

# Shrine Survival

Projekt  
**Shrine Survival**

# Über mich

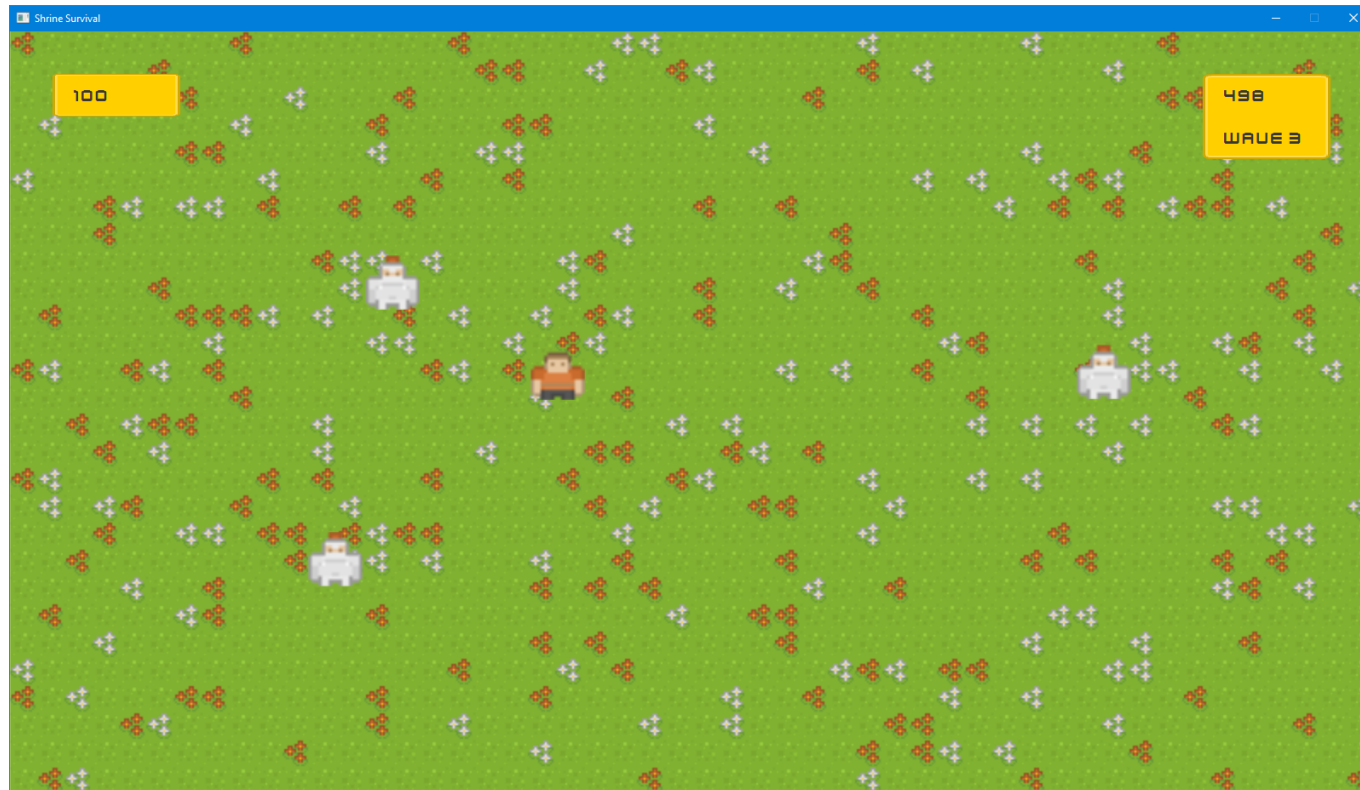
- Name: Erik Wiesinger
- Alter: 30
- Herkunft: Gmünd – Niederösterreich
- Was bisher geschah:
  - Gelernter Werkzeugbautechniker
  - 8 Jahre bei Husky KTW

# Die Idee zu Beginn

- 2D Top Down Spiel
- Wellen an Gegnern überleben
- Klassenauswahl
- Itemnutzung
- Charakter Creator
- Highscore erzielen
- Mit anderen Spielern vergleichen



