

Reimplementation des Videospiel Klassikers Mario

Ansprechpartner: Simon Eismann (simon.eismann@uni-wuerzburg.de), Raum A107/8
Lehrstuhl für Informatik II (Software Engineering)

1 Thematik

In diesem Projekt soll der Videospiel Klassiker Super Mario Bros. in Java mit einer modernen Model-View-Controller Architektur implementiert werden. Wie im Klassiker von 1987 soll die Spielfigur laufen, rennen und springen können. Während des Spielverlaufs können Münzen eingesammelt werden und Gegner durch auf ihren Kopf hüpfen ausgeschaltet werden.

2 Anforderungen an die Software

2.1 Musskriterien

- Die Spielfigur muss sich bewegen können.
- Die Spielfigur muss springen können.
- Beim Kontakt mit Gegnern und durch fallen in Abgründe verliert der Spieler.
- Gegner können durch auf ihren Kopf spingen ausgeschaltet werden.
- Es muss ein Menü vorhanden sein.
- Das Spiel muss Soundeffekte besitzen.
- Im Laufe der Entwicklung wird es eine unerwartete Requirement Änderung geben.
- Es muss eine Model-View-Controller Architektur mit strikter Aufgabentrennung verwendet werden.

2.2 Abgrenzungskriterien

- Es sollen keine Graphiken designed werden.
- Es muss kein eigener Sound designed werden.
- Es

2.3 Umgebung / Technologien

- Das Spiel soll in Java implementiert werden.
- Für die GUI soll Java FX verwendet werden.

3 Regeln der Arbeit

3.1 Allgemein

Jedes Mitglied muss mitarbeiten und eigene Verantwortung haben. Der Kern der Applikation muss allen Mitgliedern bekannt sein. Ein gemeinsames Treffen mit dem Betreuer sollte regelmäßig stattfinden. Der Kurs hat 10 ECTS Punkte – das entspricht 10 Stunden Arbeit pro Person pro Woche!

3.2 Code

Der Code muss sauber und ästhetisch geschrieben werden. Es *muss* ein Git Repository angelegt werden. Die Einarbeitung in Git ist Bestandteil des Projektes. Zusätzliches Werkzeug (Issue tracker, wiki etc.) sind optional aber empfohlen. Der Betreuer kann mit dem Code nur dann helfen, wenn der Code im Git eingecheckt ist (Vereinbarung).

3.3 Abgabe

Bestandteile der Abgabe sind:

- Code nur über Git
 - Code muss im Git der Universität liegen
 - Projekt muss der folgenden Gruppe zugeordnet sein:
<https://gitlab2.informatik.uni-wuerzburg.de/softwarepraktikum/>
- Benutzerhandbuch (Markdown im Git, optional noch ein L^AT_EX Dokument)
 - Benutzung
 - * Kommandozeilenbefehle
 - * GUI
 - * Menüs
 - Installation
 - * Voraussetzungen
 - * Einrichtung
 - * Deployment
- Developerhandbuch - Pflichtenheft (L^AT_EX oder HTML im Git)
 1. Software Requirements Specification (SRS) - Funktionale und Nichtfunktionale Anforderungen
 2. Spezifikation der Funktionalität
 3. Entwurf Softwarearchitektur mit Komponenten und Schnittstellen
 4. Klassendiagramme + Datenbankdiagramme
 5. Typische Nutzungsszenarien / Case Studies (auch im Benutzerhandbuch erlaubt)
 6. Testfälle - automatisierte und manuelle Testverläufe
 - Hier muss mit dem Betreuer besprochen werden, welche Tests es gibt
 - Wie diese durchzuführen sind
 - Und der Betreuer muss diese am Ende abnehmen
 7. Zeitplan der Arbeit
- Folien (mit Video)

3.4 Verlauf

Treffen 0: *HEUTE!* Austausch der Kontakte, Festlegen von regelmäßigen Terminen und Fragen.

Woche 1-2: Erstellen des Pflichtenhefts:

- Software Requirements (SRS), Funktionalitäten definieren
- Git Repository anlegen und aktiv verwenden
- Technologien auswählen (Programmiersprache, Bibliotheken, APIs)
- Diagramme (Architektur, Datenbank, ...)
- Zeitplan erstellen

...

Woche N-2: Abgabe des Codes, Testfälle für Musskriterien. Besprechung des Vortrags.

Woche N-1: Folien + Video. Letzte Korrekturen des Codes.

Woche N: Abschlussveranstaltung. Vortrag und Abstimmung.

3.5 Durchfallkriterien

Der Kurs wird für die ganze Gruppe nicht bestanden, wenn:

- nicht alle Funktionalitäten (Musskriterien) implementiert sind und die nicht implementierten Musskriterien nicht sinnvoll verteidigt werden.

Deadline Woche N-1

- Abgabetests für Musskriterien fehlen und die fehlenden Tests können nicht sinnvoll verteidigt werden. Manuelle Tests sind auch erlaubt aber müssen spezifiziert werden.

Deadline Woche N-1

Die endgültige Entscheidung über Bestehen / Durchfallen wird immer vom Betreuer getroffen.