Projektplan

Zur Seminararbeit im W-Seminar „Game-Development“

Thema: Tetris in Form eines kompetitiven Online-Multiplayer Browser Game

Name: Adrian Rarov

Inhaltsverzeichnis

1. Spielidee

Quelle: https://de.wikipedia.org/wiki/Tetris

Tetris ist ein puzzleartiger Computerspiel-Klassiker, bei dem man herunterfallende „Tetrominos“ (Aus vier Quadraten bestehende Bausteine) nacheinander in einem 10x20 Feld in 90° Schritten richtig drehen und von der Mitte nach rechts oder links verschieben muss, damit sie am unteren Rand horizontale, möglichst lückenfreie Reihen bilden. Die Steine fallen über die Zeit immer schneller, um die Schwierigkeit zu erhöhen. Im Original verschwinden Reihen, sobald sie vervollständigt sind und bringen dafür Punkte. In meinem Spiel verschwinden diese nicht, sondern bleiben um dem Spieler zahlreiche Möglichkeiten zu bieten, bis zu 4 Mitspielern durch verschiedene Attacken zu schaden. Je mehr Reihen übereinander gefüllt sind, desto stärker sind die Attacken. Das Ziel des Originals bleibt erhalten: Wenn dein Turm über das Spielfeld hinausragt, bist du draußen! Bei meinem Spiel ist der Gewinner der letzte, bei dem dies passiert.

Ziel der Arbeit ist es, Tetris in ein kompetitives Online-Multiplayer Browser Game zu verwandeln, sodass 2-5 Spieler über das Internet gegeneinander spielen können.

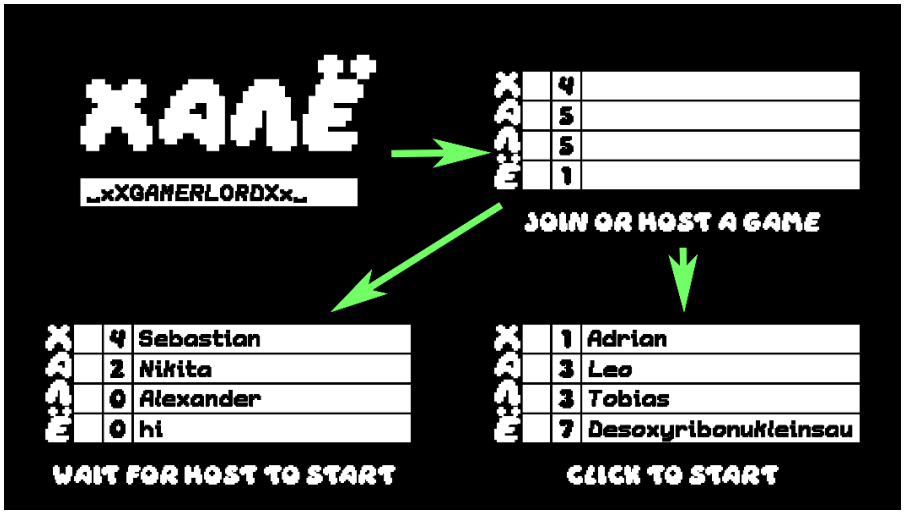
Falls es zeitlich möglich ist, möchte ich darüber hinaus mehrere Spielmodi implementieren, z.B. einen 2 gegen 2 Team-Modus.

