Die Hausaufgaben müssen von jedem Studierenden einzeln bearbeitet und abgegeben werden. Für die Hausaufgabe sind die aktuellen Informationen vom Blog `https://seblog.cs.uni-kassel.de/ws2021/programming-and-modelling/` zu berücksichtigen.

Abgabefrist ist der 17.12.2020 - 23:59 Uhr

Abgabe

Wir benutzen für die Abgabe der Hausaufgaben Git. Jedes Repository ist nur für den Studierenden selbst sowie für die Betreuer und Korrektoren sichtbar.

Für die Hausaufgabe benötigt ihr **das bestehende MiniRPG-Repository**. Falls noch nicht geschehen, kann es über folgenden Link angelegt werden:

`https://classroom.github.com/a/5LkDnrR7`

Nicht oder zu spät gepushte (Teil-)Abgaben werden mit 0 Punkten bewertet.

Abgaben, die nicht lauffähig sind, werden mit 0 Punkten bewertet!

Commit Message-Vorgaben

In dieser und zukünftigen Hausaufgaben führen wir eine Vorgabe für die Commit-Messages ein.

Solche Commits, welche als Abgabe einer Hausaufgabe dienen, werden mit der Nachricht: ”Final HA<Number>” versehen. Für die Abgabe von Teilaufgaben (wie in Aufgabe 1 dieses Blattes), wird folgende Konvention eingeführt ”HA<Number> finished Task <TaskNumber>”.

Falls in Folge der Commits ein Fehler auffällt, welchen ihr zeitlich gesehen noch korrigieren könnt, so verwendet für den Folge-Commit wieder die Message ”Final HA<Number>” oder ”HA<Number> finished Task <TaskNumber>”.

Das Ignorieren der Vorgaben wird mit 0 Punkten bewertet!

Projekte deren GUI nicht mit FXML-Dateien umgesetzt sind, werden mit 0 Punkten bewertet!

Vorbereitungen

Anpassungen der build.gradle

In dieser Hausaufgabe wird das Oberflächenframework JavaFx, sowie die Testfx-Library verwendet. Hierzu soll die bestehende build.gradle erweitert werden. Die hinzuzufügenden Zeilen sind auf unserem Blog aus dem .zip-Archiv zu entnehmen. In diesem Archiv befinden sich noch weitere Dateien, welche für diese Hausaufgabe genutzt werden, mehr hierzu in den entsprechenden Aufgaben.

Anschließend sollte ein Gradle Refresh durchgeführt werden.

Aufgabe 1 - Fxml (25P)

In dieser Aufgabe soll die Grundlage zur Umsetzung der in der vorherigen Hausaufgabe erstellten Mockups geschaffen werden.

Wir verwenden für die Erstellung unserer Oberflächen den SceneBuilder. Lade dir den SceneBuilder unter folgendem Link herunter:

`https://gluonhq.com/products/scene-builder/`

Der Scenebuilder speichert seine Dateien im fxml-Format. Für jedes Mockup muss eine eigene fxml-Datei erstellt werden. Die zwei erstellten fxml-Dateien sind im Modul src/main/resources im Package de.uniks.pmws2021 abzulegen. Beim Erstellen der Package-Ordner treten je nach IDE Probleme auf. Achte darauf, dass im resources-Ordner die korrekt geschachtelte Ordnerstruktur erstellt wurde und nicht nur ein Ordner mit dem Namen "de.uniks.pmws2021" existiert.

Für diejenigen, die ihre eigenen Mockups nicht umsetzen möchten, stellen wir folgende Mockups als Arbeitsgrundlage zur Verfügung:

