SWT

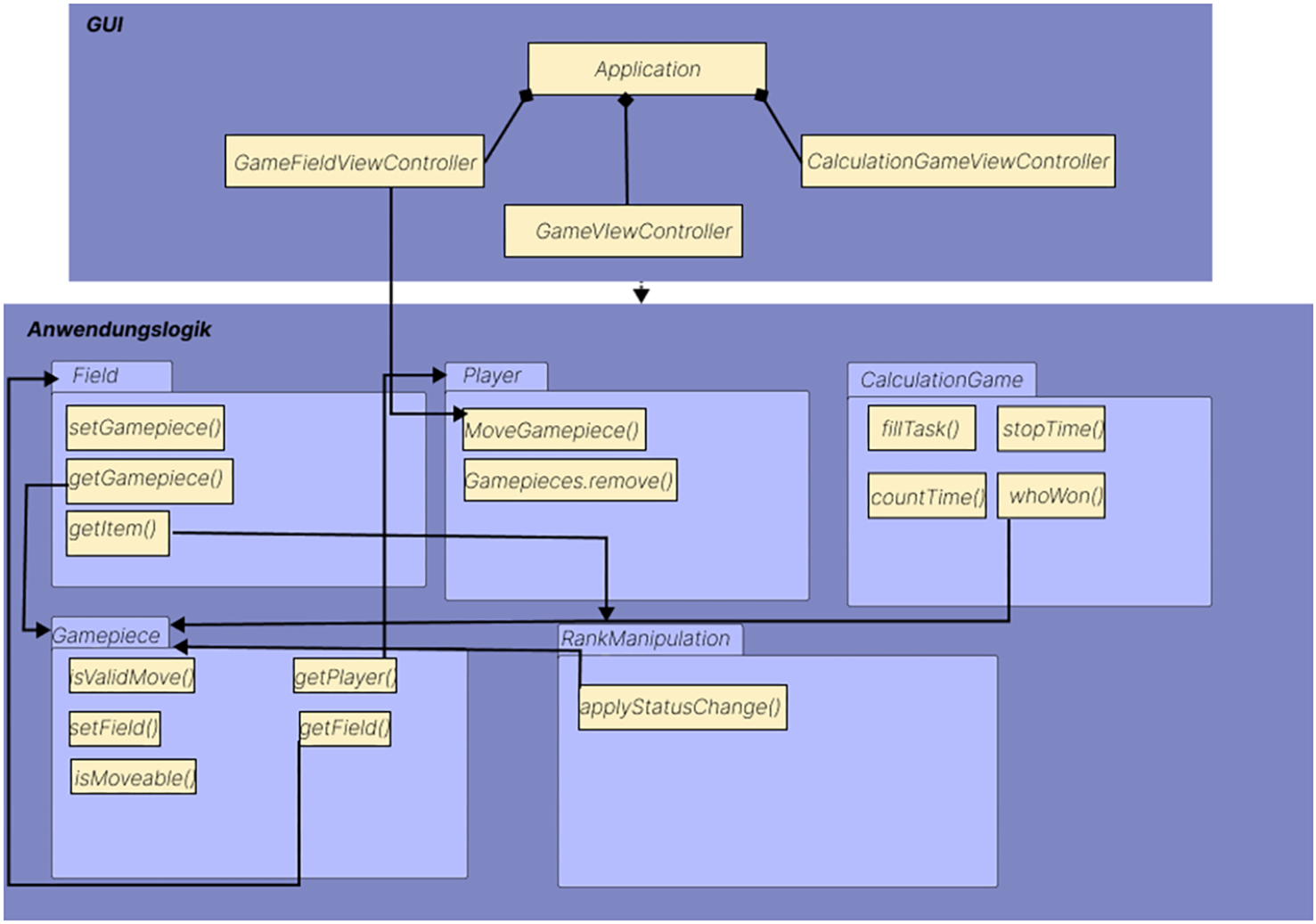
Julian Andrle, David Rossel, Sadia Miah

Juni 2023

Spezifikationen

Verweisung hierbei auf unsere definierte Spezifikationen in der Version 1.0 für alles Vorhergehende.