2. Beschreibung der Funktionalitäten

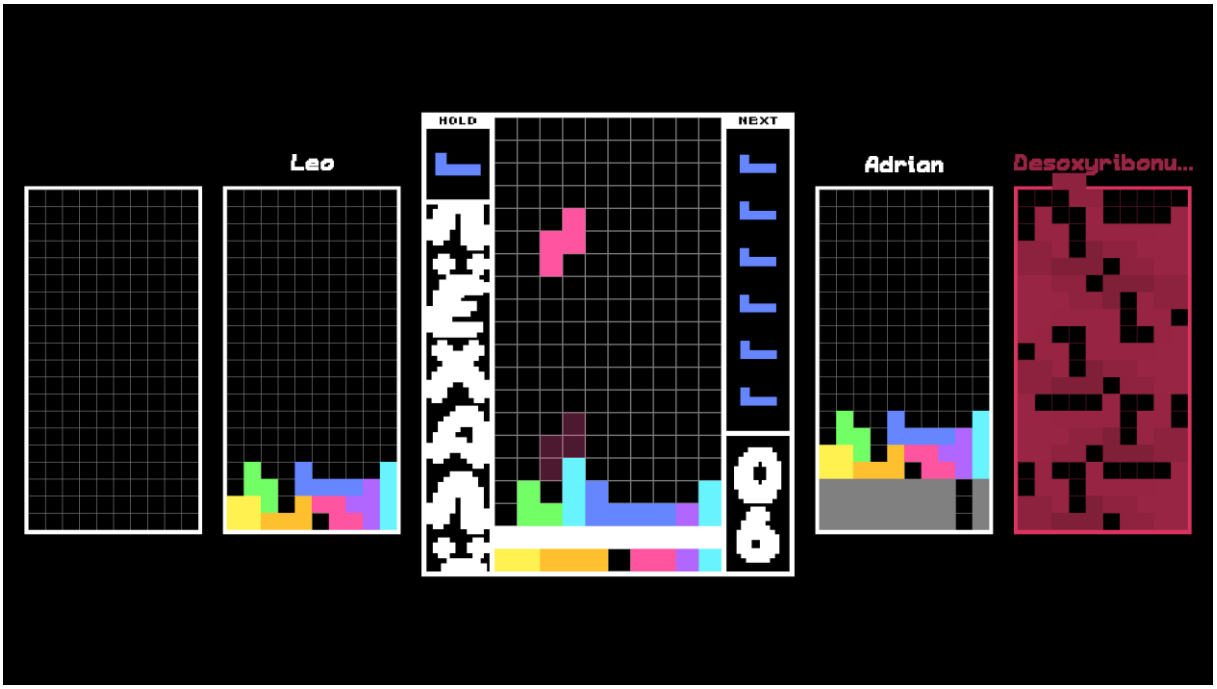
2.1. Startbildschirm, Pairing der Spieler

Nach Eingabe der Start-URL im Browser erscheint ein Startbildschirm, auf dem man in einem Rechteck seinen Nicknamen eingeben kann. Sobald der Spieler diesen eingibt und ENTER drückt, wird dem Spieler in der „Lobby“ ermöglicht, links entweder einen 4-stelligen Zahlencode einzutippen um dem Spiel eines Freundes beizutreten, oder den rechts angezeigten Code des eigenen Spiels Freunden zu senden, damit diese dem eigenen Spiel beitreten.

Entscheidet der Spieler sich Freunde zum eigenen Spiel einzuladen, werden die Namen der Mitspieler angezeigt, sobald sie beitreten. Dadurch taucht auch der Knopf „CLICK TO START“ unten auf, mit dem der Spielleiter das Spiel starten kann. Falls der Spieler dem Spiel eines Freundes beitreten möchte, wird er automatisch nachdem er die 4 Zahlen des Spiels eingetippt hat zu diesem weitergeleitet. Dadurch wird der eigene Code aktualisiert, die Namen der Mitspieler werden angezeigt und man wird darauf hingewiesen, auf den Spielleiter zu warten.



2.2. Spielmechanik beim laufenden Spiel



In der Mitte des Bildschirms wird das eigene Tetris-Spiel dargestellt. An den Seiten sind die Spielfelder der bis zu 4 Mitspieler, mit den dazugehörenden Nicknamen darüber.

