Creating Breakout - Part 1

Adapted from Basic Projects: Game Maker by David Waller

Getting Started

1. Launch Game Maker Lite. Game Maker Studio will not include the correct files needed, so be sure to choose Lite and not Game Maker Studio.

2. If a window prompting an upgrade opens, simply close it.

3. Under File, check that Advanced Mode is enabled.

4. On the left side of the program is a resource tree. By default, it includes predefined categories for different resource to be made and stored. Along the top is a toolbar that contains shortcut icons for creating and saving elements. Just like most other programs, hovering over an icon will list display will explain what it is used for.

5. When launching Game Maker, a new file is usually started automatically, however simply choosing File>New will create another one.

6. When saving for Game Maker, create a new folder for the project, and save any resources in the folder along with the Game Maker file to make locating all the resources simple. Saving can be done in the file menu or by clicking on the save icon on the toolbar.

7. Games made in Game Maker can be played on any computer that has Game Maker installed, similar to the way that Word documents can be viewed and edited on any computer that Word is installed on. This can be limiting, so once a game has been completed, publish it. this creates an executable file (.exe) that can be played on any computer, just like saving a word document as a PDF or JPG would enable any computer to view the file.

Terminology and Concepts

Objects: elements with a game that the player reacts with or that are used to create the environment.

Sprites: a graphic that represents the object. For example, there will be a ball used as an object in breakout, but a sprite (graphical image) must be created to represent the ball. Sprites are easy to change during the game. Think of Super Mario... when the player gets a star, sprite that represents Mario temporarily blinks in different colors. If the player gets a mushroom, the sprite that represents Mario grows, but can easily shrink back down if he gets hit by a turtle shell.

Events: things that happen in a game. For example, Mario colliding with a turtle shell is an event. Mario Eating a mushroom is an event too. If the event isn't present when the game starts, there has to be a creation event when it appears (Mario hitting a box that makes the mushroom appear is a creation event that creates the mushroom).

Actions: this is simply when something happens. For example, Mario colliding with the turtle shell causes an action of him shrinking or dying. Or, a ball hitting a wall could cause a rebound and sound of a bounce to play.

Rooms: are just environments in which the game is played. Games can have one room, like Tetris, or multiple room (like different levels).

Before making a game, some decisions must be made. What objects will be used? How will the sprites that represent these objects look? What events and actions will each object have? In Breakout, the objects, actions and events are:

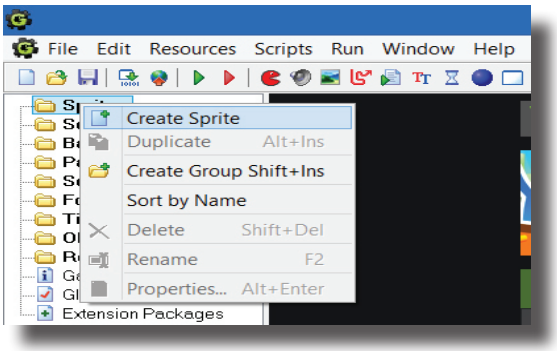
Object	Events	Actions
Bat	Cursor Keys Pressed	Bat moves right or left
Wall	None	None
Barrier	None	None
Ball	Collision with wall	Bounce
	Collision with bat	Move in a random direction
	Collision with barrier	Move in a random direction, barrier is destroyed, score increases by 1 point

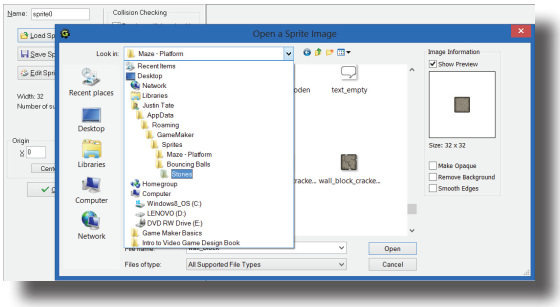
It's time to create Breakout!

