Dokumentation Schiffe versenken

14.06.2022

1 Teammitglieder

Robert Darminow Martr: 2210084

Sophie Knoll Martr: 2210082

Marcel Weber Martr: 2210085

2 Einleitung

- Schiffe versenken ist ein Spieler gegen Spieler Spiel, bei welchem die Spieler auf einem 10x10 Feld die gegnerischen Schiffe treffen müssen
- Jeder Spieler platziert zu Beginn insgesamt sieben Schiffe in sein Feld
- Die Spieler schießen abwechselnd auf die Felder und dürfen nochmal schießen, wenn sie getroffen haben
- Ziel ist es, alle Schiffe des Gegners zu treffen

3 Wochenbeschreibung

Das Projekt wurde immer in Teamarbeit aller Mitglieder programmiert, weswegen keine nennenswerte Aufgabenverteilung im Folgenden aufgeführt wird

Woche 1

- Entscheidung für das Spiel Schiffe versenken

Woche 2

- Konzepterstellung der Implementierung
- Klassen: Player, Ship, Playground, Game
- Methoden: checkWinner, checkHit

Woche 3

- Implementierung der Klasse Playground mit dem Spielfeld als Attribut
  - Spielfeld ist ein 10x10 Array mit JButtons
  - Im Konstruktor werden die Buttons direkt erstellt und angepasst
- Implementierung der Methode initialGUI
  - Erzeugt einen Frame mit einem GridBagLayout
  - Labels für die Bezeichnungen der Felder und Buttons des Spielfeldes werden in den Frame an die richtige Position hinzugefügt

AIS Programmieren 2 – Praktikum SS 2022
Gruppe B

Woche 4

- Wir haben die Klasse BattleshipFrame erstellt, in welche wir die Methode initialGUI verschoben haben
- Das Layout des Frames haben wir zu einem BorderLayout verändert
    - Im Center Panel werden die Playgrounds mit dem GridBagLayout platziert
    - Im South Panel haben wir ein Chatfenster hinzugefügt

Woche 5

- Implementierung der Ship Klasse mit den Attributen status und size
- Implementierung der Game Klasse mit einer shipList, um die Schiffe zu speichern und zwei Playgrounds
  Zudem wurde die Methode game erstellt
    - Jedem Button wurde ein MouseListener hinzugefügt, welcher auf Klick die Methode placeShip ausführt
- Die Methode placeShip hat den Button testweise lediglich schwarz gefärbt
- In der Klasse MainMenu wird dann die Methode initialGUI und game ausgeführt

Woche 6

- Die game Methode wurde angepasst
    - Der Spieler soll die Anfangsposition und die Endposition eines Schiffs anklicken und dann wird die Methode placeShip ausgeführt
    - Zudem wurde eine globale Variable turn implementiert, mit welcher die aktuelle Spielphase geprüft werden kann
    - Die Schiffgrößen werden in Abhängigkeit von der Variable turn gesetzt, sodass erst zwei 4er, dann drei 3er und zuletzt drei 2er Schiffe platziert werden
- Die placeShip Methode wurde umgeschrieben
    - Die übergebenen Koordinaten werden verglichen, um herauszufinden, ob das Schiff vertikal oder horizontal gesetzt wird
    - Es wird sichergestellt, dass das Schiff richtig platziert wird, egal in welcher Richtung Start- und Endpunkt zueinander liegen
    - Die einzelnen Positionen werden in jedem Durchlauf gespeichert und am Ende wird ein neues Ship erstellt, welches diese Postionen und die aktuelle Größe beinhaltet
- Die Methode shipDestroyed wurde angefangen zu erstellen
    - Überprüft von jedem Schiff alle Postionen, ob jede Postion schon getroffen wurde und falls ja wird dieses Schiff aus der shipList entfernt

AIS Programmieren 2 – Praktikum SS 2022
Gruppe B

Woche 7

- Die changeButtons Methode wurde implementiert
    - Wenn der Spieler einen Startpunkt ausgewählt hat, werden alle Buttons grau gefärbt und deaktiviert, welche nicht als Endpunkt geeignet sind
    - Als gültiger Endpunkt gelten maximal vier Buttons, wovon jeweils einer links, rechts, oberhalb und unterhalb des Startpunktes liegt
    - Wenn der Spieler einen Endpunkt gewählt hat, werden alle Buttons wieder aktiviert und blau gefärbt, sofern dort kein Schiff liegt und auch alle Nachbarbuttons eines Schiffs bleiben deaktiviert
- Die Methode hasNeighbor überprüft, ob um einen Button herum ein Schiff liegt

Woche 8

- Implementierung der Methode hit
    - Vergleicht die übergebene Position auf welche geschossen wurde mit der Schiffliste, ob ein Schiff an dieser Position liegt
    - Falls ja wird der Button Orange gefärbt und shipDestroyed ausgeführt
    - Falls nein wird der Button weiß gefärbt und deaktiviert
- Anpassung der Methode shipDestroyed
    - Überprüft für alle Schiffe, ob alle Positionen schon Orange gefärbt sind und färbt diese dann rot
    - Bei einem zerstörten Schiff werden zudem alle Nachbarn gefärbt und deaktiviert, da dort kein Schiff mehr sein kann
- Implementierung der Methode disableNotPlaceable
    - Stellt beim Setzen der Schiffe sicher, dass alle Buttons deaktiviert werden, für die keine Möglichkeit besteht, das aktuelle Schiff zu setzen
- Implementierung der Klassen BattleshipServer, Server und Client
    - Provisorische Verbindung zwischen Server und Client

