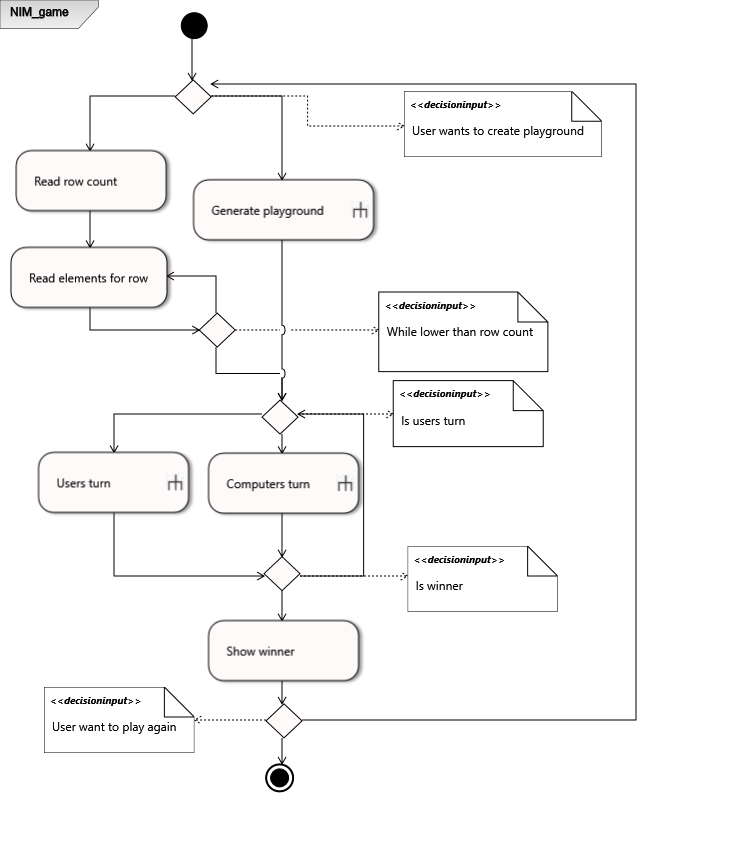
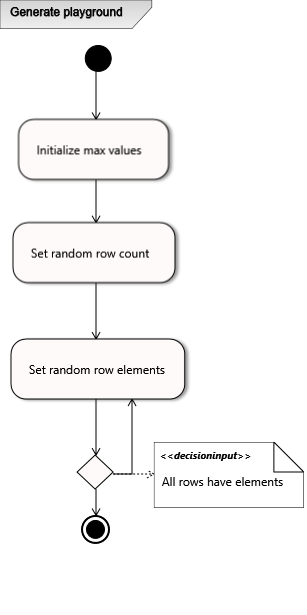
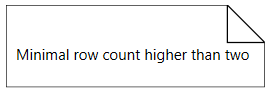
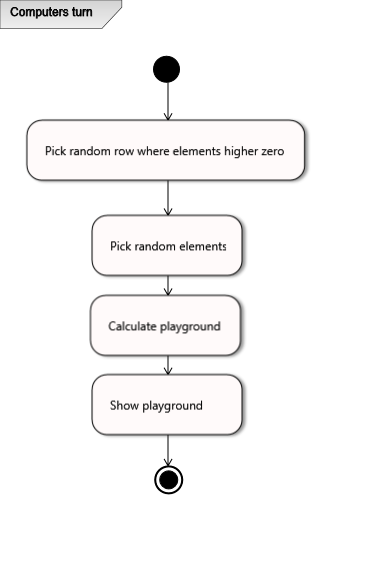
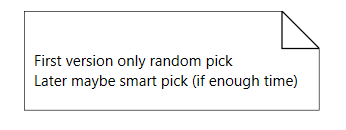
Analyse NIM-Spiel

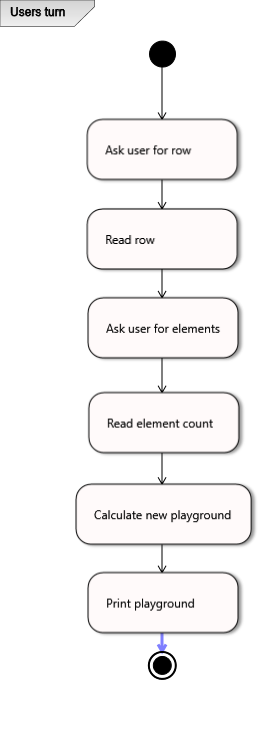
Das Dokument dient als Einstieg in das Projekt «NIM-Spiel» für das Fach BM\_P1. Es soll als Grundlage für die Programmierung dienen und die Implementation in ein C-Programm erleichtern.

# Aktivitätsdiagramm









# Implementationsreihenfolge

Die Implementation der Funktionen soll in etwa so ablaufen:

1. IsValidYesNoInput
2. GeneratePlayground
3. ManuallyCreatePlayground
4. UsersTurn
5. ComputersTurn
6. ShowWinner

# Datenspeicherung

Um den Stand des Spiels zu speichern, sollen folgende Daten zwischengespeichert werden:

* Wer ist am Zug? (Computer oder Benutzer)
* Wie viele Reihen sind vorhanden?
* Wie viele Elemente sind pro Reihe vorhanden?

Ebenso sollen folgende Konstante Werte gespeichert werden, um ein korrektes Spiel zu initialisieren:

* Minimale und maximale Anzahl Reihen
* Minimale und maximale Anzahl Elemente

# Test

Folgende Funktionen sollen mit Blackbox-Tests getestet werden:

|  |  |
| --- | --- |
| IsValidYesNoInput | Es wird kontrolliert, ob der Input «y» oder «n» ist.  Dies kann mit Hilfe der Anzeige kontrolliert werden (bei Falscheingabe soll eine Fehlermeldung erscheinen). |
| GeneratePlayground | Es wird kontrolliert, ob ein gültiges Spiel initialisiert wird.  Dies kann mit Hilfe der Anzeige kontrolliert werden. |
| ManuallyCreatePlayground | Es wird kontrolliert, ob nach dem manuellen Eingeben von einem Spiel ein korrekter Spielaufbau gemacht wird.  Dies kann mit Hilfe der Anzeige kontrolliert werden. |
| UsersTurn | Wird die korrekte Anzahl Elemente aus der angegebenen Reihe entfernt?  Dies kann mit der Anzeige kontrolliert werden. |
| ComputersTurn | Wird eine korrekter Zug vom Computer gemacht?  Dies kann mit der Anzeige kontrolliert werden. |
| ShowWinner | Es soll geprüft werden, ob auch der korrekte Gewinner gefunden und ausgegeben wird.  Dies kann mit der Anzeige kontrolliert werden. |