

Projektarbeit

1. Rahmenbedingungen

- 2er Gruppen
- Abgabe bis zum **22.01.2023 - 23:59 Uhr**
- Unity **2021.3.14f1** LTS
- Abgabe als Upload auf Stud.IP (Ordner: **Hausarbeit Abgabe**)
 - **Zip-Datei**
 - Name der Datei: "Gruppe-**X**.zip", wobei X durch den Buchstaben ihrer Gruppe zu ersetzen ist (s. Stud.IP)
 - **Inhalte der Zip-Datei**
 - Ordner des Unity Projekts mit genau 3 Unterordnern:
 - Assets
 - Packages
 - Project Settings
 - Windows-Build
 - Readme.md bzw. Readme.txt, mit Inhalt:
 - Gruppenname
 - Name aller Gruppenmitglieder
 - ggf. Erklärungen (z. B. Steuerung)

HINWEIS

Sollte die ZIP-Datei für Stud.IP zu groß sein, kann alternativ auch eine Textdatei mit einem Downloadlink zu ihrer Abgabe bei Stud-IP hochgeladen werden. Bitte stellen Sie hierbei sicher, dass der Downloadlink für mindestens 2 Wochen gültig ist.

2. Aufgabenbeschreibung

Erstellen Sie ein vollständiges Spiel, welches hauptsächlich im UI (d. h. unterhalb einer Canvas-Komponente) stattfindet. Das Spiel soll einen kurzen aber vollständigen Gameloop enthalten. Das bedeutet, eine Spielrunde muss ohne Abstürze zu einem regulären Abschluss gelangen können.

Grafik und Ton

Es dürfen sowohl eigene als auch Creative-Commons lizenzierte Grafiken und Sounds genutzt werden. Beachten Sie dabei die Lizenz jedes fremden Assets und geben Sie bei Bedarf Autor und Quelle des Assets mit an. Platzieren Sie dazu eine **Textdatei "Quellen"** in Ihrem Projekt.

Grafisch soll das Spiel ein einheitliches Aussehen aufweisen.

Das Spiel muss **Soundeffekte** nutzen; mindestens muss ein Sound bei der Betätigung eines Buttons abgespielt werden. Die Nutzung von **Musik** ist optional.

Menüstruktur

Das Spiel muss ein **Hauptmenü** mit mindestens den folgenden Funktionen aufweisen:

- Spiel starten
- Optionen
- Credits
- Spiel beenden

Darüber hinaus muss das Hauptmenü den Titel Ihres Spiels anzeigen.

Durch Auswahl von Optionen oder Credits soll das Hauptmenü ausgeblendet und das jeweilige Untermenü eingeblendet werden. Aus den Untermenüs muss in das Hauptmenü zurück navigiert werden können.

Das Optionsmenü muss **mindestens einen Slider und ein Dropdown** enthalten.

Der Slider soll die Lautstärke der Soundeffekte regulieren. Das Dropdown soll sinnvoll eingesetzt werden, z. B. für Video Optionen (Auflösung, FPS Cap, etc.) oder für Spieleinstellungen (Schwierigkeitsgrad, Anzahl an Fragen, etc.).

Das Credits-Menü muss mindestens die vollen Namen aller Studierenden enthalten, die an dieser Projektarbeit mitgewirkt haben.

Das Hauptmenü und das Optionsmenü können beliebig um weitere Elemente und Untermenüs erweitert werden.

Vorgabe: Quiz

Bei der Auswahl von "Spiel starten" gelangen Spieler zu einem Quiz, bei dem sie eine festgelegte Anzahl Fragen beantworten müssen. Für jede richtig beantwortete Frage erhalten die Spieler Punkte.

Nachdem die Spieler alle Fragen beantwortet haben, soll ihr Endpunktestand angezeigt werden. Sie sollen außerdem die Möglichkeit haben, direkt noch einmal zu spielen oder zum Hauptmenü zurückzukehren.

Sie sind frei in der Gestaltung des Aufbaus und Inhalts Ihres Quiz-Spiels.

Sie müssen jedoch mehrere Arten der Interaktion implementieren, durch welche Spieler Fragen beantworten können. Hierbei steht die Verwendung verschiedener Layouts und UI-Elemente im Vordergrund.

