

8. JUNI 2023



```
Tic Tac Toe
  | x | x
  +---+
  o | o |
  +---+
  o |   | x
```

M319 LB02 TIC TAC TOE PROJEKT

DRUCKDATUM: 08.06.2023

Kontaktinformation:
E-Mail: julian.brecht@ict.csbe.ch

JULIAN.BRECHT
BIS22 MODUL319 LB02

Inhaltsverzeichnis

Inhaltsverzeichnis	1
Einleitung.....	3
Planung.....	3
Userstories.....	3
Rollen.....	3
Story 1.....	3
Story 2.....	3
Story 3.....	3
Story 4.....	3
Story 5.....	3
Story 6.....	3
Story 7.....	3
Problem und Lösungstabelle	4
Klassen.....	4
Main.....	4
DrawField.....	4
ResetField	4
XwinCheck	4
OWinCheck	4
DrawCheck.....	4
Methoden.....	5
main.....	5
resetToStart	5
drawField	5
checkXWin	5
checkOWin.....	5
drawCheck	5
PAP.....	6
Realisierung und Erklärung.....	9
Menüs.....	9
Klassen.....	9
Main.....	9
DrawField.....	9
ResetField	9
XwinCheck	9

OWinCheck	9
DrawCheck	10
Datenmodel	10
Tabellenverzeichnis	11
Abbildungsverzeichnis	11
Quellenverzeichnis	11

Einleitung

Dies ist meine Dokumentation des Tic Tac Toe-Schulprojekts im Modul 319. Das Projekt wurde mit Java umgesetzt und zielt darauf ab, das klassische Spiel Tic Tac Toe auf Konsolenbasis zu realisieren. In Tic Tac Toe versuchen zwei Spieler der Reihe nach, eine Reihe von drei Symbolen in einer vertikalen, horizontalen oder diagonalen Linie zu platzieren.

Planung

In diesem Abschnitt geht es um die vorher erstellte Planung zum Projekt.

Userstories

Rollen

1. Spieler

Story 1

Als Spieler möchte ich das Spiel starten können, um gegen einen anderen Spieler zu spielen.

Story 2

Als Spieler möchte ich das ich mein Symbol in das von mir gewählte Kästchen platzieren kann, damit ich das Spiel spielen und gewinnen kann.

Story 3

Als Spieler möchte ich vor jedem Zug das Feld sehen, damit ich meine Strategie ändern und ausführen kann.

Story 4

Als Spieler möchte ich das mein Symbol nicht durch das meines Gegners auf meinem Feld ersetzt werden kann, damit das Spiel fair bleibt.

Story 5

Als Spieler möchte ich, dass das Spiel erkennt, wer gewonnen und wer verloren hat, damit der Gegner und ich wissen wer gewonnen oder verloren hat.

Story 6

Als Spieler möchte ich, dass das Spiel erkennt, wenn es unentschieden ist, damit wir bei Bedarf ein neues Spiel starten können.

Story 7

Als Spieler möchte ich, dass ich meinen Namen eingeben kann, damit das Spiel personalisierter wird.

Problem und Lösungstabelle

Problem	Lösung
Der Spielmodus muss gewählt werden können.	Int Variabel die den Spielmodus trackt.
Die Spieler müssen Ihre Namen eingeben können.	Dies wird mit 2 String Variablen gehändelt die beim Start mit den Namen der Spieler befüllt werden.
Das Feld muss ausgegeben werden in der Konsole.	Zwei ineinander verschachtelte for Schleifen die das Feld ausgeben.
Die Symbole müssen gesetzt werden können und entsprechend gespeichert werden.	2D char Array mit welchem man die Koordinaten und so die Position der Symbole speichern kann.
Die Spieler müssen nach der Reihe die Symbole setzen können.	Int Variabel die trackt welcher Spieler an der Reihe ist.
Spieler dürfen die Symbole nicht ersetzen können auf bereits belegten Feldern.	Schleife die im Array prüft, ob das Feld bereits belegt ist und wenn ja zur neuen Eingabe auffordert.
Das Spiel muss erkennen, ob jemand gewonnen hat.	2 Funktionen eine die Prüft, ob X gewonnen hat und eine andere die Prüft, ob O gewonnen hat.
Das Spiel muss erkennen, ob es Unentschieden endet.	Eine Funktion, die prüft, ob das Feld voll ist ohne dass jemand 3 Symbole in einer Reihe hat.

Tabelle 1 Problem / Lösung

Klassen

Hier erfasse ich die im Voraus geplanten Klassen, die es meiner Meinung nach benötigt, um das Spiel zu realisieren.

