Artem Kägi

Supercomputing Systems AG

Technoparkstrasse 1

CH-8005 Zürich

IPA 2022

"Endless Runner" Spiel für Android

Inhaltsverzeichnis

Inhaltsverzeichnis

[1 Teil 1 2](#_Toc98171733)

[1.1 Vorwort 2](#_Toc98171734)

[1.2 Aufgabenstellung 2](#_Toc98171735)

[1.2.1 App 3](#_Toc98171736)

[1.2.2 Gameplay 3](#_Toc98171737)

[1.2.3 Backend 3](#_Toc98171738)

[1.3 Projektaufbauorganisation 5](#_Toc98171739)

[1.4 Vorkenntnisse 6](#_Toc98171740)

[1.5 Vorarbeit 6](#_Toc98171741)

[1.5.1 In den letzten 6 Monaten 6](#_Toc98171742)

[1.6 Arbeitsprotokoll 7](#_Toc98171743)

[1.6.1 Mo 28.02.2022 7](#_Toc98171744)

[2 Glossar 8](#_Toc98171745)

# Teil 1

## Vorwort

Diese Dokumentation gehört zur IPA von Artem Kägi, welcher bei der Firma Supercomputing System AG angestellt ist. In diesem Dokument ist die Arbeit und das Vorgehen dokumentiert.

Die Dokumentation ist in 2 Teile gegliedert

Im ersten Teil wird die deteillierte Aufgabenstellung und der Ablauf der Arbeit aufgezeigt. Hier werden die vorhandenen Vorkenntnisse und die benutzten Mittel aufgezählt

Im zweiten Teil wird das Projekt auch wirklich Dokumentiert und beschrieben, es wird auch erwähnt welche Probleme aufgetreten sind.

Der Ganze Sourcecode und Assets werden in einem Git repository Hochgeladen.

Durch die ganze Arbeit hinweg wurde die Projektplanungsmethode IPERKA eingesetzt.

## Aufgabenstellung

Das Ziel der IPA ist ein Android Spiel vom Typ "Endless Runner". Das Grundkonzept dieser Spiele ist folgendes: Der Avatar bewegt sich automatisch durch eine Spielwelt. Der Spieler steuert seinen Avatar und versucht den darin vorkommenden Hindernissen so lange wie möglich auszuweichen. Je länger der Spieler dies schafft, umso mehr Punkte werden gesammelt. Dazu stehen dem Spieler diverse Bewegungen wie z.B. springen, ducken, nach Links bzw. Rechts ausweichen etc. zur Verfügung. Sobald der Spieler mit einem Hindernis kollidiert, ist das Spiel zu Ende. Das Ziel des Spieles ist es also, so lange wie möglich zu überleben, d.h. Kollision zu vermeiden.

Die Mobile-Applikation wird mit einem zentralen Backend kommunizieren. Das Backend verwaltet die Spieler und deren erreichte Punkte. Damit können die Spieler ihre erreichten Highscores miteinander vergleichen.

Der Spieler kann während seinem Run eine Währung, sogenannte Coins, sammeln. Mit diesen kann er "Skins" kaufen. Diese Skins erlauben es dem Spieler das Aussehen seines Avatars im Spiel zu verändern.

Für die Assets im Spiel (z.B. Avatar, Skins, Objekte in den Levels) können bestehende Assets oder selbst erstellte Assets verwendet werden.

Die Mobile-Applikation schickt die erreichten Highscores an ein zentrales Backend, wo diese mit dem Spielerpseudonym persistiert werden. In der Mobile-Applikation sind neben den Highscores auch die freigeschalteten Skins einsehbar.

Voraussichtlich verwendete Technologien: C# (Unity) für Android Mobile Spiel, Python REST Server für Backend und MySQL Datenbank im Backend.

### App

Menusystem: Die App hat ein Menu mit den Inhalten "Spiel Starten", "Highscore" und "Skins".

* Spiel Starten: startet einen neuen Run.
* Highscore: zeigt die Top 10 Highscores über alle Spieler an. Der eigene Highscore wird separat angezeigt. Wie diese Highscores dargestellt werden, wird während der IPA evaluiert.
* Skins: Alle im Spiel vorhandenen Skins sind hier dargestellt. Bei jedem Skin ist ersichtlich, ob der Spieler ihn bereits freigeschaltet hat oder nicht. Skins können in diesem Menu mit den gesammelten Coins freigeschaltet und aktiviert werden.

### Gameplay

Die Spielwelt wird während dem Spiel zufällig generiert.

