzte den Spieler zurück auf Anzahl_Felder-1
  if (spielerY < 0) {
    spielerY = ANZAHL_FELDER -1;
  }
}
```

TECHNISCHE UMSETZUNG

CODE-AUFBAU (PROCESSING)

- **Bewegen des Gegners und Aktualisierung der Gegner Position**

```
 * Bewege den Gegener
 */
void moveEnemy() {
  int direction = (int)random(4);

  switch(direction) {
  case 0:
    enemyY--;
    break;
  case 1:
    enemyX++;
    break;
  case 2:
    enemyY++;
    break;
  case 3:
    enemyX--;
    break;
  }
}

 * Aktualisiere Position des Gegners beim Verlassen des Boards
 */
void handleEnemyCrossingEdge() {
  // Wenn die X-Position des Spielers grösser als das Feld ist (rechts), setzte den Spieler zurück auf 0
  if (enemyX > ANZAHL_FELDER-1) {
    enemyX = 0;
  }
  // Wenn die X-Position des Spielers kleiner als das Feld ist (links), setzte den Spieler zurück auf Anzahl_Felder-1
  if (enemyX < 0) {
    enemyX = ANZAHL_FELDER -1;
  }
  // Wenn die X-Position des Spielers grösser als das Feld ist (unten), setzte den Spieler zurück auf 0
  if (enemyY > ANZAHL_FELDER -1) {
    enemyY = 0;
  }
  // Wenn die Y-Position des Spielers kleiner als das Feld ist (oben), setzte den Spieler zurück auf Anzahl_Felder-1
  if (enemyY < 0) {
    enemyY = ANZAHL_FELDER -1;
  }
}
```

- **Aufzeigen des GAME OVER Screens**

```
 * Zeiege den GameOver screen
 */
void displayGameOver() {
  fill(255, 0, 0);
  textSize(128);
  text("GAME OVER", width / 8, height / 2);
  fill(0);
  textSize(64);
  text("Score: " + score, width / 3.5, height / 2 + 70);
}
```

FAZIT UND AUSBLICK

ERREICHTE ZIELE

Ich bin mit diesem Projekt erfolgreich in die Welt des Programmierens eingestiegen, und habe natürlich vieles auch von diesem Projekt mitgenommen und dazu gelernt.

Viele Kenntnisse konnte Ich mitnehmen durch dieses Projekt, und konnte mir eine Grundbasis zum Programmieren aufbauen.

VERBESSERUNGEN & ERWEITERUNGEN

Für mich ist das Projekt erstmal abgeschlossen und Ich kann mich darauf freuen nächste Projekte wie dieses zu programmieren und abzuschliessen.

ANHANG

BILDER ZUM PROJEKT

Schwarz: Spieler Rot: Gegner Violett: Belohnung