1. **Launch Game Maker Lite** (NOT Studio)

2. A new game already has been created.

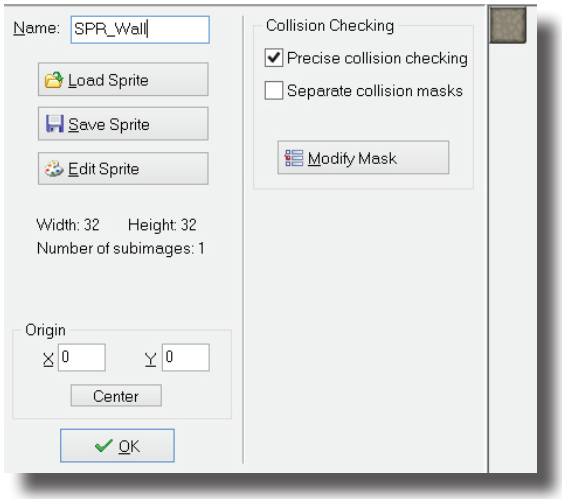
3. Create a new sprite. this can be done by **right clicking on the Sprites folder** in the resource tree or by clicking on the **new sprite icon** in the tool bar (it looks like a red pac man). This will launch a new dialogue box.

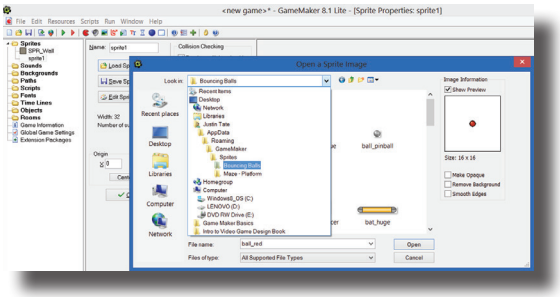




4. In the dialogue box, click on the **Load Sprite button**. Find the **wall_ block.png** file and choose open once it's been selected. Here it was located under GameMaker>Sprites>Maze-Platform>Bouncing Balls>-Stones.

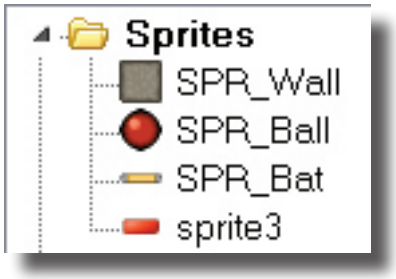
5. Change the **Name to SPR_Wall**. Always name sprites SPR_ to identify that it's a sprite of the wall, not an object of the wall (which should be named starting OBJ_). Leave the other aspects as they are and click **ok.** The wall should now appear in the resources on the left in the Sprites folder as SPR_Wall.

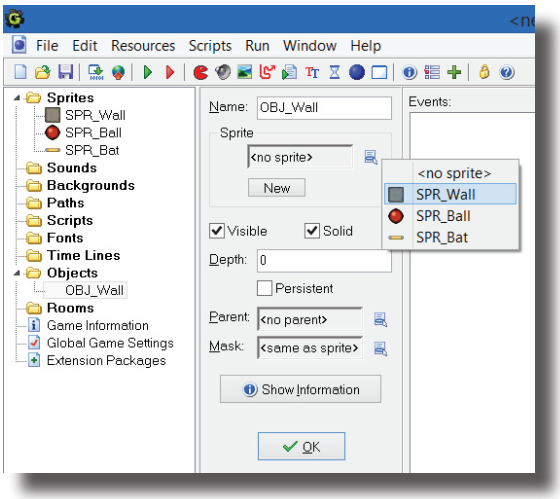




6. Create another **new sprite** (red pac man icon in toolbar or right click on the sprites folder). Click on the **Load Sprite** button. Now choose the **ball_red** for the bouncing ball in Breakout. It may require some searching to locate. In this case it was located under GameMaker>Sprites>Bouncing Balls. Change the **name to SPR_Ball** and click ok.

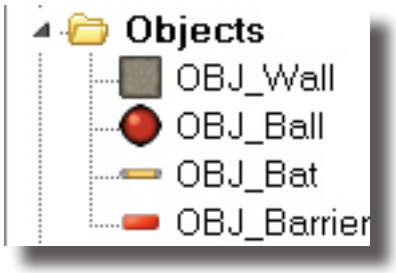
7. Two more sprites are required for a simple Breakout game. **Create another sprite**, load the **bat_large** and name it **SPR_Bat**. Click ok to close the dialogue box. Make a fourth sprite, name it **SPR_Barrier**, and load the **stone_normal_red** file. Four sprites, Wall, Bat, Ball, and barrier should be listed along the left side of the screen.



