Games Shield assessment
Minitris

Jacob Sowah (jks17)

This was My second proper coding project, and I used it to understand more about the process of developing software. Firstly, I realised how useful it is to have a plan going in. I started this a couple of days after it was given, and I threw my self in to it. I just started programming little bits here and there not really doing anything cohesively. It got to the point where integrating all the little systems got to much and I had to scrap that and start anew. The second attempt was a lot more methodical. Not only did I follow the instructions closer, but also planed out what systems had to be perfected first before I could move on the others.

The main part of the coding that took me the most time was the clearing a line function. This was bad for me because at the start I believed that I could make my 2D array Y then X so that I could just copy the entire line down once ate a time. I realise now that this could not be done and not all languages are as nice and forgiving as python. This meant at every turn I would think that all my variables where X,Y not Y,X. it got to the point where I just had to change everything and use the two for loops for that function instead of one. After that I took more time and made sure to comment my work so that I fully understand what I was doing and what I had to do next. I also did a lot more research into the limitations not only of the Language but of the shield Aswell. This research helped me later because I was hoping to use the Text function on the shield to display a lot more and even though it was said in lectures that string manipulation was much different here than in other languages. I still believe that I could crack it. I was wrong. I was able to understand most of it but using did not know how much I could put on the shield. this saved a lot of time near the end of the project because I already knew that I should not try to put to much on the screen and instead put the score in the serial port.

Now there are a couple of problems with the code that I could not figure out how to fix. The main one is in the fullDrop function on line 95 there is a call to serial.println. this was here to help me visualize the line so I could correct the function, and the thought was to remove it at the end. However, when it Is removed the blocks do not show when they hit the bottom, and I have no idea why. I have gone through everything and every possibility and no matter what I do I cant get rid of it. My best guess is that there is something going on with the memory allocation that is making it glitch.

List of Accomplished Tasks

This is the list of tasks I believe I have done:

1. Using the state machine model
2. Having suitable movement
3. Being able to cleat rows
4. Being able to duplicate the bottom row to increase the tower
5. Make the DOWN button drop the falling block down to where it would land

ID	Test	Pass/Fail
1	Game Starts in “START STATE”	Pass
2	Game goes from Start to Playing state on FIRE	Pass
3	In Playing, a yellow block falls	Pass
4	The yellow block turns red and stops at the bottom	Pass
5	The yellow block turns red and stops above a red block	Pass
6	The yellow block can be moved left/right	Pass
7	The yellow block CANNOT be moved off-screen	Pass
8	The Yellow Brick does a “Fast Fall” where it goes to the lowest point it could go on DOWN	Pass
9	When there is a full row of red bricks, that line clears	Pass
10	When a line clears, all other blocks fall by one tile	Pass
11	The bottom row of tiles can randomly duplicate	Pass
12	When the line of blocks duplicates all other read blocks go up by one tile	Pass
13	The rate at which the yellow block falls increases as the game goes on	Pass
14	The game goes into a Pause state on FIRE	Pass
15	When in Pause State the game goes into Playing state on FIRE	Pass
16	The game goes into an End State when a red block is in the Position where the Yellow Block Is supposed to spawn	Pass
17	The score is Printed to the serial monitor in the end state	Pass
18	The game goes to the Start State on Fire after 1 second	Pass
19	If the game is again put into the playing state from the Start State, grid, score and LVL will all be reset	Pass
20	The Game Will store the high score on the EEPROM and display it in the main display in an extra state	Fail
21	The Game can have a two yellow blocks falling together	Fail

6. Make the FIRE button pause the game by adding a Paused state
7. Add a scoring system and output the score to the serial monitor
8. Make the game get progressively faster

The Test Table