

1 GUI erstellen(Youran Wang)(fertig am 10.7.2023)

Um die GUI zu zeichnen, benutze ich das API in Packet java.awt. Das heißt, das GUI hat von JFrame und JPanel erstellt werden. Während Spiel gibt es 4 Jframes. Sie sind BeginnMenü, Definitionsmenü, Spielfeld und Winners.

1.1 BeginnMenü

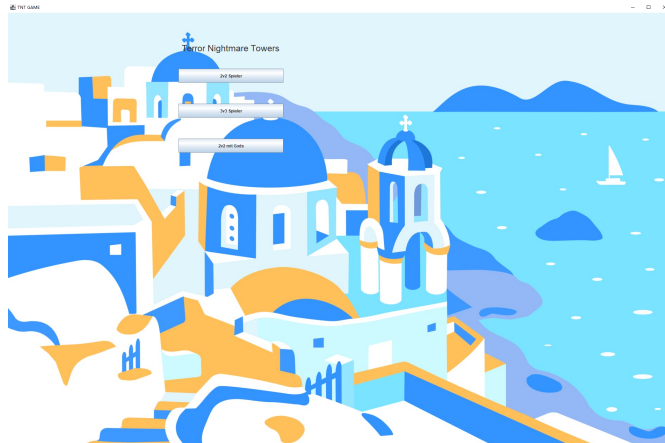


Abbildung 1: BeginnMenü

Es hat 3 Buttons die sind 3 SpieleModel. "2v2 Spiele" bedeutet, das 2er-Spieler durchgeführt werden. "3v3 Spiele" bedeutet, das 3er-Spieler durchgeführt werden. "2v2 mit Gods" bedeutet, das 2er-Spieler mit Gods durchgeführt werden. Die 3 Models sollen miteinander keinen Einfluss haben. Außerdem gibt es noch eine Text mit "TNT" Name auf dem Frame.

1.2 Definitionsmenü

1.2.1 2er-Spiele



Abbildung 2: Definitionsmenü

Es hat 8 Buttons. Spieler sollen die Name eingeben. Auf Button "Names" wird eine kleine Fenster öffnen, um die Name einzugeben. Danach wird diesen Fenster selbst schließen. Spieler sollen die Farben wählen. Auf Button "Farben" wird eine kleine Fenster öffnen, um die Farben zu wählen. Danach wird diesen Fenster selbst schließen. Jetzt bietet es rot blau gelb und grün an. Wichtig ist Spieler muss Größe des Field wählen. Mit dem Button "Größe" wird Spielfeld als $n \times n$ definieren. Button "Anfangen" wird das Spiel zum Spielfeld leiten, und das Spiel fängt an.

1.2.2 3er-Spiele

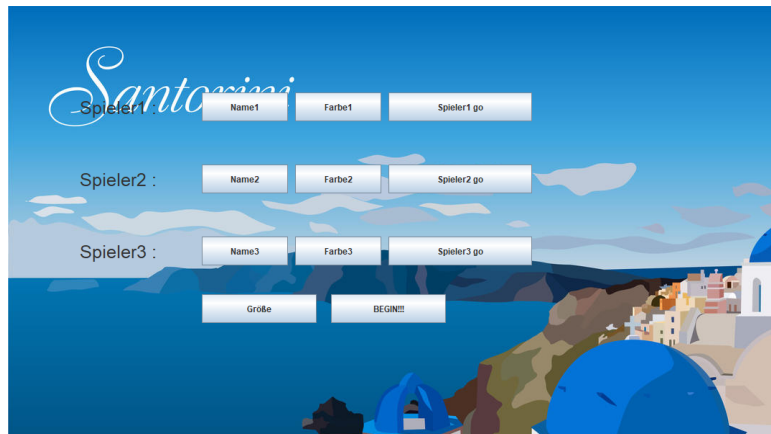


Abbildung 3: Definitionsmenü

Es hat 11 Buttons. Spieler sollen die Name eingeben. Auf Button "Names" wird eine kleine Fenster öffnen, um die Name einzugeben. Danach wird diesen Fenster selbst schließen. Spieler sollen die Farben wählen. Auf Button "Farben" wird eine kleine Fenster öffnen, um die Farben zu wählen. Danach wird diesen Fenster selbst schließen. Jetzt bietet es rot blau gelb und grün an. Wichtig ist Spieler muss Größe des Field wählen. Mit dem Button "Größe" wird SpieleField als $n \times n$ definieren. Button "Anfangen" wird das Spiel zum Spielfield leiten, und das Spiel fängt an.

