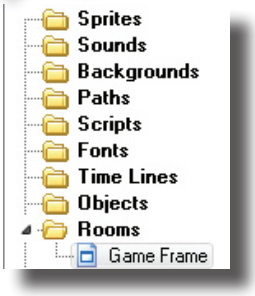
Maze Puzzler Alpha

1. Launch GameMaker Lite.
2. Create a new room named Game Frame.
3. Save the Game as LastName_Maze_Alpha.

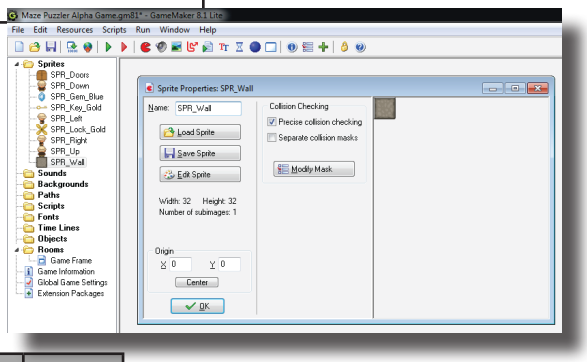


Sprites

4. Create the following Sprites:

Name	Image File
SPR_Doors	Doors.png
SPR_Down	Explorer_down_strip8.png
SPR_Gem_Blue	Gem_blue_sparkle_strip_32.png
SPR_Key_Gold	Key_gold_sparkle_strip32.png
SPR_Left	Explorer_left_strip8.png
SPR_Lock_Gold	Lock_gold_sparkle_strip32.png
SPR_Right	Explorer_right_strip8.png
SPR_Up	Explorer_up_strip8.png
SPR_Wall	Wall_block.png

5. Save the changes to the game.

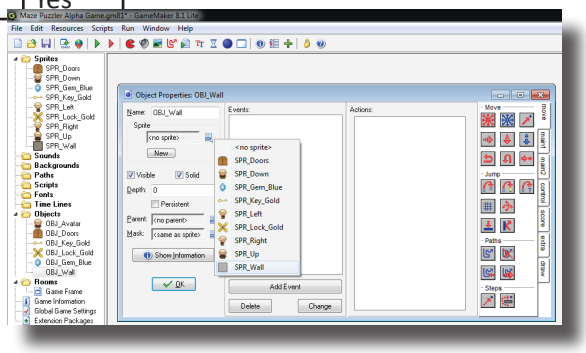


Objects

6. Create the follow Objects:

Name	Sprite	Visible	Solid
OBJ_Avatar	SPR_Down	Yes	No
OBJ_Doors	SPR_Doors	Yes	Yes
OBJ_Key_Gold	SPR_Key_Gold	Yes	No
OBJ_Lock_Gold	SPR_Lock_Gold	Yes	No
OBJ_Gem_Blue	SPR_Gem_Blue	Yes	No
OBJ_Wall	SPR_Wall	Yes	Yes

7. Save the changes.

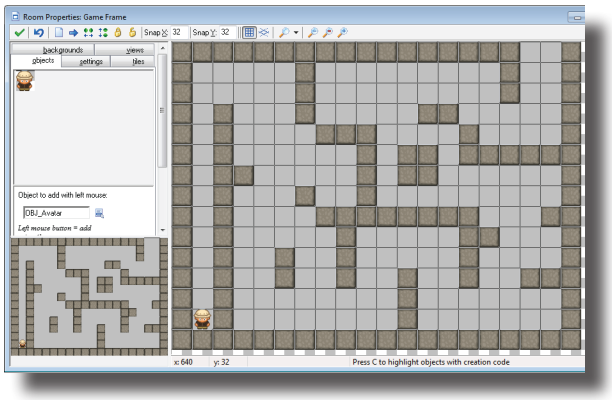


Level Design

Just like in other games, there will need to be objects in the room to create the environment and challenges. The same thing will have to happen in this game.