Main

Hauptklasse in der das Spiel ausgeführt wird.

DrawField

Klasse die die Methode zum Darstellen des Feldes beinhalten wird

ResetField

Klasse die die Methode zum zurücksetzen des Feldes beinhalten wird.

XwinCheck

Klasse welche prüft, ob X horizontal, vertikal oder diagonal gewonnen hat.

OWinCheck

Klasse welche prüft, ob O horizontal, vertikal oder diagonal gewonnen hat.

DrawCheck

Klasse welche prüft, ob das Feld voll ist und somit ein unentschieden zustande kommt.

Methoden

Hier erfasse ich meine im Voraus geplanten Methoden, die meiner Meinung nach benötigt werden, um das Spiel zu realisieren.

main

Soll den Gameloop enthalten und sich um die Menüsteuerung kümmern.

resetToStart

Soll das Feld wieder in den Ursprungszustand zurückversetzen.

drawField

Soll das Feld ausgeben mit allen Symbolen, die gesetzt wurden.

checkXWin

Soll prüfen, ob X horizontal, vertikal oder diagonal gewonnen hat.

checkOWin

Soll prüfen, ob O horizontal, vertikal oder diagonal gewonnen hat.

drawCheck

Soll prüfen, ob alle Felder belegt sind und ausgeben das es Unentschieden ist.

