Assessment 1 – Matrix Tetris

Taylah Embra

Table of Contents

Overview

Matrix Tetris is a game that allows the user to manipulate matrices represented by blocks, and stack them in a similar way to Tetris, but without the falling and strict placement. Players can move and manipulate a block, and once it is placed a new block will be generated. Each block placed increases the player's score, and if any blocks fall off the platform, the player will lose the game.

The game allows the player to manipulate the block preview by moving it (moving the player), transposing, multiplying by a scalar, multiplying by the next block, adding the next block, multiplying by a custom matrix, reflecting, and rotating.

Background

The game uses 3 by 3 matrices, which translate to 3d objects composed of cubes arranged in a square, and stacked based on the numbers given in the matrix, similar to a 3d Tetris object. Two examples are shown below.

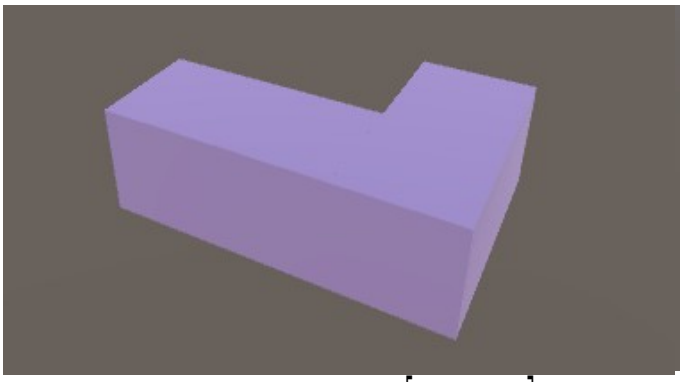

Figure 2: 3D representation of $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

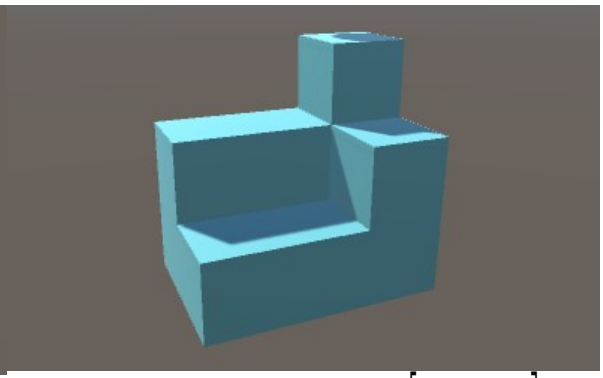

Figure 1: 3D representation of $\begin{bmatrix} 2 & 2 & 3 \\ 1 & 1 & 2 \\ 0 & 0 & 0 \end{bmatrix}$

Height of these objects is capped at 3, creating objects that are at maximum a 3 by 3 cube.

Matrix operations are important in the game as they provide a way to edit the generated blocks to allow the user to fill space. This includes addition, scalar multiplication, matrix multiplication, transpose, reflection, and rotation (of matrix elements, not reflection and rotation matrices).

Matrix addition is as follows, adding each element in each matrix together:

$$\begin{bmatrix} A & B & C \\ D & E & F \\ G & H & I \end{bmatrix} + \begin{bmatrix} J & K & L \\ M & N & O \\ P & Q & R \end{bmatrix} = \begin{bmatrix} A+J & B+K & C+L \\ D+M & E+N & F+O \\ G+P & H+Q & I+R \end{bmatrix}$$

Scalar multiplication is as follows, multiplying each element by the scalar:

$$x \times \begin{bmatrix} A & B & C \\ D & E & F \\ G & H & I \end{bmatrix} = \begin{bmatrix} x \times A & x \times B & x \times C \\ x \times D & x \times E & x \times F \\ x \times G & x \times H & x \times I \end{bmatrix}$$

Matrix multiplication is as follows, taking the dot product of each row by each column and placing them in the <row, col> element of the new matrix.

