Extra Features

1. Next Block
A Next Block indicator was added to the right of the screen, and labeled so. The shape rotates about the z axis only, as to not confuse the player by flipping horizontally and appearing as a different piece. This feature is on by default, and can be toggled via the Display – Show Next menu item. This feature required a little change to game.cpp/game.h to allow the return of the next piece id, and a copy of its definition.

2. Levels
Levels have been added to extend game play. There are 10 levels, labeled 0 through 9, and the only change is that the new playable pieces descend the well at a faster rate. The starting level is initially 0. This can be changed via the Level menu, and choosing the desired level. This does not immediately change the level however, a new game must be started.

There is a secondary option that can be toggled below the level selection menu options, called Incremental Levels. If this option is enabled, the current level will increment by one every 10 lines cleared (or, if starting at a level other than 0, by clearing 10 * (level + 1) lines, and then 10 thereafter). This option is not enabled by default.

This feature required several changes to game.cpp/game.h, as the scores, levels, and line information is stored here.

3. Score
A scoring system has been introduced, similar to a legally distinct game. At level 0, the score for a single line is 40 points, double line is 100, triple line is 300 and quadruple is 1200. For levels other than 0, it is the same, but the scores are multiplied by (level + 1).

4. Status
The Player's status in the game is now recorded and displayed at the top of the game screen. This includes indicators for the current Level, Score and Lines cleared. This feature is on by default, and can be toggled via the Display – Show Text menu item.

5. Active Piece
This feature causes the current playable piece to pulse white in a smooth fashion. This feature uses colour interpolation between the source colour (the normal colour of the block) and white. This option is normally off by default, but can be enabled via the Display – Show Active menu item. This required a small change in game.cpp/game.h to allow the active piece to be identified uniquely on the board.

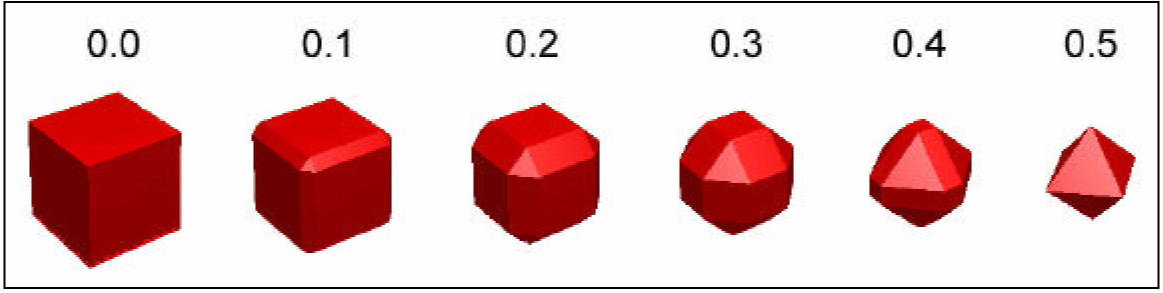
6. Pause

Pause allows the game and animations to be stopped, but rotation, scaling and persistence/gravity effects to continue, and be un-paused. Press P or select the Application – Pause menu item.

7. Beveled Cubes

Be default, the blocks are formed from unit cubes. By selecting one of the options from the Filet menu, the cubes can be given a more pleasing appearance. The first menu item is marked is no bevel, and is the cube. The most extreme beveling is 0.5, which results in an octahedron. The most pleasing choice is 0.2, marked as 'Good'

Below is an illustration of the various filet values, and the blocks it generates:



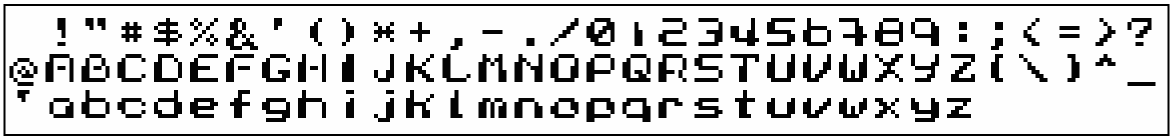
When no beveling is selected, the geometry is a simple cube, any other option uses a parameterized beveled cube construct.

8. Well Effects

The well's blocks rotated on the y axis, slightly out of phase with each other, in one column, and the same phase but opposite direction, in the other column. The bottom row of cubes rotate on the x axis in a similar fashion as well. When Lighting is enabled, and the view reset to normal, this provides an interesting reflection effect. Further, the higher the level, the faster the rotation and the more frantic the mood of the game.

9. Text

Text is rendered out of blocks, and the font is defined via a byte array inside the program (to avoid the need for the only outside file). The 8x8 font is defined as 1024 bytes, to represent the first 128 characters of the ASCII code. Each character is represented by 8 bytes, one for each row, and form a bitmap of each character, which are easily plotted with cubes. Here is the font set:

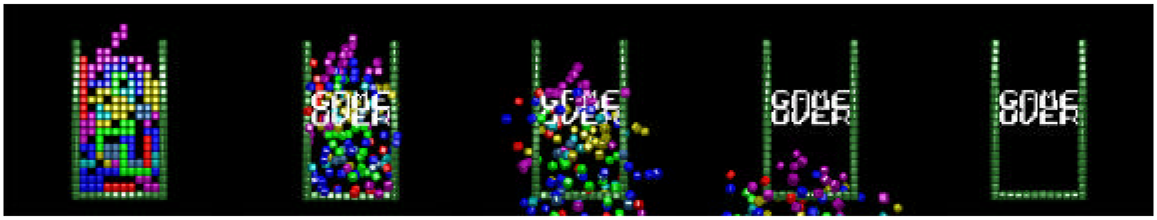


```
 !"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_
`abcdefghijklmnopqrstuvwxyz
```

10. Losing the Game

When the game is lost due to the well overfilling, the individual cubes blow out from the well in random directions in the XZ plane, and fall with slow gravity downwards. At a specified point in the Y plane, they vanish. A game over message appears, and the game awaits for the user to quit or start a new game.

Below are a few frames from a loss animation:



Changes/Notes:

Game Change

game.cpp was altered to prevent the moving of the last active piece when the game has been lost.

Also, a fix was considered that would have prevented the user from moving a piece that had been ‘dropped’ until the tick had elapsed, but this could have been considered purposeful and there was no definitive choice that could be made given the specification, so it was left unchanged.

Rotation

The rotational angle for any axis for the world-rotation specification increases as the mouse is moved to the right. Also, rotation and scaling at the same time is not done (rotation is done alone), however, if there is a persistent rotation (gravity) active, scaling can be done, as it does not interfere with the rotation at this point.

Multicolour mode

When the program is invoked, a colour lookup table is created and initialized with 44 random colour values, to apply to the faces of the beveled cube. When the filet is 0 (cube mode is selected), then only the first 12 entries of this table are used. Having multicolor mode on and switching between beveled cubes and normal cubes may cause the colours to appear to jump.