8. Open the Room Properties by double clicking on Game Frame in the resource tree.

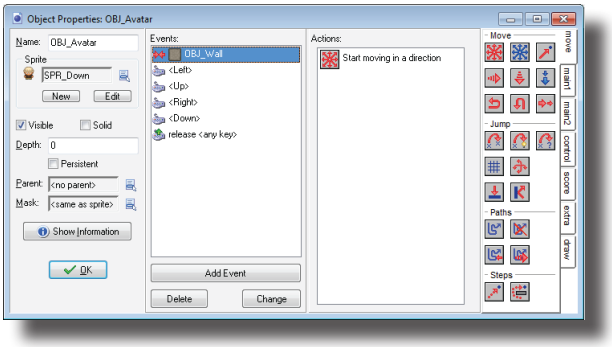
9. If the snap is not already 32 x 32, change it accordingly.
10. In the object tab, check that the object to add is OBJ_Wall. If not, change it and then place wall objects in the room. The room should have only one exit along the edge of the room.
11. Place OBJ_Avatar in the bottom left corner of the room. This way he has to maneuver to the opposite corner of the room to find the exit.
12. Save the game.



Player Movement

Open OBJ_Avatar's properties by double clicking on it in the resource tree. In the new window, program the following logic. Test it and if it works, save the game.

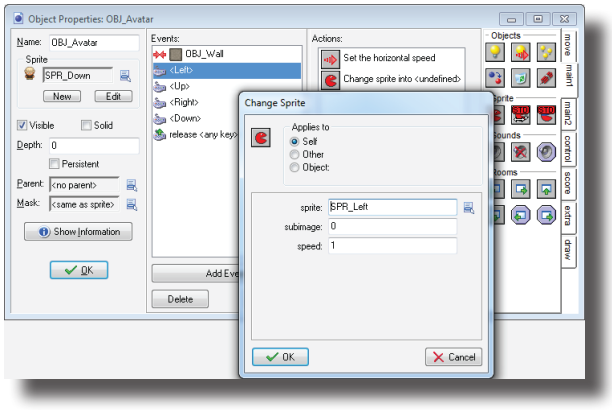
- If the avatar collides with the wall, Then the avatar movement stops.
- If the player presses the down arrow, Then set speed to 10 in the +Y direction.
- If the player presses the up arrow, Then set speed to 10 –Y direction.
- If the player presses the right arrow, Then set speed to 1- in the +X direction.
- If the player presses the left arrow, Then set speed to 10 –X direction.
- If the player releases any key, Then the avatar movement stops.



Avatar Animation

When testing the actions, they should have moved the correct direction. While they moved as they should, it didn't look good. The avatar was walking down regardless of the direction that he was moving. This needs to be changed.

13. If it is not still open, double click on OBJ_Avatar to open its properties.
14. Select the left arrow key event.
15. Drag the Change Sprite icon into the actions area
16. Check the self box so the avatar is the object that changes.
17. From the Sprite pull down menu, choose the SPR_Left. Doing this will change the sprite to SPR_Left each time the player presses the left arrow.
18. The subimage should be left at zero and the speed should be 1.



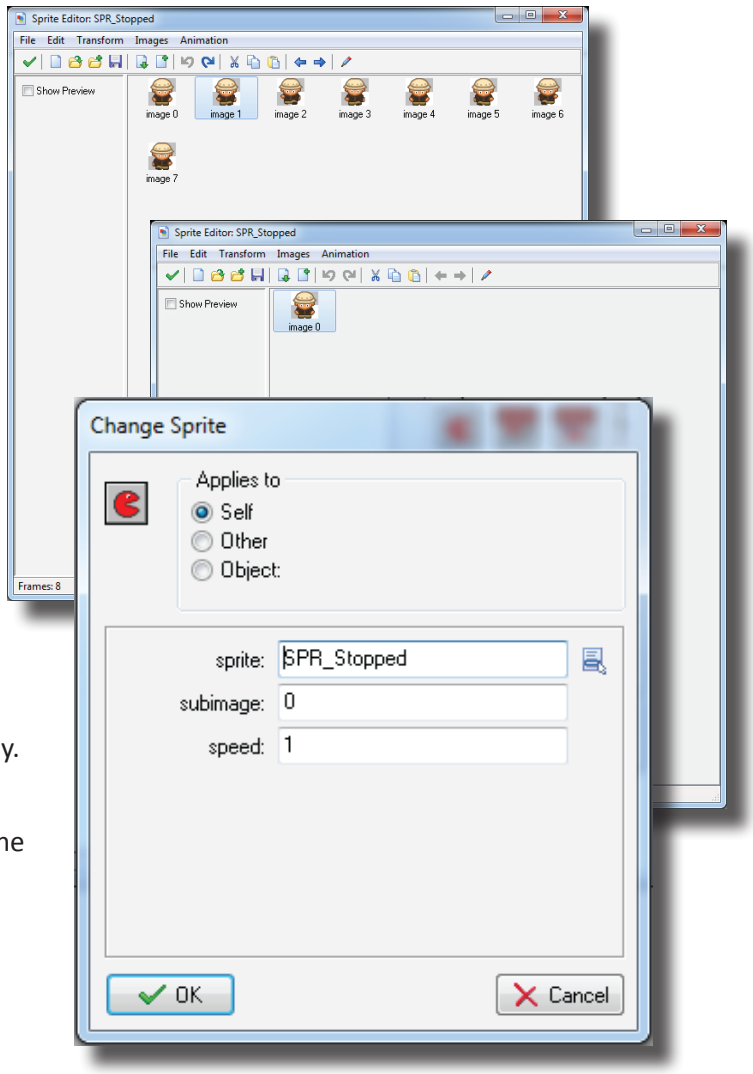
19. Click ok to close the dialogue box.
20. Repeat this programming for the up, down and right arrows and their corresponding sprites.
21. Test the game to see if the avatar now changes correctly for each arrow key.

