

Dokumentation Lost Track

1. Inhalt des Projekts

Lost Track ist ein webbasiertes Musikadventure, dass das Entdecken neuer Musik auf das nächste Level bringt. Der Spieler reist über Kontinente hinweg durch verschiedenste Länder, um die dort ansässige Musik kennenzulernen. Er findet dabei Songs in Form von Schallplatten auf, die er dann in seinem Spotify Account speichern kann. Dies wird mithilfe der Spotify Web API zur Musikbereitstellung, Construct 3 als Spieleengine und einem MEAN-Backend realisiert.

2. Was wurde geleistet?

- Ideenfindung [1]
 - Regelmäßige Brainstorm-Sessions abgehalten
 - Erste Mockups und Skizzen erstellt
 - Moodboards erstellt
 - Inspiration aus vergleichbaren Projekten gezogen (z.B. "Forgotify")

- Umsetzung [2]
 - Flussdiagramme erstellt
 - Gemeinsame Coding-Sessions abgehalten
 - Pair-Programming abgehalten
 - Videokonferenzen abgehalten
 - Regelmäßige Evaluation von Zielen durchgeführt
 - Ziele verworfen + neue erstellt
 - Fortschritte dokumentiert
 - VM eingerichtet und konfiguriert
 - Arbeitsumgebung eingerichtet
 - Construct 3:
 - Erstellung von Pixelart-Assets
 - Erstellung von Gameplay und Gamedesign
 - Erstellung Navigationsstruktur
 - Erstellung der GUI
 - Animieren der Grafikelemente
 - Backend:
 - User Login mit Spotify Account
 - Spotify API-Calls
 - Webserver Entwicklung
 - Playlist-Erstellung im User Account
 - Einbindung der Datenbank

[1] Für Mockup siehe Game Dokumentation

[2] Siehe Kommentare im Quellcode

[3] Siehe Anhang Unterpunkt Projektmanagement

- Frontend:
 - Spotify Web API & Webplayback-SDK Implementierung
 - Clientseitige Javascript-Logik
 - Oberfläche für Landingpage
 - Prototypen-Implementierung des Spotify Webplayers
 - Verknüpfung der Construct-Logik mit Spotify-SDK/API und Serverseite

- Präsentation
 - Erstellung der Präsentationsfolien und Plakate
 - MM/MI-Präsentation gehalten
 - Präsentation auf der Medianight WS 19/20

- Projektmanagement [3]
 - Projektbeschreibung
 - Projektmandat
 - Risikomanagement-Listen
 - Projektlösungsansatz
 - Projektplan gemäß Prince2
 - Produktstrukturplan
 - Produktbeschreibung
 - Produktflussdiagramm
 - Produktzeitplan

3. Probleme?

- Client-side
 - Spotify Webplayback SDK zum Laufen bringen (Doku ist vorhanden, aber wichtige Schritte werden ausgelassen)
 - Construct im Projekt eingliedern (Server und Spotify Web API ansprechen)
 - Pseudo JS Code beim Export
 - Scope und Aufrufbarkeit von Construct-Variablen

- Server-side
 - Proxyserver-Einrichtung für https
 - VM zum Laufen bringen
 - Eigenes Cover bei Playlist-Erstellung
 - Songvorschläge funktionieren nicht bei neu erstellten Spotify Accounts aufgrund mangelnder Daten

[1] Für Mockup siehe Game Dokumentation

[2] Siehe Kommentare im Quellcode

[3] Siehe Anhang Unterpunkt Projektmanagement

- Construct 3
 - Zeitaufwendige Erstellung von Assets
 - Aufgrund mangelnder Vorkenntnisse langsamer Fortschritt
 - Zu viele globale Variablen
 - Zeitaufwendiges Einarbeiten in den Editor
 - Abhängigkeit von Internetverbindung, da Construct 3 eine Webapplikation ist

4. Probleme im Projektmanagement?

- Öfters Leerlauf von Mitgliedern durch Pipeline-Abhängigkeit im Workflow
- Mini-Änderungen mussten aufwendig eingepflegt werden wegen API/SDKs
- Unklarheiten wurden nicht sofort aus dem Weg geschafft
- Nicht alles wurde für jeden eindeutig dokumentiert
- Selbst gesetzte Deadlines wurden nicht eingehalten
- Mangelhafte Erstellung eines Zeitplans
- Zu groß gewählter Projektrahmen

5. Was haben wir durch das Projekt gelernt?

- Vertiefung WebDev 2 Kenntnisse
 - Node.js
 - Routes
 - JQuery
 - Express
 - URL Params
 - Environment Variablen
 - Browser Security Policies (CORS, Website Einstellung)
- Spotify Webplayback SDK
 - Player aus URL Parameter initiieren, der auch von anderen Geräten aus steuerbar ist
 - angeforderte Songs abspielen, pausieren, überspringen
 - Infos zum aktuellen Song anfordern und anzeigen
- Construct 3 Editor
 - Layouts mit mehreren Layern erstellen und miteinander verbinden
 - Eventsheets erstellen
 - Skripte einbinden
 - mit dem internen Asset Editor umgehen
 - Funktionen korrekt benutzen
 - Lokale und Globale Variablen korrekt benutzen
 - Navigationsstrukturen erstellen

[1] Für Mockup siehe Game Dokumentation

[2] Siehe Kommentare im Quellcode

[3] Siehe Anhang Unterpunkt Projektmanagement

- mehrere Steuerelemente einbinden (Tastatur, Maus, Touch)
- 2D-Sprites animieren
- Team Arbeit
 - Lösungen und Kompromisse finden
 - Termine einhalten
 - genaue Kommunikation, Unklarheiten sofort ansprechen
 - sich auf alle Eventualitäten (Probleme) vorbereiten
- NodeJS Server
 - Routing mit Express
 - Eigene API Endpunkte entwickeln
 - OAuth Datenfluss für Login
 - NPM Pakete, die Entwicklung erleichtern (z.B. Nodemon)
 - Komplexere Web-Requests mit Authentifizierung zu nutzen
 - Debuggen von Web-Apps
 - Datenbank strukturieren und anbinden
 - Web-App auf einer VM zum Laufen zu bringen
- Projektmanagement Tools
 - Slack als offizielle Kommunikationsplattform
 - Bot installiert, der alle im Slack über Commits etc. informiert
 - Trello, um Projektplanung festzuhalten
 - Snapchat für Videokonferenzen

[1] Für Mockup siehe Game Dokumentation

[2] Siehe Kommentare im Quellcode

[3] Siehe Anhang Unterpunkt Projektmanagement