# Screen Shots and Code Shots Name: Kevin Baron

**Note: Final Project and Evaluation are due on Wed, Nov 23.**

You need a screen-shot and a code-shot for each of your **main** scenes and code.

## Directions:

1. Make a ProjectImages Folder
2. Launch your Scratch Project
3. Right click a blank part of your code area. Name and Save image to your new folder
4. Right click the stage area. Name and Save image to your new folder
5. Insert your images into this Word Document
6. Insert, Picture
7. Type an explanation

|  |  |
| --- | --- |
|  | SPRITE: Red Dude |
|  | Start  Where the red dude is is represented by a room #, which starts off as 1. He begins with $0 and there are 4 Enemies Out and about (in the final version there are 145 enemies to begin with), in various other rooms. His ghost effect is related to whether or not he is flashing (he flashes a couple of times when he touches an enemy. He should always be in front of all other sprites (“go to front”) and he always points (aims) towards the mouse pointer.  Speed  He always moves SPEED steps and the Z key toggles the speed between 4 and 8. |
|  | Movement  The W key moves the red dude up by SPEED (set above) steps unless touching black, in which case he move SPEED steps in the opposite direction.  The same technique is used for A, S, and D in their respective directions. |
|  | Respawn  If ever you were to get stuck on something black and unable to move, you could always press X to go back to room 1.  Life  You begin with 3 lives. When you have more than one life and you touch an enemy (condition is below script block—to be inserted once I include all enemies (Enemy1, Enemy2, Enemy3, 4, 5, and 6)), you lose a life and flash. When you have 0 lives, you become a ghost and are transparent. |
| 🡨Opened door  🡨Closed door  🡨Partially opened door | Doors  There is a list of all the rooms that contain the door that appears at the top of the screen. If you are in one of those rooms, when the door is in the open costume (#13) (Costume of door changes as you get closer to it) and you press space, you will go to a new room specified by the list “Door Top Goes To:” (which is the above room on the map for top doors). Then the message “room change” is broadcast so that other scripts which reactivate when entering a new room can do their functions. Then you go to the opposite door (the opposite of top is bottom. Between this and the room change it appears as though you have travelled from one room to its adjacent room. The reason there is the “change y by 20” is to put you just beyond the reach of the bottom door. This way, in the instant when the room changes and you haven’t yet released your finger from the space key, you will not activate the script for the new door (which would be the bottom one in this case). To activate that script you must nudge yourself just a little bit to be in the range for the door’s costume # to be 13 (which is required to activate the script.  There are corresponding script blocks for what happens at Door Bottom, Door Right, and Door Left, each with their own respective lists. |
|  | SPRITE: Laser |
|  | Start  When the game begins, the costume is set to 1 (a red rectangle that looks like a laser). It always sets its costume to a variable “Costume#” since variables are more easily manipulated than costume switching in the next script block. |
| 🡨Laser  🡨Part of the explosion | Shooting  When the flag is clicked, the laser hides itself and shows the two variables Ammo and $ in the upper left corner of the screen. Ammo is set to 30.  As long as you have ammo left (0<Ammo) and you are in walk mode rather than run mode (speed is 4 rather than 8), when you click the sound will play, the laser will automatically go to wherever you, the red dude, are at on the screen, point toward the mouse, make sure it looks like a laser (costume# = 1), and show. Because room 1 is a free room, you will not lose ammo when you shoot in it. Otherwise, Ammo is decreased by 1.  Then the laser makes sure it is on the front and visible and it keeps moving 20 steps until it hits the edge or the color black. Then it explodes by quickly running through its other costumes that look like fiery explosions and then going backwards in the costumes (this is the part where having Costume # as a variable is handy). But the laser hides itself before turning back into costume 1 again so that you cannot see the laser again (that wouldn’t make sense). Because the whole script must run through before it can be activated again, you must wait until one laser completely explodes before firing another one. |
|  | SPRITE: Map |
| 🡨In room 1, In room 10🡪 | When the game begins, it goes to a specific location on the screen, shows, and then always sets its costume to the variable “room.” There are 52 rooms, so it has 52 costumes, and the current room is always shown by the orange box. The room numbering starts with 1 in the upper left, then goes to the right (so room 2 is the second teal square to the left on the top row) and goes back to the left for the next row (so that the second map image shown is room 10). The room on the bottom right is room 52. |
|  | Disappearance  This function uses the distance from the red dude to change the ghost effect so that as you get closer, the map fades away and doesn’t get in your way. The same function is used for all other indicators and gauges in the upper right corner. |
|  | SPRITE: Lives |
|  | This is pretty much the same thing as the scripts for the map, but this is combined into one block, which is necessarily better or worse. The lives indicator goes to a specified location next to the map. The costume # is based on how may live you have. Costume 1 is all blank lives (0 lives), so that is why there is a + 1 involved.  The disappearance function is the same. Notice that the outline is not quite black so the laser will not explode on it. |
