

Manual Testing for MainPanel.runContinuous()

Test Case 1: Ensure in 15*15 panel, the status of cells will become that in Figure 2 after clicking “Run Continuous” button with the initial status of cells in Figure 1.

Preconditions:

The game is started with the modified runContinuous() method. The game runs with argument 15. A 15*15 world is presented.

Execution Steps:

1. Click cells on the panel which will look the same as the one in Figure 1.

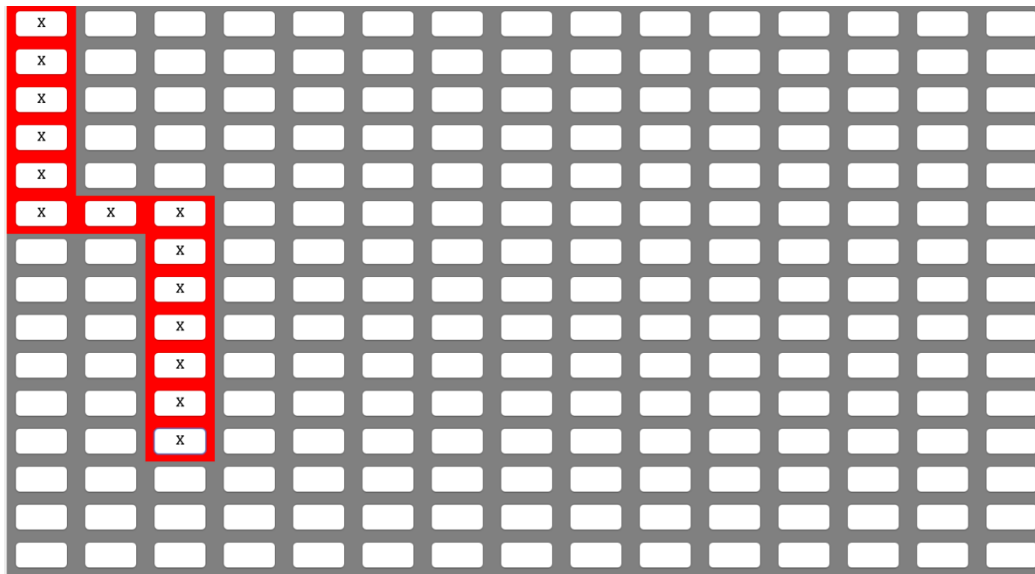


Figure 1

2. Click “Run Continuous” button.
3. Wait until status of cells doesn’t change any more, and then click “Stop” button.

Postconditions:

The status of cells on the panel is exactly the same as those when the program runs with original runContinuous() method which is shown in Figure 2.