$$\begin{bmatrix} A & B & C \\ D & E & F \\ G & H & I \end{bmatrix} \times \begin{bmatrix} J & K & L \\ M & N & O \\ P & Q & R \end{bmatrix} = \begin{bmatrix} A \times J + B \times M + C \times P & A \times K + B \times N + C \times Q & A \times L + B \times O + C \times R \\ D \times J + E \times M + F \times P & D \times K + E \times N + F \times Q & D \times L + E \times O + F \times R \\ G \times J + H \times M + I \times P & G \times K + H \times N + I \times Q & G \times L + H \times O + I \times R \end{bmatrix}$$

The transpose of a matrix is as follows, flipping the matrix over its diagonal, or swapping the rows and columns.

$$\begin{bmatrix} A & B & C \\ D & E & F \\ G & H & I \end{bmatrix}^{T} = \begin{bmatrix} A & D & G \\ B & E & H \\ C & F & I \end{bmatrix}$$

Reflection across x and y are as follows, flipping the matrix across either x or y (x corresponding to rows, y to columns).

$$\begin{bmatrix} A & B & C \\ D & E & F \\ G & H & I \end{bmatrix}^{ReY} = \begin{bmatrix} C & B & A \\ F & E & D \\ I & H & G \end{bmatrix}$$

$$\begin{bmatrix} A & B & C \\ D & E & F \\ G & H & I \end{bmatrix}^{ReX} = \begin{bmatrix} G & H & I \\ D & E & F \\ A & B & C \end{bmatrix}$$

Rotation clockwise or anti-clockwise is as follows:

$$\begin{bmatrix} A & B & C \\ D & E & F \\ G & H & I \end{bmatrix}^{RoCW} = \begin{bmatrix} D & A & B \\ G & E & C \\ H & I & F \end{bmatrix}$$

$$\begin{bmatrix} A & B & C \\ D & E & F \\ G & H & I \end{bmatrix}^{RoACW} = \begin{bmatrix} B & C & F \\ A & E & I \\ D & G & H \end{bmatrix}$$

Controls

The game’s movement is controlled by the wasd keys, and the mouse. The mouse controls the view of the player, and wasd keys controls the actual movement of the player. The escape key will pause these controls to allow the user to interact with the user interface. Left click will place the current block.

The player can also use the scroll wheel to bring the block preview closer or further from the player.

Implementation

The game uses matrix and block class to represent the matrices. These blocks can then be manipulated by the player.

Matrix Class

This class contains matrix elements, and functions for matrix operations. Below is the class diagram.

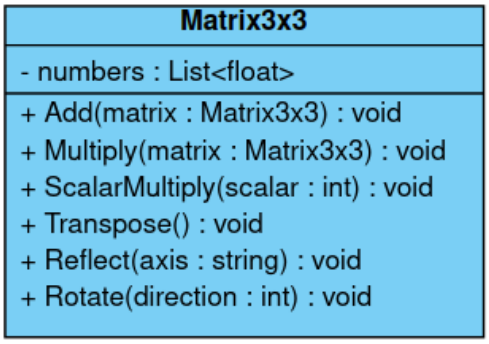

Figure 3: Matrix class diagram

Block Class

This class controls a block object, and has functions to change shape, call the matrix functions for operations, and trigger and react to events. Below is the class diagram.

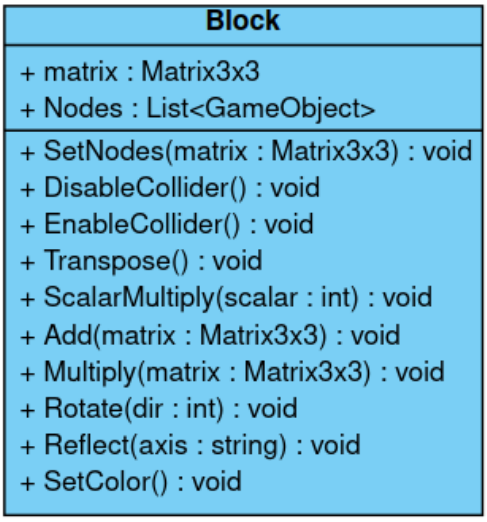

Figure 4: Block class diagram

Demonstration