



**THM**

TECHNISCHE HOCHSCHULE MITTELHESSEN

**CAMPUS  
FRIEDBERG**

**IEM**

Informationstechnik-  
Elektrotechnik-Mechatronik

# Präsentation

## Rush B

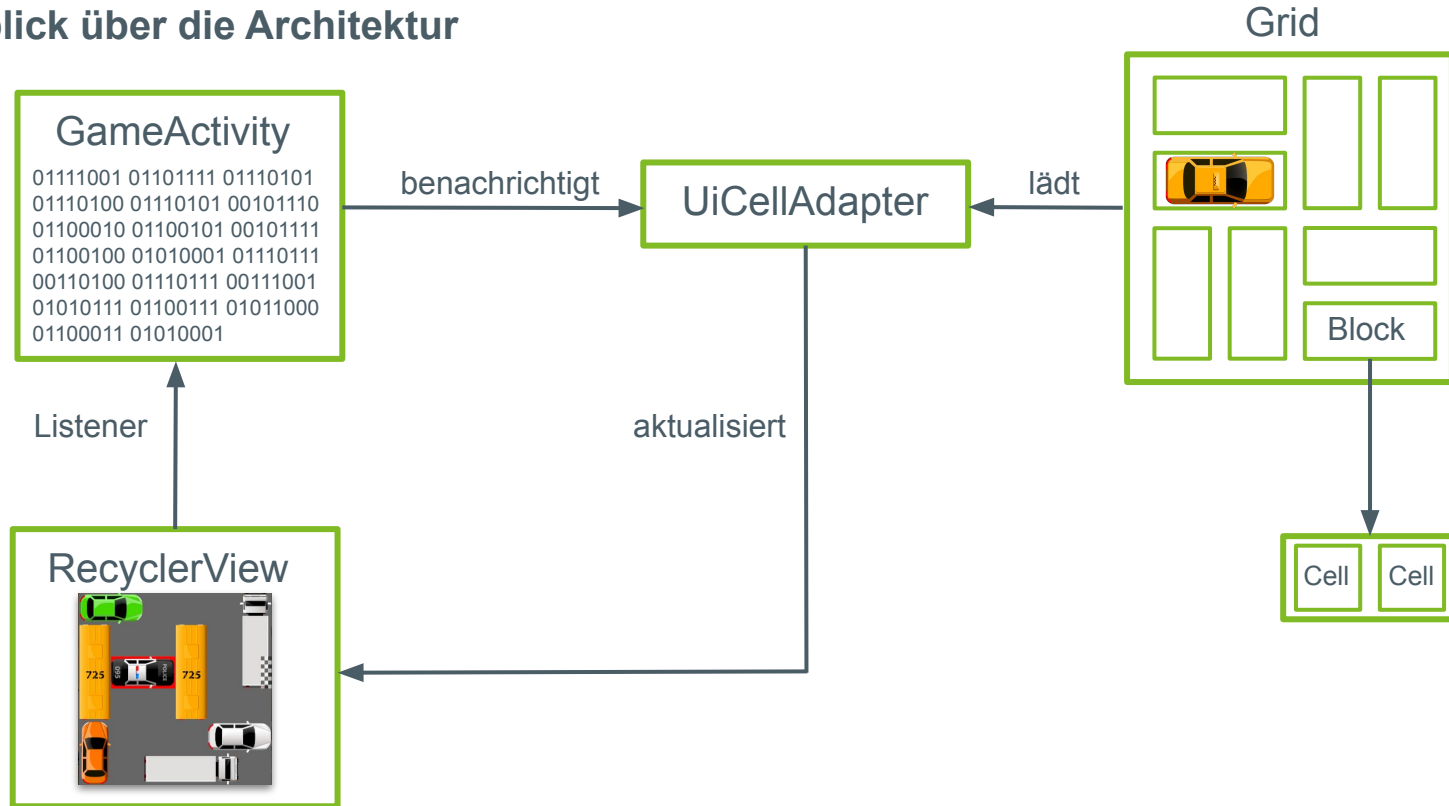
Silas Buchwald, Noah Stürtz, Adrian Werner

## Idee

- Spiel für zwischendurch
- Orientiert an Puzzle Game “Rush Hour”
- verschiedene Spiel-Modi
  - Level-Modus (Must-Have)
    - vorgefertigte Level in einer Übersicht
    - unterschiedliche Schwierigkeiten
    - zurücksetzen des Spielfeldes
  - Time Breaker Modus (Nice-To-Have)
    - Zeitvorgabe - Schaffe so viele Levels wie möglich
    - automatisch generierte Spielfelder
    - Leaderboard / Highscore

