

Dokumentation & Projekttagebuch

Innovation Lab 1
Jahr 2021

Projekt: **Raspberry PI Arcade Game**

Team: **Gruppe 26**

1. Allgemeine Informationen

Projektname: Raspberry PI Arcade Game

Supervisor: Markus Petz

Innovation Lab 1, Wintersemester 2021

Projektteam:

Nicholas Hoeller, if19b043@technikum-wien.at, Projektleitung

Miriam Boehm, if20b066@technikum-wien.at

David Xing, if19b124@technikum-wien.at

Marko Zepic, if20b256@technikum-wien.at

Stefan Zuza, if19b136@technikum-wien.at

Management-Summary des Projektes

Das Ziel dieses Projekts ist die Erstellung eines Arcade Automaten mithilfe eines Raspberry PI. Im Wintersemester 2021 wollen wir unseren Core in der „RetroArch“ API ausführen und implementieren ihn mithilfe eines Sample Codes namens „Skeletor“ von Libretro.

Rahmenbedingungen und Projektumfeld

Bei der Implementierung sind vor allem ein paar wichtige Punkte zu beachten. In der Datei der eigentlichen Implementierung muss die „libretro.h“ Header-Datei eingebunden sein, damit die notwendigen Funktionen für den Core implementiert werden können. Aus diesem Grund stehen uns nur zwei Programmiersprachen zur Auswahl, nämlich C und C++. Darüber hinaus muss der Code als DLL-Datei kompiliert werden, da dies das vorgegebene Format für Cores im RetroArch ist.

Semester-Roadmap

Während des jetzigen Semesters (Wintersemester 2021) haben wir vor allem den Fokus darauf gelegt, ein im RetroArch ausführbares Programm zu entwickeln, welches eventuell sogar schon Benutzereingaben verarbeiten und darauf dementsprechend reagieren kann.

Dazu ist es unserer Meinung nach am allerwichtigsten, sich im vorhinein mit der Materie vertraut zu machen und sich eventuell Dokumentationen oder bereits geschriebene Cores anzuschauen. Womöglich lässt sich bei der Recherche auch ein Tutorial finden, welches den Entwicklungsprozess etwas genauer erklärt.

Collaboration & Tooling

Link zum GitHub Respository unseres Projekts: <https://github.com/Miriaana/InnoLab-Raspberry-Pi-Arcade-Game>

2. Projekt-Kurzbeschreibung

Das Hauptziel ist, am Ende des Semesters einen Core zu haben, welcher korrekt in RetroArch reingeladen werden kann und auf Eingaben eines Benutzers reagiert, beispielsweise die Bewegung eines Objekts mit den Pfeiltasten bzw. mit WASD. Mithilfe dieses Cores haben wir dann am Ende ein sehr stabiles Fundament für die weitere Entwicklung in den darauffolgenden Semestern, nämlich für ein richtiges Arcade Spiel, wie z.B. Pong.

Zudem ist mit einem funktionierenden Core auch eine der größten Herausforderungen bereits überwunden, da ab diesem Punkt der Fokus auf die Implementierung der Spiellogik und auf das Inputhandling gelegt werden kann.

Für die Programmierung des Cores werden wir uns vor allem mit den Dokumentationen und Tutorials von Libretro vertraut machen, da es auf der Website einen extra Bereich bezüglich Development gibt, in welchem auch einige GitHub Repositories verlinkt sind. Parallel dazu, werden wir auch erste Implementierungsversuche in Visual Studio und RetroArch starten, um schon mal Hands-on Erfahrung zu sammeln.

3. Spezifikation der Lösung

Anfangs haben wir uns fast ausschließlich mit der Recherche und dem Lesen von Dokumentationen gewidmet. Dabei sind wir auf der Libretro Website auf eine sehr hilfreiche Doku gestoßen (URL: <https://docs.libretro.com/development/cores/developing-cores/>), auf welcher wir einen Sample Code namens „Skeletor“ gefunden haben. Dieser Beispielcode besteht aus einer fertigen Projektstruktur für den RetroArch Core und enthält nur die nötigsten Funktionen. In der dazugehörigen Header-Datei fanden wir dann so gut wie alle wichtigen Informationen für die eigentliche Implementierung. Als API haben wir uns für das Emulator Frontend „RetroArch“ entschieden, da man hier ganz einfach Cores in Form einer DLL-Datei reinladen kann.

Für unseren nächsten Schritte haben wir uns dann vorgenommen, den Sample Code genauer anzuschauen bzw. Stück für Stück die jeweiligen Funktionen zu implementieren und diesen Code zum Testen in den RetroArch zu laden, um zu überprüfen, ob der Code erwartungsgemäß funktioniert.