Bausteinsicht



Beschreibung

Wir haben im GUI-Paket die Klasse Applikation, die in Verbindung mit den Klassen GameFieldViewController, GameViewController und CalculationView-Controller steht.

Dann haben wir eine Anwendungslogik mit der Klasse Player, die die Methoden Gamepieces.remove() und MoveGamepiece() besitzt. Der GameFieldController ruft MoveGamepiece() auf, um die Position der Spielfigur zu aktualisieren.

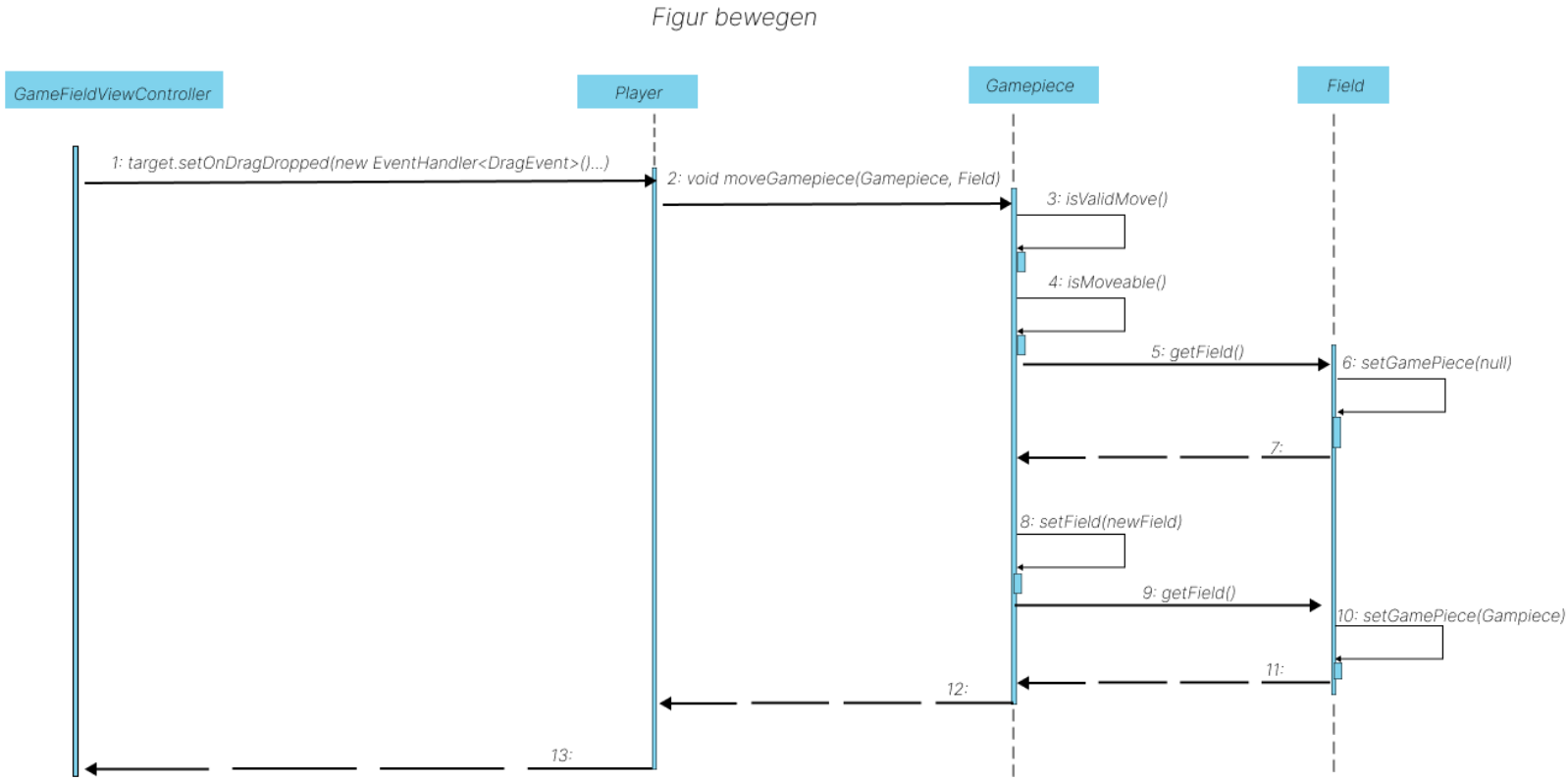
Es existiert die Klasse RangManipulation mit der Methode applyStatusChange, die die Klasse Gamepiece beeinflusst.

