

# Studienleistung 2

Zuletzt bearbeitet von Jürgen Hahn

## Studienleistung-02

### Wichtige Informationen zur Bearbeitung der Aufgabe

- Informationen zur Entwicklungsumgebung *IntelliJ IDEA*
- Informationen zum Im- und Export von Projekten
- GraphicsApp

### Starterpaket

Ein vorbereitetes Starterpaket zur selbständigen Implementierung der Aufgabe finden Sie hier.

### MemoryApp

Im Rahmen dieser Aufgabe implementieren Sie das bekannte Spiel Memory.

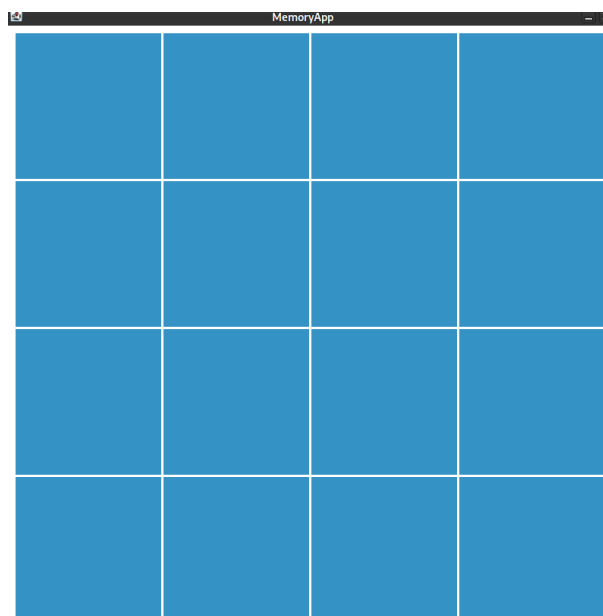
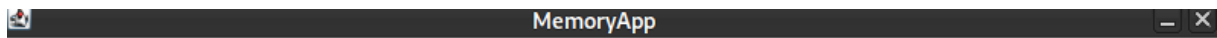


Abbildung 1: 4x4 Memory

Wie auf dem obigen Bild dargestellt, beinhaltet das Memoryspiel dieser Aufgabe 16 Karten, die auf vier Reihen a vier Karten aufgeteilt werden. Zu Beginn jedes Spiels die Karten per

Zufall verteilt. Per Mausklick wird eine Karte aufgedeckt. Sind zwei Karten aufgedeckt, soll überprüft werden, ob deren Bilder zusammenpassen. Passen sie zusammen, verschwinden die Karten. Ansonsten bleiben sie so lange aufgedeckt bis sie durch erneutes Klicken wieder umgedreht werden. Am Ende des Spiels soll angezeigt werden, wie oft zwei Karten verglichen wurden.



You completed Memory with 14 comparisons!

```
{ width=50% }
```

Alle benötigten Bilder befinden sich im Ordner `data/assets`, der im Starterpaket enthalten ist.

Folgende Anforderungen muss das Memoryspiel für diese Aufgabe erfüllen:

- \* Die Klasse `MemoryApp` muss als Einstiegspunkt für Ihr Memoryspiel verwendet werden
- \* Teilen Sie Ihre Anwendung in sinnvolle Komponenten ein und legen Sie entsprechende Klassen dafür an
- \* Trennen Sie die Daten von Objekten (z.B. Karten) von deren Darstellung
- \* Verwenden Sie die Klasse `java.io.File` anhand von `File[] files = new File(PATH_TO_ASSETS).listFiles()`, um ein Array zu erhalten, dass alle darin befindlichen Dateien enthält
- \* Verwenden Sie die Methode `getAbsolutePath()`, um den absoluten Pfad einer Datei zu bekommen

als `java.io.File` Objekt vorliegt \* Verwenden Sie `java.util.Random` und `Random.nextInt()`, um beim Spielstart die Karten zu \*mischen\*, also die Bilder auf zufällige Art und Weise den Karten zuzuteilen \* Um Mausklicks zu verarbeiten, benutzen Sie `ur.mi.oop.events.GraphicsAppMouseListener` und überschreiben sie die Methode `onMousePressed` \* Um zu überprüfen, ob eine Karte angeklickt wurde, verwenden Sie die Methode `hitTest(int x, int y)`, des angeklickten `GraphicObject`, z.B. (`Rectangle`) \* Verwenden Sie sinnvolle Datenstrukturen (z.B. `ArrayList`) \* Praktizieren Sie `Decomposition`