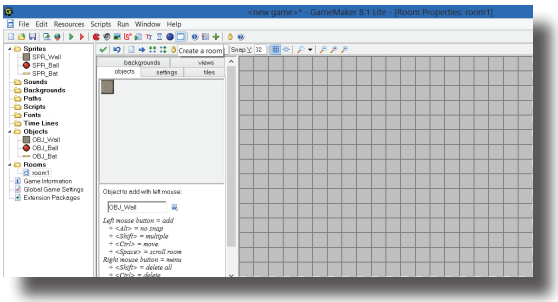


8. Now, create an object for the first sprite. To do this click on the **blue ball icon** in the tool bar or **right click on the objects folder** in the resource tree. **Name** the object **OBJ_Wall**. Now click o the **menu icon** at the right of the <no sprite> window. Here the sprites have just been loaded should appear. Choose **SPR_Wall**. Check **the visible and solid** boxes and click **ok** to close the dialogue box.

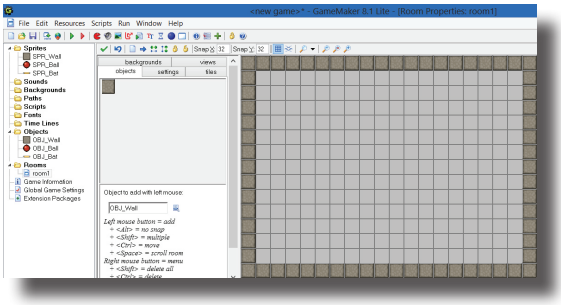
9. Repeat the same process to create objects for the ball, barrier, and bat. Once completed, they should appear at the left in the Objects folder.



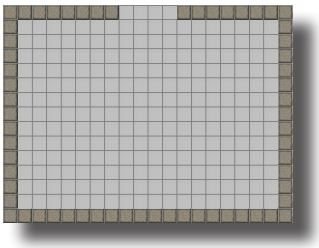
10. All the resources needed are now in the game, they just need a room to reside in. Create a new room by **right clicking on the room folder** and choosing **new room** or clicking on the **new room icon** (looks like a white box with blue banner across the top). This will open another new dialogue box.



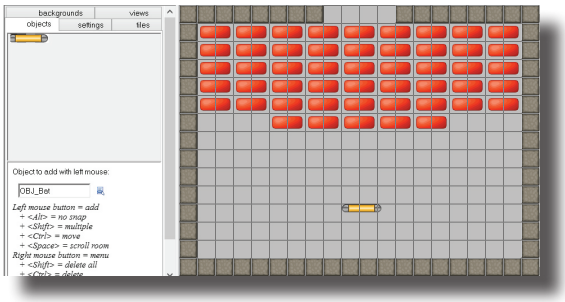
11. Maximize the new dialogue box. A grid has been laid over the room to make placing objects easier.



12. The first thing to place in the room is a wall along the outside of the room. Select the objects tab. In this tab, select the OBJ_Wall by clicking on the menu icon under the Object to add section. Now the sprite that represents the wall object should appear. Left click all along the perimeter of the room to add a wall all the way around the room. Holding shift and dragging will allow for quicker placement. To remove a misplaced wall, right click on the wall and click delete.

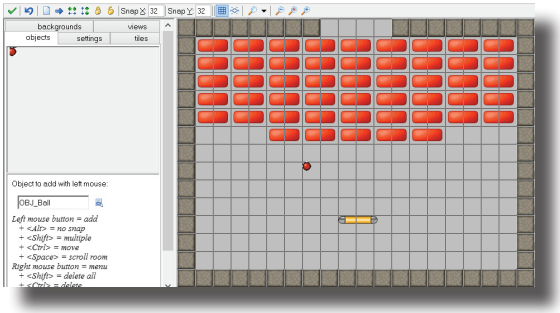


13. Remove four wall blocks from the center of the top wall so the ball can leave the room. To do this, right click on the wall and delete it.



14. Change the object to OBJ_Barrier and let click in the room to place EXACTLY 50 barriers in the room. This is important for scoring purposes.