1.3 Spielfield

1.3.1 3er-Spiele

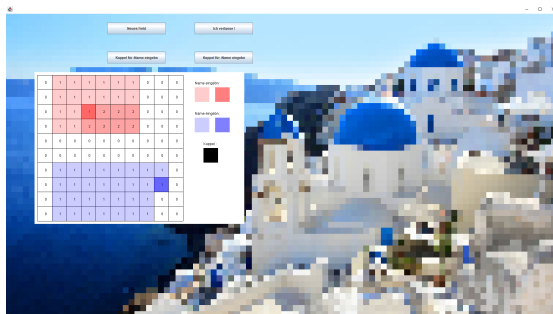


Abbildung 4: Spielfield

Die größewählbar SpielField mit $n \times n$ ausgezeigt. Die Namen und Farben der Spielern und Kuppel wird als Label auf recht Seite des SpielFeild ausgezeigt. Es gibt Buttons für verschiedene Spieler, um die Kuppels addieren. Das Spiel können immer neues Field erstellen, und die alle behalten wird auch gelöscht, wenn Spieler "Neues Field" drückt. Spieler kann verlassen, und alle Fenster schließen. Die Hoch des Gebäude wird als Nummer ausgedrückt.

1.4 Winners

Es wird angezeigt, wenn eine Spieler gewonnen hat. Und man das Button mal drückt, wird die alle Fenster schließen.

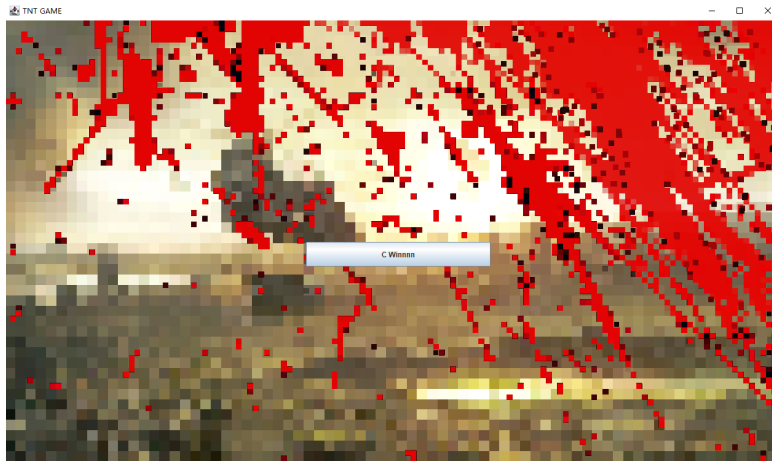


Abbildung 5: Winners

2 Maussteuerung(Youran Wang)(fertig am 15.6.2023)

Um das Spiel mit Maus zu steuern, benutze ich das Interface `MouseListener`. Und implementiere ich das Methode `mousePressed`, um die Ereignis abzuheören.

3 Regel prüfen(Youran Wang)(fertig am 15.7.2023)

3.1 Laufen Regel

Ich erstelle einige `ArrayList`, um die alle `POsition` den `Movement`, `Put` und `Kuppel` zu speichern. Jedes Drücken sollen die Längen den `Arraylisten` prüfen und vergleichen. Dann mache ich eine `State Machine`.

3.1.1 2erSpiel

Behalten	MoveA	vergleichen	MoveB	PutA	vergleichen	PutB
MoveA		=			=	
PutA		>			=	
MoveB		>			>	
PutB		=			>	

3.1.2 3erSpiel

	MoveA	MoveB	MoveB	MoveC	PutA	PutB	PutC	PutC
MoveA	=			=		=		=
PutA	>			=		=		=
MoveB	>			=		>		=
PutB	=			>		>		=
MoveC	=			>		=		>
PutC	=			=		=		>

Vor dem Bewegen und Bauen wird prüfen, ob jemand schon gewonnen hat. Die erste `Movement` und Bauen sind freiwählbar für alle Spieler. Deshalb das Regel wird nicht auf erste Schritt gewendet.