Die Klasse Gamepiece besitzt die Methoden isValid(), setField(), isMoveable(), getPlayer(), die den Spieler in der Klasse Player zurückgibt, und getField(), die das Feld in der Klasse Field zurückgibt.

CalculationGame besitzt die Methoden fillTasks(), stopTime(), countTime() und whoWon(), die auf Gamepiece zugreifen und den Gewinner zurückgeben.

Zuletzt haben wir die Klasse Field mit den Methoden setGamepiece(), die von Gamepiece die Figur zurückgibt, und getItem(), die uns RangManipulation zurückgibt.

Laufzeitsicht



Beschreibung

1.Der GameFieldViewController erkennt, dass eine der Spielfiguren bewegt wird und ruft den Handler des Objekts auf, um auf den Player zuzugreifen.

2.In der Player-Klasse wird nun die Methode "moveGamepiece" aufgerufen, um die Figur zu bewegen, wenn der Spielzug valide ist. Hierfür benötigt die Methode das Gamepiece und das neue Feld.

3.Zunächst wird geprüft, ob die Figur gemäß ihrer Eigenschaften diesen Zug machen darf.

4.Anschließend wird geprüft, ob die Figur selbst einem Statuseffekt einer Falle unterliegt und ob sie sich bewegen kann.

5.Wenn die Figur sich bewegen darf, wird das aktuelle Feld, auf dem sie steht, angesprochen.

6.Das in Punkt 5 angesprochene Feld (alte Position) wird gelöscht bzw. die alte Position der Spielfigur wird geändert. Dies muss nicht nur für die Spielfigur geändert werden, sondern auch in jedem einzelnen Field.

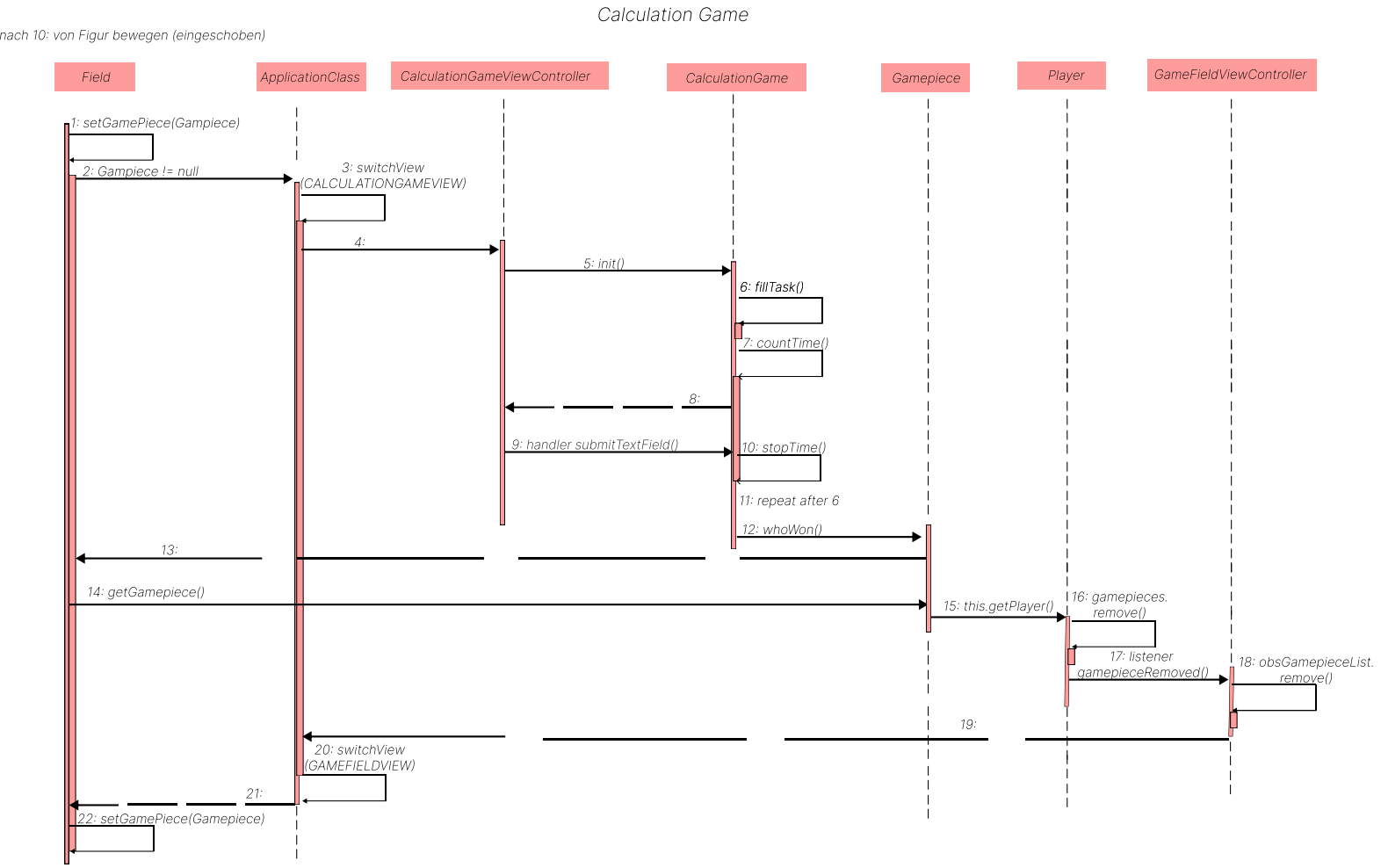
7.Rückgabe des Werts an gamepiece.