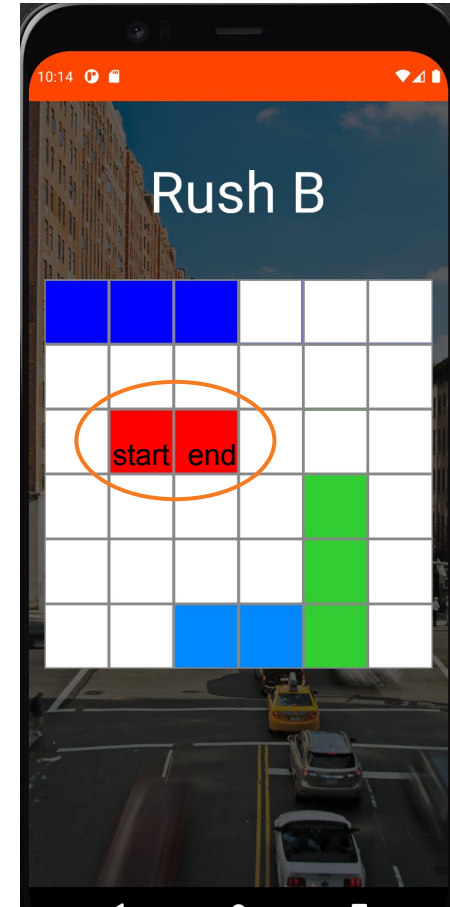


## Überblick über die Architektur



## Überblick über die Architektur

| Block  |
|--|
| <ul style="list-style-type: none"> <li>- id : Int</li> <li>- orientation : Orientation</li> <li>- start : Position</li> <li>- end : Position</li> <li>- color : Colors</li> <li>- type : String</li> <li>- isStart : Boolean</li> </ul>                              |
| <ul style="list-style-type: none"> <li>- contains(p : Position) : Boolean</li> <li>- contains(block : Block) : Boolean</li> <li>- isValid(width : Int, height : Int) : Boolean</li> <li>- eachPos() : List&lt;Position&gt;</li> <li>- toString() : String</li> </ul> |





**THM**

TECHNISCHE HOCHSCHULE MITTELHESSEN

**CAMPUS  
FRIEDBERG**

**IEM**

Informationstechnik-  
Elektrotechnik-Mechatronik

# Live-Demo

## Code: Generation

```
do {
    when (orientation) {
        Block.Orientations.HORIZONTAL -> {
            do {
                position = Position(Random.nextInt(0, width - size), Random.nextInt(0, height))
            } while (position.y == start.start.y)
        }
        Block.Orientations.VERTICAL -> {
            position = Position(Random.nextInt(0, width), Random.nextInt(0, height - size))
        }
    }
}

block = if(size == 2) {
    val type = cars[Random.nextInt(cars.size)]
    Block(size, orientation, position, color, type)
} else {
    val type = trucks[Random.nextInt(trucks.size)]
    Block(size, orientation, position, color, type)
}
while ((list.any { it.contains(block) } || !block.isValid(width, height)) && counter < limit)
```



## Code: Move

```
when(block.orientation) {  
  Block.Orientations.HORIZONTAL -> {  
    when(direction) {  
      true -> {  
        if(list.none { it.contains(Position(block.end.x+1, block.start.y))} && block.end.x+1 < width ) {  
          block.start.x++  
          block.end.x++  
        }  
      }  
    }  
    (...)  
    (...)  
  }  
}
```



## Code: JSON

```
{
  "difficulty": "Beginner",
  "blocks": [
    {
      "length": 2,
      "orientation": "HORIZONTAL",
      "x": 1,
      "y": 2,
      "type": "police_car",
      "start": true
    },
    {
      "length": 3,
      "orientation": "VERTICAL",
      "x": 0,
      "y": 0,
      "type": "yellow_bus"
    },
    (...)
  ]
}
```





## Herausforderungen / Schwierigkeiten

- KTX / GTX Library
- schwere Level generieren
  - rechenintensiv & komplex
- Speicherung von generierten Leveln
  - Copy by reference
- Rotieren von Grafiken
  - doppelte Grafiken für vertikal und horizontal

## Ausblick von “Rush B”

- Speicherung von generierten Leveln
- Shop mit Sternensystem
- Drag & Drop
- mehr vorgefertigte Level
- Level-Lösungen



## Fazit zur Motivation

- Spiel für zwischendurch ✓
- Android-App Programmierung lernen ✓
- Kotlin lernen ✓
- prozedurale Generierung von Spielfeldern / Leveln üben ✓
- für eigenes Portfolio ✓



**THM**

TECHNISCHE HOCHSCHULE MITTELHESSEN

**CAMPUS  
FRIEDBERG**

**IEM**

Informationstechnik-  
Elektrotechnik-Mechatronik

**Vielen Dank für die Aufmerksamkeit.  
Gibt es noch Fragen oder Anmerkungen?**