3.2 Movement Regel

- 1.Spieler kann nicht auf gegene Field bewegen.
- 2.Spieler muss auf nähr 8 Felder bewegen. Also oben, unter, links, recht, obenlinks, obenrecht, unterlinks und unterrecht.
- 3.Spieler kann nicht auf einen Field bewegen, wo höher als Spieler 2 ist.
- 4.Spieler kann nicht auf eine Kuppel bewegen

```
/**
 * Its the Methode to check, if A can move to here
 *
 * @return if a Move Right
 */
1个用法 A.kurz
public static boolean ifAMoveRight() {
    boolean ifamoveright = true;
    // if the position was been choose by player 2 and player 3
    if(!ifInList()==false && !ifInList()==false) {
        // if there is a Kuppel
        if (SpielField_Ser.field[SpielField_Ser.o][SpielField_Ser.b] < 9) {
            //if there is near to the last move
            if (
                (SpielField_Ser.o == (moveIN_Ser.player1.move.x[moveIN_Ser.player1.move.x.length - 1] - 1) && SpielField_Ser.b == (moveIN_Ser.player1.move.y[moveIN_Ser.player1.move.y.length - 1])) ||
                (SpielField_Ser.o == (moveIN_Ser.player1.move.x[moveIN_Ser.player1.move.x.length - 1] + 1) && SpielField_Ser.b == (moveIN_Ser.player1.move.y[moveIN_Ser.player1.move.y.length - 1])) ||
                (SpielField_Ser.o == (moveIN_Ser.player1.move.x[moveIN_Ser.player1.move.x.length - 1]) && SpielField_Ser.b == (moveIN_Ser.player1.move.y[moveIN_Ser.player1.move.y.length - 1] - 1)) ||
                (SpielField_Ser.o == (moveIN_Ser.player1.move.x[moveIN_Ser.player1.move.x.length - 1]) && SpielField_Ser.b == (moveIN_Ser.player1.move.y[moveIN_Ser.player1.move.y.length - 1] + 1)) ||
                (SpielField_Ser.o == (moveIN_Ser.player1.move.x[moveIN_Ser.player1.move.x.length - 1] + 1) && SpielField_Ser.b == (moveIN_Ser.player1.move.y[moveIN_Ser.player1.move.y.length - 1] + 1)) ||
                (SpielField_Ser.o == (moveIN_Ser.player1.move.x[moveIN_Ser.player1.move.x.length - 1] - 1) && SpielField_Ser.b == (moveIN_Ser.player1.move.y[moveIN_Ser.player1.move.y.length - 1] - 1)) ||
                (SpielField_Ser.o == (moveIN_Ser.player1.move.x[moveIN_Ser.player1.move.x.length - 1] - 1) && SpielField_Ser.b == (moveIN_Ser.player1.move.y[moveIN_Ser.player1.move.y.length - 1] + 1)) ||
                (SpielField_Ser.o == (moveIN_Ser.player1.move.x[moveIN_Ser.player1.move.x.length - 1] + 1) && SpielField_Ser.b == (moveIN_Ser.player1.move.y[moveIN_Ser.player1.move.y.length - 1] - 1)) ||
                (SpielField_Ser.o == (moveIN_Ser.player1.move.x[moveIN_Ser.player1.move.x.length - 1] + 1) && SpielField_Ser.b == (moveIN_Ser.player1.move.y[moveIN_Ser.player1.move.y.length - 1] + 1)) ||
                (SpielField_Ser.o == (moveIN_Ser.player1.move.x[moveIN_Ser.player1.move.x.length - 1] - 1) && SpielField_Ser.b == (moveIN_Ser.player1.move.y[moveIN_Ser.player1.move.y.length - 1] - 1)) ||
                (SpielField_Ser.o == (moveIN_Ser.player1.move.x[moveIN_Ser.player1.move.x.length - 1] - 1) && SpielField_Ser.b == (moveIN_Ser.player1.move.y[moveIN_Ser.player1.move.y.length - 1] + 1)) ||
                (SpielField_Ser.o == (moveIN_Ser.player1.move.x[moveIN_Ser.player1.move.x.length - 1] + 1) && SpielField_Ser.b == (moveIN_Ser.player1.move.y[moveIN_Ser.player1.move.y.length - 1] - 1)) ||
                (SpielField_Ser.o == (moveIN_Ser.player1.move.x[moveIN_Ser.player1.move.x.length - 1] + 1) && SpielField_Ser.b == (moveIN_Ser.player1.move.y[moveIN_Ser.player1.move.y.length - 1] + 1)) ||
            ) {
                ifamoveright = true;
            } else {
                System.out.println("Can't move there");
                ifamoveright = false;
            }
        } else {
            System.out.println("Can't move there");
            ifamoveright = false;
        }
    } else {
        System.out.println("Can't move there");
        ifamoveright = false;
    }
}
return ifamoveright;
}
```

