Projektplanung

Projekttitel: SchiffeVersenken

Entwickler: Christian Bauer

**Start der Entwicklung: 16.02.2021 – Zielabschluss: 26.02.2021**

Projektpfad: C:\Users\christian.bauer\source\repos\SchiffeVersenken

# Ziel, Regeln und Anforderungen

**Schiffe Versenken Überblick**

|  |  |
| --- | --- |
|  |  |

|  |  |  |
| --- | --- | --- |
|  | **Regeln und Vorgehen des Spieles** | **Hinweise** |
| 1 | Willkommensfenster: „Möchtest du Schiffe versenken spielen?“ | Button click |
| 2 | Der Spieler muss insgesamt 4 Schiffe mit den folgenden Größen auf einem 8x8 Felder großen Spielplan verteilen. | Datenformat des Spielplanes im Backend: 2D Array |
|  | Regeln/Ablauf im Detail:  „Welches Schiff möchtest du setzen? Wähle zwischen 2, 3, 4 und 5:" Es erscheint über dem Spielplan ein Window mit den Auswahlmöglichkeiten.  Das aufliegende Window verschwindet nach der erfolgreichen Auswahl. Über dem eigenen Feld erscheint ein Hinweis: „Wo möchtest du das Schiff mit der Länge X positionieren? Klicke das jeweilige Feld an.“  Die gesetzten Schiffe dürfen sich 1. nicht überkreuzen, 2. nicht berühren und 3. nur vertikal/horizontal gesetzt werden. Alle möglichen Felder werden z.B. Grün hervorgehoben.  „Schiff neu setzen? Oder Position des Schiffes so belassen?“ | Alle bereits gesetzten Schiffe stehen nicht mehr zur Verfügung (Datenformat Liste)  Fehlermeldungen integrieren  Window mit Auswahlmöglichkeit |
| 3 | Computerschiffe werden für den Spieler unsichtbar auf dem Plan verteilt. | Random Klasse |
| 4 | Optional: Schwierigkeitsgrad auswählbar. Passt die Intelligenz des Computers an.   * Einfach: Schussposition zufällig gewählt * Mittel: Computer zielt zufällig, weiß aber wann er getroffen hat und sucht im Anschluss nur umliegende Felder ab * Schwer: Computer berechnet Wahrscheinlichkeiten für die Schiffsposition | Start mit einfacher Variante |
| 5 | Festlegung: Spieler beginnt. | Optional: Zufallsauswahl. |
| 6 | Beide Spielpläne erscheinen.  **Runde i:**   * Der Spieler wählt ein gegnerisches Feld per Klick aus (z.B. B2) * Wenn Treffer: Rotes X erscheint, Spieler darf seinen Zug in Runde 2 wiederholen * Andernfalls daneben: Blaues O erscheint. Der Computer ist an der Reihe.   **Runde i+n:**   * Der Computer schießt (je nach Intelligenz/Schwierigkeitsgrad) ein Feld auf dem Plan des Spielers ab. * Hier gelten dieselben Regeln wie beim Spieler (Treffer = Rotes X + Zug wiederholen, Daneben = Blaues O + Spieler ist an der Reihe) | Button Klick |
| 7 | Es werden so viele Runden gespielt, bis der Spieler oder der Computer die Schiffe des jeweiligen Gegners abgeschossen hat. | If/else |
| 8 | Game over:  „Glückwunsch Sie haben den Computer in X Runden geschlagen.“ ODER „Schade, vielleicht gewinnen Sie beim nächsten Mal.“  Window erscheint: „Möchtest du noch eine Runde spielen oder das Programm beenden?“ | Button click |
| 9 | ~~Optional: In einer Datenbank wird gespeichert wie viele Runden der Spieler gebraucht hat um zu gewinnen. Der Score wird gespeichert. In einem erneuten Spiel kann man selbst versuchen besser zu werden und seine Rundenzahl unterbieten.~~ |  |

# Geschätzte Zeit- und AP-Planung

|  |  |  |  |
| --- | --- | --- | --- |
| **AP** | **Inhalt** | **Zeit** | **Abschluss-**  **datum** |
| 1 | Projekt planen | 8 h | 17.02 |
| 2 | Grundlagenwissen aneignen zu folgenden Themen:   * Ableitung einer Basisklasse (Inheritance) – Tutorial: <https://www.udemy.com/course/complete-csharp-masterclass/learn/lecture/9567022#overview> * Oberflächenprogrammierung mit Windows Forms – Grundlagentutorials: <https://www.youtube.com/watch?v=RyfP0GE6jv4> * Event handler – Tutorial und Überblick: <https://www.udemy.com/course/complete-csharp-masterclass/learn/lecture/20606578#overview> | 8 h | 19.02 |
| 3 | Softwarestruktur planen | 4 h | 22.02 |
| 4 | Programmieren, Fehler testen (in mehreren Iterationen) | 16 h | 26.02 |
|  | **Gesamt** | **36 h** |  |

# Entwicklung des Projekts