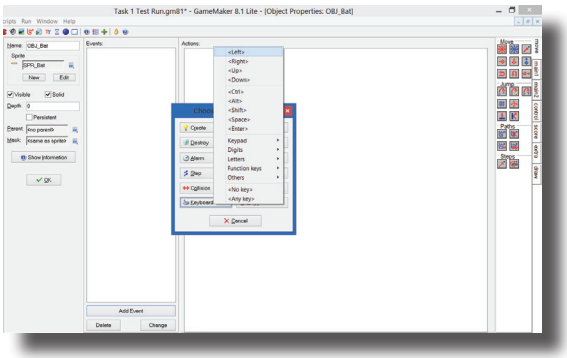
15. Now change the objet to add to OBJ_Bat. place one bat in the room approximately 3/4 the way down.



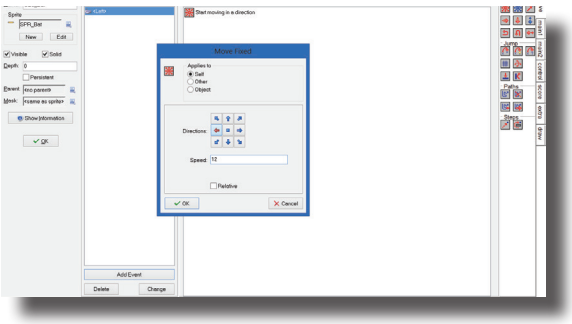
16. There is one final object to add to the room. Change the object to add to the OBJ_Ball and place one ball in the room. Click on the green check check in the top left corner of the room dialogue box to save the changes that have been made to the room.

17. Now, save the game as Breakout_"your name"_Task 1. Once it's saved, it would be a good idea to back it up on a thumbdrive, Dropbox, Google Drive or some other location too.

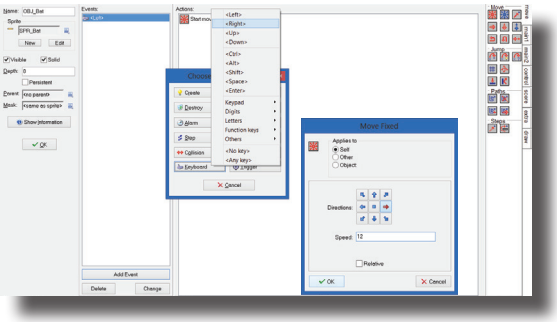
18. It's time to program how all of the actions and events will occur and take place. Double click OBJ_Bat. This will open a dialogue box for the bat that includes it's sprite, actions and events.



19. Click on the Add Event button. On the new dialogue box, choose keyboard, and finally, left. The event appears in the list and indicates that it is triggered by the left arrow on the keyboard. now the event needs an action on the event will do nothing.

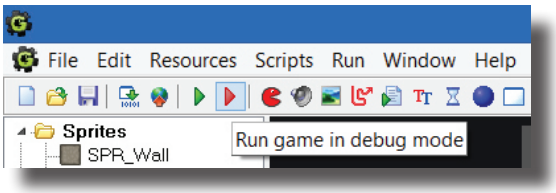


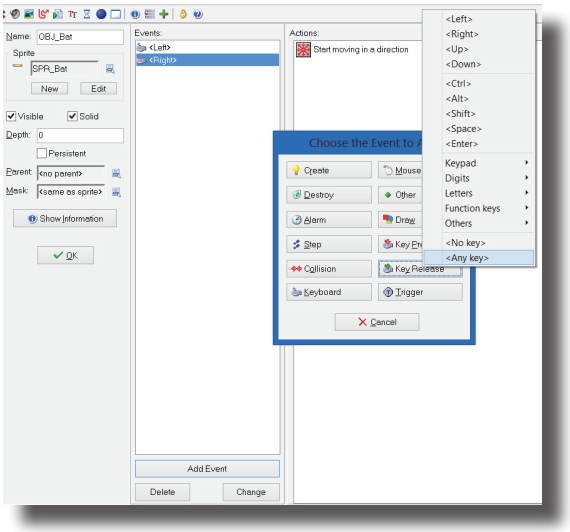
20. Select the move tab on the right. From this tab, click and drag the move fixed icon to the actions list. This will make a new dialogue box appear. In this new box, select only the left arrow (making it turn red) and set the speed to 12. Click ok to close the OBJ_Bat dialogue box.



21. Now make the bat move right by creating a New Event>Keyboard>Right, and dragging Move Fixed into the actions. This time, choose only the right arrow and use speed of 12 again.

