

Einleitung und Idee

Bei dem Praktikum für Computergrafik haben wir uns für eine Variante des Videospielklassiker Tetris. Grund dafür war der Vorschlag eines Kommilitonen. Da wir bei einer Suche im Internet eine Repository gefunden haben, welches unserer Idee ähnelte, haben wir uns an diesem Repository orientiert. Aus dieser Quelle stammen auch die Dateien, die wir verwendet haben.

Umfang

Das 3D Spiel umfasst eine grafische und logische Implementation von Tetris. Es ist dabei möglich die Steine entlang der 3 Achsen X, Y, Z zu rotieren, die Steine entlang der horizontalen Achsen zu verschieben, während diese von selbst nach unten Fallen. Mittels der Leertaste lässt sich der Fallprozess beschleunigen. Für die fallenden Steine wird eine semitransparent Vorschau in der Ziellage dargestellt. Für eine bessere Übersicht über das Spielfeld wird an der von der Kamera aus hinteren Wand und am Boden ein Gitternetz dargestellt. Zusätzlich ist es möglich die Kamera um 45 bzw. 90 Grad zu drehen.

Tastenkombinationen

- x – Rotieren um die x Achse
- X – Rotieren um die x Achse (Gegenrichtung)
- y – Rotieren um die y Achse
- Y – Rotieren um die y Achse (Gegenrichtung)
- c – Rotieren um die z Achse
- C – Rotieren um die z Achse (Gegenrichtung)
- p, P – Pause ein/ausschalten
- q, Q – Beenden
- (Pfeiltasten) – Bewegen in die entsprechende Richtung
- (Leertaste) – Stein nach unten setzen
- Linke/ Rechte Maustaste – 90 Grad Kamera drehen (Taste halten für 45 Grad)

Die Entwicklung

Bei der Entwicklung haben wir uns zuerst mit der theoretischen Umsetzung des Spielkonzeptes beschäftigt. Das die tatsächliche Darstellung (Rendering des Spiels) haben wir ans Ende verschoben. Zu Beginn haben wir eine 3D Datenstruktur erstellt welche das Spielfeld repräsentiert. Diese Struktur *Grid.cpp* bietet damit die Verwaltungsschnittstelle für das Spielbrett. Dort werden unter anderem Methoden angeboten um Spielsteine (Tetrominos) hinzu zu fügen, volle Ebenen zu löschen und die Verfügbarkeit von Operationen zu testen. Viele Methoden wurden zu diesem Zeitpunkt noch nicht vollständig ausgearbeitet und nur als Dummys vorbereitet. Um diese Lücken zu schließen haben wir uns als nächstes an die Erstellung der Tetris-Steine gesetzt. Dazu werden in der *ModelGenerator.cpp* die Spielsteine erstellt. Per Zufallszahl kann dann ein Stein aus der Menge der Steine gezogen werden. Die Farbe wird statisch pro Blocktyp vergeben.

Für die Bewegung und Rotation des aktiven Tetromino wird die *TetrominoTransforms.cpp* genutzt. Damit kann ein Stein um den Pivotpunkt rotiert und entlang der Achsen bewegt werden. Auch das "fallen lassen" des Steins wird hier behandelt.

Zum Ende des Projektes haben wir begonnen das Rendering umzusetzen. Dazu wird in der *ViewGrid.cpp* in Abhängigkeit des Kamerawinkels die Darstellung des Gitters berechnet. Es wird zur besseren Übersichtlichkeit des Spielfeldes nur das Gitter auf dem Boden und an der jeweiligen Rückwand angezeigt.

Der Kern, der 3D Darstellung liegt in der *View.cpp* Datei. Dort werden OpenGL und Buffer initialisiert und vorbereitet. Außerdem übernimmt die Klasse das Handling der Tastatureingaben und zeichnet die Buffer der Tetromino, der bereits platzierten Steine und des Gitters.

