2.2 Problemanalyse und Realisation

Die folgende Analyse und Beschreibung bezieht sich auf das Spiel TicTacToe, welches als JavaFX-Anwendung umgesetzt werden soll. Für eine abschließende Diskussion des Spiels sind weit mehr Unterpunkte notwendig. Auch stellt dieses Beispiel nicht die perfekte Lösung dar, sondern lediglich eine valide Lösung.

2.2.1 GUI-Darstellung des Spielfelds

Problemanalyse

Das Spielfeld bei TicTacToe besteht aus einer $3 \times 3-$Tabelle, wobei in jeder Zelle eines der Spieler-Symbole platziert werden kann. Ist ein Spieler am Zug, so kennzeichnet er seine Zelle, indem er sie anklickt. Die darzustellenden Symbole sind X und O (bzw. ein Kreis). Bei Größenveränderung des Fensters soll sich das Spielfeld automatisch anpassen.

Realisationsanalyse

Für das Spielfeld eignet sich ein **GridPane** hervorragend. Hier können in Zeilen/Spalten weitere JavaFX-Komponenten angeordnet werden, welche die Zellen repräsentieren.
Für die Zellen bieten sich verschiedene Komponenten an:

1. **Labels**
   Die darzustellenden Symbole sind unkompliziert als Buchstaben darstellbar. Allerdings lässt sich die automatische Größenanpassung schwer realisieren. Außerdem ist bei einem Label nicht sofort ersichtlich, dass der Benutzer damit interagieren (= das Label anklicken) muss, um sein Symbol zu platzieren.
2. **Buttons**
   Genau wie beim Label können die Symbole als Buchstaben dargestellt werden. Ein weiterer Vorteil ist, dass die Zellen dadurch für den Benutzer sofort als „klickbar“ erkennbar sind. Auch die automatische Größenanpassung ist gut realisierbar (wenn auch nicht optimal).
3. **ImageViews**
   Mithilfe von ImageViews können vorgefertigte Bilder dargestellt werden. Optisch lässt sich mit dieser Lösung das beste Resultat erzielen. Durch ein geschickt gewähltes Bild zur Repräsentation einer leeren Zelle, kann dem Nutzer vermittelt werden, dass die entsprechende Zelle „anklickbar“ ist. Positiv zu bemerken ist außerdem die leichte Austauschbarkeit der Bilder für ein anderes Design. Nachteil ist jedoch, dass für jedes Symbol ein eigenes Bild kreiert werden und als Datei dem Projekt beigefügt werden muss.

Realisationsbeschreibung

Das Spielfeld wird durch ein GridPane mit je drei Zeilen/Spalten repräsentiert. In jeder Zelle des GridPanes befindet sich ein Button, welcher immer den maximal zur Verfügung stehenden Platz einnimmt. Wird ein Button geklickt, werden über eine EventHandler-Methode Vorgänge in der Logik angestoßen und schlussendlich das entsprechende Symbol des Spielers auf dem Button in Textform dargestellt. Wenn die Größe des Fensters verändert wird, verändert sich nur die Button-Größe, nicht jedoch die Größe der Schrift auf dem Button.
Da das Design für die Aufgabenerfüllung nicht besonders ansprechend oder variabel gestaltet werden musste, wurde an dieser Stelle auf den Mehraufwand verzichtet, welcher für die Umsetzung mit ImageViews nötig gewesen wäre.