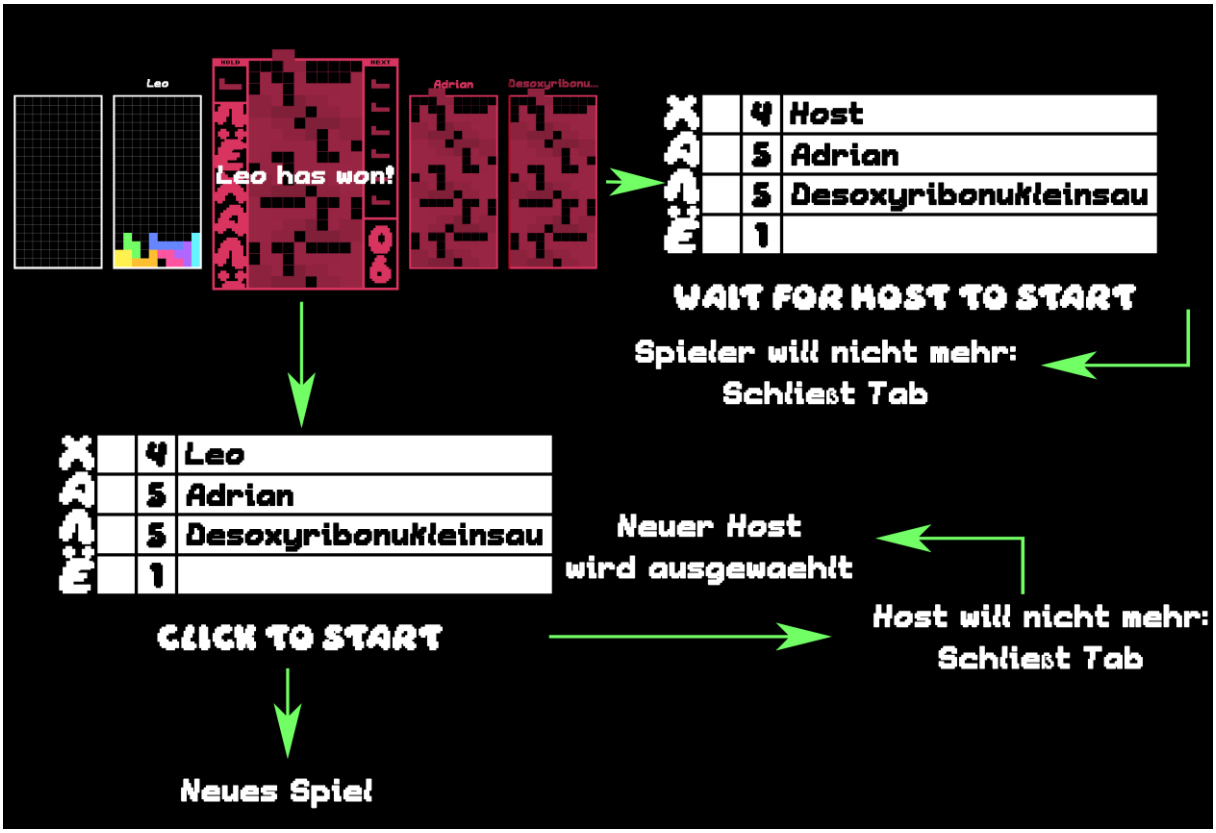
Der Spieler spielt hier Tetris mit der Tastatur. Man kann mit den Pfeiltasten den Stein nach rechts, links und unten bewegen, drehen und sofort fallen lassen. Die Anzahl der bereits gefüllten Reihen wird rechts angezeigt, unter der Warteschlange seiner 6 nächsten Tetrominos. Die Tetrominos werden nicht einfach zufällig ausgewählt, sondern sind immer in 7er Paketen geteilt, in dem jedes Tetromino in einer zufälligen Reihenfolge vorkommt. Links ist das Lager, in dem der Spieler ein Tetromino erst einlagern, und dann später gegen dieses eintauschen kann, falls ihm dieser Tetromino lieber ist.

Wenn der Spieler eine Reihe füllt, leuchtet diese weiß und kann vom Spieler einzeln per Drag&Drop zu einem Gegner gezogen werden. Dadurch wird der Turm des Gegners nach oben bewegt und eine graue gefüllte Reihe mit einer Lücke wird ganz unten eingefügt. Wird diese graue Reihe gefüllt, verschwindet sie und bewegt alle Quadrate über sie wieder nach unten. Um das Loswerden dieser grauen Reihen zu vereinfachen, ist es sehr wahrscheinlich, dass die Lücken übereinanderliegen. Jedoch können mehrere gefüllte Reihen übereinander in Form von Attacken verschickt werden. Je mehr Reihen es sind, desto stärker sind die Attacken. Diese stiften Chaos bei dem Gegner in Form von kurzzeitigem Kontrollverlust, bis hin zu riesigen Tetrominos in der Warteschlange. Dabei muss der Spieler aufpassen, dass er diese Entscheidungen schnell trifft – denn die eigenen Steine hören nicht auf zu fallen!