|  | SPRITE: Sprint Indicator |
|  | Speed is changed by pressing Z (see explanation of that script block in RED DUDE SCRIPTS). If your speed is fast (8) that is indicated by showing this sprite and if speed is slow (4), then it is hidden.  also not quite black |
|  | SPRITE: Red Ball |
|  | When you begin, you do not have a red ball (0 means “not” or “false”). As soon as Have Red Ball is changed to 1 (by buying a red ball; see script block explanation for the sprite “Buy 1 Red Ball”) it switches to the costume “small ball” and goes to the upper right of the screen and show. It is currently an indicator that you have a red ball to use whenever you want. When you press R, it goes to wherever you are and switches to the bigger costume and shows. It is no longer an indicator but is active for use of Invisibility (SEE NEXT). You now have no red ball to use so Have Red Ball is returned to 0. |
|  | Invisibility  The Red ball controls invisibility. Because it is the same color as the red dude, when it is active (costume = 2 = big ball) and you are touching it, you become invisible to the enemies (see enemies’ scripts to see their reaction to invisibility). If the conditions of costume and touching are not both met, then invisibility is 0. |
|  | When you go from one room to another, your active red ball is lost. If the red ball is in indicator mode though, it is not hidden when changing rooms.  The red ball’s fade-away script is the same as Map’s/Lives’s/Sprint’s but only applies when costume = 1 (indicator mode). |
|  | SPRITE: Scroll |
| all the different costumes | The locked door is set to locked (1 = yes = true) and the scroll is set to 5 lives, which is indicated by having costume “big block”. It only appears in room 40, and if in this room, shooting it takes away its lives. Each life you take away changes the costume to the scroll surrounded by a smaller block, until life = 0 and the scroll is bare.  If the scroll is bare, when you get close enough to it will say what the password is (for password, see next). It will repeat this every 2 seconds (2 seconds got cut off from the image on the left) until you move away from it. As soon as you leave room 40, the scroll disappears. |
|  | Password  At the beginning of the game, the password is set to a joining of 6 random characters from a to z to 0 to 9. The script uses 5 “join” pieces and each joined item is the randomth (1 to 36) letter of the string abcdefghijklmnopqrstuvwxyz0123456789 |
|  | SPRITE: Door Top |
|  | Open/Close  This door goes to the top of the screen and closes itself (costume = Closed) When you get within 75 pixels of the door it creaks open and as you get closer it opens more (each costume shows a different openness). If you are within 15 pixels of the door it automatically stays at Closed12 (which is actually Open). And if you are beyond 75 pixels, the door automatically shuts. |
|  | Show/Hide  If the room you are currently in is not in the list of rooms that Door Top should appear in, then the door hides itself. If the room is contained in the list, then the door will show. |
|  | Door Lists  When the game starts it loads the list of all the rooms Door Top should be in (Door Top Rooms) and loads a list of all the rooms that Door Top leads to (Door Top Goes To:). Notice that the lists are constructed differently. Door Top Rooms is made by adding values to it. This list has no necessary order since the only thing tested of it is whether or not a certain number is contained in it (see preceding). “Door Top Goes To:” however needs a certain order. When it is used by the red dude to determine which room to go to (see red dude script), the current room # is used to determine what the next room will be. Thus the next room is a function of the current room and this list provides the output for each input. For example, if you were in room 11 and used the top door, the red dude’s script would go to the 11th item of this list and find the number 2. It would then use 2 as the new room. 12 yields 5, 13 yields 6, 7 yields 14, and so on. Notice though that many rooms go to room 1. This is because those rooms don’t lead to another room. They are place holders because there needs to be some item in that spot of the list so that the list has 52 places and all rooms can be checked by the above-mentioned process (except for item 10, because room 10 actually does lead to room 1). To be safe I put 1 in every place-holding spot so that if something went wrong, the worst that would happen is to go back to room 1.  One very important thing I found with lists like these is to make sure to always start out by clearing them (delete all). After I made my first list and was trying things out, the whole game kept getting slower and slower and slower. I later checked my list and found that it contained about 5000 items because every time I clicked the flag it would add all the numbers, not knowing that the numbers were already there in the list. This was simply corrected by starting each list construction by clearing it.  Each door has its own particular lists about which rooms to appear in and which rooms to go to. The only differences other than that in their scripts are the locations they are set to and their orientations. |
|  | SPRITES: Help Blocks |
|  | There are 10 Help Blocks. This is the script for “Help 1.” Each one is located at a different spot and each one says a different instructional message if touching the red dude. They only show when room = 1. |
|  | VARIABLE: Remaining Enemies |
