

# Boulderdash YAMAI

Immanuel Bach, Aida Bakhtiari, Andreas Kößler, Yichen Liu, Maximilian Schleder



# Überblick

Hauptregeln, Level-/Feldklasse, Level-IO **Maximilian**

Gamecontroller, Audio **Yichen**

(Menü-)GUI **Aida**

Levelregeln, Beispiellevel, Leveleditor (*+Immanuel*),  
KI(*+Immanuel*) **Andreas**

Themeditor, Spielfeldanzeige **Immanuel**

# Hauptregeln | Level-/Feldklasse | Level-IO (1)

- **LevelFactory**

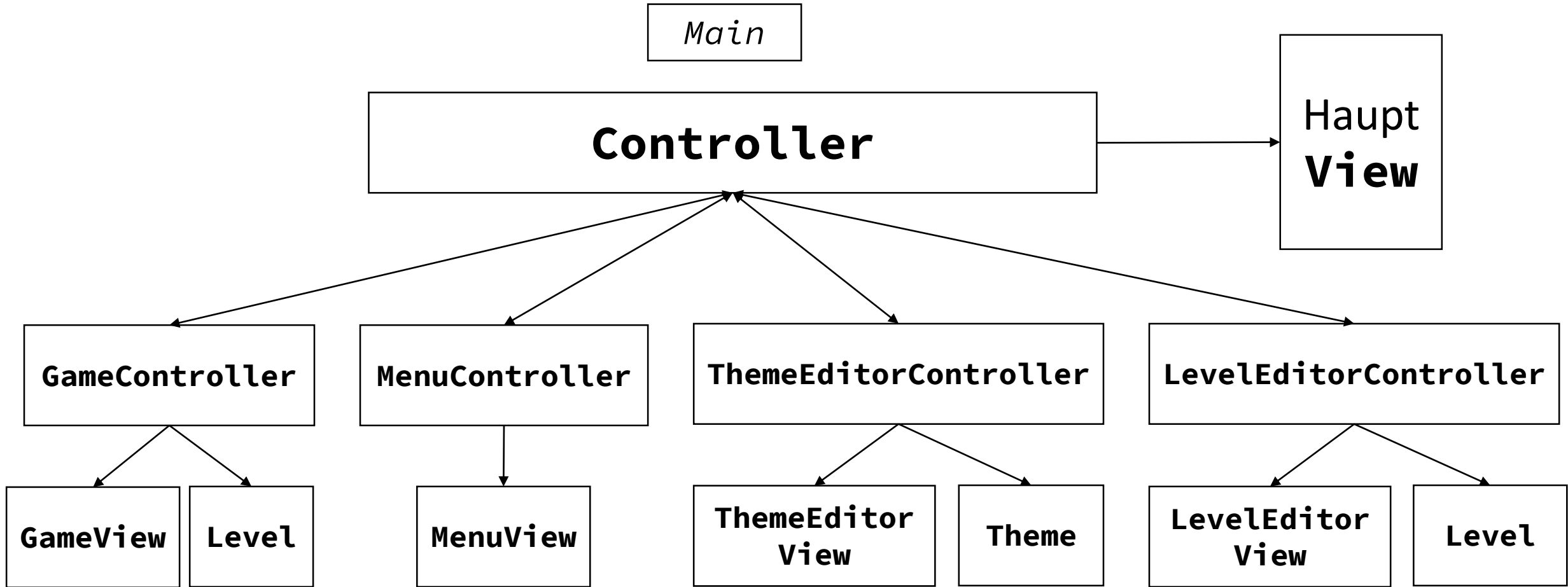
- Import: **Level** starten
- Export:
  - Savegames
  - Eigene **Level** speichern (**LevelEditor**)

- **Level**

- Enthält **Felder**
  - Methoden zur Manipulation von Spielfeldinhalten
- Hauptregeln
  - fest als Java-Code implementiert
- **LevelRules**

# Hauptregeln | Level-/Feldklasse | Level-IO (2)

- **Level** (Forts.)
  - Reset von Zusatzwerten
  - Gewinnen/Verlieren
  - Aktuelle Medaillen berechnen



Nur eine Stage:

- Zurück zum Menü und wieder zum Spiel :
  - Immer prüfen, ob es vorher schon einen **MenuController** oder **GameController** gibt
- Alert Fenster
  - Alert Fenster für Exit → **MenuController** geupdatet, Listener entfernen
  - Alert Fenster für Restart → **GameController** geupdatet, Listener entfernen und updaten
- **ThemeEditorController**
- **LevelEditorController**

