

Technische Dokumentation
zum Projekt „Stock-Bälle-Klang“

im Fach

Audio-Video-Programmierung

bei Herrn Plaß und Herrn Sudau
im Wintersemester 2017/2018

Gruppenmitglieder:

Volkan Demiroglu [2096856]

John Adebiyi [2102775]

Shayan Keshmiri [2239412]

Tamara Schnarf [2248996]

Datum: 22.01.2018

1. Einleitung

Im Rahmen des Wahlpflichtfaches Audio-Video-Programmierung wurde das Spiel „Stock-Bälle-Klang“ entwickelt, welches in erster Linie zur Unterhaltung dienen soll und durch „Guitar Hero“ inspiriert ist.

Im Folgenden wird kurz auf die Zielsetzung vor Projektbeginn, die Vorgehensweise und technische Umsetzung im Backend sowie Frontend und schließlich auf das Projektergebnis eingegangen.

2. Zielsetzung vor Projektbeginn

Zu Beginn des Projektes war das Ziel, dass Spieler, zu einem selbst ausgewählten Track, im richtigen Moment, herunterfallende Noten spielen. Über eine Website sollte das Spiel gespielt werden können.

Zum Spielen der Noten sollten die Besucher vier sogenannte „Stock-Bälle“ benutzen. Nähere Erläuterungen sind im Projektkonzept beschrieben.

3. Vorgehensweise und technische Umsetzung

Das Spiel benötigt folgende **Hardware-Umgebung**:

Einen Rechner, worüber die Website aufgerufen wird und einen Monitor o.Ä., um das Spiel überhaupt darzustellen. Selbstverständlich werden der Kinect-Sensor sowie die „Stock-Bälle“ benötigt.

Das Spiel besteht aus folgender **Software-Umgebung**:

Die Kinect Verbindung ist mit C++ und OpenCV programmiert und besteht aus zwei Header und zwei .cpp Dateien.

Das Grundgerüst der Website ist mit HTML aufgebaut, die Funktionen in JavaScript geschrieben. Außerdem wurde JQuery genutzt.

3.1. Backend

Der erste Schritt war der Aufbau einer Kinect-Verbindung, bzw. das Ansprechen der Kinect über Qt. Anschließend folgte die Farberkennung der Stock-Bälle.

Zu Beginn wurden die Farbbereiche im RGB-System beschrieben, dann jedoch nochmal ins HSV-System übersetzt, um möglicherweise verschiedene Lichtverhältnisse auszugleichen, und um Farbbereiche besser definieren zu können. Da OpenCV keine HSV-Bilder darstellen kann, mussten die RGB-Werte über binäre Verundungen von Threshold und Originalbild umgewandelt werden, um in das HSV-System zu gelangen.

Eine Herausforderung war die Definition der einzelnen Farbwerte, um Bereiche zu kennzeichnen. Des Öfteren erkannte Qt zu schnell oder zu langsam, oder in einem zu großen oder zu kleinen Bereich die entsprechende Farbe. Vor allem die Erkennung innerhalb des definierten Kreises bereitete Probleme. Letztendlich wird nun in einem Viereck innerhalb des Kreises nach der gewünschten Farbe gesucht. Damit die jeweilige Farbe als erkannt gilt,

muss eine bestimmte Menge an Pixel, innerhalb des Bereiches, den gewünschten Farbbereich enthalten. Jeder Bereich wird mit Threads parallel abgefragt.

Eine weitere Herausforderung war abschließend, die Website und das Qt-Programm miteinander kommunizieren zu lassen. Der im Kinect-Bild definierte Bereich, sollte schließlich auch dem auf der Website dafür vorgesehene Bereich entsprechen und jeweils den Score zählen, wenn die Noten in diesen Bereich fallen.

Gelöst wurde diese Aufgabe über eine lokale Datei, auf die sowohl die Webseite, als auch QT zugreift. Dabei wird über QT immer der aktuelle Zustand der Farberkennung geschrieben (Red-, Blue-, Green-, Yellow-Detected oder „null“). Dieser wird von der Website abgefragt, sobald sich die Note im akzeptierten Bereich befindet. Hierbei musste viel ausprobiert werden, um Website und QT aufeinander abzustimmen, da keine direkte Kommunikation stattfindet.

Parallel wurde die Website zusammenführbar fertig programmiert, mit selbst geschnitten Tönen versehen und zunächst über die Tastatur spielbar gemacht.

3.2. Frontend

3.2.1. Song.js

Zu Beginn gab es Schwierigkeiten mit den Audio Nodes, da diese die Musik nie abgespielt haben. Letzten Endes wurde das Problem mit html Audio Tags gelöst.

Weitere Probleme gab es beim Stoppen. Wenn vor dem Lied Stopp gedrückt wurde, dann wurden nur die einleitenden Sounds (Jubel, Drumsticks) gestoppt, das Lied selbst lief jedoch trotzdem weiter. Gelöst wurde dies durch Zuweisen der timeout-Funktion an eine Variable. Sounds und Songs wurden in JavaScript realisiert.

3.2.2. Game.js

Auf die Länge des Songs abgestimmt, wurden eine Zufallszahl zur Farbbestimmung und eine weitere zufällige Zahl für den Abstand zur nächsten Note generiert. Anhand dessen, konnten die Noten der setTimeout()-Funktion zugewiesen werden. Jede dieser Noten wurde in einem Array gespeichert, um sie später alle wieder löschen zu können, wenn man Stopp drückt.

Zu Testzwecken wurde ein keylistener implementiert, um das Spiel zunächst mit der Tastatur spielen zu können.

Der Aufbau der Website konnte, ähnlich wie im Projektkonzept aufgeführt, umgesetzt werden. In der Mitte eine Spur mit herunterfallenden Noten und rechts ein Menü, welches eine Auswahl an unterschiedlichen Songs zur Verfügung stellt und einen Play-Button. Nach Auswahl des Songs, erscheinen der ausgewählte Song sowie eine Spielerstatistik (Punkte, Streak, Multiplikator). Die Änderungen sind live während des Spiels zu beobachten.

Am Ende des Songs erscheint außerdem ein Gameover-Screen, auf welchem Punktestand, Streak und Anzahl der gespielten Noten angezeigt wird.

4. Projektergebnis

Der Spieler ist in der Lage mit „Stock-Bällen“ die Noten auf der Website zu spielen, so dass geglückte und auch misslungene Noten durch ein Audiosignal begleitet werden. Auf der Website steht auf der rechten Seite eine Auswahl an möglichen Songs zur Verfügung sowie der sich immer mit aktualisierende Punktestand des jeweiligen Spielers. Außerdem verfügt das Spiel über den eingangs erwähnten Multiplikator, um einen möglichst hohen Punktestand zu ermöglichen. Es ertönt ebenfalls ein Startsignal sowie ständige Untermalung des Spiels durch Jubel-oder Buhrufen.

Die Kinect läuft fast reibungslos und ist in der Lage Bilder aufzunehmen und an QT weiterzuleiten, wo dann die vordefinierten Bereiche und Farben der Bälle erkannt und ausgewertet werden.

5. Fazit

Rundum war es ein gelungenes Projekt mit Lerneffekt. Die im Projektkonzept bereits beschriebenen Ziele, konnten weitestgehend umgesetzt werden.

Bei Weiterentwicklung des Projektes, könnte noch eine direkte Verbindung zwischen QT und der Webseite hergestellt und das Kinect-Bild in die Website eingebunden werden.