

DinoProject

...

von Johannes Bartsch

Gliederung

1. Einleitung
2. Anforderungen
3. Begriffserklärungen
4. Vorgehen
5. Genutzte Software / Programmiersprachen
6. Demonstration
7. Ergebnisse
8. Probleme
9. Fazit
10. Fragen

Einleitung



Anforderungen

- Spiel auf HTML5-Canvas spielbar
- Startzeit: max. 3 Sekunden
- Springen und Ducken als Steuerungsarten
- “keine” Eingabeverzögerung bei gleichem Eingabegerät
- bei Websocket Verbindung: kaum Eingabeverzögerung
- Unterschiedliche Eingabemöglichkeiten
- Zugucken einer laufenden Session durch Aufruf des Links
- Übernahme Steuerung bei Verbindungsabbruch des Spielers
- Logging der Websocket Übertragungszeiten
- Responsiv für mehrere Geräte
- Umsetzung als PWA (Progressive Web App)

Begriffserklärungen



Vorgehen

1. Spiel auf Canvas erstellen
2. NGINX konfigurieren für Sessionverwaltung
3. Websocket Server mit Sessionverwaltung erstellen
4. Datenbank für Highscores und Log-Dateien erstellen
5. Backend erstellen
6. Steuerungsmöglichkeiten abstrahieren
7. Logging der Übertragungsgeschwindigkeiten

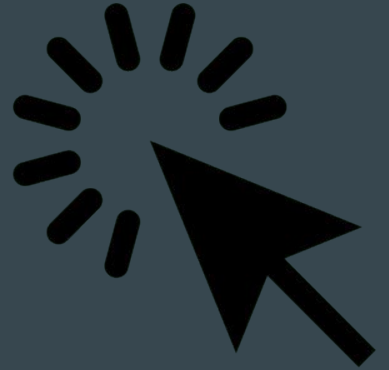


Genutzte Software / Programmiersprache



Demonstration

<http://webengineering.ins.hs-anhalt.de:32195/>



Ergebnisse

- Neues Spiel → Start neuer Session des Spiels
- Abbildung Session in URL
- Springen
- Multiplikator für Geschwindigkeit und Highscore
- Speichern Highscore für Session
- Übernahme Steuerung bei Verbindungsabbruch vom Spieler
- Logging der Übertragungszeiten
- Steuerung per Tastatur, Maus und Touch-Events
- Schnittstelle zum Anbinden neuer Steuerungsmöglichkeiten

Ergebnisse

DinoProject

Neues Spiel

```
Offset: 25 ms  
Offset: 26 ms  
Offset: 26 ms  
Offset: 24 ms  
Offset: 25 ms  
Offset: 26 ms  
Offset: 27 ms  
Durchschnitt: 25 ms
```

SCORE: 1559
HIGHSCORE: 589



Ergebnisse

```
// Input in Befehl umwandeln
function setInput(eventType, startEvent) {
    var key_state = (eventType == startEvent) ? true : false;

    controller.up = key_state;
}
```

```
// Tastatureingabe
window.addEventListener("keydown", event => {
    if(event.keyCode == 38) {
        setInput(event.type, "keydown");
    }
});
window.addEventListener("keyup", event => {
    if(event.keyCode == 38) {
        setInput(event.type, "keydown");
    }
});
```

Probleme

- Ducken nicht möglich
- keine Assets eingefügt
- Responsivität nicht vollständig
- nur 3 Steuerungsarten
- keine vollständige PWA



Fazit

Positiv:

- Erlernen / Nachlesen neuer Technologien (Websockets, Docker, PWA)
- Vertiefen bereits bekannter Gebiete (NGINX, JS, Python, SQL)
- Großes Interesse an der Aufgabe
- Spaß an der Bearbeitung

Negativ:

- Zeitplan besser erstellen
- Zu viel Zeit an Unteraufgaben verschwendet
- Kontakt zum Professor halten

**Vielen Dank für Ihre
Aufmerksamkeit!**



Fragen?