# Levelregeln

**Rule**

**RuleElementOriginal**

**RuleElementResult**

**TokenMatcher**

erkennt einen oder mehrere Tokens  
sowie Zusatzwerte

**TokenReplacer**

Setzt neuen Token,  
setzt Zusatzwerte

# Rule → execute()

```
int i = 0; // Zähler der äußeren Schleife
while (between(startI, i, endI)) {
    int j = initJ(); // Initialisierung von j gemäß DIRECTION (Zähler der inneren Schleife)

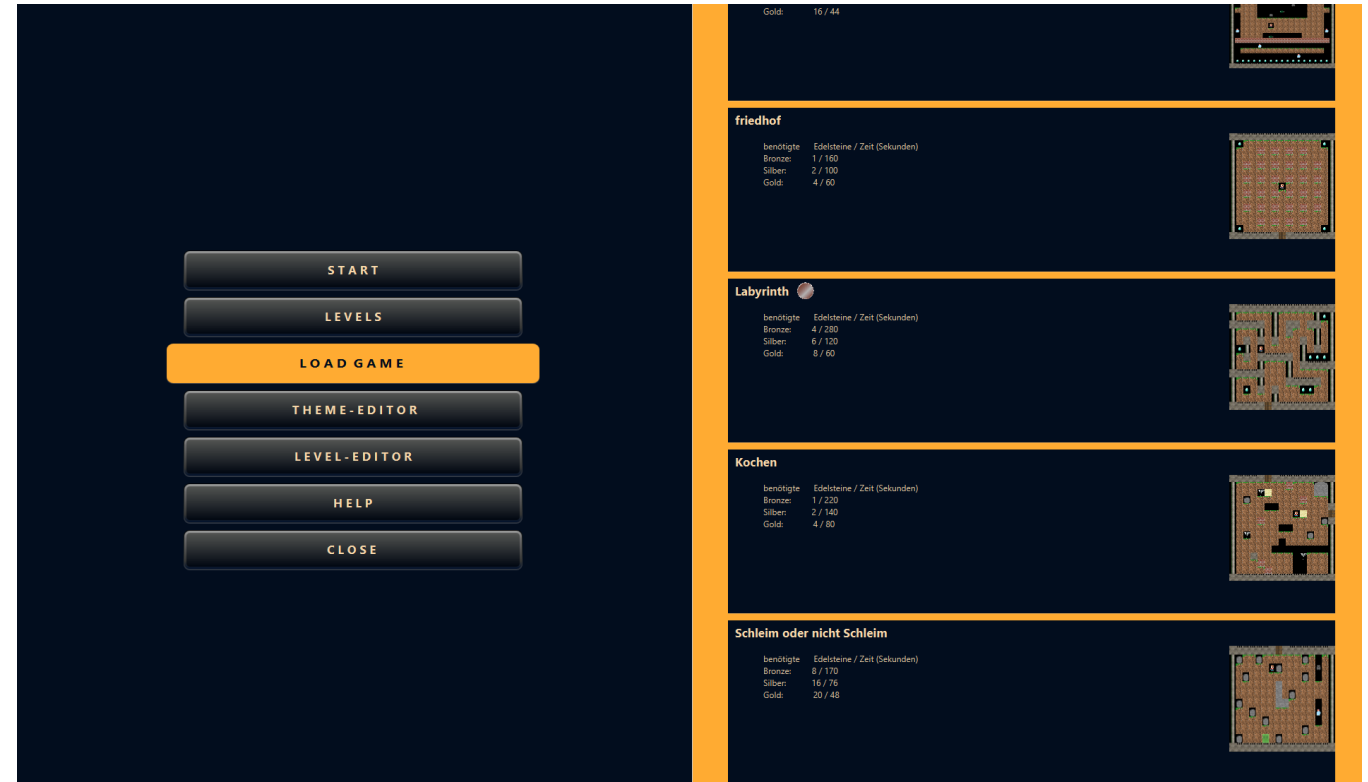
    while (between(startJ, j, endJ)) {
        List<Feld> feldList = new ArrayList();
        try {
            int y = initY(i, j);
            int x = initX(i, j);
            feldList.add(map[y][x]);
            map[y][x].getNeighboursRecursive(direction, ruleLength, feldList);

            if (matchFeldListWithOriginalList(feldList)) {
                replaceTokens(feldList)
                j = hoverOverRuleLengthJNW(ruleLength, j);
                if (!between(startJ, j, endJ))
                    throw new IndexOutOfBoundsException();
            } else {
                j = incOrDecJNW(j);
            }
        } catch (IndexOutOfBoundsException e) {
            break;
        }
        i++;
    }
}
```