Woche 9

- Programmierung des Servers wurde für unser Beispiel angepasst
    - Server hat zwei Playgrounds und einen boolean, mit welchen überpüft wird, welcher Spieler gerade dran ist, als Attribute bekommen
- In der Game Klasse wurde implementiert, dass die beiden Spieler ihre Playgrounds auf den Server schicken und den jeweils gegnerischen Playground abrufen können
- Sobald die Spieler ihre Schiffe gesetzt haben, werden die Schiffe in den rechten Playground gesetzt, in welchem auch später die Schüsse gesehen werden können
- In den linken Playground wird dann der gegnerische Playground kopiert
- Es wurde gewährleistet, dass jeder Spieler nur einmal dran ist

- Die Methoden clear, enabled, copyPlayground und shipsDestroyed wurden in der Klasse Playground implementiert
    - Clear setzt den Playground zurück auf Anfang
    - Enabled aktiviert oder deaktiviert alle Buttons des Playgrounds
    - CopyPlayground kopiert die einzelnen Attribute eines Playgrounds auf einen anderen, um ein Duplikat zu erstellen, ohne dass die Referenzen gleich sind
    - ShipsDestroyed gibt an, ob alle Schiffe des Playgrounds zerstört wurden
- Die game Methode wurde so erweitert, dass wenn ein Spieler schießt, mit der Methode shipsDestroyed geprüft wird, ob alle Schiffe zerstört sind und somit der Spieler gewonnen hat
- Dem gegnerischen Spieler wird dann bei der nächsten Aktualisierung angezeigt, dass er verloren hat

Woche 10

- Der ursprüngliche Ansatz, die gesamten Playgrounds über den Server zu versenden hat sich als problematisch dargestellt
- Implementierung haben wir so geändert, dass nur noch einzelne Integer verschickt werden
- Server hat zwei Arrays, in welchen die Positionen der Schiffe gespeichert werden und 4 Listen, in welchen die Koordinaten der Schüsse gespeichert werden
- Zusätzlich wurde die Klasse Server um alle nötigen Hilfsmethoden erweitert, welche die Positionen und Schüsse in den jeweiligen Datenstrukturen speichern und ausgeben
- In der Game Klasse wurden die Methoden sendPlayground, getPlayground und getHits implementiert
    - SendPlayground dient dazu, die Positionen der Schiffe an den Server zu senden
    - Mit getPlayground kann der Spieler dementsprechend die Positionen der gegnerischen Schiffe abrufen und zusätzlich auch alle bisherigen Schüsse des Gegners mit der Methode getHits
- Die Methode shipDestroyed wurde angepasst
    - Die Schiffe werden nicht mehr aus der Schiffliste gelöscht, sondern das Attribut destroyed der Klasse Ship wird auf true gesetzt
- Methode copyPlayground musste angepasst werden, da vorher die Schiffe der Playgrounds gleichgesetzt wurden und daher die Referenz der Schiffe gleich war
    - Anstatt der gesamten Schiffe wurden die Postionen gleichgsetzt, wodurch die Referenz nicht mehr gleich ist
- Schlussendlich wurde verbessert, dass ein Spieler nochmal schießen darf, sofern er ein Schiff getroffen hat

Woche 11

- Implementierung der Klasse MainOverlay
    - Öffnet zu Beginn des Spiels ein Fenster mit welchem ein Spiel gehostet oder gestartet werden kann
    - Der Host kann in einem Textfeld einen Port eingeben und auf den Button Host klicken
    - Der Client kann in einem Textfeld eine IP und in einem anderen Textfeld einen Port angeben und auf den Button Connect klicken
    - Danach wird abhängig davon, welcher Button geklickt wurde, ein Server oder ein Client erstellt
- Einbindung eines zweiten Threads in der Klasse BattleshipServer, welcher parallel zum Main Thread läuft
    - Methode startTimer gibt die Stunden, Minuten und Sekunden aus
    - Der Timer wurde in einem Panel hinzugefügt, welches im Osten des BorderLayouts gesetzt wurde
- Implementierung eines Buttons, zum starten des Spiels
    - Button wird aktiviert, sobald alle Schiffe platziert wurden
    - Dient dazu, den gegnerischen Playground abzurufen, sofern dieser schon existiert
- Implementierung der Methode lostConnection in der Game Klasse
    - Methode gibt bei einer Exception aus, dass die Verbindung verloren wurde und schließt das Fenster

Woche 12

- Implementierung der inneren Klasse ImageFilter der Klasse BattleshipFrame
    - Einbindung eines Bildes mit dem Namen shipImage, wobei es egal ist, ob es ein png, jpg oder jpeg Bild ist
- Das Textfeld wurde mit einer Scrollbar erweitert

4 Bedienungsanleitung

- Das Spiel wird mit der main Methode der MainMenu Klasse gestartet
- Dann öffnet sich ein Fenster und ein Spieler muss einen Port eingeben und das Spiel hosten
- Der andere Spieler muss diesen Port und die IP-Adresse des Spielers eingeben, welcher den Server hostet
- Danach öffnet sich ein neues Fenster und beide Spieler können ihre insgesamt sieben Schiffe platzieren
- Nach der Platzierung fängt der Host an zu schießen und das Spiel endet, wenn ein Spieler alle gegnerischen Schiffe getroffen hat

5 Klassendiagramm