Abbildung 6: MoveRegel

3.3 Put Regel

- 1.Spieler kann nicht auf gegene Field platzieren.
- 2.Spieler muss auf nähr 8 Felder platzieren. Also oben, unter, links, recht, obenlinks, obenrecht, unterlinks und unterrecht.
- 3.Spieler kann nur bauen, wo gleich hoch wie Spieler.
- 4.Spieler kann nicht auf eine Kuppel platzieren

```
/**
 * Its the Methode to check, if A can put to here
 *
 * @return if a Put Right
 */
1个用法 A.kurz
public static boolean ifAPutRight() {
    boolean ifaputright = true;
    // if the position was been choose by player 2 and player 3
    if(!ifInList()==false && !ifInList()==false) {
        // if there is near to the last move
        if (
            (SpielField_Ser.o == (moveIN_Ser.player1.move.x[moveIN_Ser.player1.move.x.length - 1] - 1) && SpielField_Ser.b == (moveIN_Ser.player1.move.y[moveIN_Ser.player1.move.y.length - 1])) ||
            (SpielField_Ser.o == (moveIN_Ser.player1.move.x[moveIN_Ser.player1.move.x.length - 1] + 1) && SpielField_Ser.b == (moveIN_Ser.player1.move.y[moveIN_Ser.player1.move.y.length - 1])) ||
            (SpielField_Ser.o == (moveIN_Ser.player1.move.x[moveIN_Ser.player1.move.x.length - 1]) && SpielField_Ser.b == (moveIN_Ser.player1.move.y[moveIN_Ser.player1.move.y.length - 1] - 1)) ||
            (SpielField_Ser.o == (moveIN_Ser.player1.move.x[moveIN_Ser.player1.move.x.length - 1]) && SpielField_Ser.b == (moveIN_Ser.player1.move.y[moveIN_Ser.player1.move.y.length - 1] + 1)) ||
            (SpielField_Ser.o == (moveIN_Ser.player1.move.x[moveIN_Ser.player1.move.x.length - 1] + 1) && SpielField_Ser.b == (moveIN_Ser.player1.move.y[moveIN_Ser.player1.move.y.length - 1] + 1)) ||
            (SpielField_Ser.o == (moveIN_Ser.player1.move.x[moveIN_Ser.player1.move.x.length - 1] - 1) && SpielField_Ser.b == (moveIN_Ser.player1.move.y[moveIN_Ser.player1.move.y.length - 1] - 1)) ||
            (SpielField_Ser.o == (moveIN_Ser.player1.move.x[moveIN_Ser.player1.move.x.length - 1] - 1) && SpielField_Ser.b == (moveIN_Ser.player1.move.y[moveIN_Ser.player1.move.y.length - 1] + 1)) ||
            (SpielField_Ser.o == (moveIN_Ser.player1.move.x[moveIN_Ser.player1.move.x.length - 1] + 1) && SpielField_Ser.b == (moveIN_Ser.player1.move.y[moveIN_Ser.player1.move.y.length - 1] - 1)) ||
            (SpielField_Ser.o == (moveIN_Ser.player1.move.x[moveIN_Ser.player1.move.x.length - 1] + 1) && SpielField_Ser.b == (moveIN_Ser.player1.move.y[moveIN_Ser.player1.move.y.length - 1] + 1)) ||
            (SpielField_Ser.o == (moveIN_Ser.player1.move.x[moveIN_Ser.player1.move.x.length - 1] - 1) && SpielField_Ser.b == (moveIN_Ser.player1.move.y[moveIN_Ser.player1.move.y.length - 1] - 1)) ||
            (SpielField_Ser.o == (moveIN_Ser.player1.move.x[moveIN_Ser.player1.move.x.length - 1] - 1) && SpielField_Ser.b == (moveIN_Ser.player1.move.y[moveIN_Ser.player1.move.y.length - 1] + 1)) ||
            (SpielField_Ser.o == (moveIN_Ser.player1.move.x[moveIN_Ser.player1.move.x.length - 1] + 1) && SpielField_Ser.b == (moveIN_Ser.player1.move.y[moveIN_Ser.player1.move.y.length - 1] - 1)) ||
            (SpielField_Ser.o == (moveIN_Ser.player1.move.x[moveIN_Ser.player1.move.x.length - 1] + 1) && SpielField_Ser.b == (moveIN_Ser.player1.move.y[moveIN_Ser.player1.move.y.length - 1] + 1)) ||
        ) {
            // if there is a Kuppel and the last move was high enough to reach this field
            if ((SpielField_Ser.field[SpielField_Ser.o][SpielField_Ser.b] == SpielField_Ser.field[moveIN_Ser.player1.move.x[moveIN_Ser.player1.move.x.length - 1]][moveIN_Ser.player1.move.y[moveIN_Ser.player1.move.y.length - 1]]) &&
                (SpielField_Ser.field[SpielField_Ser.o][SpielField_Ser.b] < 9)) {
                ifaputright = true;
            } else {
                ifaputright = false;
                System.out.println("Can't put there");
            }
        } else {
            ifaputright = false;
            System.out.println("Can't put there");
        }
    } else {
        ifaputright = false;
        System.out.println("Can't put there");
    }
}
return ifaputright;
}
```

