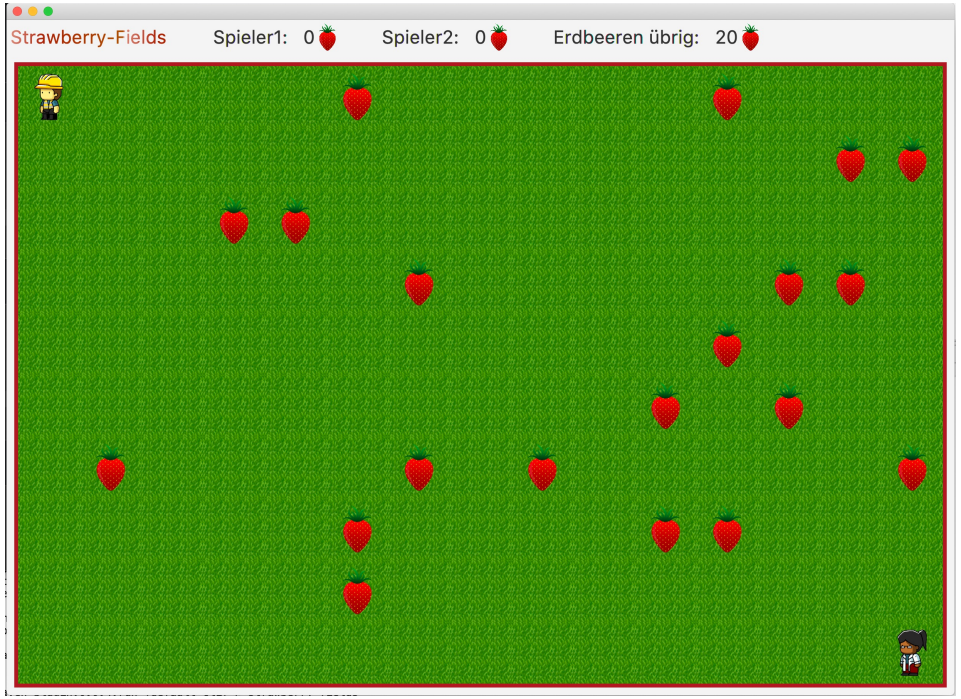
Applikations Entwickler

Java auf dem Weg zum Lehrabschluss

Strawberry-Fields

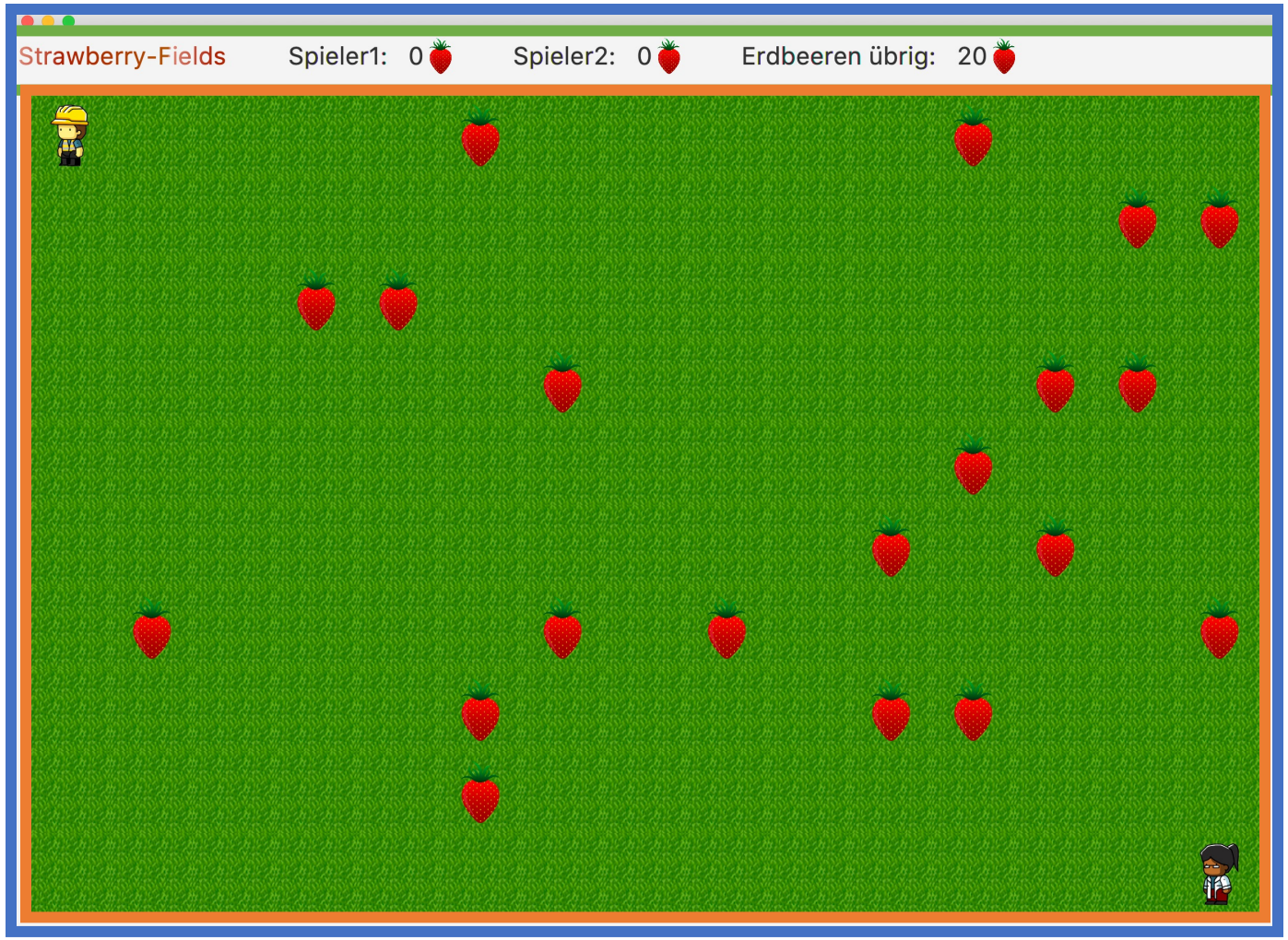
- Spiel mit GUI
  - 2 Spieler
  - Erdbeeren sammeln
  - Wer zuerst mehr Erdbeeren gesammelt hat gewinnt
  - Spiellogik beliebig erweiterbar