Der Avatar bewegt sich automatisch durch die Spielwelt. Der Run ist zu Ende, sobald der Avatar mit einem Hindernis kollidiert. Der Spieler hat folgende Aktionen zur Auswahl:

* Springen: Der Avatar weicht einem Hindernis aus, in dem er darüber springt.
* Sliden: Der Avatar weicht einem Hindernis aus, in dem er sich darunter durch bückt.
* Links: Der Avatar weicht einem Hindernis aus, in dem er nach links ausweicht.
* Rechts: Der Avatar weicht einem Hindernis aus, in dem er nach rechts ausweicht.
* Coins sammeln: kollidiert der Avatar mit einem Coin so wird dieser aufgenommen.
* Sterben: kollidiert der Avatar mit einem Hindernis ist der Run zu Ende. Der Spieler wird aufgefordert seinen Namen einzugeben. Wenn der Spieler will, kann der erreichte Highscore öffentlich gemacht werden.
* Punkte: der Spieler erhält Punkte je länger der Run dauert. Pro Frame, das der Avatar rennt, erhält der Spieler einen Punkt.

### Backend

Das Backend bietet eine REST Schnittstelle mit den folgenden Operationen

* Highscore erfassen und abrufen
* Coins erfassen und abrufen
* Skins freischalten und abrufen

Dieses REST API genau zu spezifizieren ist Teil der IPA.

Das Backend wird mit Python umgesetzt. Es verbindet mit einer MySQL Datenbank.

Die Datenbank speichert alle notwendigen Daten, um die Informationen in den Menüpunkten Highscore und Skins darstellen zu können.

## Projektaufbauorganisation

**Lehrbetrieb und Durchführungsort:**

Supercomputing Systems AG

Technoparkstrasse 1

8005 Zürich

043 456 16 00

[Info@scs.ch](mailto:Info@scs.ch)

**Kandidat:**

Artem Kägi

Agnesstrasse 8

078 705 64 27

[artem.kaegi@scs.ch](mailto:artem.kaegi@scs.ch)

**Berufsbildner:**

Christian Walder

Benedict Schule

Militärstrasse 106

8004 Zürich

044 298 17 83

[christian.walder@benedict.ch](mailto:christian.walder@benedict.ch)

**Verantwortliche Fachkraft:**

Pascal Iselin

Supercomputing Systems AG

Technoparkstrasse 1

8005 Zürich

079 215 62 37

[pascal.iselin@scs.ch](mailto:pascal.iselin@scs.ch)

**Nebenexpert:**

Rinoy Manavalan

[rinoy.manavalan@gmail.com](mailto:rinoy.manavalan@gmail.com)

078 226 80 19

**Hauptexpert:**

Marko Tepeluk

[martepeluk@gmail.com](mailto:martepeluk@gmail.com)

076 497 58 88

## Vorkenntnisse

Kandidat hat bei der SCS schon Handy Apps entwickelt und zu einer auch ein Python Backend erstellt. Er hat Interesse für Spiele Entwicklung und hat in seiner Freizeit schon mit Unity Spiele konstruiert. Für ein weiteres Projekt hat er auch eine MySql Datenbank mit einem Python Rest interface verbunden.

* Unity: Unregelmässig mit Unity, PC-Spiele erstellt. Noch nie ein Handygame entwickelt.
* Python: Rest Schnittstelle für Flutter App schon einmal erstellt.
* MySql: Datenbank für ein Python Backend erstellt.

## Vorarbeit

Für die IPA wurden folgende Massnahmen vorgenommen:

* Unity wurde auf dem Arbeitslaptop installiert
* Es wurde ein «Testspiel» auf ein Android Handy gebracht.

### In den letzten 6 Monaten

Entwicklung von Handy Apps und Python Applikationen:

* OFS: Eine App mit einem Python Rest backend
* .Net Applikation für SBB intern.

Werkzeuge:

* JetBrains: Rider, Intellij, PyCharm
* MySql Workbench

## Arbeitsprotokoll

Starttermin: 28.02.2022

Abgabedatum: 15.03.2022

|  |  |
| --- | --- |
| Mo 28.02.2022 | |
| Ausgeführte Arbeiten | * Zeitplan erstellt * Aufbau der Dokumentation definiert * Leitfragen und Aufgabenstellung studiert |
| Ungeplante Arbeiten |  |
| Erreichte Ziele | * Der Zeitplan steht * Aufbau der Dokumentation ist klar |
| Beanspruchte Hilfestellungen |  |
| Aufgetretene Probleme |  |
| Reflexion | Ich konnte gut in meine IPA starten, jedoch habe ich ein paar fragen und Unsicherheiten. Die fragen kann mir hoffentlich mein Hauptexperte morgen beim besuch beantworten. Meine grösste Unsicherheit bezieht sich auf die Dokumentation, ich bin mir unsicher, wie ich dieses Umsetzen soll. Ich habe es jetzt nach Leitfaden gemacht, jedoch endet der Streng definierte teil hier und ich muss mir was ausdenken. Ich hoffe, dass auch das beim Gespräch mit dem Hauptexperten morgen etwas klarer wird. |

# Glossar

|  |  |
| --- | --- |
| Begriff | Erklärung |
| Unity | Ein Tool zum Entwickeln von Spielen oder auch eine Game Engine genannt. In diesem Projekt wird vor allem in Unity gearbeitet, um das Spiel zu entwickeln. |