Implementieren sie dazu **mindestens 3** der folgenden Typen von Quizfragen:

- **Single Choice:** Spieler müssen die richtige Antwort aus mehreren vorgegebenen Antwortmöglichkeiten wählen.
- **Multiple Choice:** Spieler müssen alle richtigen Antworten aus mehreren vorgegebenen Antwortmöglichkeiten wählen. Hierbei kann eine beliebige Anzahl der Antwortmöglichkeiten richtig sein, auch jede oder keine der Optionen.
- **Slider:** Spieler müssen den richtigen Zahlenwert durch Bewegen eines Sliders eingeben. Eventuell muss die Lösung nicht exakt richtig sein, sondern nur innerhalb eines bestimmten Bereichs liegen.
- **Input:** Spieler müssen die richtige Antwort in ein Eingabefeld eingeben. Zeichenlänge und zulässige Zeichen sollten sinnvoll limitiert sein.
- **Single Choice - Image:** Spieler müssen das richtige Bild aus mehreren vorgegebenen Bildern auswählen.

Es steht Ihnen frei, sich weitere Typen von Quizfragen zu überlegen und zu implementieren.

Achten Sie in jedem Fall darauf, den Spielern **audiovisuelles Feedback** zu ihren Antworten zu geben. Spieler müssen nachvollziehen können, ob ihre Antwort richtig ist oder welche die richtige Antwort gewesen wäre.

Alternative Spielideen

Das Quiz ist eine mögliche Vorgabe unsererseits, die vieles von dem abdeckt, was wir in ihrer Prüfungsleistung sehen wollen.

Wenn Sie eine eigene Spielidee haben und umsetzen möchten, dürfen Sie uns diese gern vorstellen (schreiben Sie uns dazu im Discord an). Ihr Spiel muss in jedem Fall die über dem Abschnitt "Vorgabe: Quiz" genannten Anforderung erfüllen und darüber hinaus mehrere verschiedene UI-Komponenten nutzen. Wir entscheiden dann im Einzelfall, ob wir Ihre Idee für die Prüfung zulassen.

3. Bewertungskriterien

- Erfüllung der Aufgabenbeschreibung
- Aufbau des UIs (Hierarchie)
- Bedienbarkeit der UI Elemente
- Skalierbarkeit des UIs bei verschiedenen Bildschirmauflösungen und Seitenverhältnissen (mindestens 16:9, 16:10 und 21:9)
- Strukturierung und Dokumentation des Projektes
- Benennung in Unity und im Code
- Angemessene Verwendung von Unity eigenen Funktionen und Komponenten (z. B. Prefabs, Scriptable Objects, etc.)
- Code-Struktur

Hinweise zum Code

- Nutzen Sie in längeren Skripten Kommentare und/oder Regions um Ihren Code zu strukturieren.
- Halten Sie sich beim Schreiben von C# Code an die von Microsoft dokumentierten Konventionen. Achten Sie besonders bei der Benennung von Klassen, Feldern und Methoden darauf, dass ihre Namen auch ihre Funktionen beschreiben und nicht gar im Widerspruch zu diesen stehen.
- Fehlerbehafteter oder verbesserungswürdiger Code sollte als solcher erkennbar sein! Vermerken Sie eventuell bekannte Bugs als Kommentare an entsprechender Stelle im Code. Fügen Sie bei Bedarf weitere Kommentare hinzu, um auf mögliche Verbesserungen oder Erweiterungen bestimmter Abschnitte Ihres Codes hinzuweisen.
- Dokumentieren Sie die Funktionsweise von Methoden, wenn diese schwer nachzuvollziehen sind. Verwenden Sie Tooltip-Attribute für serialisierte Felder, deren Bedeutung nicht selbsterklärend ist.
- Beschränken Sie den Funktionsumfang von Klassen und Methoden auf einen sinnvollen Rahmen. Vermeiden Sie die Verwendung von "Gott-Klassen".
- Schreiben Sie wiederverwendbaren Code und vermeiden Sie doppelten Code. Benutzen Sie Komponenten sofern möglich mehrfach. Vermeiden Sie es, mehrere Skripte zu erstellen, die nahezu identische Aufgaben erfüllen und sich nur minimal unterscheiden.
- Achten Sie auf Skalierbarkeit und Erweiterbarkeit Ihrer Skripte. Eine minimale Veränderung oder Erweiterung von Anforderungen sollte nicht dazu führen, dass Skripte komplett neu geschrieben werden müssen.

Fremd-Code

- Falls Sie an irgendeiner Stelle fremden Code benutzen, müssen Sie diesen durch einen Kommentar inklusive Quellenangabe als solchen identifizierbar machen. Vermeiden Sie die Verwendung komplett fremder Klassen.
- Unity- oder .NET-Klassen sowie Packages aus der Unity Registry zählen selbstverständlich nicht als Fremd-Code.

Asset Store

- Die Nutzung von Assets aus dem Unity Asset Store ist **nicht** erlaubt.