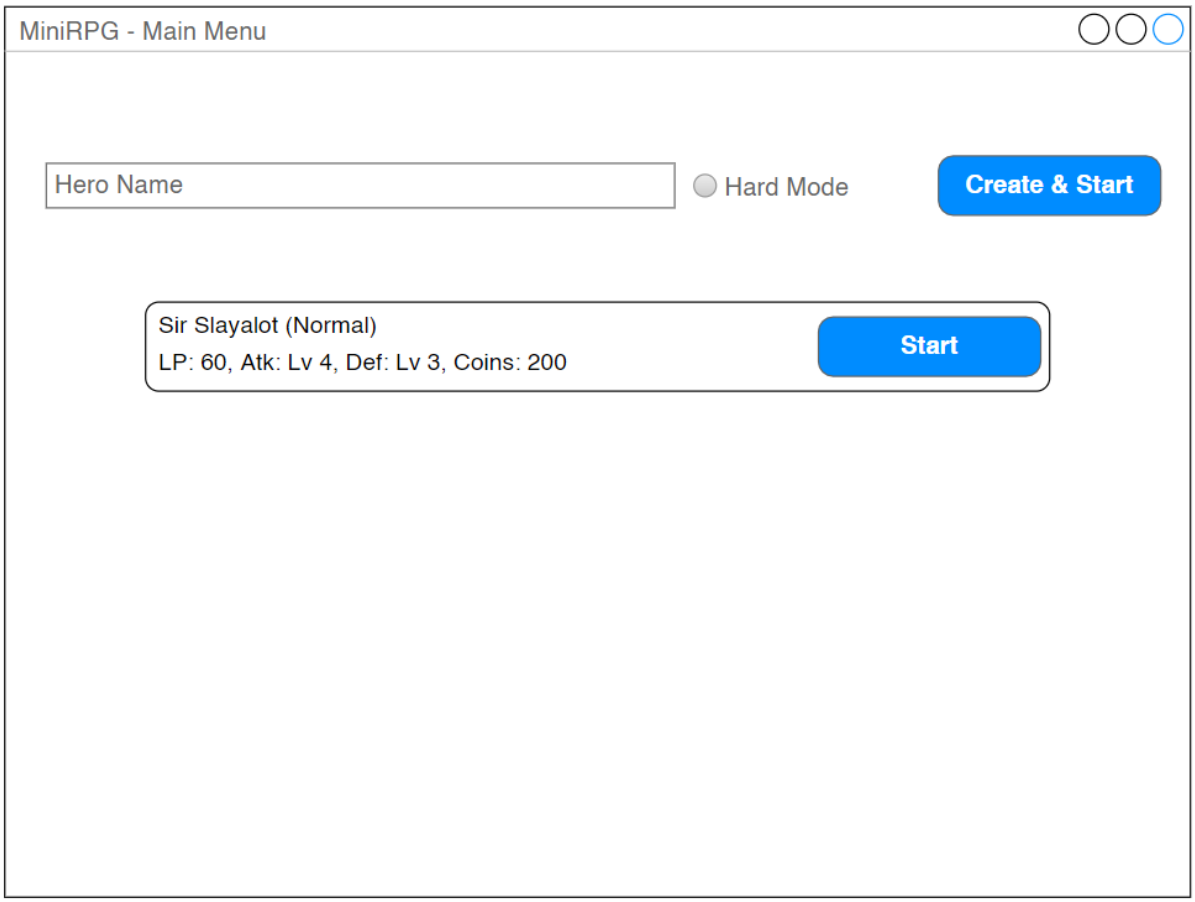

Abbildung 1: Hero Screen

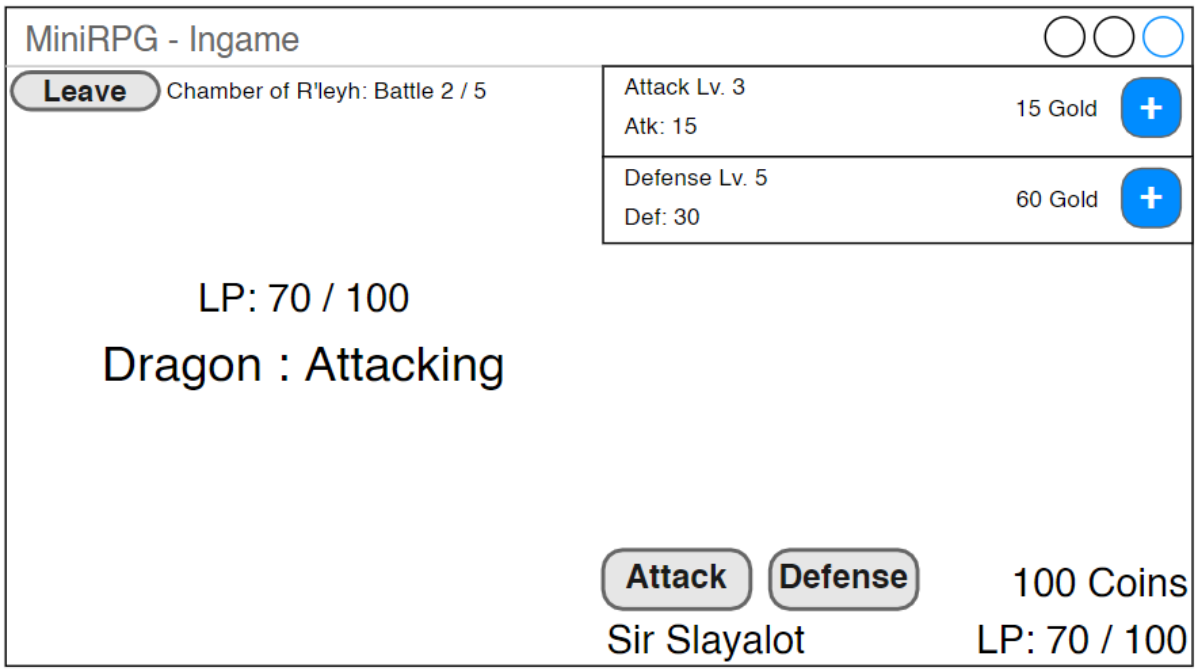

Abbildung 2: Dungeon Screen

Beachte, dass nun ein **Leave-Button** im Dungeon Screen hinzugekommen ist. Dieser soll auch in eigenen Entwürfen und den dazugehörigen Mockups vorhanden sein. Eventuell ist also eine Anpassung der Lösung der letzten Hausaufgabe vonnöten.