22. Save the game.

Tuning the Avatar Animation

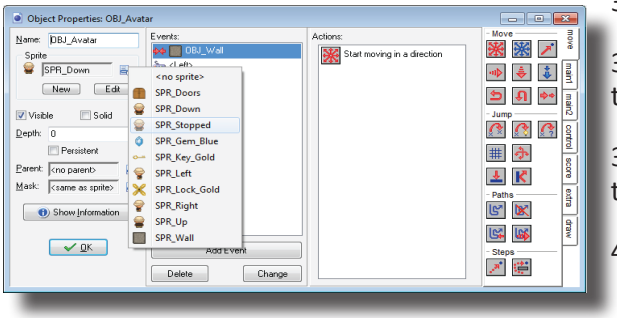
As it exists now, the avatar changes direction but continues to march on even when he has stopped moving. This doesn't emulate real life.

23. Duplicate SPR_Down.
24. Open the duplicates properties and rename it SPR_stopped. This will be the sprite that is used when the player stops moving.
25. Click on the Edit Sprite button.
26. Select image 1 (the second character).
27. Click on the scissors icon to cut out the select image.
28. Repeat this process for all but the first image (image 0)
29. Click on the green check to close the properties.
30. Choose Ok to close the sprite properties box.
31. Open OBJ_Avatar's properties.
32. Select the release <any key> event.
33. Select the event for when the player releases any key. Drag the Change Sprite icon into the actions area.
34. Change sprite to SPR_Stopped by choosing it from the pull down menu.
35. Test the game by playing it.
36. Save the new programming.



Final Animation Tuning

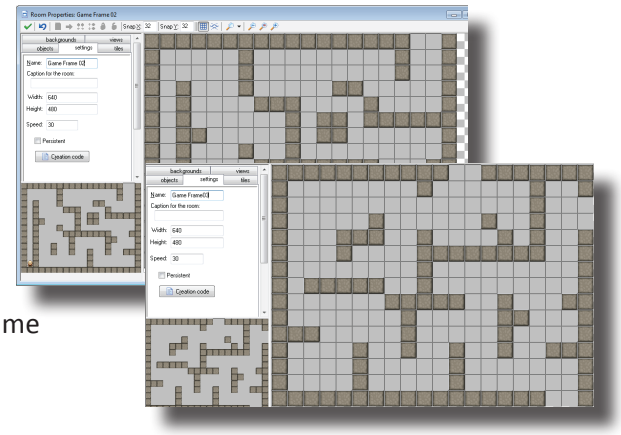
The avatar now stops marching when he is standing still, but there is still one more little glitch. Before the player presses any keys, the avatar is standing in place with his feet moving. It's time to fix this final piece of the animation.



37. Double click on OBJ_Avatar if it isn't still open.
38. Just below the sprites name is a pull down menu for selecting the sprite. From that pull down, choose SPR_Stop.
39. Test the game. The animation should now be perfect, even at the beginning of the game.
40. Save the game.

