Schiebepuzzel

Teamprojekt Woche 4 JavaScript Weiterbildung cimdata.

# Namen aller Beteiligten

* Pedro Chincoa
* Yung-Hsin Zöllner

# Kurzbeschreibung

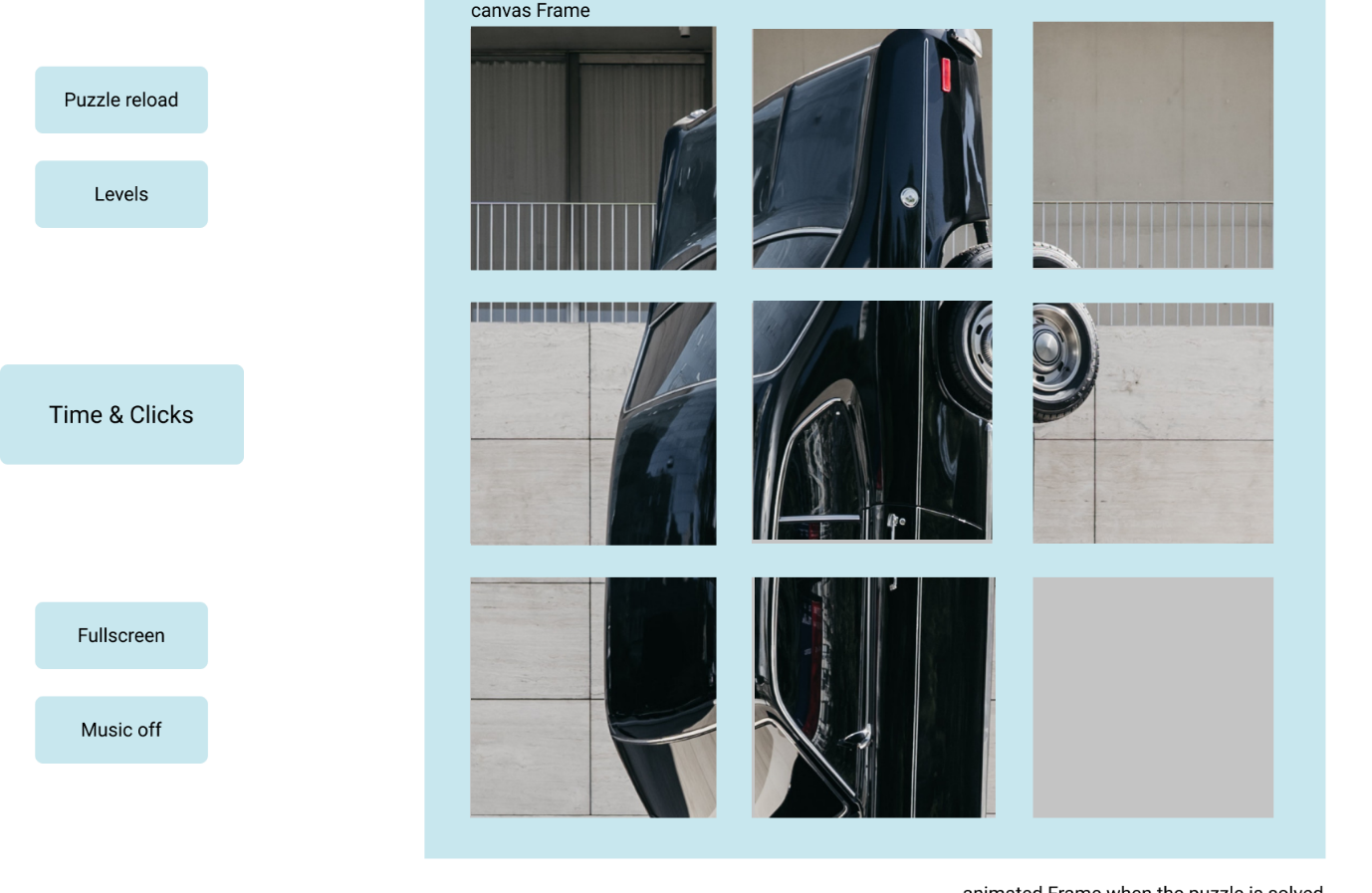
Erstellen eines Picture Puzzle Game (Schiebepuzzel).

