Project 4: Tetris Game

Introduction

CS177 - Spring 2016
Project 4 is a group assignment

- You have been divided into groups of 2-3 students
- Project starts on: April 05
- Project is due on: April 26
- Your group has 3 weeks to complete the project
Tetris is a tile matching puzzle game

- Tetris was originally designed and programmed by Russian game designer Alexey Pajitnov.
- It was first released on June 6, 1984.
A demo video of Project 4 is available here...

- [https://www.youtube.com/watch?v=Ts6EaEqlDnk](https://www.youtube.com/watch?v=Ts6EaEqlDnk)

- Let’s watch the program in action (live demo)
Gameplay is controlled using the keyboard

- Left-arrow: Move left
- Right-arrow: Move right
- Up-arrow: Rotate
- Down-arrow: Move down faster
- Space key: Move down very fast
There are lots of Internet resources on Tetris

• If you haven’t seen or played Tetris before:

  http://tetris.com/play-tetris/
Game overview

current_tetrimino.coordinates:
[(220, 60), (220, 100), (220, 140), (220, 180)]

tetris_window.landed_objects:
[((220, 780), (255, 165, 0)), ((260, 780), (255, 165, 0)), ((260, 700), (0, 255, 255)), ((260, 740), (0, 255, 255)), ((300, 740), (0, 255, 255)), ((300, 780), (0, 255, 255)), ((180, 780), (255, 165, 0)), ((20, 780), (0, 255, 0)), ((140, 740), (255, 165, 0)), ((140, 780), (255, 165, 0)), ((100, 780), (255, 165, 0)), ((20, 740), (0, 255, 0)), ((60, 740), (0, 255, 0)), ((100, 740), (0, 255, 0)), ((60, 780), (0, 255, 0)), ((420, 740), (0, 255, 0)), ((420, 780), (0, 255, 0)), ((380, 780), (0, 255, 0))]

NUM_BLOCKS_X = 11

NUM_BLOCKS_Y = 20

BLOCK_SIZE = 40
There are 7 ‘tetrimino’ shapes

- Tetrimino_I
- Tetrimino_J
- Tetrimino_L
- Tetrimino_O
- Tetrimino_S
- Tetrimino_T
- Tetrimino_Z

- All tetriminos are composed of 4 squares
Components of the game
You will create 5 tetriminos

- Tetrimino_I and Tetrimino_J are provided in the Project 4 code skeleton
- You will generate the coordinates and create the other Tetriminos using the window size and block size.
Initializing Tetrimino_L
Initializing Tetrimino_O
Initializing Tetrimino_S
Initializing Tetrimino_T
Initializing Tetrimino_Z
Rotating a tetrimino changes its coordinates and ‘state’

- Pressing the up-arrow key will rotate the tetriminos.
- Depending on a tetrimino’s shape, it will have 1, 2 or 4 states reflecting different rotated positions.
Rotating Tetrimino_I

State = 0

State = 1
Rotating Tetrimino_J

State = 0

State = 1

State = 2

State = 3
Rotating Tetrimino_S

State = 0

State = 1
Rotating Tetrimino_T

State = 0
State = 1
State = 2
State = 3
The left-arrow key moves a tetrimino to the left

- When the left-arrow is pressed, we should update the tetrimino’s coordinates one grid to the left.
- If the movement would make the tetrimino go outside of the window, the coordinates should remain unchanged.
Moving tetrimino left

Before

After
Moving tetrimino left

Before

After
The right-arrow moves a tetrmino to the right

- When the right-arrow is pressed, we should update the tetrmino’s coordinates one grid to the right.

- If the movement would make the tetrmino go outside of the window, the coordinates should remain unchanged.
Moving tetrimino right

Before

After
Moving tetrimino right

Before

After
Tetriminos will constantly move down the grid

- Pressing the down-arrow key will make the tetrimino move even faster.

- We should update the tetrimino coordinates.

