SOFTWAREENTWICKLUNG 2

Sommersemester 2021/2022

DRINK OR DARE

Kiara Krug kk167 - Mobile Medien

Gabriela Gerhardt gg037 - Mobile Medien

https://gitlab.mi.hdm-stuttgart.de/kk167/drinkordare

1. Kurzbeschreibung

Unser Projekt ist ein einfaches Trinkspiel für das zweite Semester des Studiengangs Mobile Medien an der Hochschule der Medien.

Anfangs wählt man zwischen zwei Schwierigkeitsgraden: Soft und Hard.
Als nächstes gibt man alle Spielernamen an, die in dieser Runde mitspielen.
Danach muss man nur noch auf den Start Button drücken und man kann mit dem Trinkspiel anfangen.

Es gibt bei den 2 Schwierigkeitsgraden jeweils Wahrheitsfragen und Pflichtaufgaben.
Auf der Karte erscheint der Spielername der dran ist und eine zufällige Frage oder Aufgabe.
Es können zwischen zwei und 6 Personen spielen. Mit dem „zurück“ Button gelangt man zu der vorherigen Seite um seine Eingabe zu verbessern, falls ein Fehler unterlaufen ist. Ziel des Spiels ist es, Spaß mit seinen Mitspielern zu haben, die Spieler besser kennenzulernen und natürlich Alkohol zu trinken.