Es ist ein Spiel, bei dem der Spieler aus 8 fragmentierten Teilen eines Fotos durch schieben das Bild in die richtige Ordnung bringt.

# Anforderungen und Fähigkeiten der Software

* HTML
* CSS
* JavaScript
* P5.js Library

# Ergonomisches Konzept (GUI)



# Terminliche Einschätzung der Arbeitsschritte

|  |  |  |  |
| --- | --- | --- | --- |
| # | Arbeitspaket | Definition of Done | Zeit-schätzung |
| 1 | HTML Struktur erstellen | - HTML5 Seite mit  - 3x3 DIVs für Puzzel  - (leere) style.css eingebunden  - (leere) main.js eingebunden  - Laden-Button  - DIV für Zähler der Züge | 1h |
| 2 | Bilder vorbereiten | - 5 Bilder sind vorbereitet  - Jedes Bild wird in 9 gleichgroße Quadrate zerteilt  - Ablage in Unterordner /images  - Auflösung hoch genug für FullHD Darstellung (1920 x 1080) | 2h |
| 3 | Lade Puzzelbild in Canvas | - Nutzer klickt "Reload image" und  - 8 Puzzelteile werden in Canvas geladen  - Puzzelteile werden geordnet dargestellt  - eine freie Stelle ist zufällig angeordnet  - Canvas skaliert mit Seitengröße (responsiveness)  - #6 wird ausgeführt | 8h |
| 4 | Puzzellogik implementieren, Event Listener hinzufügen | - Puzzelteile können angeklickt werden  - Wenn neben Puzzelteil die freie Stelle ist, springt das Puzzelteil dorthin  - keine Animation, einfacher Positionswechsel | 5h |
| 5 | Sound vorbereiten | Sound gefunden für  - Bild laden abgeschlossen  - Mischen  - Puzzelteil verschieben  - Gewonnen | 1h |
| 6 | Mischen implementieren | - Button "Mischen" wird aktiviert nachdem ein Bild geladen wurde  - Wenn Button "Mischen" gedrückt wird  - ertönt "Mischen"-Sound  - werden 8 Puzzelteile und freie Stelle gemischt (ohne Animation) | 4h |
| 7 | Gewinnanimation (Konfetti) implementieren | von oben herab fällt buntes Konfetti | 3h |
| 8 | Spielende implementieren (alle Puzzelteile richtig angeordnet) | - Spielende nach jedem Nutzerklick auf ein Puzzelteil prüfen  - wenn alle Puzzelteile richtig angeordnet wurden, Spielende erreicht, dann  - Button "Music off" deaktivieren  - Musik stoppen  - Gewinn-Sound abspielen  - trigger Gewinn-Animation (Konfetti) #7 | 10h |
| 9 | Implementiere "Time and Clicks" Button | - Im click event listener der Puzzelteile wird jeder Zug gezählt  - Button "Time and Clicks" kann gedrückt werden  - HTML wurde ein DIV Element hinzugefügt  - CSS formatiert dieses DIV Element über dem Canvas und versteckt es  - Nach Drücken von "Time and Clicks" wird das DIV Element über dem Canvas sichtbar  - Im DIV Element werden Spielestatistiken angezeigt:  - Zeit  - Anzahl Züge | 5h |
|  | Zeitmessung implementieren | - Sobald "Puzzle reload" gedrückt wurde und Mischen abgeschlossen ist (also das Spiel beginnt), startet ein Timer oder die Startzeit wird gespeichert  - Sobald das Spiel beendet wurde, stoppt der Timer oder die Endzeit wird genommen und die Differenz mit Startzeit berechnet  - Es wurde in JavaScript eine Variable hinzugefügt, die die gemessene Zeit speichert  - Nach Spielende wird die gespeicherte Zeit mit #9 angezeigt | 1h |
|  | (optional) Angeklickte Puzzelteile sliden mit Animation | angeklickte Puzzelteile neben der freien Stelle bewegen sich in einer geschmeidigen Animation | 5h |