# Assembly Project: Tetris

Yehyun Lee
Hisham

April 1, 2024

## 1 Instruction and Summary

1. Which milestones were implemented?

   Last Week:
   - Milestones 1, 2, 3 has been completed!

   For the snapshot of progress take a look at our Github page. Please contact yehyun.lee@mail.utoronto.ca for access to private repo.
   https://github.com/YehyunLee/AssemblyTetris

   This week:
   - Milestones 4 and 5 completed!
   - We have 4 easy and 2 hard features:

   Easy features:
   - Gravity
   - Speed of gravity increase gradually
   - Pause
   - Different colours of tetrominoes

   Hard features:
   - Full set of tetrominoes
   - [Extreme Difficulty] Full row, will lead to powerup of clearing all that row. This may happen simultaneously for multiple rows for those who are full!

2. How to view the game:

   (a) Note: tetris.asm is our main code with bug when row clear happen. This bug happen not often, and minor bug. beta.asm is bug free but in beta mode.

   We will explain this in-person!

   (b) Run the tetris.asm file in a MIPS IDE such as Saturn or Mars (try beta.asm too!)

Figure 1: caption

3. Game Summary:

   - Pressing ‘W” would rotate the tetrominoes 90 degrees clockwise. ‘A”, and ‘D” move left or right.

- Pressing ‘S” would decrease the position of incoming tetrominoes by 1 until it collides
- If it collides, a new tetromino is loaded into the screen.
- For this game version, the program is design to go on until program stops or you press ‘Q” to terminate.
- Program can be paused, i.e., pressing ”P”, and resume with repeated ‘P”.
- We have randomized tetrominoes, all 7, being created, with different shapes and colours!
- Gravity is added, and the speed of gravity also increases based on a number of tetrominoes created thus far.
- We have row clear, so when row is full, we clear that row!
- We didn’t add sound so we could play during the lecture. Hope you enjoy :D

```
# Major variables:
# lw $s0 for paint (sw)
# li $s1 for paint counter (need this for general use)
# li $s2 for what TETRO, ex) O, J, T, using int; refer to image.
# li $s3 for what ANGLE ex) 0 is default, 1 is one 90 roration upto 3.
# li $s4 OTetrominoX
# li $s5 OTetrominoY
# lw $s6, ADDR_DSPL
# lw $s7, ADDR_KBRD
# a3 for collision
```

## 2 Attribution Table Last Week

| Yehyun Lee (1008992217) | Aung Zwe Maw (1008604949) |
|---|---|
| [MEDIUM] Coded the background: grid and 3 walls | [HARD] Implemented Original tetromino drawings (as well as colour) |
| [HARD] Designed and Coded Collision Logic | [HARD] Also created every possible rotated tetromino drawing |
| [HARD] Movement W, A, S, D | ←Hisham helped me debugging with CodeTogether until 2AM. |
| [EASY] Coded Keyboard Input (Quit as well) | [MEDIUM] Rotation is consisten at a consistent point of origin |
| [HARD] General Game Loop Logic Flow: Saving Tetrominoes Information, Loading and Handles All Game States | [HARD] Rotation code so screen reloads with rotated Tetromino |

## 3 Attribution Table This Week

| Yehyun Lee (1008992217) | Aung Zwe Maw (1008604949) |
|---|---|
| Fixed colour, rotation, row clear bug | Random Added |
| [Ez] Gravity Added | Worked on rotation base point bug TA mentioned last week, but rollback. |
| [Ez] Gravity Speed Increase: based on Num of Tetromino | Fixed collision bug and also ensured that rotations now work with fixed angle |
| [Ez] Pause added: Hisham then improved the design of pause. | ←Pause design improvement, pause now spells ‘Paused” |
| [Hard] Row Clear: Worked Together. Code didn’t work, thus I spent a long time debugging and improve the code. Big thanks to Hisham for coming up with initial code! This was the most extremely difficult part. Fixed even, odd row bug, repeating bug, detection bug, and shift bugs. | [Hard] Row Clear: ←Hisham came up with initial logic for row clear; both contributed hard work. Hisham added additional grid update work needed for Row Clear. |

