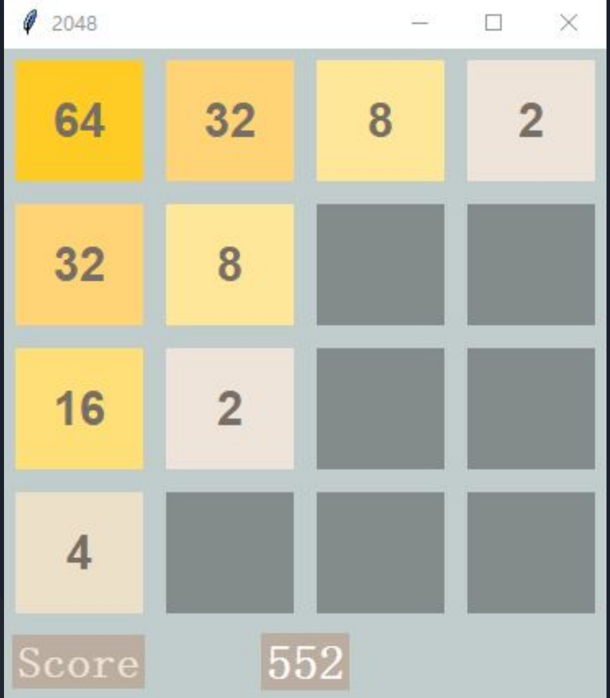
AMS 561 Group Project — 2048



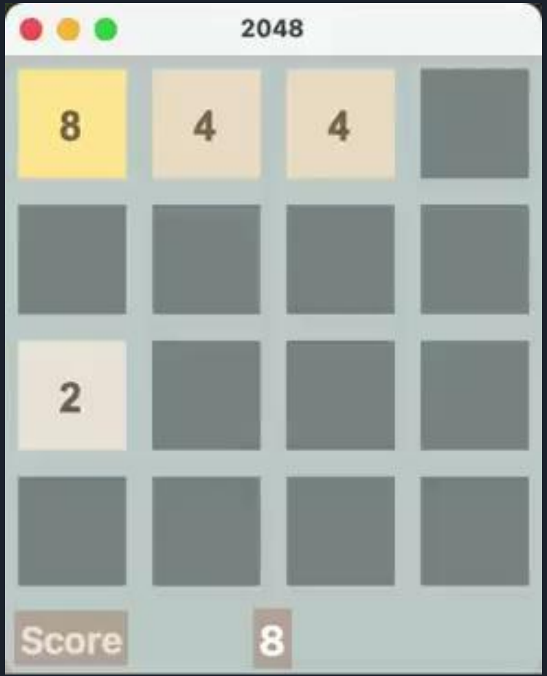
64	32	8	2
32	8		
16	2		
4			

Score 552

Team members: Qiang Wang, Sikun Wang, Yizhen Jia

Overall Idea and significance

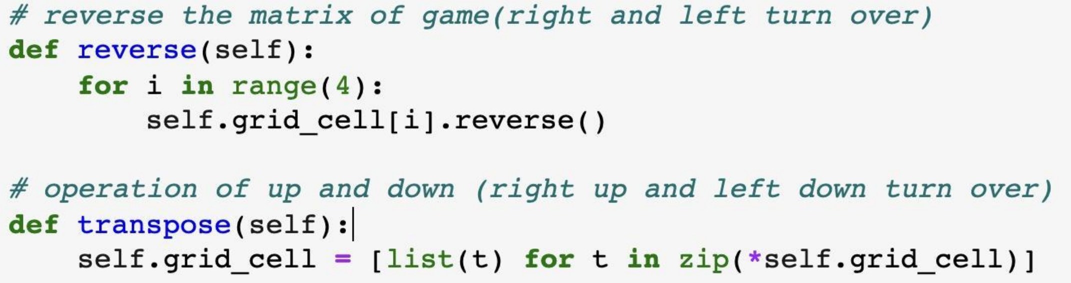
- 4*4 grid;
- Open with random two blocks with 2 or 4;
- Use keyboard to control;
- Blocks are moving in same direction;
- Two same blocks merge to their sum when colliding;
- New block form per operation.
- Victory goal : Get a "2048" block!



8	4	4	
2			

Score 8

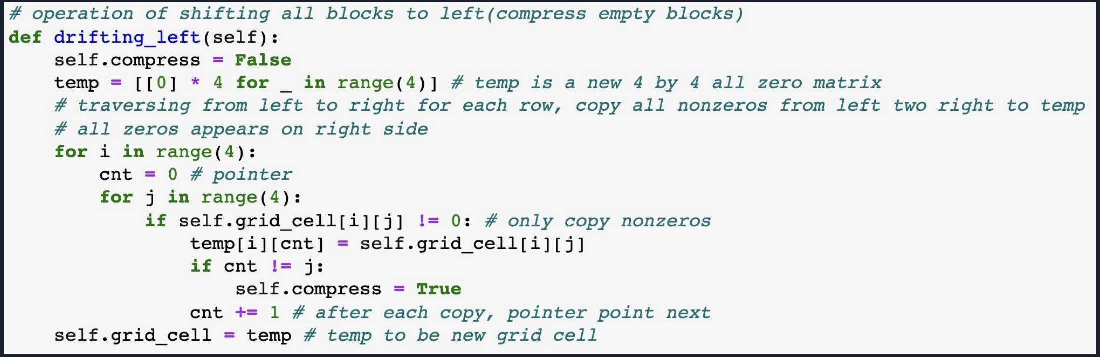
Techniques and Tools used



```
# reverse the matrix of game(right and left turn over)
def reverse(self):
    for i in range(4):
        self.grid_cell[i].reverse()

# operation of up and down (right up and left down turn over)
def transpose(self):
    self.grid_cell = [list(t) for t in zip(*self.grid_cell)]
```