8.Das Feld der Spielfigur wird auf die neue Position im gamepiece selbst gesetzt.

9.Danach wird auf das Feld zugegriffen.

10.Und das neue Feld erhält das gamepiece.

11., 12. und 13. sind Rückgaben.



Beschreibung

In der Abbildung ist ein Sequenzdiagramm für das Ausführen vom CalculationGame dargestellt. Als Voraussetzung gilt, dass wir bereits die Sequenz beschrieben haben wie sich eine Figur bewegen lässt. Hier klinken wir im 10. Schritt ein.

(1:) setGamePiece wird in der Klasse Field aufgerufen und um einen Wettkampf aufzurufen muss die Voraussetzung erfüllt sein, dass das Feld bereits eine Spielfigur innehält.

(2:) Gamepiece != null falls in der setGamePiece Methode bereits eine Spielfigur erkannt wird wird hier zunächst die (3:) switchView Methode in der Application Class angestoßen um dort das Spiel darzustellen. Somit wird auch gleichzeitig eine neue (4:) CalculationGameView und deren Controller erstellt und der View hinzugefügt. In der (5:) init methode werden dann zwei CalculationGames

Instanzen initialisiert, die jeweils einen Spieler verwalten.

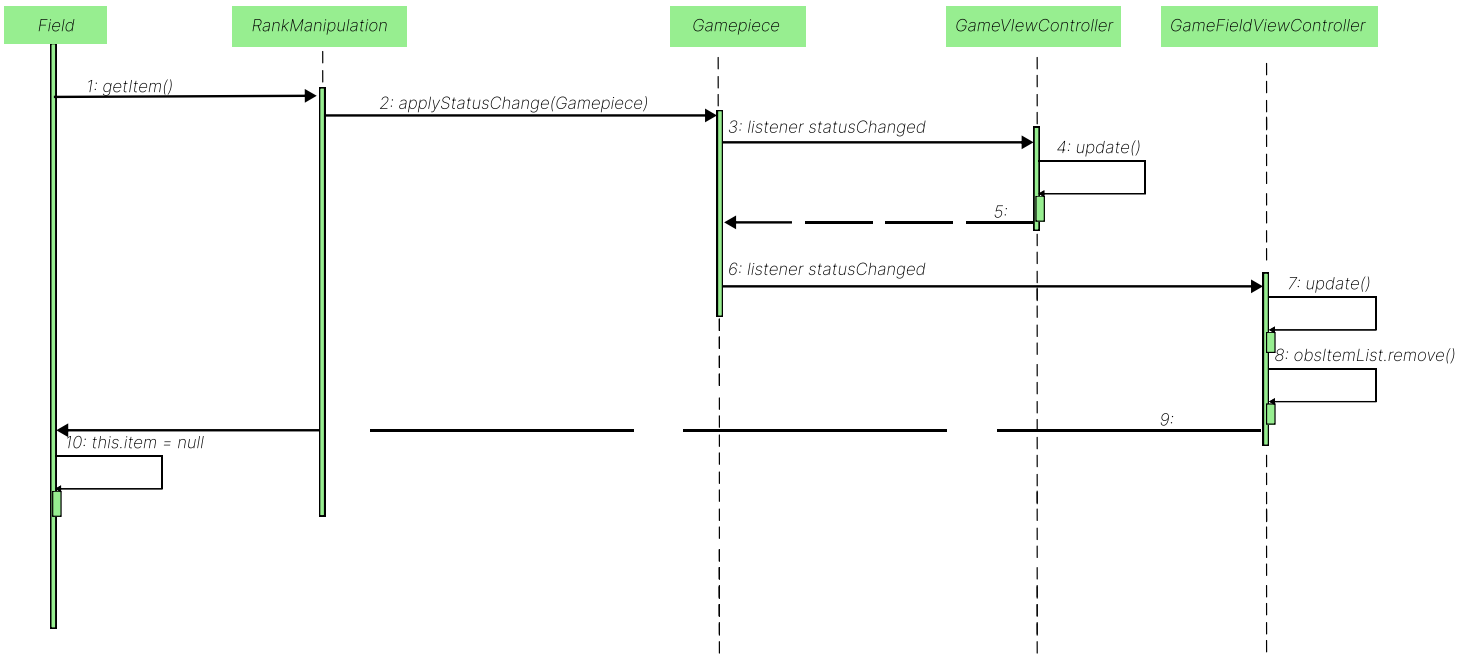
In jeder Instanz wird einmal durch folgenden Durchlauf iteriert: (6:) Mit der fillTask() Methode wird dann die Rechnung aus der HashMap gezogen und gleichzeitig die (7:) countTime() Methode nebenläufig angestoßen, die die verstrichene Zeit zählt. (8:) Jetzt wird das ganze anhand von Properties wieder zurück an den Controller gegeben und dargestellt. Der Controller gibt dann nach den Klick auf den Submit Button die Eingabe an die CalculationGame Instanz weiter und es wird hier nun die Richtigkeit der