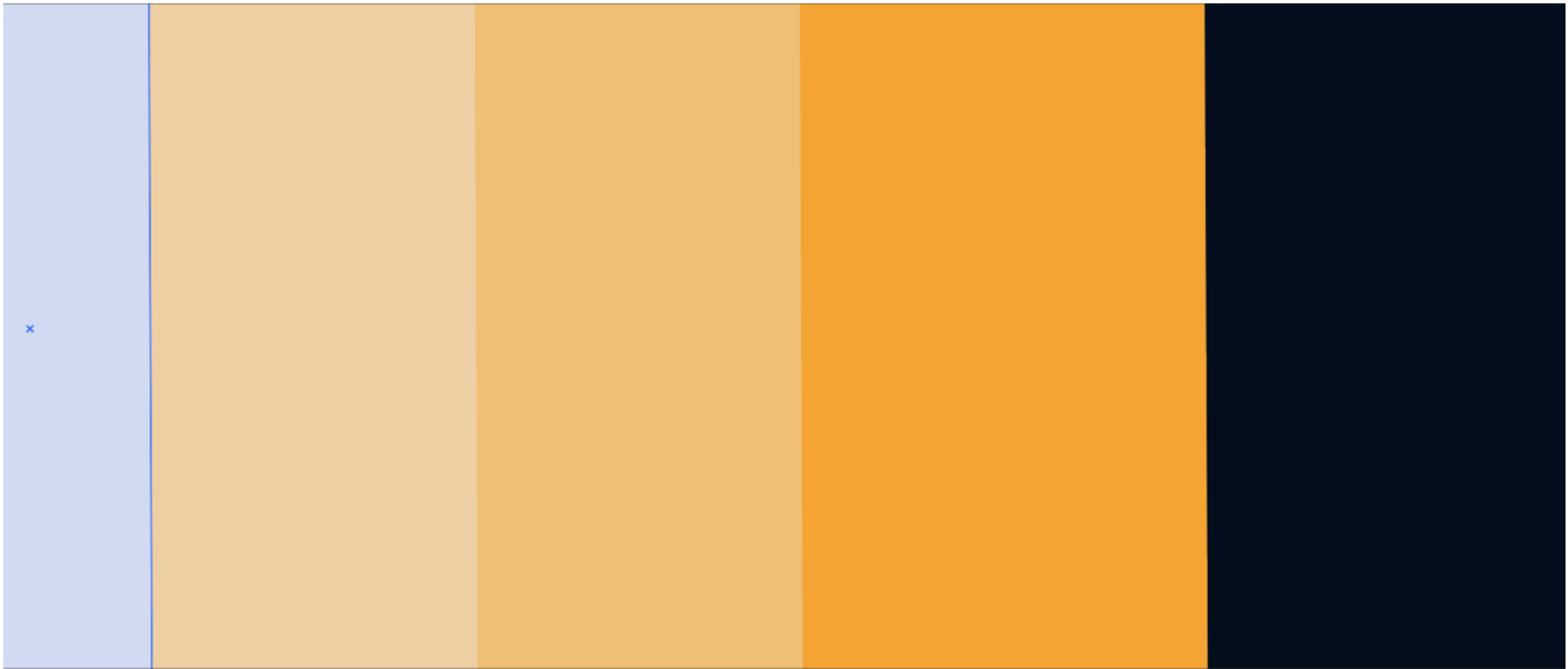


# GUI und Styling

- Stage vs Scene
- Menu :
  - 2 SPALTEN Layout (ContentFrame)
  - linke Spalte : fixiert
  - rechte Spalte : austauschbar
- LevelItem