Room for More

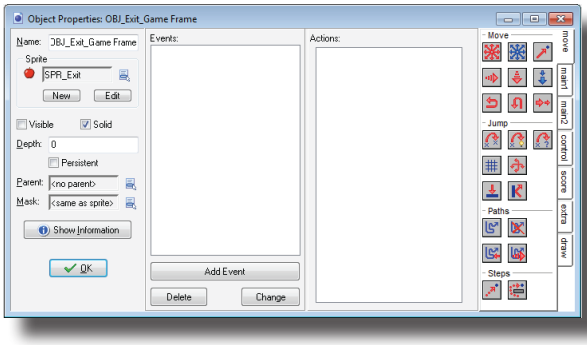
When the player finds his or her way to the exit, nothing happens. What should happen? The player should move on to a new room.



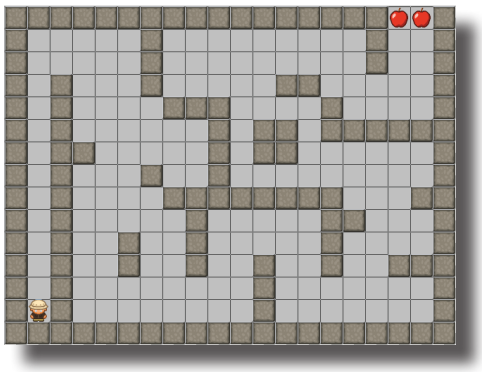
41. Duplicate the Game Frame room.
42. Change the rooms name to Game Frame 02.
43. Change the maze so the entrance to the second room lines up with the exit from the first room.
44. Create an exit that will lead to a third room.
45. Repeat the same process to make the third room, naming in Game Frame03. Just be sure to make the entrance/exit line up.
46. Save the changes.

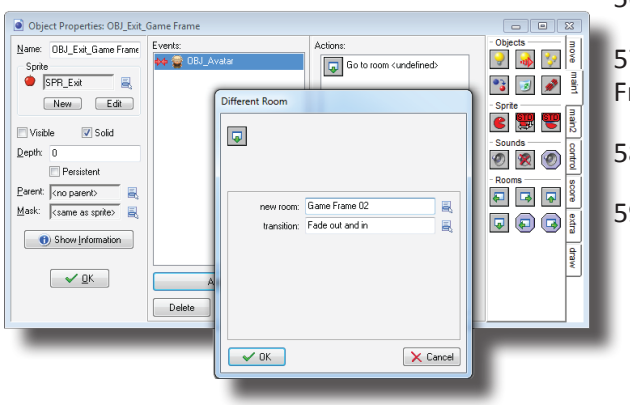
Entering New Rooms

The game has other rooms, but there is no programming to allow the player to move from one room to another. The following steps will explain how to make this happen.



47. Create a new sprite with SPR_Exit as its name and any new, existing sprite: an apple for example.
48. Create a new object. Name it OBJ_Exit_Game Frame and use the new sprite. Check the box for solid, but not visible. This will ensure that a random apple will not show up in the game play.
49. Open the room properties for Game Frame and add instances of OBJ_Exit_Game Frame to completely bock the exit.
50. Now go back to OBJ_Exit_Game Frame's properties.
51. Add an event for colliding with the avatar.
52. Next, drag in the different room button and drop it into the actions area.
53. When the dialogue box appears, select Game Frame02 from the pull down menu for new room/
54. Choose any transition from the transition pull down menu.
55. Next open the room properties for Game Frame02.
56. Move the OBJ_Avatar object to the correct starting spot.
57. Repeat this process for moving from Game Frame02 to Game Frame03.
58. Test the changes.
59. Save the game.





Skill Practice

These next few bits of programming are things that were learned in previous lessons, so the directions will be a bit more vague than the rest of the lesson. If they are too vague, try going back to previous lessons for more detail.

60. Rename each of the rooms to be Level 1, Level 2 and level 3.
61. Add a splash screen to welcome the player to the game and describe how to play the game.
62. Place a few instances of OBJ_Gem_Blue on each level of the game.
63. Add programming so that If the avatar collides with a blue gem, Then the blue gem should be destroyed And a sound should play And points should be added to the players score.
64. Program in a high school table to show scores when the game is over.
65. Add a score caption to show lives and score (lives will come later).
66. Give the collision between the wall and avatar a sound.
67. Test the game to make sure that it functions properly.
68. Save the changes and turn them in.