CSE 60166 – Final Project Report (Fall 2014)

Sandipan Banerjee

Main Idea: The main idea of this project is to implement a simple 3D tetris game (brick game) with effects and sounds using python and OpenGL. There are 10 distinct textured brick patterns, whose orientation can be manipulated by the user in Y and Z directions. The pattern is generated randomly for the incoming brick; its texture also changes once it hits the floor. A scoring scheme has also been introduced where each stationary brick not hitting the roof adds 1 point to the score. Multiple views have also been implemented for user convenience. The game is lost once a brick hits the roof; it can be reset at any time. The only thing remaining, however, is the vanishing of the bricks (bonus score) when all the positions at a level on the floor is completed.

Main functions implemented:-

1. Drawing of the main 3D single brick
2. Collating such single bricks to make different brick shapes
3. Animating the bricks, i.e. making them come down from the roof to the floor
4. Wrapping texture around the incoming brick
5. Controlling the velocity of the brick by the user
6. Manipulating the movement of the brick in both X and Z direction by the user
7. Rotation of the brick in Y and Z direction by the user
8. Putting the bricks in their proper positions and keeping them stationary once they hit the floor
9. Changing brick texture once it hits the floor

10. Calculating the Y limit of the incoming brick based on the position of the bricks below, so that no two bricks (textures) overlap each other
11. Generating sounds and effects for each different scenario
12. Generating multiple views (3 different views) based on the users choice
13. A scoring scheme which adds 1 point for each brick becoming stationary without hitting the roof
14. A resetting mechanism handled by the user
15. A game over mechanism which ends the game once a stationary brick touches the roof
16. Taking screenshots of the current frame
17. Resizing window as per the users choice
18. A quitting function as per the user's choice as well

<u>Steps to run FinalProject.py</u>:-

1. Install 'Pygame' which can be found in 'SandipanBanerjee_FinalProject' folder
2. Open the command prompt window (cmd)
3. Go to the project directory location (cd ../SandipanBanerjee_FinalProject)
4. Type 'python FinalProject.py' and hit enter
5. Press 'a' or 'A' to move incoming block left
6. Press 'd' or 'D' to move incoming block right
7. Press 'w' or 'W' to move block nearer to the viewer
8. Press 's' or 'S' to move block further away from the viewer
9. Press 'n' or 'N' to rotate the incoming block in Z direction
10. Press 'm' or 'M' to rotate the incoming block in Y direction
11. Press '+' to increase block velocity

12. Press '-' to decrease block velocity (but not make it zero)

13. Press 'v' or 'V' to change view

14. Press 'r' or 'R' to reset the game at any time

15. Press 'f' or 'F' to save current frame as a ppm file

16. Press 'q' or 'Q' key to quit

<u>Technical challenges faced</u>:-

1. Making the different shapes with the single block. I made 10 different shapes with 1, 2, 3 and 4 such single blocks adding them together into different shapes.
2. The hardest part about the project for me was to calculate the exact limit or height the incoming block can traverse downwards so that it doesn't go inside another block on the floor. I made a list of all the blocks on the floor, calculated the permissible height for each block based on their position and shape, checked if the incoming block was on top of it and then changed its limit accordingly. I literally spent 6-7 hours fixing that.
3. The other problem was to learn how to generate multiple views at different angles for the user's convenience. There were very few Python examples on the internet but luckily I understood them and made multiple views (straight, sideways and looking down). I used glViewport( ) parameters to partition the main screen into different parts and manipulated the view with the gluLookAt( ) function.
4. The only thing remaining is the vanishing or breaking of bricks once all the positions at a level on the floor is filled i.e. bonus scoring. I have an idea on how to do it using a grid structure, and hope to implement it before the final submission on the 17th.

Things I learned from the project:-

1. Making different structures (different brick patterns) from one simple 3D structure (single brick)
2. Parameterizing objects based on other objects when calculating the object parameters to change the parameters of other objects
3. Generating multiple views, splitting the viewport using glViewport( ) and gluLookAt( ) functions
4. Calibrating the frustum to get the desired orientation

Things I plan to implement before the final submission: Although I am very happy with the work done for the tetris game at present, I hope to include two more things before the final submission.

1. Add a grid structure to make bricks vanish once all the positions at a level on the floor is filled
2. Show relevant text like score and dialogues on the screen rather than the cmd window

Thanks for going through this report. Hope it works.