2.3. Spielende

Verliert ein Spieler, weil ein platziertes Quadrat über sein 10x20 Spielfeld rausragt, schaut er zu und wartet bis jeder zu Ende gespielt hat. Ist das Spiel beendet, wird der Sieger kurz gewürdigt und die Spieler werden zurück zur „Lobby“ weitergeleitet, bei dem man nach Wunsch entweder das Spiel einfach verlassen kann, indem man das Fenster schließt

(Spielername wird selbstverständlich von der Liste entfernt usw.), oder weiterspielt, indem man im Spiel bleibt während der Spielleiter das nächste Spiel startet. Falls der Spielleiter sein Fenster schließt, wird ein Anderer zum Spielleiter erklärt.



3. Technische Realisierung

3.1. Frameworks

Das Client-Programm liegt in Javascript vor und wird im Browser ausgeführt. Der Javascript-Code wiederum entsteht durch Kompilierung eines Typescript-Programms (siehe https://www.typescriptlang.org/). Als clientseitige Bibliothek wird die Game-Library Phaser.js (siehe https://phaser.io/) verwendet.

Serverseitig wird node (https://nodejs.org/en/) als Laufzeitsystem verwendet, Javascript-Code auszuführen, der – wie im Client – aus Typescript erzeugt wird. Zudem finden die Bibliotheken Express (siehe https://expressjs.com/de/) zur Programmierung der http-Serverfunktionalität und ws (siehe https://github.com/websockets/ws) zur Programmierung des WebSocket-Servers Anwendung.

3.2. Zuständigkeiten von Client und Server

Der Server führt eine Liste von Spielern zusammen mit den entsprechenden WebSocket-Objekten (Nickname, WebSocket-Objekt) zudem verwaltet er alle offenen Spieleinladungen zusammen mit den von ihm vergebenen Codes (Code, Spieler) und alle laufenden Spiele (Spieler1, Spieler 2, Zustand).

Der Server speichert alle Informationen lediglich als Objekte im Arbeitsspeicher.

Der Client stellt abhängig vom Spielzustand den jeweiligen Bildschirm (siehe die Skizzen oben) dar, sendet die Benutzereingaben zum Server und bekommt von dort Bescheid, wie sich der Spielzustand ändern soll.

3.3. Technische Herausforderungen

- Um das Eintippen des Namens oder Codes zu ermöglichen, muss man einen kleinen Umweg in Kauf nehmen. Direkt in Phaser ist dies nicht möglich zu erstellen, jedoch kann man einen sogenannten „Dom“ integrieren, der HTML Objekte über der Phaser Scene platziert. Und in HTML ist mit CSS ein stilisierte Input-Box einwandfrei erstellbar.
- Die Organisation der Messages, die der Server und die Clients verschicken/erhalten gestaltet sich immer schwerer, da man leicht der Überblick verliert, wer was genau erhalten soll.
- Man übersieht schnell, wie viele eigentlich Messages nötig sind um bestimmte Funktionen zu integrieren. Wenn man einen Code eintippt, erhält man nicht direkt die Bestätigung dem Spiel mit diesem Code beizutreten. Wenn man sich zum Beispiel vertippt hat, soll man benachrichtigt werden, dass es dieses Spiel nicht gibt. Oder falls das Spiel bereits voll ist, soll man dies erfahren und nicht beitreten können.
- Die verschiedenen Spielbildschirme werden als einzelne Phaser Scenes realisiert
- Es kann jederzeit passieren, dass ein Client nicht mehr verfügbar ist, weil beispielsweise eine Spielerin/ein Spieler das Spiel bewusst abbricht, der Browser abstürzt, der Rechner keinen Strom mehr hat, …
  Diesen Fall soll der Server bemerken (WebSocket wird geschlossen) und den Clients, die sich mit diesem Spieler in einer gemeinsamen Runde befinden, Bescheid sagen, damit diese z.B. den Namen aus der Mitspielerliste in der Lobby entfernen können

4. Weiteres Vorgehen, Meilensteine

- **Woche 1 – 2:** Umsetzung der Serverstruktur (Code, Freunden beitreten, …)
- **Woche 3 – 4:** Serverstruktur optimieren und für Tetris-Spielzustandsübertragungen vorbereiten (Fusion mit Leos Tetrisspiel)
- **Woche 5 – 6:** Multiplayerelemente in Tetris integrieren (Anzeigen der gegnerischen Spielstände, Versenden der Attacken, …)
- **Woche 7 – 8:** Optimierung und Implementierung aller Attacken und Maximierung des Spielspaßes ☺
- **Woche 9:** Abschluss des gesamten Spiels