|  |  |  |  |
| --- | --- | --- | --- |
| Method/Variable | Requirements | Difficulties | Functions |
| Place Block () | 1. Will need to place a new block at a certain location on the grid  2. Keeps track of it visually and within the block map data structure  3. Has to connect to the location of the old block | 1. Being able to keep track of the blocks and how that relates spatially to the grid on screen. (i.e. the location of a block on the grid is stored in the variable. Like block 3 = some coordinate on the grid)  2. Keeping the order in place of the blocks and making sure when one is added the order is not messed up. | addBlock () similar to push() in a queue |
| Remove Block () | 1. Removes the block after it has been hit by the ball. The first block added is the first removed and so on.  2. Makes the following block the new first in the queue, but doesn’t alter anything else. | 1. Being able to remove just the first block without messing anything else up.  2. Resetting the “first block” of the queue.  3. Having the method know when the level is completed and stop removing blocks or if the user places the blocks incorrectly and then they have to restart the level. | 1. Similar to the .pop () method in a queue |
| Block Map | 1. Stores the location of all the bricks within one given level. (ex. Block 4 = coordinate 350 by 250).  2. Connect the blocks to each other when a new one is added. | 1. Keeping an organized map of all the blocks without altering the order. | 1. Similar to a queue as the first thing added is the first thing removed when the ball moves around the screen. |
| Level Count/Brick Count | 1. Keeps track of which level the user is on and how many bricks they have to place (they are equal to each other) | 1. Not really that hard | 1. Increment ++ level each time the block map size is equal to 0 or in other words the level is completed. |
| Skin | 1. Variable for the skin of the blocks and their appearance  2. When the user reaches 100 blocks, they unlock a new skin.  3. There must also be a variable storing all the different kinds of skins which could resemble an array. | 1. Visual elements of making the skin different colors. | 1. Some kind of setColor function to the visual blocks which can be changed **if** the block count == 100. |

Priority:

1. Block Map, AddBlock(), Remove Block()
2. Level Count/Brick Count, Skin

The game itself is not very complex to play but there are some difficult elements. There are a lot of visual aspects of the code and variation. For example, a ball moving continuously on the screen and bouncing off bricks at different angles is random and seems very variable. Does the ball always bounce off the blocks the same? Does it matter if it hits the center of the block or the corner? How do you determine how the ball bounces of each block and moves to the next block? All of these factors can influence the complexity of the program.