# CSS-Styling



# Leveleditor

ME

MUD

STONE

GEM

EXIT

WALL

BRICKS

PATH

EXPLOSION

SLIME

SWAPLING

BLOCKLING

XLING

GHOSTLING

FIRE

NORTHTHING

EASTTHING

SOUTHTHING

WESTTHING

POT

SIEVE

SAND

Levelname

friedhof

Levelgröße

15 x 15

Schwierigkeit

5

Ziele

Gems

1

2

4

Zeit

800

500

300

friedhof.json

Speichern

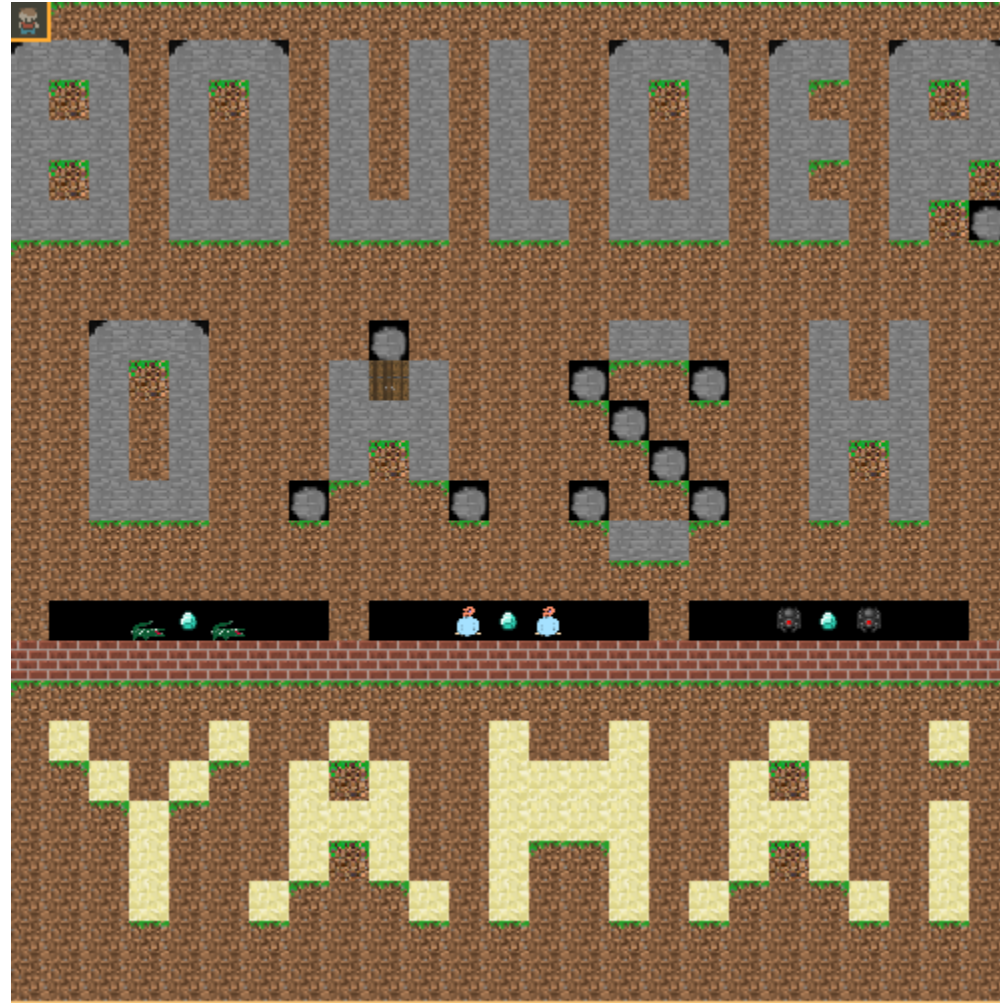
Zurück zum Menü

Property	Wert
Kein Content in Tabelle	

Property

Hinzufügen

# Beispiellevel



# KI

- Interface **AI** -> **InputDirection** getNextMove()

Vorgehen von **Robot**:

- Falls ein letztes Ziel vorhanden, setze den Weg dorthin fort
- Wenn nicht, dann: Suche den nächsten GEM
- Falls GEM vorhanden, finde einen Pfad, der mit einem **sicheren** Schritt beginnt
- Falls kein GEM vorhanden, suche einen Weg zum nächsten EXIT
- Falls kein EXIT gefunden, laufe zu einem zufälligen Ort
- Falls kein sicherer Pfad gefunden, bleibe stehen (und stirb wahrscheinlich)

# Themes (Themeeditor) und flüssige Animationen

- Theme enthält
  - Für jedes Token
    - für verschiedene Situationen (Feld hat einen Nachbarn, Zwei Nachbarn ...)
      - Bilder für unterschiedlichen Positionen (oberer Nachbar, unterer Nachbar , ...)
- Flüssige Animationen
  - Innerhalb eines Ticks
  - Erweiterungen der Transitionklasse

# Bugs/Verbesserungsmöglichkeiten

- **Hauptregeln**
  - Man wird nicht von herunterfallenden GEMs erschlagen, wenn man sie gerade einsammelt
- **KI**
  - Schließt sich manchmal ein
  - Bekämpft Gegner nicht aktiv
- **Leveleditor**
  - Keine Unterstützung für Levelregeln
  - Begrenzte Kartengröße, da die Anzeige zu klein ist
- **Themeeditor**
  - Bilder können nicht mehr gelöscht werden
- **Menü**
  - Viele überflüssige Levelimports
- **Animation**
  - Hardgecodete Animationsgeschwindigkeit, dadurch Vorspulen nicht möglich