game-view.fxml

game-view:
Hält die beiden anderen Views zusammen

field-view:
Zeigt das Spielfeld an

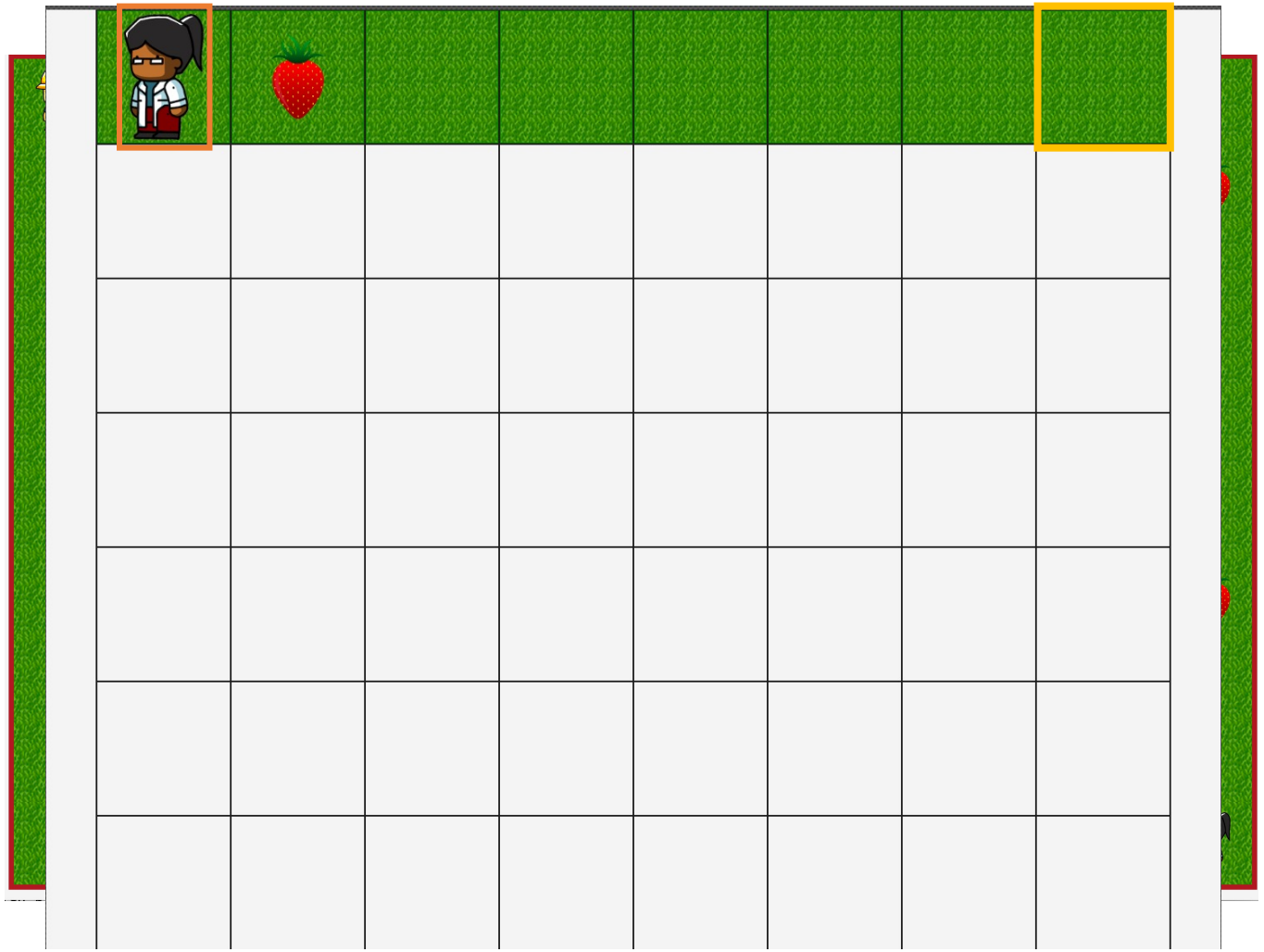


game-info-view:
Zeigt den Spielstand an

field-view.fxml

item-view:
Stellt ein Item am Spielfeld dar

ImageView stellt ein Image dar



square-view:
Stellt ein Rechteck am Spielfeld dar.