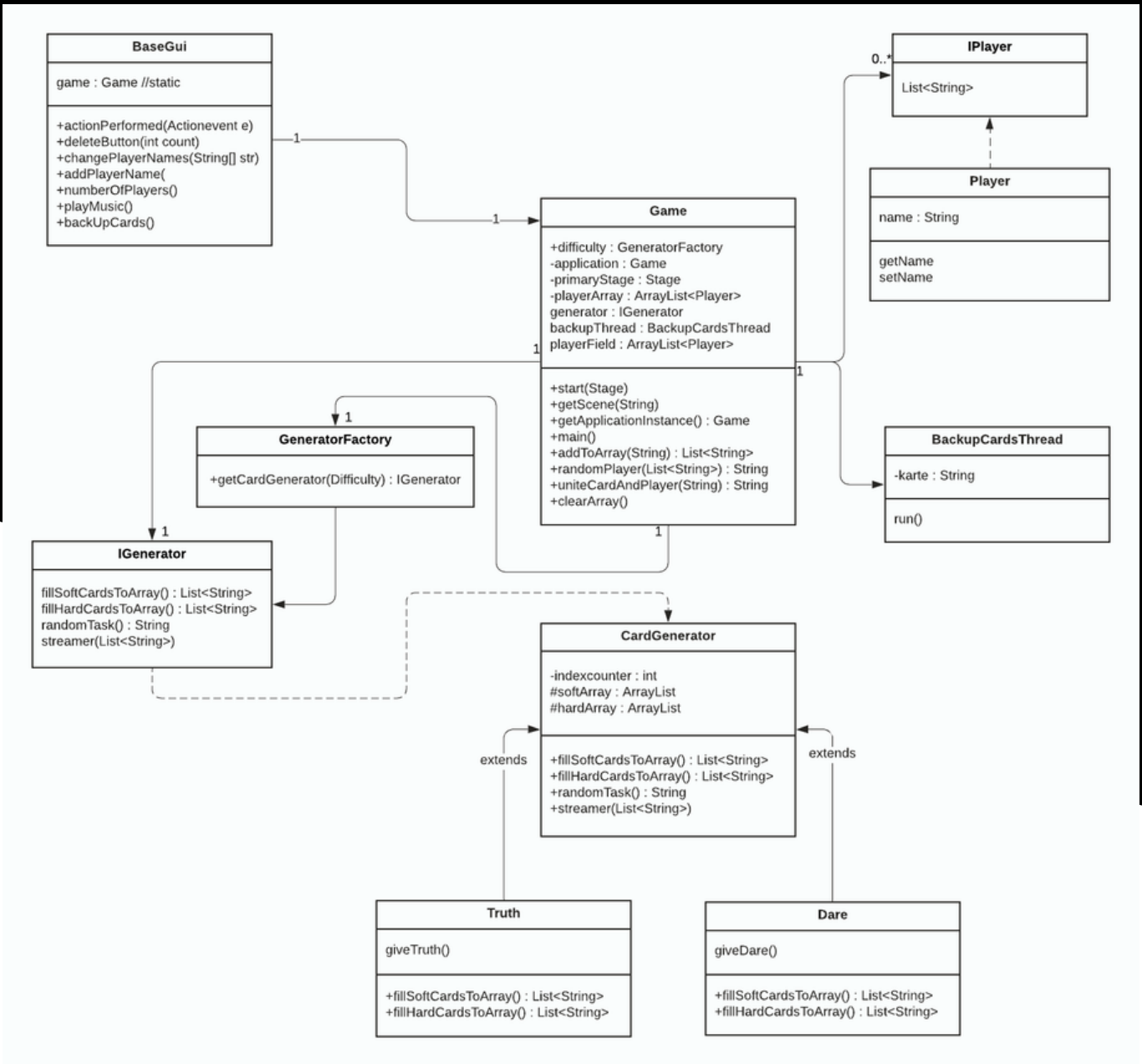
2. Startklasse

Die Main-Methode befindet sich in der Klasse Game.java

3. Besonderheiten

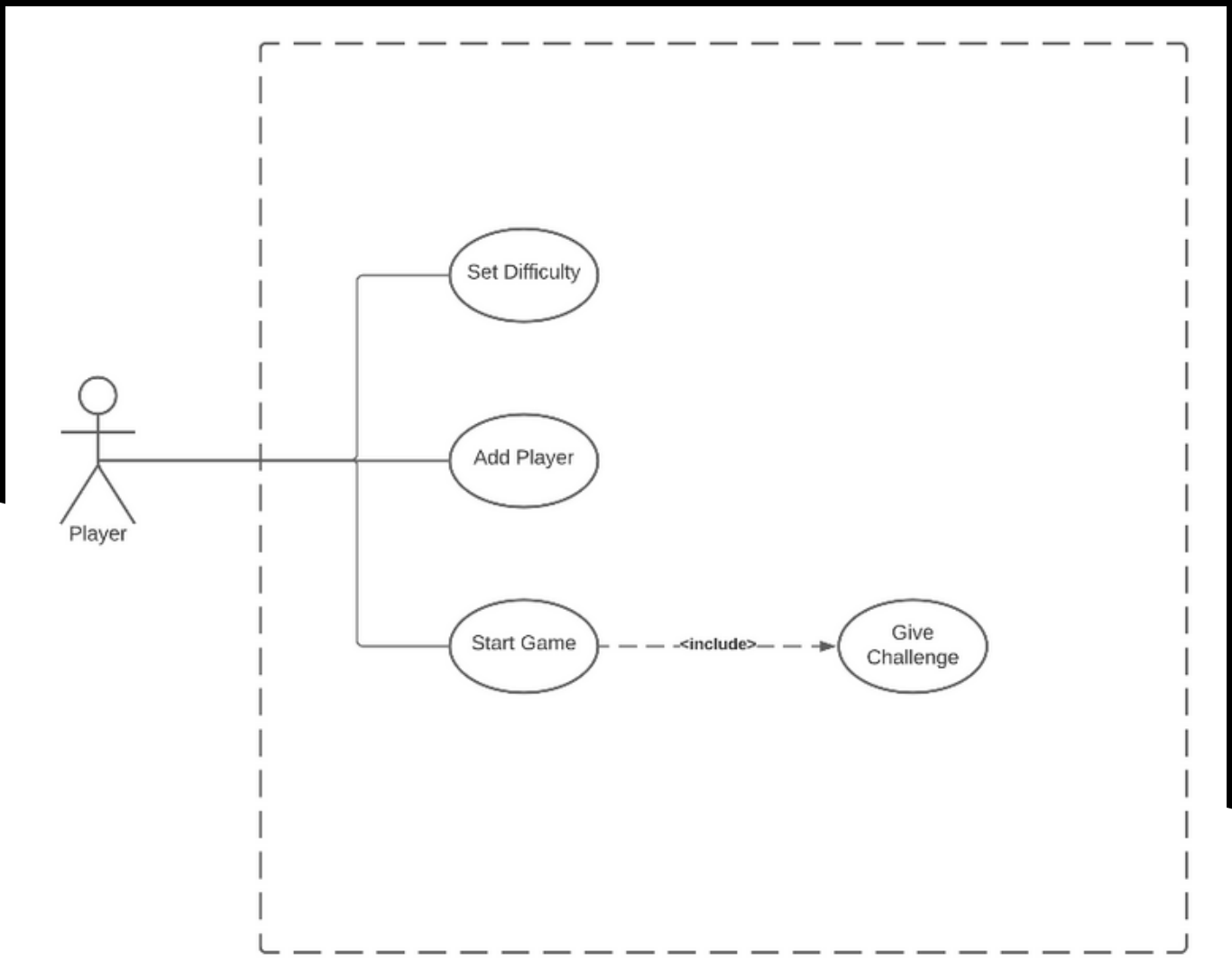
Das UML beinhaltet nur die Logik für das GUI.
Wir nutzen JDK 16.
Das Programm sucht sich die 60 Challenges aus der ausgewählten Kategorie heraus. Da auch Challenges öfter dran kommen können, muss man das Spiel mit dem Stop Button beenden, wenn man fertig ist.

4. UML - Klassendiagramm



Das UML-Klassendiagramm finden Sie unter anderem auch im git-Repository in dem Ordner: Nachdenkzettel mit dem Dateinamen "classDiagramDrink-Or-Dare.pdf"

5. UML Use-Case-Diagramm



Das UML Use-Case-Diagramm finden Sie unter anderem auch im git-Repository in dem Ordner: Nachdenkzettel mit dem Dateinamen "usecaseDrink-Or-Dare.pdf"

6. Stellungnahme

Architektur

Unsere Architektur unterteilt sich in insgesamt acht Packages. Hierzu gehören abstractClasses, exceptions, factory, gui, interfaces, player, resources, und threads. In den jeweiligen Packages befinden sich die entsprechenden Klassen für die Logik und das GUI. Unsere Tests befinden sich im test package unter "Project".

