3D Tetris

Alexander Toews

Introduction

In this project we develop and explore a 3D adaptation of the classic 2D Tetris game. This work is motivated by the fact that 3D Tetris embodies many core concepts from EE267. Manipulating the blocks requires 3D transforms and successful gameplay requires careful consideration of depth cues. This game was developed in Unity/C# with the help of two tutorials[1,2] teaching the basics of game development.

Gameplay

At the start of the game the user sees an empty platform in outer space with a transparent plane hovering above it. Blocks fall down through the transparent plane towards the platform one at a time. The goal is to prevent these shapes from piling up above the transparent plane. When this happens, the game is over. A player's score has two parts: the number of blocks on the platform and the elapsed time when the game ends.

The player can use the keyboard to shift and rotate each block as it falls in order to achieve a compact packing pattern. The D/A and W/S keys translate the block +/- one grid step along the x and z dimensions respectively. Left/right and up/down arrow keys rotate the block 90 degrees about the x and z dimensions respectively. All transformations are performed with respect to world coordinates to ensure an intuitive user experience.

Game Design

Camera and Lighting

The play area is illuminated by an overhead directional light pointing straight down (-y). Shadows created by this light are very helpful for judging the x-z position of a block as it falls. An additional point light source is positioned near the player's head to add further lighting contrast in the scene. The camera view is locked to the player's head. The player can move their head around to get different perspectives on the play area and gather depth information from the resulting parallax.

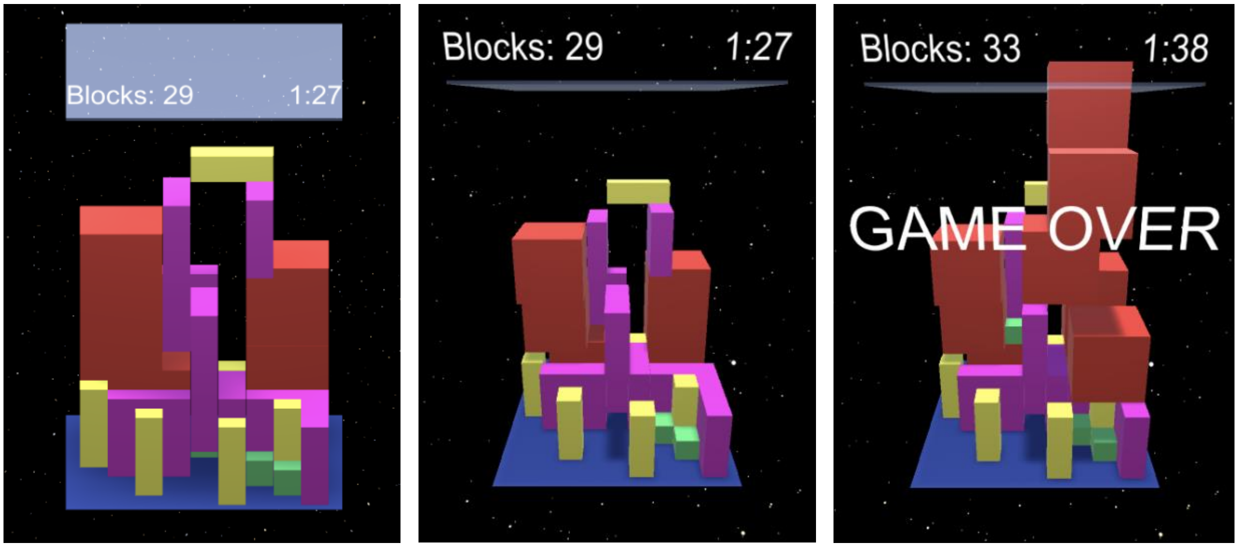
Blocks

Four distinct blocks exist in this game. Each block is a rectangular box with a distinct color from the rest. The shapes were intentionally made simpler than the original Tetris game (which includes "T" and "L" shaped pieces) because the added dimension makes the game considerably more difficult even without such pieces. Each new block instance is randomly chosen from the four possible shapes and initialized with a random orientation and x-z position on a grid.

Block motion is constrained in several ways. Blocks fall with constant speed and freeze immediately when they come into contact with anything. The center of each block is is restricted to the volume between the platform and the ceiling plane. Finally, blocks are not permitted to intersect one another. Any user input which would violate these constraints is ignored.

Discussion

Playing this Tetris game highlights the importance of depth cues for fast and accurate depth perception. For example, I found playing the game on my computer screen with an orthographic projection nearly impossible. It is very difficult to judge the depth of the block in with only occlusion and shadows available as depth cues. Perspective projection makes gameplay quite a bit easier thanks to the depth information provided by both perspective and relative size. Naturally an immersive 3D display provides the most fluent user experience. I used a Windows Mixed Reality head-mounted display for this purpose.



Screenshots of the same game scene in Unity with orthographic (left) and perspective (center) projections. Right: “Game Over” screen.

References

[1] Roll-A-Ball Tutorial https://learn.unity.com/project/roll-a-ball-tutorial
[2] Space Shooter Tutorial. https://learn.unity.com/tutorial/introduction-to-space-shooter#