Committe und pushe die neuen Dateien abschließend auf den main-Branch.

Achte darauf, das Repository der aktuellen Hausaufgabe zu verwenden.

Aufgabe 2 - JavaFx (30P)

Aufgabe 2.1 Vorgegebene Klassen kopieren

Um einen einheitlichen Startpunkt zur Implementierung einer JavaFx-Applikation zu gewährleisten, findet ihr in dem in der Vorbereitung genannten Archiv vier Java-Klassen.

```
de.uniks.pmws2021.Launcher
de.uniks.pmws2021.StageManager
de.uniks.pmws2021.controller.HeroScreenController
de.uniks.pmws2021.controller.DungeonScreenController
```

Füge diese in der vorgeschriebenen Ordnerstruktur in dein Projekt ein.

Aufgabe 2.2 Funktionalität

In dieser Aufgabe soll eine Verbindung zwischen fxml-Dateien und Controller geschaffen werden. Implementiere hierzu die Methoden der aus der vorherigen Aufgabe kopierten Klassen. Durch Kommentare in den Methoden wird deren Funktionalität bereits vorgegeben. Nach Bearbeitung der Aufgabe sollte es möglich sein, durch das Klicken des CreateStart-Buttons vom Hero Screen in den Dungeon Screen zu gelangen. Ebenso sollte dies umgekehrt durch den Leave-Button möglich sein.

Weiterhin sollen an dieser Stelle die **Fenstertitel** umgesetzt werden, die in den Mockups der vorherigen Aufgabe zu sehen sind (MiniRPG - Main Menu und MiniRPG - Ingame). Wie zuvor sollen eigene Mockups falls nötig angepasst werden.

Committe und pushe die Änderung abschließend auf den main-Branch.

Bei der Bewertung wird vor allem darauf geachtet, dass die JavaFx-Applikation samt Scenewechsel funktoniert.

Achte darauf, das Repository der aktuellen Hausaufgabe zu verwenden.

Aufgabe 3 - TestFx (30P)

Als Abschluss für diese Hausaufgabe soll die Gesamtheit der implementierten Teile der vergangenen Teilaufgaben getestet werden. Dies geschieht wie in der Vorlesung gezeigt mithilfe der bereits hinzugefügten TestFx-Library.

Aufgabe 3.1 Vorgegebene Klassen kopieren

Um einen einheitlichen Startpunkt zur Implementierung eines TestFx-Testes zu gewährleisten, findet ihr in dem in der Vorbereitung genannten Archiv eine Java-Test-Klasse:

`de.uniks.pmws2021.StageManagerTest`

Füge diese in der vorgeschriebenen Ordnerstruktur in dein Projekt ein.

Aufgabe 3.2 Implementierung

Die changeViewTest-Methode soll folgenden Ablauf implementieren:

- Initialen Fenstertitel prüfen.
- Hero-Name aus FXML in das dafür vorgesehene Eingabefeld eingeben. Bei Verwendung der vorgegebenen Mockups wäre dies Sir Slayalot, bei eigenen Mockups der dort verwendete Name.
- Create&Start-Button klicken, um ein neues Spiel mit dem neuen Hero zu starten.
- Neuen Fenstertitel prüfen.
- Hero-Name aus FXML (s.o.) im entsprechenden Label prüfen.
- Leave-Button klicken.
- Fenstertitel erneut prüfen.
- Prüfen, dass das Eingabefeld für den Hero-Name leer ist.

Committe und pushe die Änderung abschließend auf den main-Branch.

Achte darauf, das Repository der aktuellen Hausaufgabe zu verwenden.

Anhang

Es folgt eine Auflistung hilfreicher Webseiten und weiterer Erklärungen zu den Themen dieser Hausaufgabe. Die Links sind als Startpunkt zur selbstständigen Recherche angedacht. Das Durcharbeiten der folgenden Quellen ist kein bewerteter Anteil der Hausaufgaben.

JavaFx

- Erklärung zur Launcher-Klasse:
  `https://stackoverflow.com/a/52631238/10526222`
- JavaFx und CSS:
  `https://guigarage.com/2016/02/javafx-and-css/`

TestFx

- TestFx auf GitHub:
  `https://github.com/testfx/testfx`