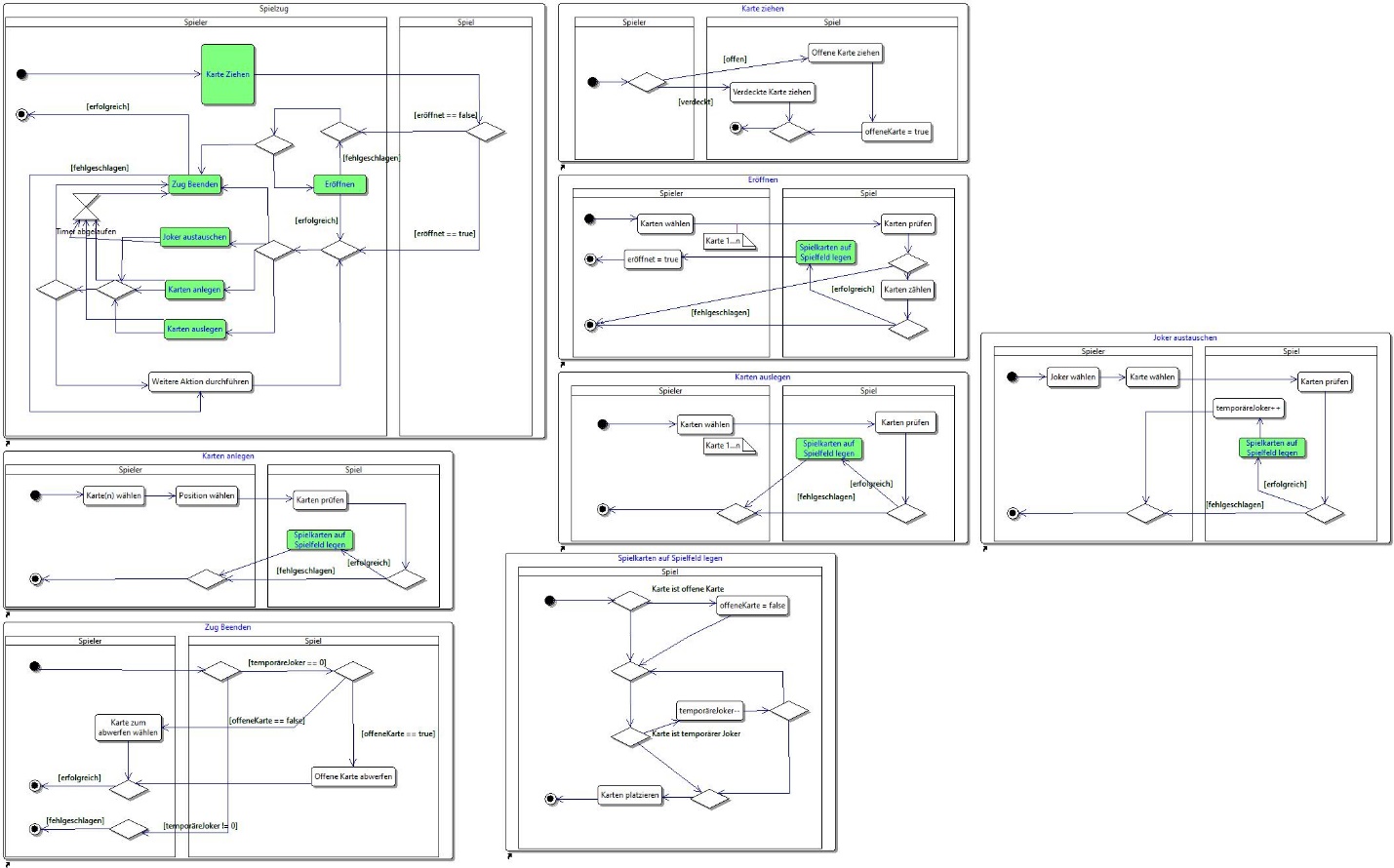
SWE-Labor Analyse Teil 2

# Activity Diagramm Spielzug



(repository <https://github.com/wetzu/swe-labor>)

# Systemoperationen im Use Case „Spielzug durchführen“

|  |  |
| --- | --- |
| **Name** | (Karte) karteZiehen(boolean verdeckt) |
| **Verantwortlichkeit** | Möchte ein Spieler vom (verdeckten) Restkartenstapel ziehen, muss geprüft werden welche Karten noch nicht “im Spiel” sind und eine zufällige gewählt werden  Möchte ein Spieler vom (offenen) Ablagestapel ziehen muss geprüft werden, ob er die Erlaubnis dazu hat. Ist dies der Fall muss die darunterliegende Karte wieder für alle Spieler dargestellt werden  In beiden Fällen wird dem Spieler die Karte auf die Hand gegeben |
| **Referenzen** | Usecase: Karte Ziehen Funktionen: F10 Eigenschaften: E2, E3, E4 |
| **Bemerkungen** | Karten vom Ablagestapel dürfen theoretisch nicht auf die Hand genommen werden, sondern direkt benutzt, hierzu wäre jedoch ein weiterer “Kartenraum” zum Zwischenspeichern nötig. Stattdessen kann man den Spieler die Karte auf die Hand nehmen lassen und zwingen wieder abzulegen, sofern er keinen Nutzen für die Karte findet  Zwischenzeitig kann es vorkommen, dass der Ablagestapel leer ist (Karte wurde aufgenommen, jedoch noch nicht wieder abgelegt) |
| **Ausnahmen** | Ist der Restkartenstapel leer wird das Spiel beendet |
| **Vorbedingungen** | Spieler hat noch keine Karte gezogen |
| **Nachbedingungen** | Die Anzahl der Karten auf einem der Stapel hat sich um 1 verringert |