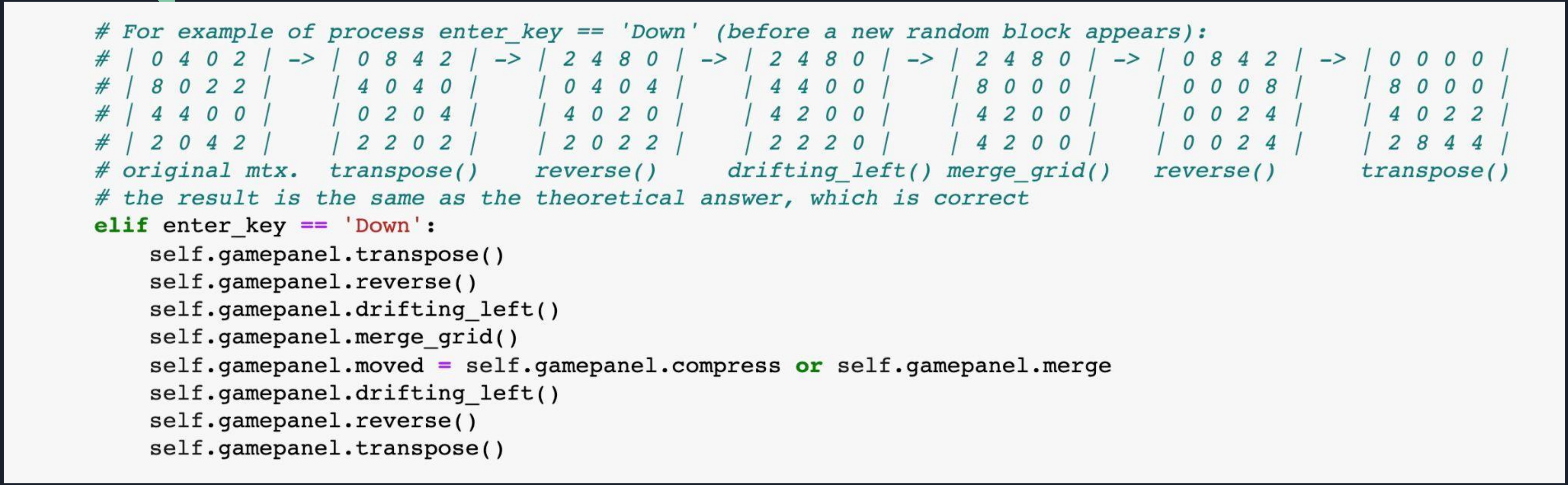
— reverse() and transpose() func



```
# operation of shifting all blocks to left(compress empty blocks)
def drifting_left(self):
    self.compress = False
    temp = [[0] * 4 for _ in range(4)] # temp is a new 4 by 4 all zero matrix
    # traversing from left to right for each row, copy all nonzeros from left two right to temp
    # all zeros appears on right side
    for i in range(4):
        cnt = 0 # pointer
        for j in range(4):
            if self.grid_cell[i][j] != 0: # only copy nonzeros
                temp[i][cnt] = self.grid_cell[i][j]
                if cnt != j:
                    self.compress = True
                cnt += 1 # after each copy, pointer point next
    self.grid_cell = temp # temp to be new grid cell
```

— drifting_left() func

Techniques and Tools used. Cont



```
# For example of process enter_key == 'Down' (before a new random block appears):
# | 0 4 0 2 | -> | 0 8 4 2 | -> | 2 4 8 0 | -> | 2 4 8 0 | -> | 2 4 8 0 | -> | 0 8 4 2 | -> | 0 0 0 0 |
# | 8 0 2 2 |    | 4 0 4 0 |    | 0 4 0 4 |    | 4 4 0 0 |    | 8 0 0 0 |    | 0 0 0 8 |    | 8 0 0 0 |
# | 4 4 0 0 |    | 0 2 0 4 |    | 4 0 2 0 |    | 4 2 0 0 |    | 4 2 0 0 |    | 0 0 2 4 |    | 4 0 2 2 |
# | 2 0 4 2 |    | 2 2 0 2 |    | 2 0 2 2 |    | 2 2 2 0 |    | 4 2 0 0 |    | 0 0 2 4 |    | 2 8 4 4 |
# original mtx.  transpose()    reverse()     drifting_left() merge_grid()   reverse()      transpose()
# the result is the same as the theoretical answer, which is correct
elif enter_key == 'Down':
    self.gamepanel.transpose()
    self.gamepanel.reverse()
    self.gamepanel.drifting_left()
    self.gamepanel.merge_grid()
    self.gamepanel.moved = self.gamepanel.compress or self.gamepanel.merge
    self.gamepanel.drifting_left()
    self.gamepanel.reverse()
    self.gamepanel.transpose()
```

Application of all functions

(Example of “DOWN”)

Challengings and Cool thing

- Original code was too messy, found a new way and connect functions using class to make it better
- Scoreboard algorithm and design
- Operation Sounds

Cool: Similar to the original game, but has our own essence.

Team expertise level and what we learned during the project

- Members of the team have elementary programming skills.
- Improvement of Game Algorithm Design including class, pygame, tkinter.
- Resources Searching
- Communicate and cooperate with each others.