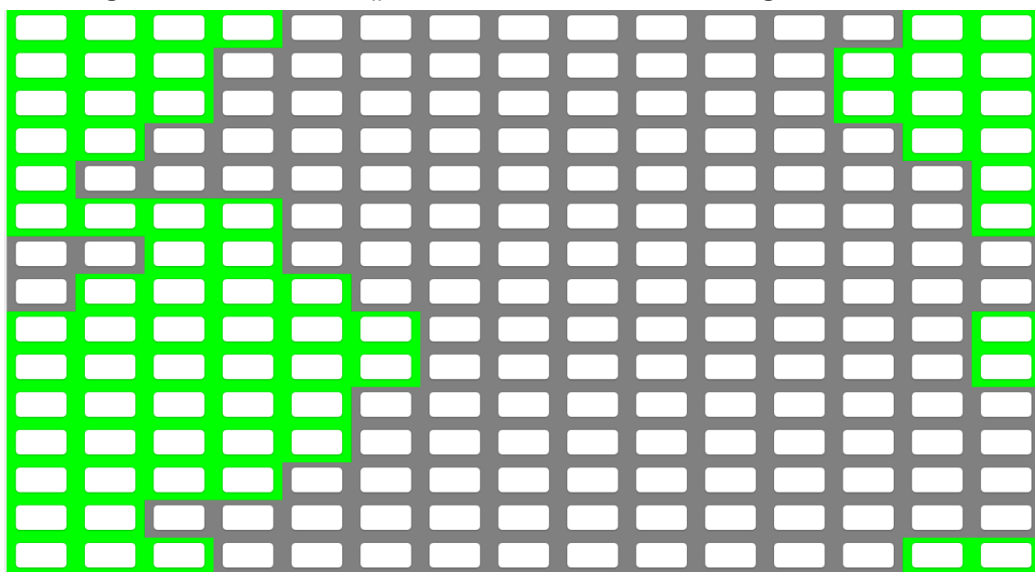


Figure 2

Test Case 2: Ensure in 15*15 panel, the status of cells will become that in Figure 3 after clicking “Run Continuous” button without initializing status of any cells.

Preconditions:

The game is started with the modified runContinuous() method. The game runs with argument 15. A 15*15 world is presented.

Execution Steps:

1. Click “Run Continuous” button.
2. Click “Stop” button after 10 seconds.

Postconditions:

The status of cells on the panel is exactly the same as those when the program runs with original runContinuous() method which is shown in Figure 3.

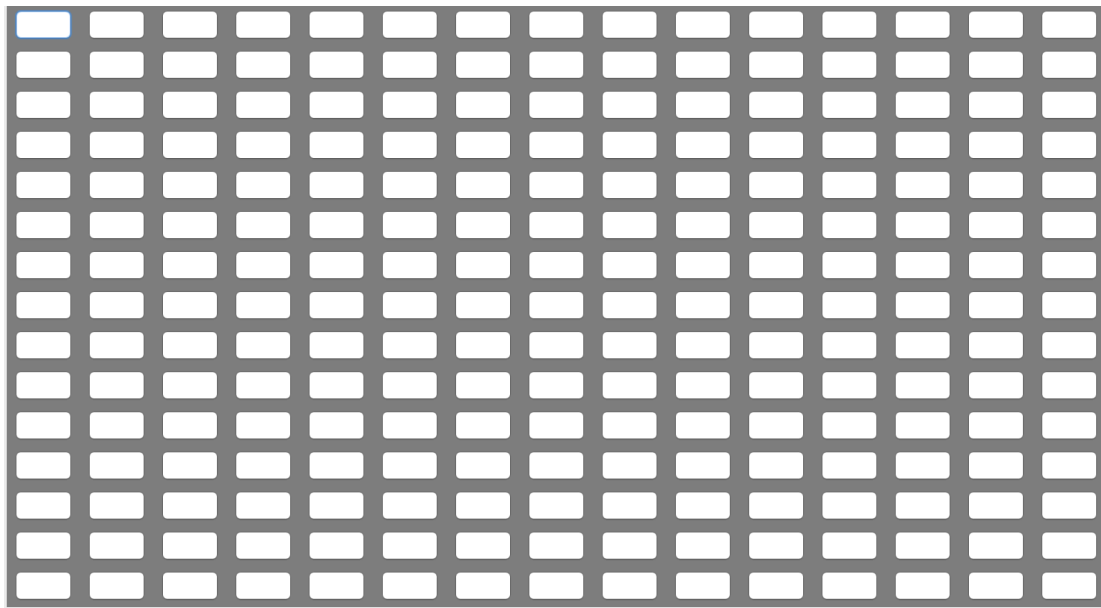


Figure 3

Case3: Ensure in 15*15 panel, the status of cells will become that in Figure 3 after clicking “Run Continuous” button without initializing status of any cells.

Preconditions:

The game is started with the modified runContinuous() method. The game runs with argument 15. A 15*15 world is presented.

Execution Steps:

