Computer Programming Homework
Tetris Battle

Overview

Tetris is a game where you have to place different bricks to complete lines. The bricks are generated randomly and lines will be eliminated once they appeared. You are able to control one brick at one time. Once a brick descends to the bottom and be placed, another new brick will show from the top. In this homework, you will revise the program to be keyboard-driven using Vpython.

Instructions

You will have three files: game.py, settings.py, and utils.py.

1. Game.py is the core engine of this game.
   The basic requirement is to read lines 56 to 63 to complete the keyboard event.
2. Settings.py contains information of the game, you could change parameters below to have different experience of the game.

```
# The width and height of the game board.
# Game speed.
BOARD_WIDTH = 10
BOARD_HEIGHT = 20
speed_rate = 1
```

3. Utils.py contains useful functions to check the rules.
   The most important code here is collide(), only this function is needed to meet basic requirements.

```
def collide(piece, px, py):
    """ Check if the position(px,py) collides with the board
    @param piece: the shape of the block
                  px: x of the new position
                  py: y of the new position
    @returns: True or False
    """
```

For example, to check if the brick can decline down or not, use the code below.

```
# px and py are position of current brick
if not collide(piece, px, py + 1):
    # (px, py+1) is a valid position
```

Requirements

Complete the following keyboard-driven event to meet the basic requirement.

1. Press ‘up’ to rotate the current brick.
2. Press ‘left’ and ‘right’ to shift the current brick.
3. Press ‘down’ to speed up the current brick.
4. (Plus) Press ‘backspace’ to pause.

It is recommended to add new features to make your own Tetris game and demo it on the class.

Hints

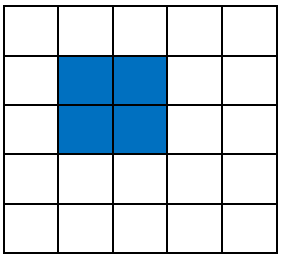
1. One time one brick is under control. The following part will help you render the Vpython objects (In game.py, line 67), so **concentrate on how to change the position of the brick.**

```
# Change the position
for i in xrange(4): focus[i].pos = vector(px, py) + piece[i]
```

   Px and py are the position of the current brick, for example, set py = py + 1 if you want to descend the brick.

2. **Piece** shows the coordinate of every new brick.
   In this case, piece will be

```
[(1, 1), (1, 2), (2, 1), (2, 2)]
```



   For another case:

```
[(1, 0), (1, 1), (1, 2), (2, 2)]
```