Einen großen Durchbruch erlangten wir, indem wir die Dokumentationen und Tutorials beiseite legten und uns hauptsächlich mit der Header-Datei von „Skeletor“ beschäftigten. Nach ein paar Stunden Arbeitsaufwand haben wir es nämlich geschafft einen Core erfolgreich in RetroArch reinzuladen. Dieser wird im Debugging Fenster der API auch korrekt erkannt und ausgegeben. Innerhalb kurzer Zeit haben wir es auch geschafft, ein bewegbares Objekt zu spawnen, nachdem der Core geladen wird.

Kurz vor der Abgabe fanden wir beim Recherchieren im Internet eine sehr hilfreiche Anleitung zu Pong. Dies haben wir ebenfalls erfolgreich implementiert.

4. Auslieferung

GitHub Repository: <https://github.com/Miriaana/InnoLab-Raspberry-Pi-Arcade-Game>

Installation unter Windows:

- 1) RetroArch installieren (Link: <https://www.retroarch.com/?page=platforms>)
- 2) **OPTIONAL:** In RetroArch unter Settings > Logging > Loggin Verbosity aktivieren. Es wird ein zusätzliches Fenster geöffnet mit Debugging Informationen
- 3) Einstellungen der Controls unter Settings > Port 1 Controls
Hier können Sie die benutzerspezifischen Controls einstellen
- 4) Pong.dll aus dem GitHub Repository runterladen (Link: <https://github.com/Miriaana/InnoLab-Raspberry-Pi-Arcade-Game/blob/main/bin/Pong.dll>) und im „download“ Ordner des Dateipfads von RetroArch speichern
- 5) Um den Core in RetroArch zu installieren, navigieren Sie zu Main Menu > Load Core > Install or Restore a Core > Pong.dll
- 6) Nun taucht Pong.dll in der Core Liste auf (Main Menu > Load Core > Pong.dll) und beim erneuten Auswählen kann dieser geladen werden. Danach kann der Core im Main Menu gestartet werden.

5. Unser Projekt-Tagebuch

Sprint 1 (10.11.2021)

Beim ersten Sprint haben wir uns vorgenommen hauptsächlich zu recherchieren und Dokumentationen zu lesen, um uns mit der Materie vertraut zu machen. Dazu haben wir uns noch vorgenommen, uns mit RetroArch und Lakka genauer zu beschäftigen, da es zu dem Zeitpunkt noch unklar war, welche API wir benutzen. Im Laufe des Sprints hat sich dann aber RetroArch als die bessere Option herausgestellt. Außerdem hat ein Teammitglied noch versucht seinen eigenen Raspberry PI aufzusetzen und diesen für ein paar Testversuche startklar zu machen.

Sprint 2 (24.11.2021)

Nun haben wir uns als Ziel gesetzt RetroArch auf unseren Rechnern zu installieren und uns damit ein wenig vertraut zu machen. Dazu haben wir als reine Testversuche bereits fertige Cores inklusive Spiel reingeladen um zu sehen, wie der ganze Prozess und das Gameplay aussehen könnte. Außerdem wollten wir schon erste Implementierungen starten, indem wir uns nach der Development-Anleitung von Libretro richten, haben dies aber leider nicht geschafft, da wir ein wenig überwältigt von der Komplexität und Menge des Sample Codes waren.

Sprint 3 (08.12.2021)

Aufgrund von Stress wegen anderer LVs, haben wir in dieser Phase einen ziemlichen Zeitmangel gehabt und konnten somit sehr wenig am Projekt weiterarbeiten. Wir haben es lediglich geschafft, dass wir uns weiterhin mit dem Sample Code auseinandersetzen und einen kurzen Einblick in CMake machen.

Sprint 4 (15.12.2021)

In Sprint 4 gab es aufgrund anderer LVs ebenfalls wenig Fortschritt. Hier erreichten wir einen ziemlichen Tiefpunkt und es kam mehr oder weniger zu einem Stillstand im Projekt.

Sprint 5 (12.01.2021)

Hier hatten wir den größten Erfolg und konnten in RetroArch ein bewegbares Objekt erzeugen. Wir haben ebenfalls versucht verschiedene Objektformen zu erzeugen. Zusätzlich werden im Debugging Fenster von RetroArch jegliche Fehlermeldungen ausgegeben und vor allem die X- und Y-Koordinaten vom Objekt.

Kurz vor der Abgabe haben wir noch geschafft ein funktionierendes Pong Spiel zu implementieren. Dazu haben wir uns online an einem step-by-step guide orientiert. Manche Funktionen mussten wir dazu jedoch manuell umschreiben und anpassen.