Abbildung 7: putRegel

3.4 Kuppel Regel

- 1.Spieler muss auf nähr 8 Fielde Kuppel platzieren. Also oben, unter, links, recht, obenlinks, obenrecht, unterlinks und unterrecht.
- 2.Spieler kann nicht auf eine Kuppel platzieren

```
/**
 * Its the Methode to check, if A can Kuppel to here
 *
 * @return if a kuppel Right
 */
1个用法 A Euzil
public static boolean ifKuppelRight() {
    boolean ifKuppelRight = true;
    // If there is near to the last move
    if ( (Spielerfeld_3er.a == (moveIN_3er.player1.move.x[moveIN_3er.player1.move.x.length - 1] - 1) && Spielerfeld_3er.b == (moveIN_3er.player1.move.y[moveIN_3er.player1.move.y.length - 1])) ||
        (Spielerfeld_3er.a == (moveIN_3er.player1.move.x[moveIN_3er.player1.move.x.length - 1] + 1) && Spielerfeld_3er.b == (moveIN_3er.player1.move.y[moveIN_3er.player1.move.y.length - 1])) ||
        (Spielerfeld_3er.a == (moveIN_3er.player1.move.x[moveIN_3er.player1.move.x.length - 1]) && Spielerfeld_3er.b == (moveIN_3er.player1.move.y[moveIN_3er.player1.move.y.length - 1] - 1)) ||
        (Spielerfeld_3er.a == (moveIN_3er.player1.move.x[moveIN_3er.player1.move.x.length - 1]) && Spielerfeld_3er.b == (moveIN_3er.player1.move.y[moveIN_3er.player1.move.y.length - 1] + 1)) ) {
        // If there is a Kuppel and the last Move was high enough to reach this Field
        if ((Spielerfeld_3er.Field[Spielerfeld_3er.a][Spielerfeld_3er.b] == Spielerfeld_3er.Field[moveIN_3er.player1.move.x[moveIN_3er.player1.move.x.length - 1]][moveIN_3er.player1.move.y[moveIN_3er.player1.move.y.length - 1]]) &&
            (Spielerfeld_3er.Field[Spielerfeld_3er.a][Spielerfeld_3er.b] < 5)) {
            ifKuppelRight = true;
        } else {
            ifKuppelRight = false;
            System.out.print("cant Kuppel there");
        }
    } else {
        ifKuppelRight = false;
        System.out.print("cant Kuppel there");
    }
    return ifKuppelRight;
}
```