Aufgabe bestätigten und die Zeit in der (10:) stopsTime() Methode gestoppt. Nachdem beide Instanzen dann mit den Werten gefüllt sind wird anschließend mit der (12:) whoWon() Methode ermittelt wer gewonnen hat und jenes GamePiece an die (13:) Field Methode returned.

In unserem Fall hat die Figur gewonnen, die in diesem Zug versucht hat auf das Field zu ziehen. Daher rufen wir jetzt in Field die (14:) getGamePiece() Methode auf um die bereits bestehende Figur zu bekommen. In der GamePiece Klasse wird dann mit (15:) getPlayer() auf den Player der Spielfigur zugegriffen und jenes Objekt dann aus der Liste von Gamepieces mit (16:) remove() entfernt.

Da wir im GameFieldView auch die Figur entfernen wollen wird durch einen Listener auf den Controller zugegriffen und das Gamepiece auch aus der (18:) ObservableList entfernt. Letztendlich wird dann wieder in der Application Class die View zurückgeschalten (20:) und dann das neue GamePiece auf das Field gesetzt.

nach Figur bewegen Sequenz

Item anwenden: Rank Manipulation



Beschreibung

1. Das Field holt sich ein Item namens RankManipulation.
2. RankManipulation manipuliert den Rang der Spielfigur, die gegeben ist.
3. Der Listener "statusChanged" wird im GameViewController verwendet.
4. Der GameViewController aktualisiert sich.
5. Das Queen Icon wird als Tower Icon zurückgegeben.
6. Der Listener "statusChanged" wird aufgerufen.
7. Der GameFieldViewController aktualisiert sich.
8. Die observableList wird aktualisiert, das Item ist nicht mehr auf dem Feld vorhanden.
9. Die Methode kehrt zurück.
10. Das Item wird auf null gesetzt und ist somit verschwunden.