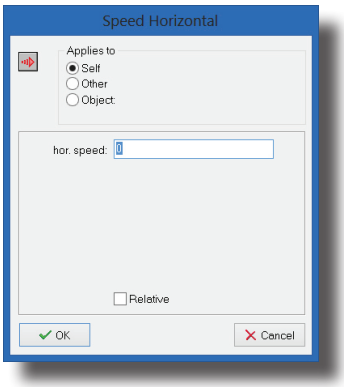
22. To test if this event and action work, click on the Run Game in Debug Mode (red play triangle in the tool bar). Test the action by pressing the left arrow and then the right arrow. The first actions are working! Press Esc to exit Debugging Mode.



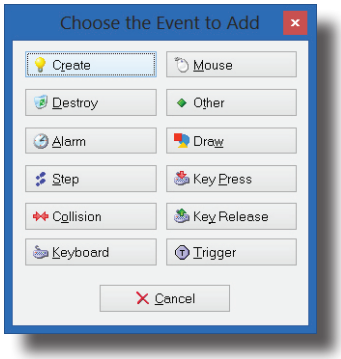


23. There's a serious issue though... the bat continues moving and actually leaves the room. To fix it, double click on the OBJ_Bat. and add an event for the bat. In the new dialogue box, click Key Release>Any Key. With this event, releasing any key, including the arrows, will stop the movement.

24. To make it stop the bat, the movement for it will need to be set to 0. Drag the Speed Horizontal icon into the actions window. A new dialogue box will appear. Here, set the speed to 0 and click ok. Click ok to exit the OBJ_Bat properties dialogue box.

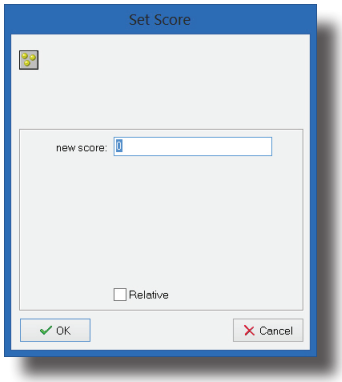


25. Test the new tweaks by clicking Run Game in Debug Mode to test if the new event fixed the problem. Now the bat should stop when the arrow is released.

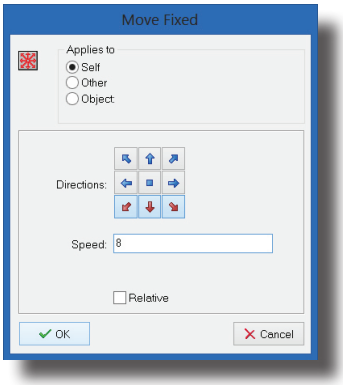


26. Now that the bat has been programmed, it's time to program the ball. Double click on OBJ_Ball. Now choose the Add Event button and make a create event. This event will now happen when the game starts.

27. Open the score tab and click and drag the set score icon into the actions area. Leave the score at zero and click ok.

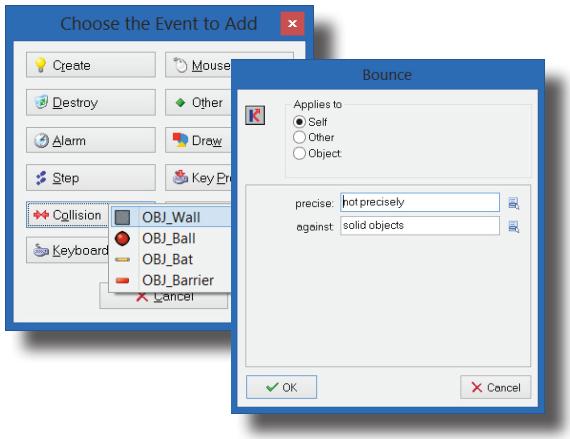


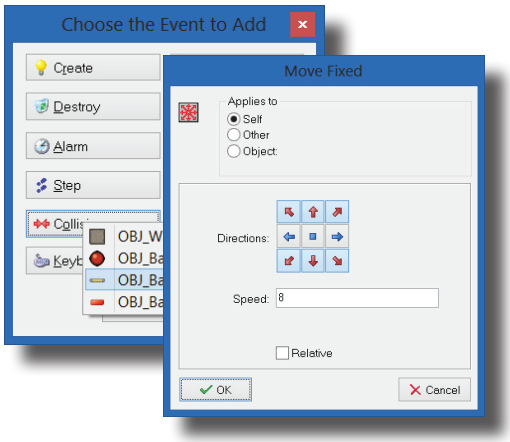
28. Now drag the Score Caption icon into the actions area. Leave these setting at their default so that the score is showing and click ok. The score will now show in the top of the window when the game is being played.