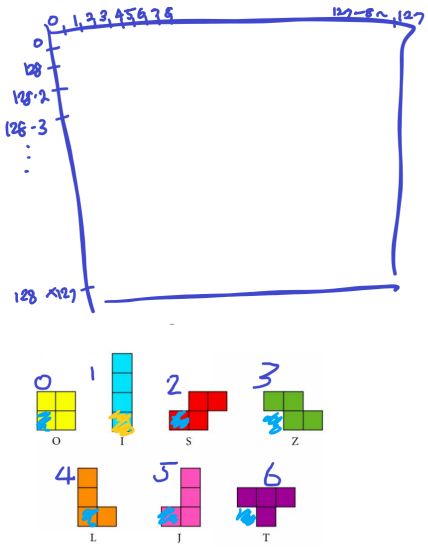


Figure 2: Draft Diagram by Yehyun

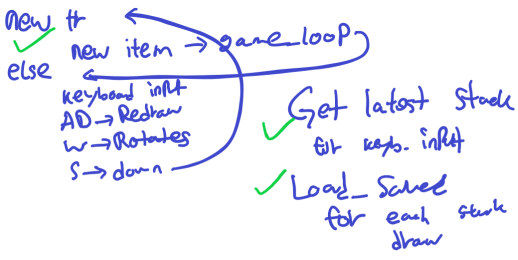


Figure 3: Draft Diagram by Yehyun: Outdated Game Logic


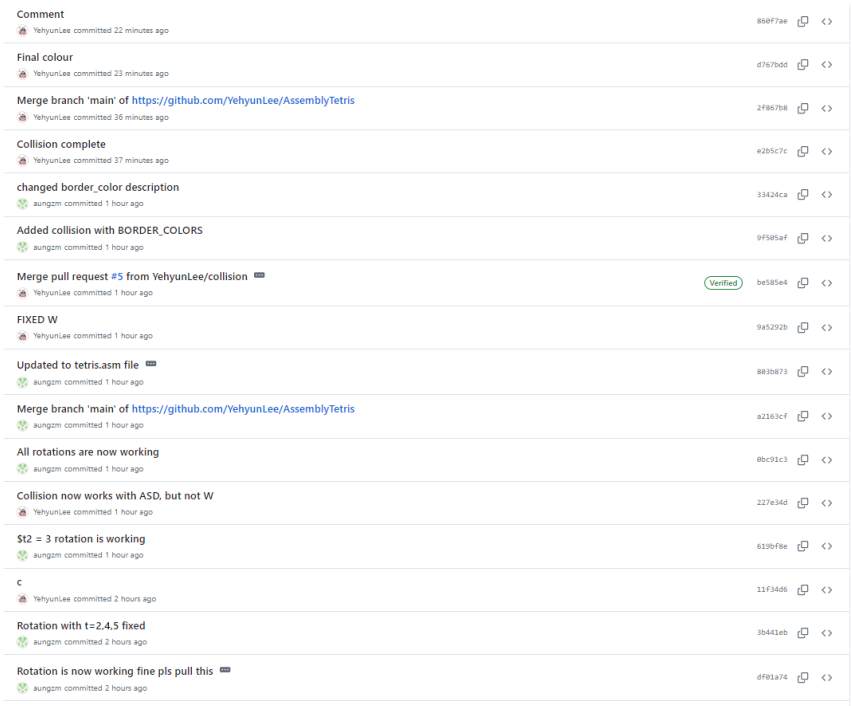


Figure 4: Snapshot of Progress Last Week

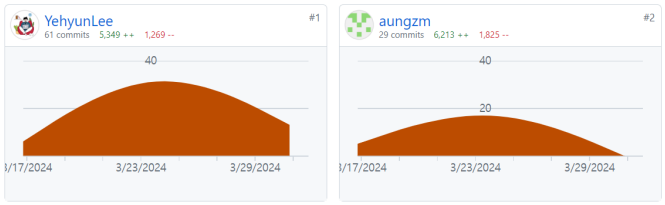


Figure 5: Snapshot of Progress This Week