|  | These script blocks are arbitrarily in the script area of the sprite “Buy 1 Laser…$1”. It probably would have made more sense to put these script blocks in the script area of Red Dude.  Anyway, when room = 1, the variable is shown, and when it hides as soon as you leave room 1.  The other script in this first block is related to the sprite “Buy 1 Laser…$1” and is irrelevant to the variable Remaining Enemies.  The second script block here simply hides the variable whenever entering a different room.  The variable is toggled by pressing the E key. |
|  | SPRITE: Buy 1 Laser |
|  | Only in room 1: If you have enough money (in this case more than $0) and you click this sprite, you will lose $1 and will gain 1 Ammo. The wait here is crucial so that 1 click will not quickly drain your money. The other buying sprites are all similar, however they will change your money by a different number (one of them a random number 0-10) and will change you Ammo by a different number (one of them a random number 0-10). Buying 1 Life will increase Red Dude Life by 1 rather than Ammo and buying a red ball will set Have Red Ball to 1. In all cases you must have at least the amount of money the item costs so that you can never have negative dollars. To choose the random cost 0-10 option you must have at least $10. For the increase life option, Red Dude Life must be less than 6 (6 is the maximum number of lives possible). |
|  | SPRITE: Random Barrier |
|  | It goes to (0,0). There are only a few rooms that it hides itself from. A list could have been used here like with the doors, but there are few enough rooms not to appear in that it was just easier to do it this way. Whenever the room changes, a random costume is picked. This means that the same room will not look the same the next time you enter it. There are many different costumes, including strange shapes and drawings, and upper or lower case letters, which are unusually shaped enough to provide good obstacle or cover. |
|  | A few different Random Barriers. There are 12 costumes total. |
|  | SPRITE: Planned Barrier |
|  | Rooms 31, 33, 22, and 36 have their own special needs. They are trick rooms with internal uncrossable walls that cause you to need to take an alternate route to get to the other side. Costume 2 is a straight line down and costume 1 is a diagonal line. Random barriers do not appear in these rooms and these do not appear in any other rooms. |
|  | SPRITE: Enemy 1 |
|  | Killing An Enemy  The first script block tells what happens when you reduce an enemy’s life to 0—when you kill it. You can only kill it if it started out alive, and once the life = 0, you earn $5 and the variable Remaining Enemies shows just long enough for you to see that you have reduced it by 1. Replacing the life with -1 removes this sprite from being involved in multiple loops of this process.  Max and Min  The second block shows the maximum and minimum number of lives a sprite can have (this will make more sense if you see the next block). |
|  | Enemy Lists  The list Enemy Rooms establishes how many lives the enemy will have in the various rooms. For example, in room 2, Enemy 1 will have a random number of lives between 1 and 3 (because 1 and 3 are the minimum and maximum). If the number of lives is set to -1 to begin with, that means that the enemy never exists in that room. Using a list like this is an extremely effective way of getting 1 enemy in any room you want. Rather than creating a different sprite for every room, the list creates multiple versions of the same sprite. However, the limit is that there is only one sprite per room. The easy solution is to make more sprites here. Because there are 6 enemy sprites total, the most number of enemies that can be in one room at one time is 6. Each enemy has its own list titled Enemy Rooms. |
|  | Room Change/Mobility  This script block reactivates every time the room changes. The first thing it does is it sets a current room. This is done so that in mid-script it can be detected when there is a room change. The first condition that must be passed is whether or not the enemy exists in the room. The Enemy Rooms list contains all this information; all that needs to be done is to input the room number to find the output of lives left in that room. The enemy goes to a random place and shows. It then repeats its movement process until it is killed or until the room number (which changes instantaneously when entering another room) is not equal to the current room number (which only changes once this script is finished). After you have entered another room, this script stops and starts over by resetting the current room. |
|  | Within this loop, though, it is always checking whether or not you are invisible.  If you are not, then it has the simple program of moving three steps towards the red dude at all times.  If you are invisible, it freaks out and moves in random directions.  Whenever the enemy is hit by the laser, it finds out how many lives it has in that room, subtracts 1 from that number (cut off from the image on the left), and then replaces the lives with this new number. The wait command here stops the enemy from moving for a brief time.  When the enemy touches the red dude, it waits then also.  When the enemy hits a wall, it just backs up 10 steps.  In the case of the enemy not having any lives, it simply hides.  All the enemies have similar programming to this, with different maximum and minimum random lives, different amounts of money for killing them, and different rooms that they appear in. Enemies 4, 5, and 6 appear in only 1 room each, move only in tight circles, appear in the same spot every time, and are not affected by invisibility (other than that they stop). These facts are all so that these enemies can only be reached via a specific route. They are in the rooms with internal walls. |
|  | SPRITE: Game Over/You Win |
|  | When you lose all your lives, game over.  If you kill all the enemies, you win. |