Abbildung 8: kuppelRegel

3.5 Win Regel

Wenn eine Spieler auf ein Field bewegt, wo mit 4 Level hoch ist.

```
/**
 * its the Methode to check, if A already won the Game
 *
 * @return if A win
 */
1个用法 A Euzil
public static boolean ifAwin() {
    if (Spielerfeld_3er.Field[Spielerfeld_3er.a][Spielerfeld_3er.b] == 4) {
        System.out.println("A win");
        return true;
    } else {
        return false;
    }
}
```

Abbildung 9: winRegel

4 Name und Farben(Youran Wang)(fertig am 20.6.2023)

Im Packet "engine" habe ich das Spieler mit Eigenschaft: Name und Farbe definieren. Und das Spieler wird als Konstruktoren erstellt.

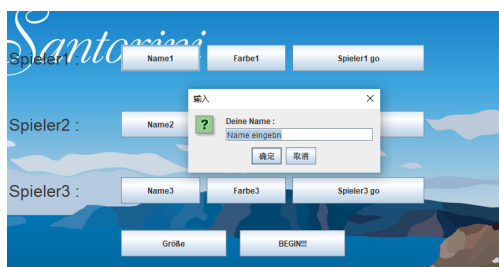


Abbildung 10: Name

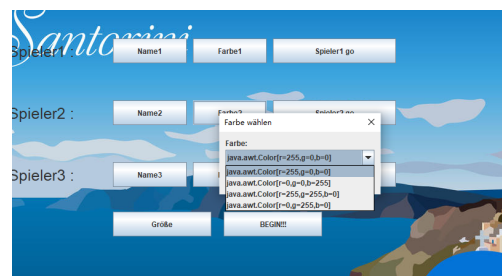


Abbildung 11: Farbe

5 Größe des Spielfeld(Youran Wang)(fertig am 27.7.2023)

Am Anfang wird das Größe des Field definieren. Und es wird als Parameter des Field. Im Spielfeld.java zeichnet das Field mit Größe $n \times n$.



Abbildung 12: Größe des Fields

6 Verlassen(Youran Wang)(fertig am 26.7.2023)

Es gibt 2 Buttons.

1. Neues Field wird das Frame neue mal erstellen. Und alle Positionen ,die im ArrayList den MoveIn.java und putIn.java gespeichert werden, wird auch gelöscht. Deshalb bekommen wir eine ganz neues Spielfeld.

2. Verlassen Button lass die alle Prozess schließen und die alle Fenster wird auch zugemacht.

```
// Start a new Field
JButton jButton_new = new JButton( text: "Neues Field");
jButton_new.setBounds( x: 350, y: 30, width: 200, height: 40);
jButton_new.addActionListener(e -> {
    // clear all parameter and shut down the JFrame
    JFrame.dispose();
    kuppel.clear();
    moveIn.clear();
    putIn.clear();
    for (int i = 0; i < Field.length; i++) {
        for (int j = 0; j < Field[i].length; j++) {
            Field[i][j] = 0;
        }
    }
    JFrame.setContentPane(check.oneMoreLook());
});
JFrame.add(jButton_new);
// close the JFrame
JButton jButton_quit = new JButton( text: "Ich verlasse !");
jButton_quit.setBounds( x: 450, y: 30, width: 200, height: 40);
jButton_quit.addActionListener(e -> {
    System.exit( status: 0);
});
JFrame.add(jButton_quit);
```

Abbildung 13: verlassen

7 Gods(Finn Hinrichs)(fertig am 1.8.2023)

8 KI(Fabio Drissen)(fertig am 30.7.2023)

9 Speichern/Laden(Tim Wiegert)(nicht fertig)

10 High Score(Tim Wiegert)(nicht fertig)