|  |  |
| --- | --- |
| **Name** | (boolean) kartenPrüfen(Karte[] cards) |
| **Verantwortlichkeit** | Überprüft, ob der Spieler beim Kartenauslegen sich an Spielregeln gehalten hält. |
| **Referenzen** | Usecase: Spielzug  Funktionen: F6, F7,  Eigenschaften: E1, E2, E3, E4, E5 |
| **Bemerkungen** | maximal vier Karten pro Satz möglich sind. |
| **Ausnahmen** |  |
| **Vorbedingungen** | Beim Auslegen darf der Spieler nur wie folgt auslegen:  Reihen: bestehen aus drei oder mehr aufeinanderfolgenden Karten einer Farbe, also mit identischem Symbol (Kreuz, Pik, Herz oder Karo)  Sätze: Sätze können aus allen Kartentypen gebildet werden, und bestehen aus mindestens drei gleichrangigen Karten |
| **Nachbedingungen** |  |

|  |  |
| --- | --- |
| **Name** | (boolean) prüfenObSpielerEröffnet(spieler Spieler) |
| **Verantwortlichkeit** | Prüft ob angegebener Spieler bereits eröffnet hat |
| **Referenzen** | Use Case: Eröffnen  Funktionen: F4 |
| **Ausnahmen** |  |
| **Bemerkungen** |  |
| **Vorbedingungen** |  |
| **Nachbedingungen** |  |

|  |  |
| --- | --- |
| **Name** | (void) Karten platzieren(Karte[] cards) |
| **Verantwortlichkeit** | Ein Spieler platziert Karten auf dem Spielfeld.  Die Karten werden von seiner Hand entfernt und dem Spielfeld hinzugefügt. |
| **Referenzen** | Usecase: Karten Auslegen, Karten Anlegen,  Funktionen: F7  Eigenschaften: E1, E4 |
| **Bemerkungen** |  |
| **Ausnahmen** |  |
| **Vorbedingungen** |  |
| **Nachbedingungen** | Danach kann das Spiel vorbei sein, wenn die Hand leer ist. |

|  |  |
| --- | --- |
| **Name** | (Event) TimerAbgelaufen() |
| **Verantwortlichkeit** | Timer wird jedes Mal neu gestartet, wenn ein Spieler sein Zug anfängt. Falls der Spieler nicht fertig ist, bevor der Timer abläuft, wird ein Event geworfen und der Server darüber benachrichtigt. |
| **Referenzen** | Use Case: Eröffnen  Funktionen: F4 |
| **Ausnahmen** |  |
| **Bemerkungen** |  |
| **Vorbedingungen** |  |
| **Nachbedingungen** | Nach dem Event wird Zug vom Spieler beendet |

|  |  |
| --- | --- |
| **Name** | (boolean) kartenZaehlen(karten: Karte []) |
| **Verantwortlichkeit** | Zählt wie viel Augen haben die Karten. Wenn sie weniger als 40 Augen dann wird false zurückgegeben ansonsten true |
| **Referenzen** | Use Case: Eröffnen, Anlegen, Auslegen  Funktionen: F4, F15 |
| **Ausnahmen** | Die Ass wird in einer Folge mit einem König als 11 und ohne König als 1 gezählt |
| **Vorbedingungen** | Der Spieler muss 13 Karte in der Hand haben und die Karten müssen geprüft werden bevor sie gezählt werden |
| **Nachbedingungen** | Wenn wahr kann der Spieler anlegen, auslegen und Joker tauschen da er eröffnet hat. Wenn falsch wird er als nicht eröffnet markiert |

|  |  |
| --- | --- |
| **Name** | kartenAnlegen(karten: Karte[], position: Position) |
| **Verantwortlichkeit** | Eine Karte von Hand nehmen und zu einer passenden Position in einem Satz oder eine Folge hinzufügen. Position bezieht sich dabei auf Satz und Karte innerhalb des Satzes |
| **Referenzen** | Use Case: Anlegen  Funktionen: F6, F8, F11  Eigenschaften: E1, E2, E3, E4, E5 |
| **Bemerkungen** | Nachdem ein Spieler ausgelegt hat, ist es möglich im selben Spielzug an diese Sätze bzw. Folgen anzulegen? |
| **Ausnahmen** | Ein Joker darf nicht neben einen Joker in einer Folge oder angelegt werden  Das Ass kann in einer Folge hinter einem König und vor die Zwei angelegt werden |
| **Vorbedingungen** | Der Spieler muss bereits eröffnet haben |
| **Nachbedingungen** | Die Anzahl von Handkarten reduziert sich nach jedem Kartenanlegen um 1 |
|  |  |

## Class Diagram