PAP

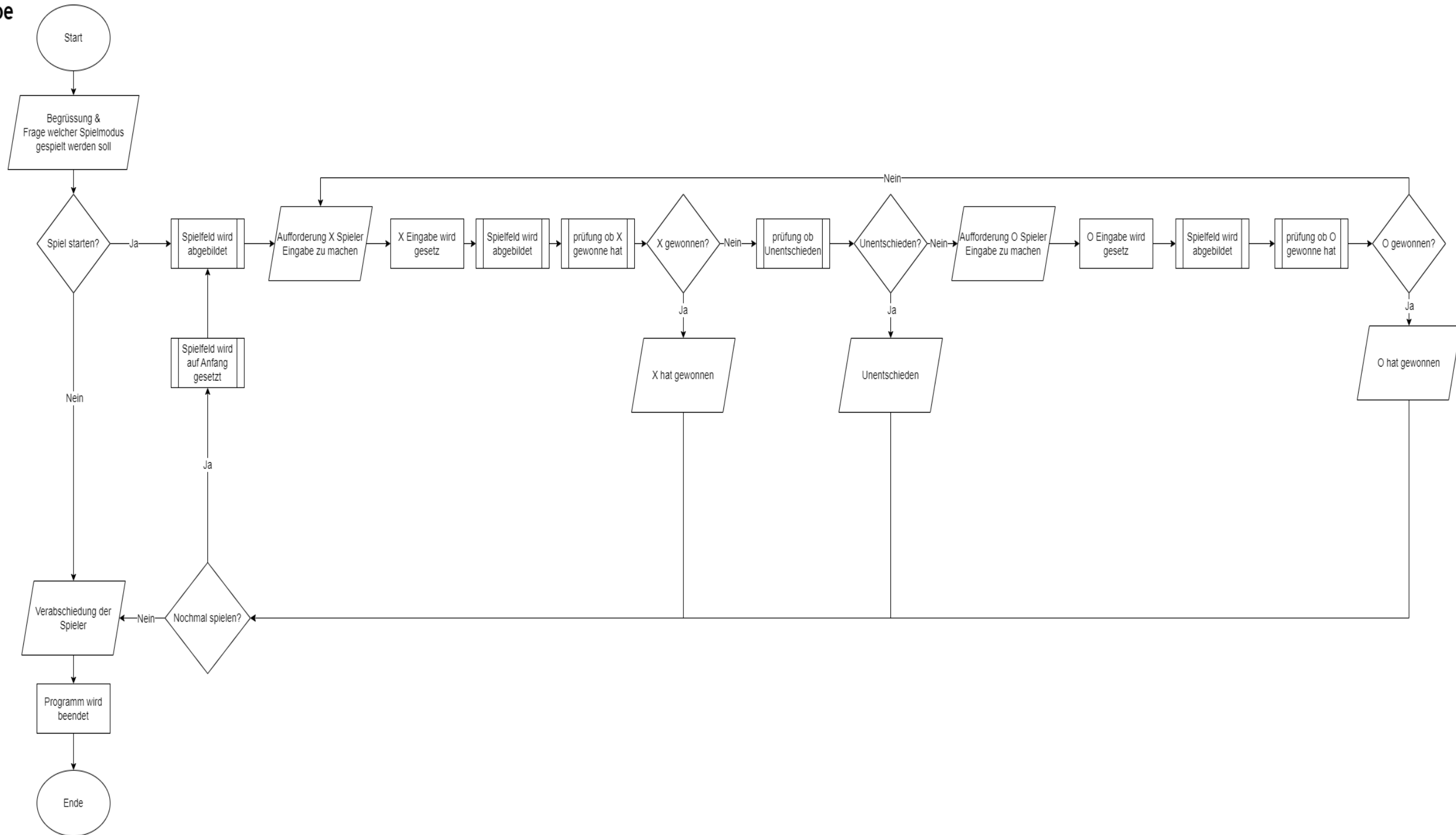
TicTacToe
PAP

Abbildung 1 Main PAP



Abbildung 2 Display Field PAP



Abbildung 3 Reset Field PAP

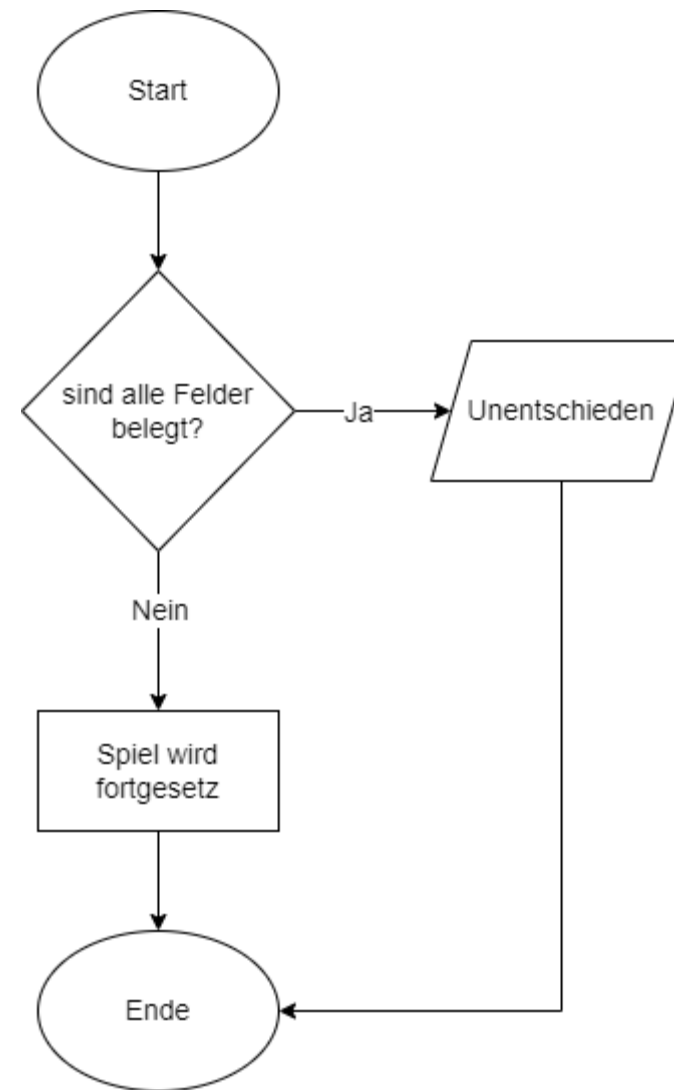


Abbildung 4 Draw Check PAP

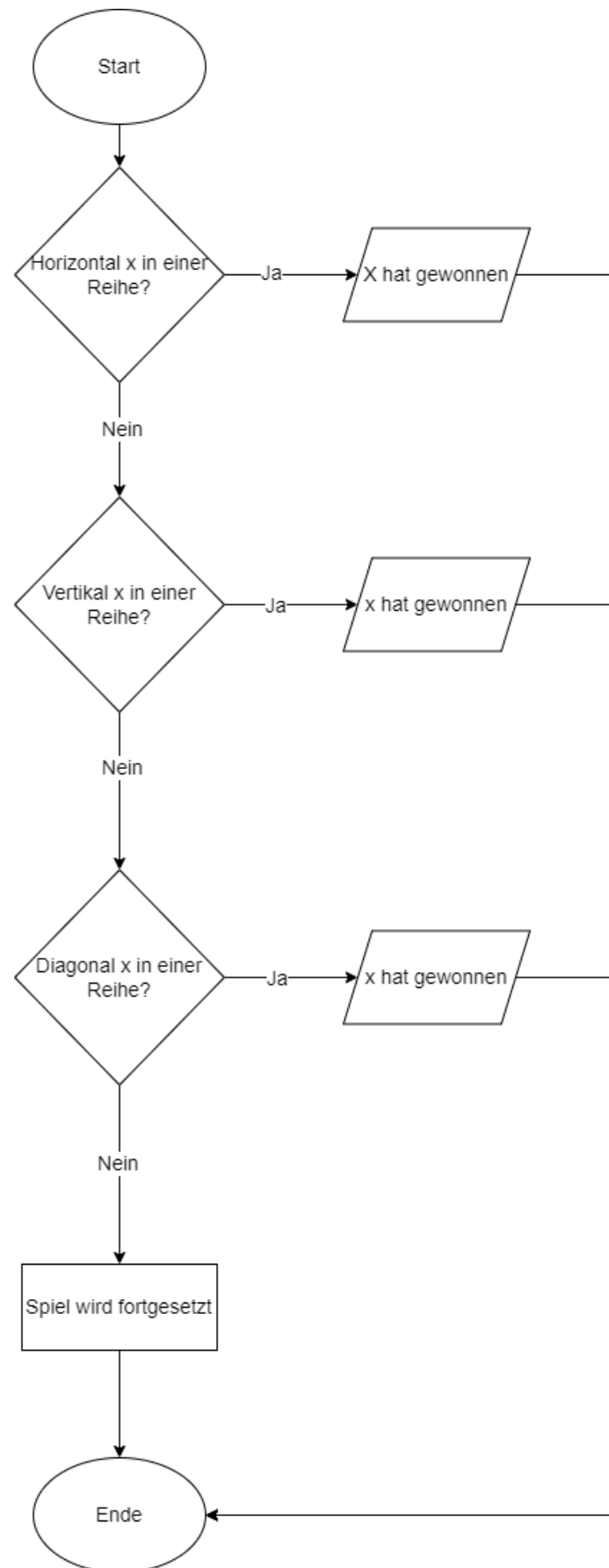


Abbildung 5 X Win PAP

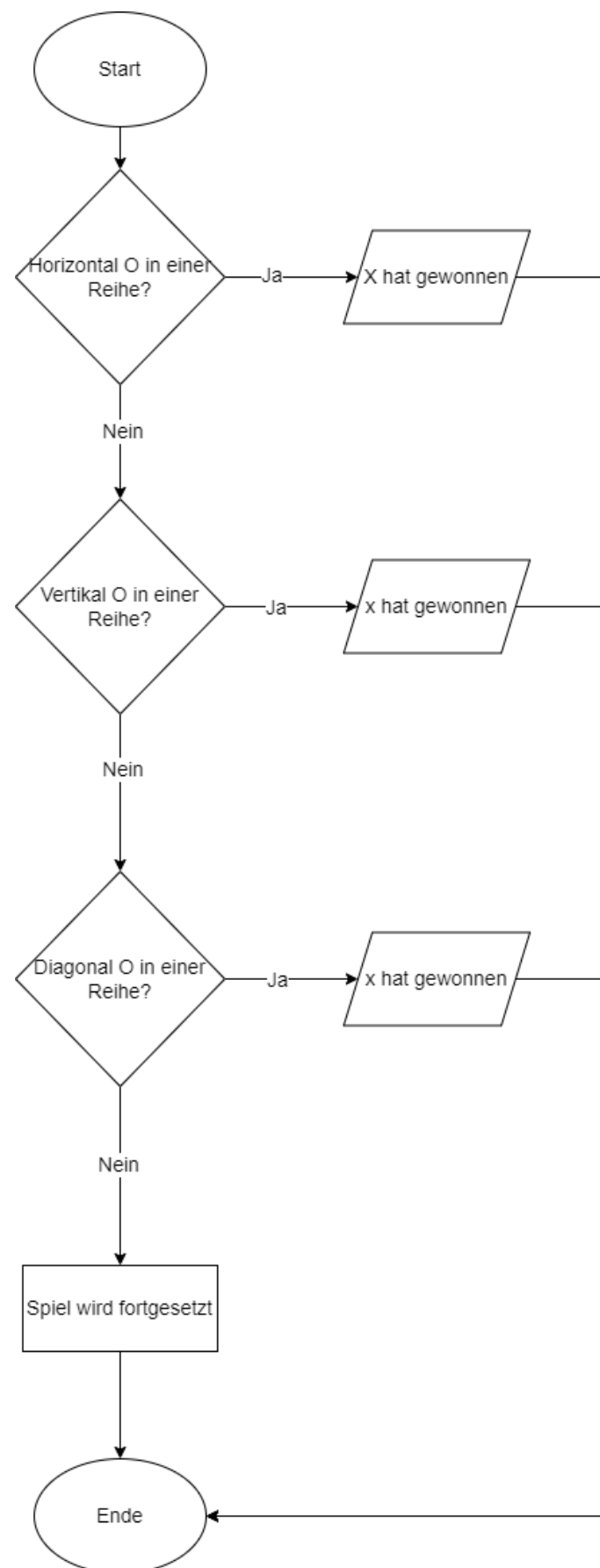


Abbildung 6 O Win PAP

Realisierung und Erklärung

Menüs

1. Hauptmenü
 - 1.1 Spiel Starten
 - 1.2 Programm beenden
2. X win Menü
 - 2.1 Neues Spiel starten
 - 2.2 Spiel beenden
3. Draw Menü
 - 3.1 Neues Spiel starten
 - 3.2 Spiel beenden
4. O win Menü
 - 4.1 Neues Spiel starten
 - 4.2 Spiel beenden

Klassen

Die Klassen konnten 1 zu 1 der Planung übernommen werden. Im Anschluss eine kurze Erklärung dieser und was Sie beinhalten.

Main

Hauptklasse in der sich die main Methode befindet. In der main Methode befindet sich der Gameloop der die anderen Klassen und dessen Methoden zusammen mit dem User Input verwaltet und so das Spiel spielbar macht und die Menüs bedienbar.

DrawField

In dieser Klasse befindet sich die „drawField“ Methode, die benötigt wird, um das Spielfeld auszugeben. Das Spielfeld wird dabei mit zwei ineinander verschachtelten for loops erstellt wobei die Position der gesetzten Symbole aus den Werten des gameArray mit eingebunden wurden. Somit lässt ich das Spielfeld immer im aktuellen Zustand ausgeben.