- If the tetrimino reaches the bottom of the grid or touches a ‘landed’ piece, it should stop moving.
Moving tetrimino down

Before

After
Moving tetrmino down

Before

After
A tetrimino stops moving when it cannot go further

- We need to know whether a Tetrimino finished its movement.
- When a tetrimino has landed, update the ‘landed’ objects and create a new tetrimino
Checking a tetrimino has landed

has_landed = False

has_landed = True
Checking a tetrimino has landed

has_landed = False

has_landed = True
Pressing the Space key forces a tetrimino to land

- Tetrimino moves very fast and lands.
- When a tetrimino has landed, update the ‘landed’ objects and create a new tetrimino.
Land a tetrimino

Before

After
After landing, update the landed objects

- After a tetrmino has landed, we need to update the rows of landed objects.
- If a row is full all the way across, remove it.
- When a full row is removed, the row(s) above it should move down.
Updating landed objects

This line will be removed.
Removing ‘full’ rows

Before

After

This line will be removed.
Updating landed objects

These lines will be removed.
Updating landed objects

Before

After

This line will be removed.
Updating landed objects

These lines will be removed.
At the start of the game, draw the grid

- This is essentially drawing a bunch of lines.
- Three variables define the grid’s appearance
  - BLOCK_SIZE
  - NUM_BLOCKS_X
  - NUM_BLOCKS_Y
Drawing the grid

NUM_BLOCKS_X = 11
NUM_BLOCKS_Y = 20

NUM_BLOCKS_X = 9
NUM_BLOCKS_Y = 15
There are three Python classes used in Project 4

**Tetrmino**
Variables:
- state (Integer)
- coordinates (List of (x,y))
- tetris_window

Functions:
- get_color() (Given.)
- move_down()
- move_left()
- move_right()
- has_landed()
- land()
- draw_tetrmino()

**TetrismoFactory**
Functions:
- get_random_object()

**Tetris_Window**
Variables:
- block_size (Integer)
- num_blocks_x (Integer)
- num_blocks_y (Integer)
- width (Integer)
- height (Integer)
- surface (Pygame window)
- landed_objects (List of ((x,y),(r,g,b))

Functions:
- all_inside()
- has_collision_with_landed_objects()
- add_tetrimo_to_landed_objects()
- update_landed_objects()
- is_game_over()
- draw_grid()
- draw_landed_objects()
Your grade is based on completing 24 tasks

• The project will be graded out of 200 points.

• Tasks are grouped together based on functionality
Your project groups are pre-assigned

- You can find your group’s email addresses in the Project 4 assignment on the course page.

- You are **not allowed** to change your group.

- If you have any questions or concerns regarding groups, please send an e-mail to:

  Sudharshan Viswanathan, viswans@purdue.edu
Get started with your group right away

- Contact your group members immediately, you have until Friday 8\textsuperscript{th} to report any issue (e.g., can’t get hold of a member, etc.)

- Every student in the group will receive the same score.

- \textbf{Suggestion:} Try to divide the tasks according to the difficulty of the task. (Not each task has the same difficulty.)
Your group can earn up to 2% extra course credit

- This part is **NOT mandatory**, and worth up to 2% overall course grade.

- Finish all 24 original tasks before attempting any extra credit, then choose the extra credit components you want to attempt

- For extra credit, you can rewrite any function as you wish and add new functions

  - *Remember – for the 24 original tasks you cannot modify the original program skeleton*
Your extra credit program is *separate* from Project 4.

- **DO NOT** submit your Extra Credit program via Blackboard.

- If you want to do the Extra Credit part, you are still expected to submit the other tasks (Task 0-24) before the deadline via Blackboard system. The Extra Credit part will be graded separately.

- You will *present* your extra credit and every team member should be familiar with the extra credit version of the program.

- All the team members needs to be present for the extra credit presentation. Every team member will get the same extra-credit.

- The grade of the team will be decided based on the **complexity of the program** and the **familiarity** of each group member with the code.
There are many options for extra credit

- **Show the current score of the game.** Scoring should be as follows:
  - If one line is removed: Give 10 points
  - If two lines are removed: Give 30 points
  - If three lines are removed: Give 50 points
  - If four lines are removed: Give 100 points

- **Introduce levels in the game.**
  - The game starts at Level 1. After 50 tetriminos has landed, you should increase the level by one.
  - When level increases, the speed of the tetriminos should get faster. (Hence, the game gets harder.)
There are many options for extra credit

- After the game is over, ask their name to add the name into the “Best Scores” table.

- You should keep the “Best Scores” table in a text file with their corresponding names and scores.

- Add sound effects when a new tetrino is generated, a row gets removed, and when game is over.
  
  - Use different sounds when one line is removed, two lines are removed, three lines are removed and four lines are removed.

- Show the landing position of the tetrino before it lands to the user. (Refer to this game: http://tetris.com/play-tetris/)
There are many options for extra credit

- In our version of the Tetris game, when we try to rotate a tetrimino object, sometimes it cannot because it would go out of the window since `tetris_window.all_inside()` returns False.

- However, when we move the tetrimino left or right and then rotate it, it rotates. Fix this for extra credit.
Questions?