Prozedurale Programmierung Projektplanung - “Tetris”

Spezifikationen

Angelehnt an die erste veröffentlichte Version von Tetris:

* Es fallen dauerhaft neue Blöcke, mit 7 verschiedenen Formen, in das Spielfeld hinein.
* Die Blöcke können während des Fallens seitlich bewegt und rotiert werden.
* Bei Drücken der unteren Bewegungstaste wird der Block automatisch unten an der gerade ausgewählten Stelle platziert.
* Ist eine Reihe des Spielfelds komplett durch Blöcke bedeckt so verschwindet diese Reihe (maximal 4 Reihen gleichzeitig möglich).
* Ziel des Spiels ist es, durch das Verschwinden dafür zu sorgen, dass die Blöcke nicht zu hoch gestapelt werden.
* Wird ein Block nicht (komplett) innerhalb des Spielfelds platziert, wird das Spiel beendet. (Game Over)
* Ein Ende des Spieles ist per se nicht zu erreichen, der Fortschritt wird durch ein Score-System dargestellt.
* Das Score System:
* Jede fertiggestellte Reihe bringt Punkte auf den Score.
* Mehrere Reihen, die gleichzeitig fertiggestellt werden, geben einen höheren Wert auf den Score, als wenn diese Reihen einzeln fertiggestellt worden.
* 4 gleichzeitig fertiggestellte Reihen ergeben einen sog. “Tetris” und damit einen größeren Bonus auf den Score.
* Die Standardgeschwindigkeit der fallenden Blöcke ergibt sich durch das aktuelle Level. Hierbei gibt es in der Theorie kein maximales Level.
* Jede fertiggestellte Reihe erhöht den Fortschritt des Levels. (Ein Tetris erhöht den Fortschritt deutlich stärker, dieses System ist unabhängig von dem Score.)

Design

* Die wichtigste Funktion des Programmes ist die `tick()` Methode, welche jeden Tick des Spiels aufgerufen wird, es sei denn es wurde noch nicht gestartet oder bereits beendet. In der Hauptstruktur `struct tetrisGame{};` befinden sich die gesamten Daten zum Spiel, diese reichen von der Map, mit der Zeilen und Spaltenanzahl, dem Scoresystem, das Levelsystem, mit Wert bis zum nächsten Level, sowie dem gerade fallenden Block und dem nächsten Block. Verliert der Spieler, so wird ein “Game Over” - Fenster angezeigt, welches auch den erreichten Score enthält. Aus diesem Fenster kann dann das Spiel beendet werden.
* void tick(); - *Wird jeden Tick aufgerufen. Ruft andere Methoden zum anzeigen und berechnen des Spiels auf.*
* void applyGravity(tetrisGame \*game); *- Wendet die Gravitation auf das momentane Feld an.*
* void checkRows(tetrisGame \*game); - *Checkt ob eine Reihe komplett mit Blöcken bedeckt ist.*
* void generateNext(tetrisGame \*game); *- Generiert den nächsten zu platzierenden Block.*
* struct position { int x; int y; };
* struct block { int type; int orientation; struct position pos; };
* struct tetrisGame { int rows; int columns; char \*map; int score; int level; int linesToNextLevel; struct block falling, struct block next };