# Aktueller Screenshot

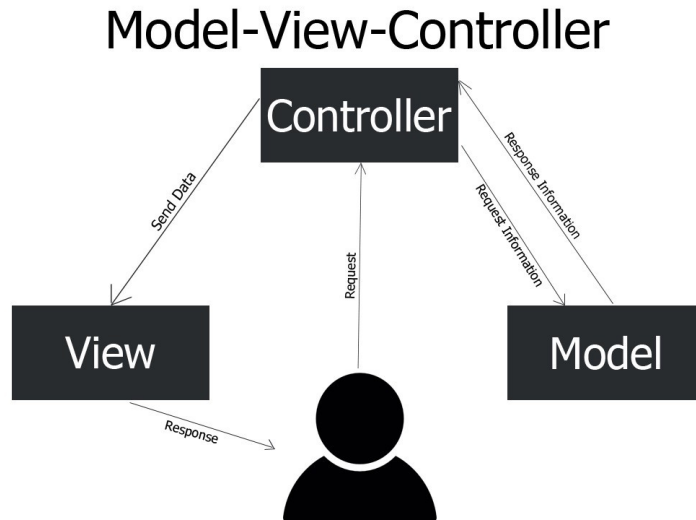


# Technologie Stack



# Architekturmuster / Design Patterns

- Model View Controller

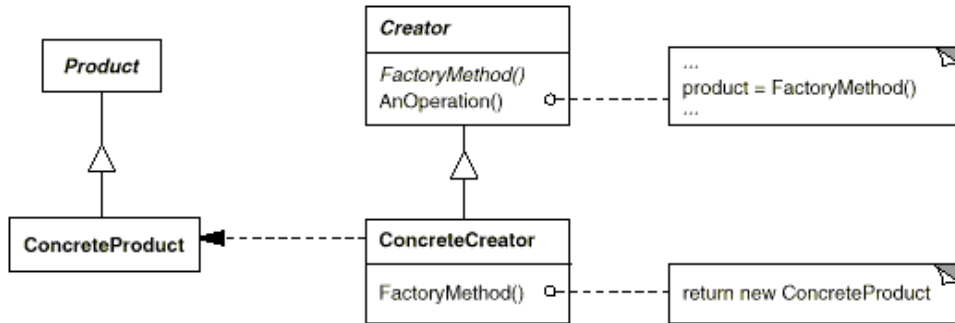


- Singleton (Datenbank Connector)



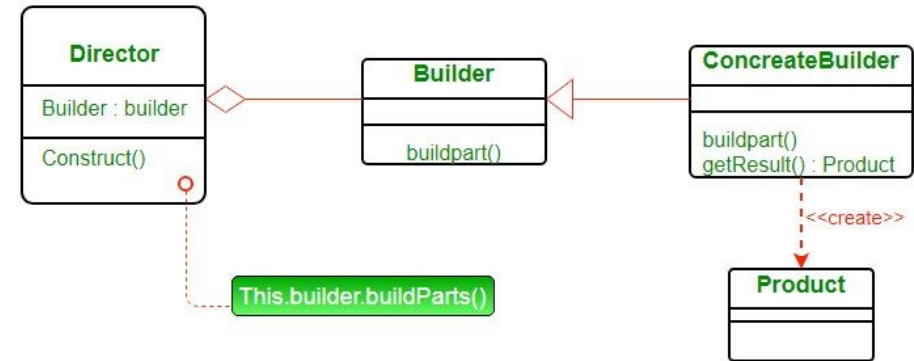
# Architekturmuster / Design Patterns

- Factory Pattern (Entity Objekt)



- Builder (Sprite Objekt)

*UML diagram of Builder Design pattern*





# Code

- JDK 14

## Enhanced Switch

- Kein break
- Multiple cases kombiniert

```
mainPane.getScene().setOnKeyPressed(event -> {  
    switch (event.getCode()) {  
        case UP -> goUp = true;  
        case DOWN -> goDown = true;  
        case LEFT -> goLeft = true;  
        case RIGHT -> goRight = true;  
        case W -> attackUp = true;  
        case S -> attackDown = true;  
        case A -> attackLeft = true;  
        case D -> attackRight = true;  
    }  
});
```

# Mehr Code

- Liste mit Lambda
- AtomicReference
  - Updated bei jedem Durchlauf

```
int totalSize = buttonList.size() * 70;
AtomicReference<Double> buttonStartY = new AtomicReference<>(
    (mainPane.getHeight() / 2) - (totalSize / 2)
);

buttonList.forEach(button -> {
    mainPane.getChildren().add(button);
    button.setLayoutX(buttonStartX);
    button.setLayoutY(buttonStartY.get());
    buttonStartY.updateAndGet(v -> v + 70);
});
```

# Assets von Kenney.nl



# Spritesheet to Canvas

- Objekt "SpriteSheet"  
Liest .png aus Resources

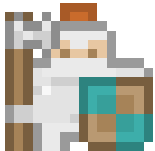
```
public SpriteSheet(String url) {  
    this.tilesize = 16;  
    this.margin = 1;  
    try {  
        image = new Image(getClass().getResource(url).toString());  
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
}
```

- Objekt Sprite  
Erstellt Canvas mit  
Tilesize

```
public Sprite(SpriteSheet spritesheet) {  
    this.spritesheet = spritesheet;  
    canvas = new Canvas(  
        spritesheet.getTilesize(),  
        spritesheet.getTilesize());  
}
```

# Sprite hinzufügen

- `Sprite.add(x, y)`
  - Errechnen der X,Y Position
  - Hinzufügen des Bildes



```
public void addSprite(int x, int y) {  
  
    double tileSize = spritesheet.getTileSize();  
    double margin = spritesheet.getMargin();  
  
    double startX = x * (tileSize + margin);  
    double startY = y * (tileSize + margin);  
  
    if (x > 1) {  
        startX += margin;  
    }  
  
    canvas.getGraphicsContext2D().drawImage(spritesheet.getImage(),  
        startX, startY, tileSize, tileSize,  
        0, 0, tileSize, tileSize);  
}
```

# Live Demo

Github

[Github.com/shuffle-coding](https://github.com/shuffle-coding)

Ende

Danke für eure Aufmerksamkeit