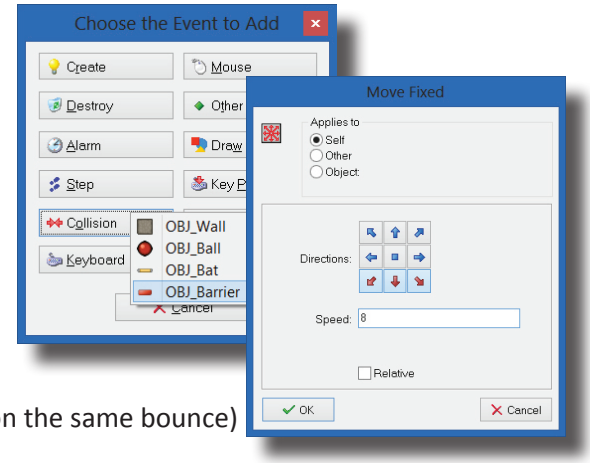
29. The ball needs to move down when the game starts so open the move tab and drag the Moved Fixed icon in. Choose all three of the down arrows and make the speed 8. Click ok to close the dialogue box.

30. Now the collisions need to be added. Click on Add Event and choose Collision. When asked, choose OBJ_Wall as the object the ball will collide with. The ball needs to bounce off the wall when it collides with it, so click and drag the Bounce icon into the actions. The default settings of not precise are fine (in fact, having the collision be more random will make the game more challenging).

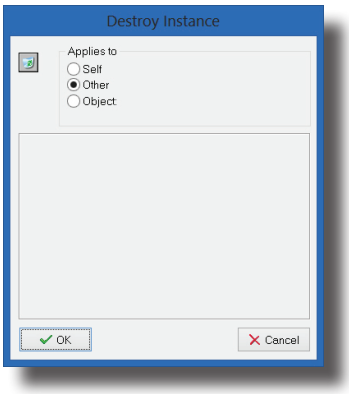




31. Next is programming the collision with the bat. press the Add Event button again and choose collision>OBJ_Bat. For the action, drag in the Moved Fix icon. In the dialogue box, choose all of the up and down arrows and set the speed to 8. Click ok to close the dialogue box.

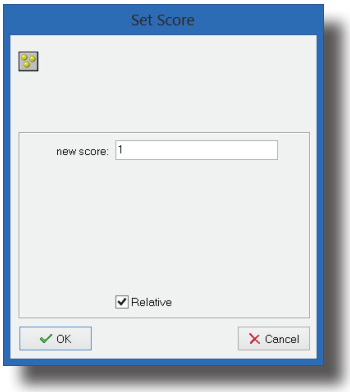


32. With collisions with the wall and the bat both programmed, the only item left for it to collide with is the barriers. These will be a little more complex since a collision will not only cause the ball to bounce off the barrier, but it will change the score and destroy or break the barrier. Again, Add a Collision event and select OBJ_Barrier as the object in which the ball will collide with. Drag in the moved fixed icon, select all of the down arrows (so the ball bounces back down instead of continuing on through and breaking more barriers on the same bounce) and set the speed to 8. Click ok to close the dialogue box.



33. The collision with the barrier still needs to destroy the barrier, so, while still in the barrier collision event, click on the Main 1 tab and drag the Destroy Instance icon into the actions area. When the new dialogue box appears, check other for the applies (the ball shouldn't be destroyed, the barrier should be) and then click ok.

34. The ball colliding with the barrier also needs to add to the score. Make sure the barrier collision event is still open, click the score tab at the right and drag in the Set Score icon into the actions area. On the new dialogue box, set the score to 1 and check relative. Relative will add 1 point each time the collision happens. Without it the score would always stay at 1! Click ok to close the dialogue box, and then ok to close the OBJ_Ball.



35. Test the new programming by clicking on Run Game in Debug Mode (the red play icon). The score should now change, barriers should disappear when hit, and the ball should bounce off of everything it hits. If it does, things are running just the way they should be!