1. Click all cells to make all cells dead which looks the same as Figure 4.

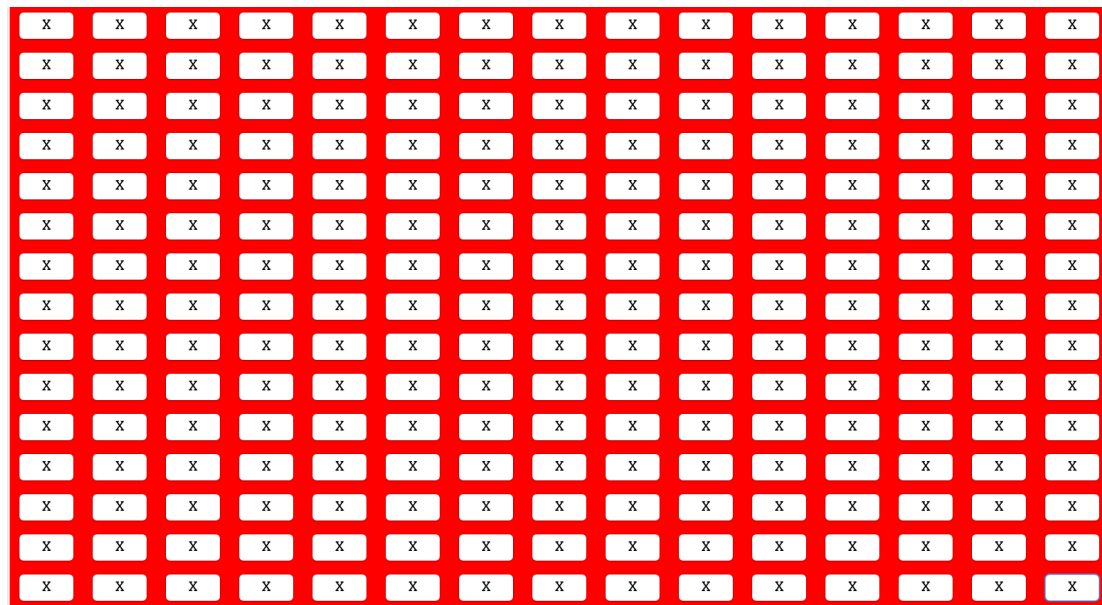


Figure 4

2. Click “Run Continuous” button.

3. Wait until status of cells doesn't change any more, and then click “Stop” button.

Postconditions:

The status of cells are all alive on the panel which is exactly the same as the result shown in Figure 5 when the program runs with original runContinuous() method.

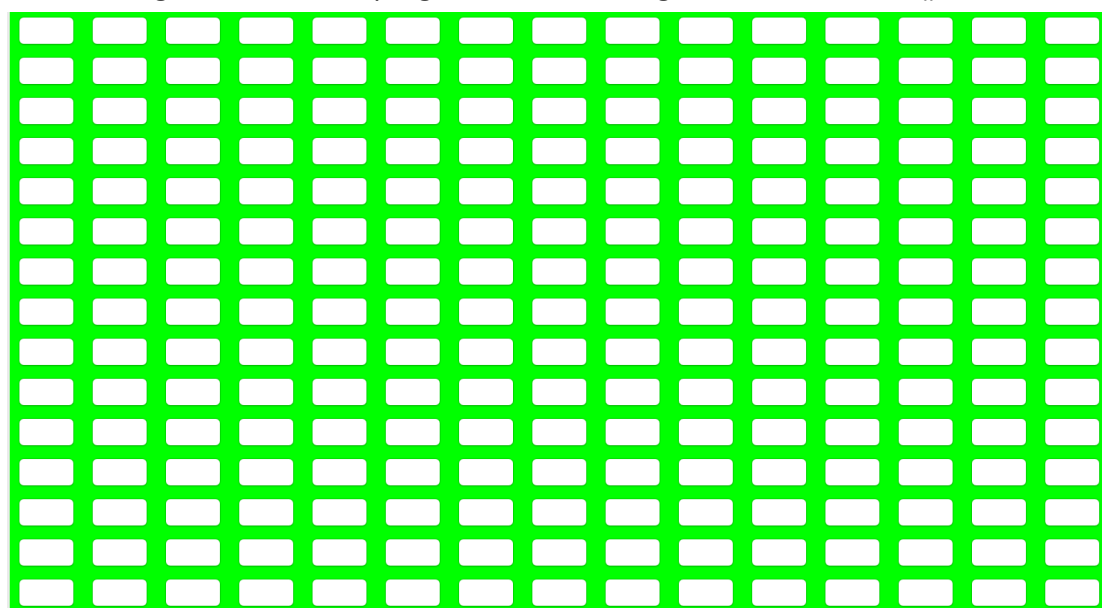


Figure 5