

Erläuterung des Projektes Ziffernspiel

Die grundlegendste Ebene des Programms, welches meine Umsetzung des “Ziffernspiels” ist, stellt die Klasse Spielfeld dar. Diese Klasse implementiert alle Funktionalitäten, die notwendig sind, damit der Computer das Spiel speichern und verändern kann. Außerdem implementiert die Klasse Funktionen, mit denen einfach Kerninformation über den Status des Spiels herausgezogen werden können.

Die einzelnen, verschiebbaren Spielplättchen sind in der zweidimensionalen Zahlenreihung `game_map` mit den Maßen 5x5 gespeichert. Dabei gibt die Zahl “0” in an, dass das Feld frei ist, und die Zahlen von Eins bis Fünf entsprechen dem jeweiligen Plättchenwert. Da ich der Meinung bin, dass das zu erzeugende Muster ein Teil der Spiellogik ist, sowie weil beides sehr eng verwoben ist, habe ich mich dadurch entschieden, dessen Implementierung in derselben Klasse umzusetzen. Das Muster wird in der eindimensionalen Zahlenreihung `goal_pattern` gespeichert.

Der Konstruktor der Klasse belegt beide Reihungen mit korrekten Werten. Um mit dem Spiel auf der Programmierenebene zu interagieren, wurden verschiedenste Methoden implementiert. So kann beispielsweise die Funktion `getCellFromCoordinate(): int[]` die x- und y-Koordinate des Feldes zurückgeben, auf dem sich gerade die Maus befindet. Die Methode `checkGameWon()` gibt einen Wahrheitswert zurück, der angibt, ob die Plättchen so verschoben wurden, dass das Muster gelöst wurde. Neben verschiedensten Getter-Funktionen gibt es auch einige Setter-Funktionen. So kann man mit der Methode `set_clicked_cell(val: int[])` eine Reihung übergeben, die als das aktuell angeklickte Feld gespeichert wird. Anschließend könnte man mit `moveCellInGamemap()` das angeklickte Feld auf das Feld verschieben, über dem sich die Maus im jeweiligen Moment befindet.

Die Klasse `UIOfGameLogic` basiert stark auf den Methoden der Klasse Spielfeld. Diese implementiert nämlich Methoden zur grafischen Darstellung dieser Klasse. So wurden Methoden zum Anzeigen des zu lösenden Musters, der Zellen der `game_map` des Spielfeldes, des angeklickten und in Bewegung befindlichen Feldes, sowie zwei weiteren Animationen in dieser Klasse implementiert.

Die Klasse `Button` ermöglicht ein einfaches Erstellen von mit der Maus anklickbaren Buttons auf der grafischen Oberfläche. Dafür muss man bei der Initialisierung der Klasse Variablen an den Konstruktor übergeben, die den angezeigten Schriftzug, die Position, die Ausdehnung und die Stärke der Abrundung der Ecken angeben.

Die Klasse `Ziffernspiel` enthält sowohl ein Objekt der Klasse `Button`, sowie `UIOfGameLogic`. Mit Hilfe der Methode “`updateGame(): void`”, der Setup-Funktion und der fortlaufend ausgeführt werdenden Funktion `draw()` werden die in den jeweilig implementierten Klassen implementierten Methoden so aufgerufen, dass das Spiel spielbar wird.

Dabei wird vor allem in der Draw-Methode unterschieden, ob das Spielfeld über die Methode `updateGame()` sich aktualisieren soll und somit responsiv sein soll, oder ob das Spiel bereits vom Spieler gelöst wurde, was zur Folge hat, dass die Animation für den Erfolg abgespielt wird. Außerdem wird in dieser Klasse ein Objekt der Klasse `Button` so implementiert, dass sie einen Button erzeugt, der wiederum das Spiel zurücksetzt.