StackPane kann mehrere Nodes übereinander stapeln

fx:id=fieldView

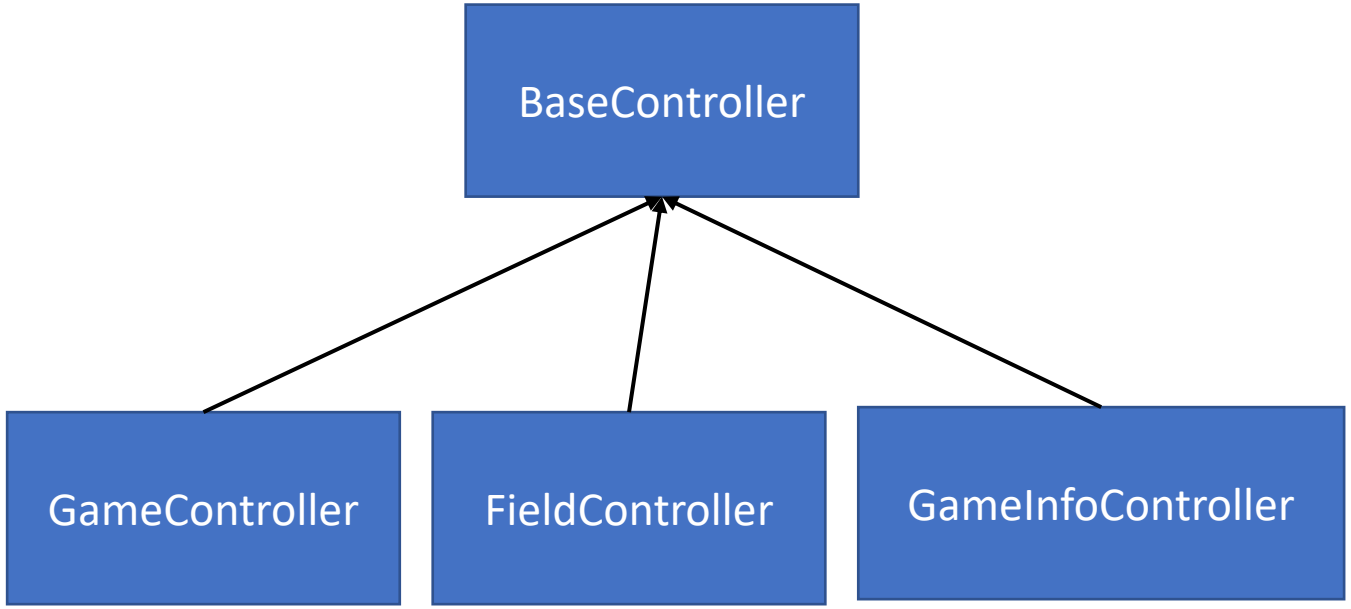
game-info-view.fxml

player1NameLabel player2NameLabel

player1PointsLabel player2PointsLabel strawberrysLeftLabel

Controllers: fx:id

- GameController
  - leer
- FieldController
  - ***GridPane fieldView***
- GameInfoController
  - **Label** player1NameLabel
  - **Label** player2NameLabel
  - **Label** player1PointsLabel
  - **Label** player2PointsLabel
  - **Label** strawberriesLeftLabel



FieldController

- ***fieldView*** dynamisch befüllen
  - void generateSquares()
  - void updateSquareAtIndex(int ***index***)
  - Image imageForItem(String item)
  - void onKeyPressed(KeyEvent event)

Model: Field

- Simples Model um Grundfunktionalität des GUI zu testen.

  - Field
    - ObservableList<Square> squares
  - Square
    - ObjectProperty<String> item;

Field -> FieldController

- Auf Änderungen im Model reagieren und im Controller den View anpassen
- Field:
  - ObservableList mit Callback initialisieren
    - Damit wird ListChangeListener aufgerufen bei Änderungen der Property
- FieldController:
  - Auf field.getSquares() Listener registrieren, und darin updateSquareAtIndex aufrufen

item bewegen

- Item finden
- Item lesen
- Item in Start-Square überschreiben
- Item in End-Square einfügen

Model: Game

- Hält den State und die Logik des Spiels und wird mit der GUI verbunden.
- State:
  - Field field
  - Position currentPosition
  - Position startPosition
  - Player
  - …

Model: Game

- Logik:
  - void initItems()
  - void handleKeyPressed(KeyCode)
  - boolean move(Position position)
  - Field getField()

- Position:
  - POJO
    - int x
    - int y

Game -> Controllers

- Alle Controller sollen **dasselbe Game** verwenden
- Idee: **BaseController**
  - **static Game game**
- **Alle Controller davon abgeleitet**