May 7, 2018

EE 356/CS 376
Project 03 – Hour Glass Simulation **Due: October 1, 2018**

For this project you will use automata to simulate sand flowing through an hour glass. When the project opens the user will see only the hour glass with all of the sand at the bottom. When the user hits the space bar, buttons will appear to turn the glass over and begin the flow of sand. Also provide a button for reset and stop. (All buttons should only be visible for 10 seconds after the user hits the space bar.)

The size of a grain of sand is up to you. You can make it as small as on pixel or as large as an 8-pixel (diameter) circle.

Simulation rules

1. Sand can only fall downward – never up or sideways. It can only fall into empty locations. In Figure 1 only positions 1, 2, or 3 or possible and only if those positions are unoccupied.

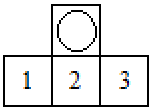

Figure 1
Falling sand grains.

2. If space 2 is open the sand grain falls downward into it.
3. If space 2 is not open but spaces 1 and 3 are both open there is a 50% probability that it will fall into either 1 or 3.
4. If space 2 is not open but one of 1 or 3 is open it will fall into the open space.
5. If spaces 1, 2, and 3 are occupied the grain of sand does not move.

To do this project you should use a bitmap and two 2D arrays where each array location represents one grain of sand. One of the arrays is for the present generation and one for the next generation. Here is one algorithm of how this might work.

1. Give Gen 0 an initial configuration. You could store a 0 for nothing there, a 1 for a grain of sand, and a 2 for a wall or edge.
2. Translate the Gen 0 array to the bitmap and assign it to an image.
3. Use Gen 0 to produce Gen 1. Since Gen 1 will be changing but Gen 0 will not you can do this with multiple threads assigned to various portions of the array.
4. When all of the threads complete their work translate the Gen 1 array to the bitmap and assign it to the image.
5. Repeat this process but this time use the Gen 1 array to create the Gen 0 array.

Alternatively, you could use a *writeable bit map*. See the notes for how to do this. There is a *GetPixel* and a *SetPixel* function in the WriteableBitMap class so you don't need an extra memory array to hold the data.

Minimum Requirements:

1. Your project must use C# WPF.
2. You must use threading (preferably a background worker) to do the calculations on the grains of sand.
3. Grains of sand can be as small as one pixel or as large as 8 pixels in diameter.
4. You will need to use a timer to display each iteration for a few milliseconds. The timer should be set to give a realistic display.

Extras:

1. Make the sand change color as it passes through the neck of the hour glass.
2. Create a *run forever* mode in which the hour glass automatically rotate 180^{o} when all of the sand is gone from one end. Exit when the user hits the space bar.
3. Do the rotation in part 2 as animation.
4. Allow the user to set the size of the hour glass and/or the number of grains of sand.
5. Allow the user to adjust the time between iterations.

Turn in the following electronically to your instructor

1. A Word document titled `EE356P3XXX.docx` (where xxx is your three initials) which contains:
    - Cover Sheet – The cover sheet should include your name, course number, project number, project title, and date handed in.
    - A short description of your project. Include an overview and any special features that you added that are not in the specifications.
2. A complete code file that can be executed. Your code file must have well documented source code.

Compress these two items into a zip file should have the title `EE356P3XXX.zip` where `XXX` are your three initials and email them to your instructor.