Clean Code

In den Kernklassen befinden sich keine public member. Unsere Objekte haben wir jeweils von einem Interface oder über eine Factory erstellt.

Tests

Wir haben insgesamt fünf Klassen aus drei unterschiedlichen Packages getestet. Unter dem Package "generators" werden die Klassen Generator Factory, Hard Generator und der Soft Generator getestet.
Im Package "gui" werden die wichtigsten Methoden des GUI getestet.
Im Package "player" überprüfen wir, ob sich das Array mit den Namen befüllt und ein Zufallsname aus diesem Array ausgesucht wird. Ein weiterer Test ist, ob dieser Name korrekt mit der Zufallskarte vereint wird.

GUI

Das GUI besteht aus insgesamt einem Frame, dessen Content sich verändert, je nach dem, welche Buttons man anklickt. Insgesamt gibt es 4 verschiedene Szenen. In der ersten Szene wählt man den Schwierigkeitsgrad und legt somit fest, welche Textdateien später ausgewählt werden. Hierzu gibt es die Buttons Soft und Hard. In der zweiten Szene kann man alle teilnehmenden Spieler eintragen oder auch löschen. Mithilfe des Submit Buttons fügt man Spieler in die Liste ein und mit den X Buttons löscht man sie wieder. Beim Löschen war es wichtig, zu beachten, dass die Einträge in der richtigen Reihenfolge bleiben und keine leeren Felder entstehen. Anschließend kann man auf der dritten Szene das Spiel beginnen, indem man auf einen Play Button klickt.
Auf der vierten Szene wird den Spielern nun angezeigt, wer an der Reihe ist, sowie die Aufgabe, die zu erfüllen ist, beziehungsweise die Frage, die zu beantworten ist. Klickt man auf den Next Button, ist der nächste Spieler dran, mit der nächsten Aufgabe/Frage. Dies kann man solange fortführen, wie man möchte. Um das Spiel zu beenden, klickt man auf den Stop Button. Nahezu alle Szenen wechselt man mit Next -oder Back Buttons. Hat man das Spiel gestoppt und möchte von vorn beginnen, sind bereits die vorherigen Spielernamen eingetragen, sodass man dies nicht noch einmal eintragen muss.

Logging/ Exceptions

Logging haben wir an allen (für uns) wichtigen Stellen eingebaut. Jedes Mal wenn wir eine Information des Benutzers anfordern, ob diese gespeichert wird oder wenn ein Objekt erzeugt wird. Logging befindet sich in nahezu allen Klassen.
Unsere eigenen Exceptions ArrayNotFilledException und IllegalStringException befindet sich im Package Exceptions, dort haben wir ebenfalls Logging verwendet und eine entsprechende Fehlermeldung als Message ausgegeben. Die ArrayNotFilledException wird geschmissen, wenn das Array nicht befüllt werden konnte. Die IllegalStringException wird bei Null oder wenn die Stringlänge 0 beträgt, geschmissen.

Threads

Im Package Threads befinden sich zwei selbstgeschriebene Threads. Der BackgroundMusicThread sorgt für eine angenehme Hintergrundmusik während des Spiels. Dieser Thread wird in der Main gestartet. Der BackUpCardsThread ruft die Methode fillTextFileWithCards() auf, welche ein TextFile erstellt, in dem jede gespielte Karte parallel abgespeichert wird. Diese Methode ist synchronized, sodass immer nur ein Thread in das Text-File schreiben kann. Sobald man den Beenden-Button drückt wird die Textdatei automatisch gelöscht.

Streams und Lambda –Funktionen

Im Package abstractClasses in der Klasse CardGenerator befindet sich der Stream. Das Programm wählt eine Zufallskategorie (Soft oder Hard). Die entsprechende Liste mit den Fragen aus der gewählten Kategorie wird an den Stream übergeben.

Factory

Da wir zwei Schwierigkeiten haben, bei der die Klassen jeweils identisch aufgebaut sind, haben wir hier eine Factory des entsprechenden Interfaces erstellt. In einem Switch-Case wird das jeweilige Objekt erstellt. Die Factory befindet sich im Package Factory, die Klasse wurde GeneratorFactory benannt.

7. Bewertungsbogen

Vornamae	Nachname	Kürzel	Matrikelnummer	Projekt	Arc.	Clean Code	Doku	Tests	GUI	Logging/Except.	UML	Threads	Streams	Profiling	Summe - Projekt	Kommentar	Projekt-Note
Gabriela	Gerhardt	gg037	41938	DrinkOrDare	3	2	3	2	3	2	2	3	2	3	25,00		1,70
Kiara	Krug	kk167	41937	DrinkOrDare	3	2	3	2	3	2	2	3	2	3	25,00		1,70

8. Profiling Analyse

Die Nachdenkzettel finden Sie unter anderem auch im git-Repository in dem Ordner: Nachdenkzettel