Bauen & Ausführen

Das Projekt kann mit *make* gebaut werden. Ausgeführt wird es dann mit ausführen der *./Tetris*

Externe Dateien

- *mat.h* – Definition von Matrizen
- *vec.h* – Definition von Vektoren
- *vertex.glsl* – Vertexshader

Quelle: <https://github.com/hnathani/3D-Tetris>,
Zugriff am 4. Januar 2019

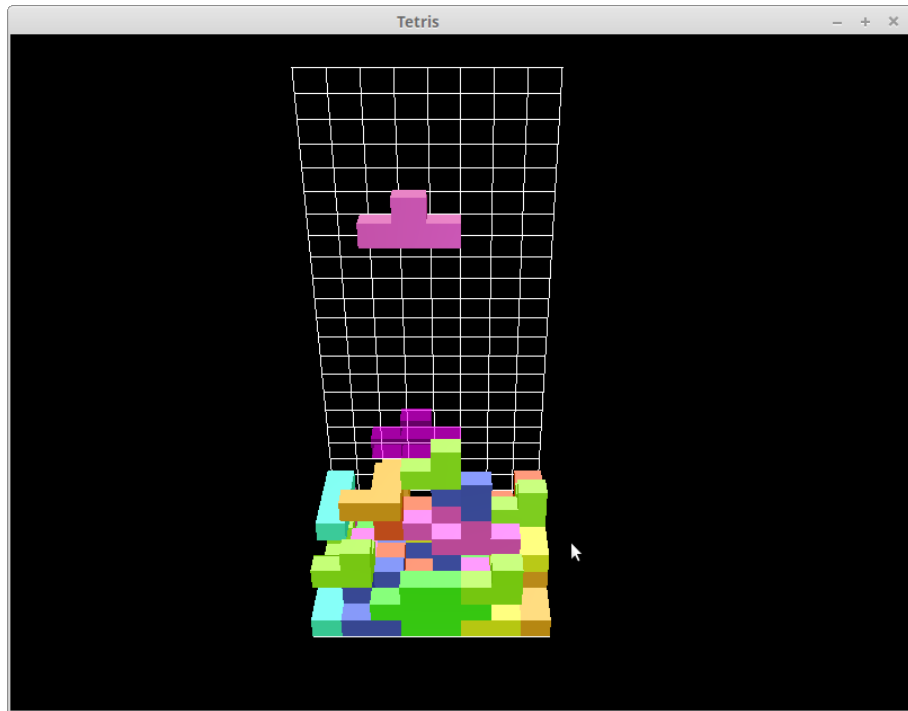


Figure 1: Aktives 3D Tetris Spiel

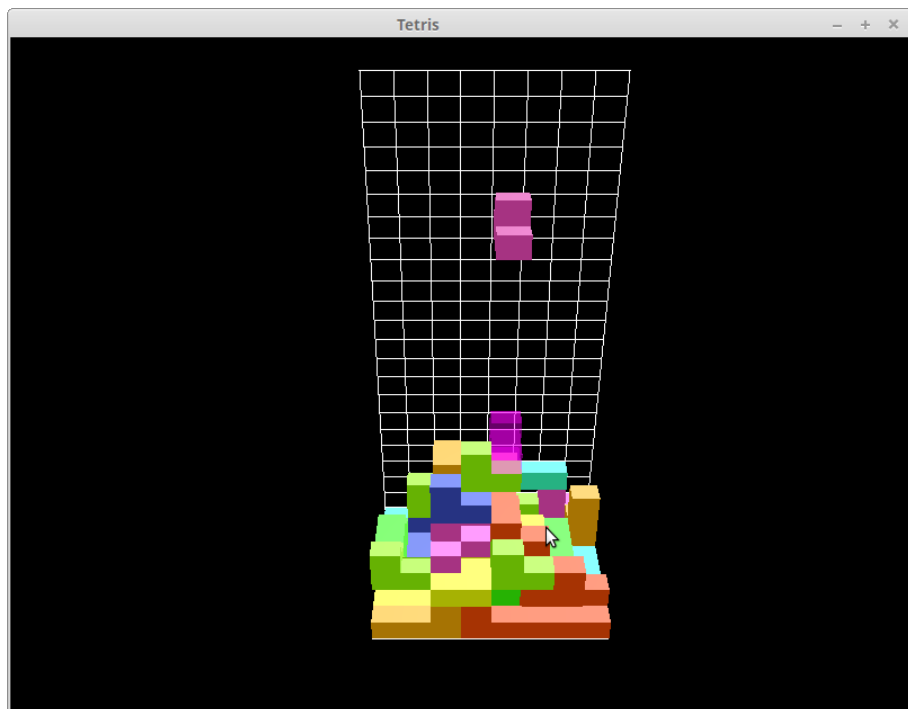


Figure 2: Aktives 3D Tetris Spiel um 90 Grad rotiert

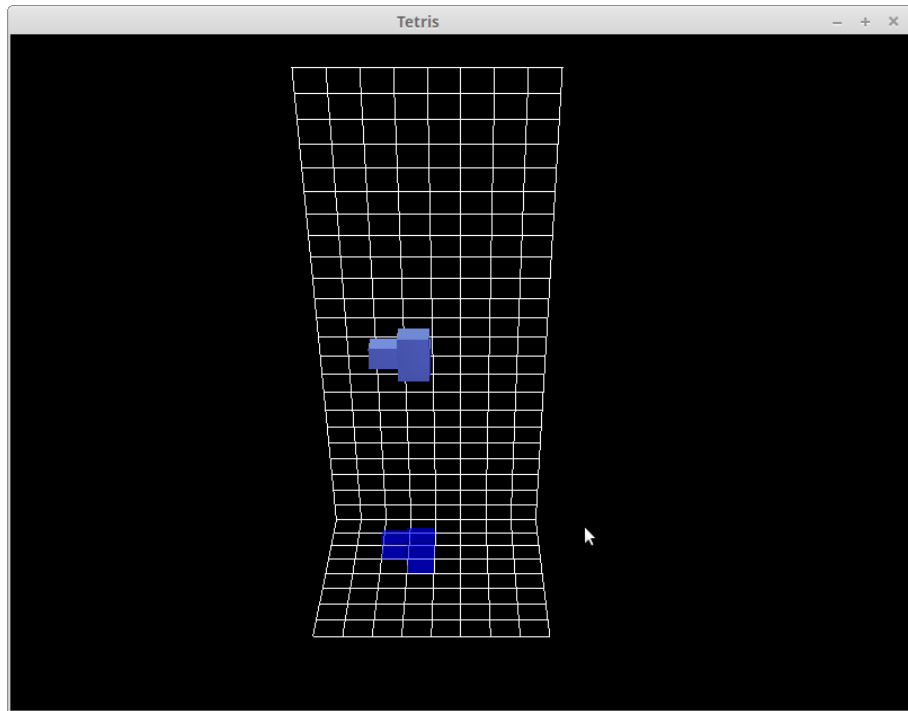


Figure 3: Spiel kurz nach dem Start

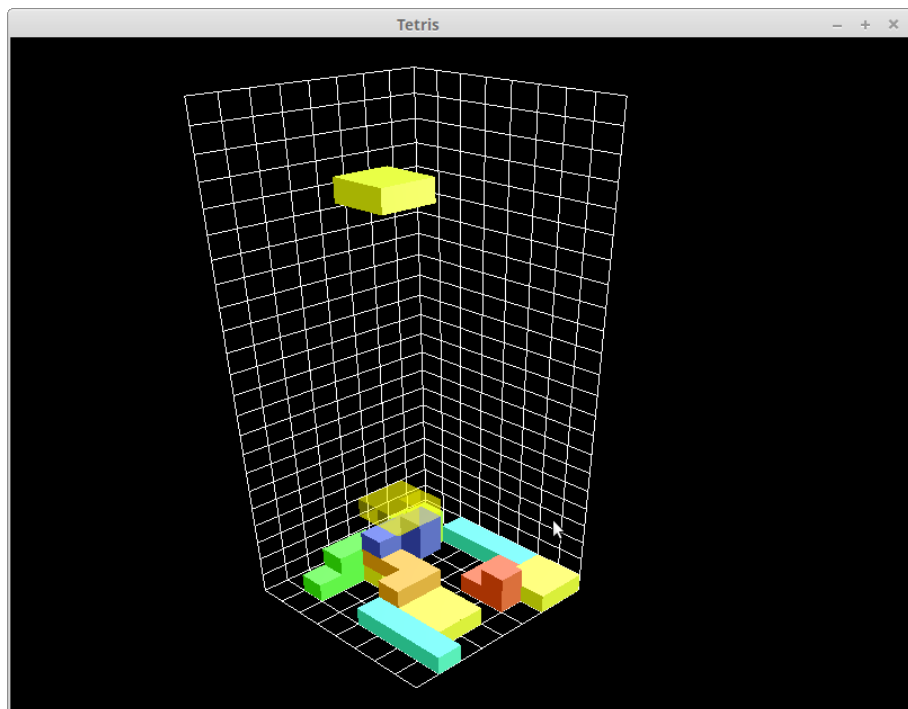


Figure 4: Ansicht aus einem 45 Grad Winkel