ResetField

In dieser Klasse befindet sich die resetToStart Methode, die alle Positionen des gameArray auf leere char Elemente zurücksetzt, um das erste Feld zu initialisieren und ein neues Spiel starten zu können.

XwinCheck

In dieser Klasse befindet sich die Methode checkXWin die prüft ob im gameArray entweder horizontal, vertikal oder diagonal drei X stehen. Wenn dies so ist, setzt diese Methode die Variable gameState auf 1, was in einer anschliessenden Prüfung im Gameloop dafür sorgt, dass das X win Menü aufgerufen wird.

OWinCheck

In dieser Klasse befindet sich die Methode checkOWin die prüft ob im gameArray entweder horizontal, vertikal oder diagonal drei O stehen. Wenn dies so ist, setzt diese Methode die Variable gameState auf 2, was in einer anschliessenden Prüfung im Gameloop dafür sorgt, dass das O win Menü aufgerufen wird.

DrawCheck

in dieser Klasse befindet sich die drawCheck Methode, welche prüft, ob alle Felder des gameArray belegt sind. Sobald dies so ist, wird die Variabel gameState auf 3 gesetzt und das Draw Menü wird aufgerufen.

Datenmodel

Um das Spielfeld zu generieren und Werte für diese Speichern zu können, wird ein zweidimensionales char Array namens „gameArray“ gebraucht.

Es wird ein char Array verwendet, da man immer nur ein Symbol pro Position speichern muss.

Spielernamen sind in zwei Strings gespeichert mit den Namen „playerName1“ und „playerName2“.

Es wurden Strings verwendet da ein Name mehr als ein einzelnes Symbol ist.

Tabellenverzeichnis

Tabelle 1 Problem / Lösung	4
----------------------------------	---

Abbildungsverzeichnis

Abbildung 1 Main PAP	6
Abbildung 2 Display Field PAP	7
Abbildung 3 Reset Field PAP	7
Abbildung 4 Draw Check PAP	7
Abbildung 5 X Win PAP	8
Abbildung 6 O Win PAP	8

Quellenverzeichnis

Internet:

Java Logo im Titelblatt

https://cdn.icon-icons.com/icons2/2415/PNG/512/java_original_wordmark_logo_icon_146459.png

22.05.2023 19:52

<https://www.w3schools.com/java/default.asp>

https://www.w3schools.com/java/java_try_catch.asp

https://www.tutorialspoint.com/java/java_documentation.htm

Bücher:

Herdt Campus Java 9 Grundlagen Programmierung

Autor Eimar Fuchs

1. Ausgabe 1. Aktualisierung Mai 2021

ISBN: 978-86249-645-7

Mündliche Überlieferungen:

Unterricht von Manuel Sollberger