

Creating Breakout - Alpha

Adapted from Basic Projects: Game Maker by David Waller

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.

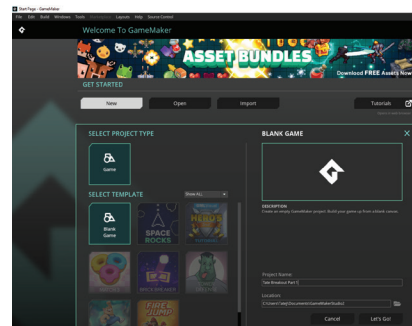
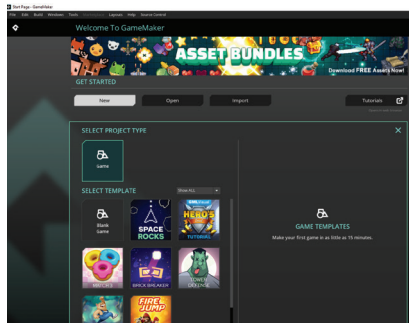
Rooms: are just environments in which the game is played. Games can have one room, like Tetris, or multiple room (like different levels).

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:

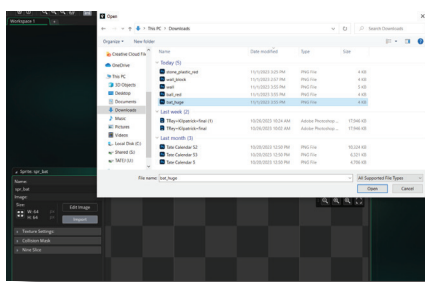
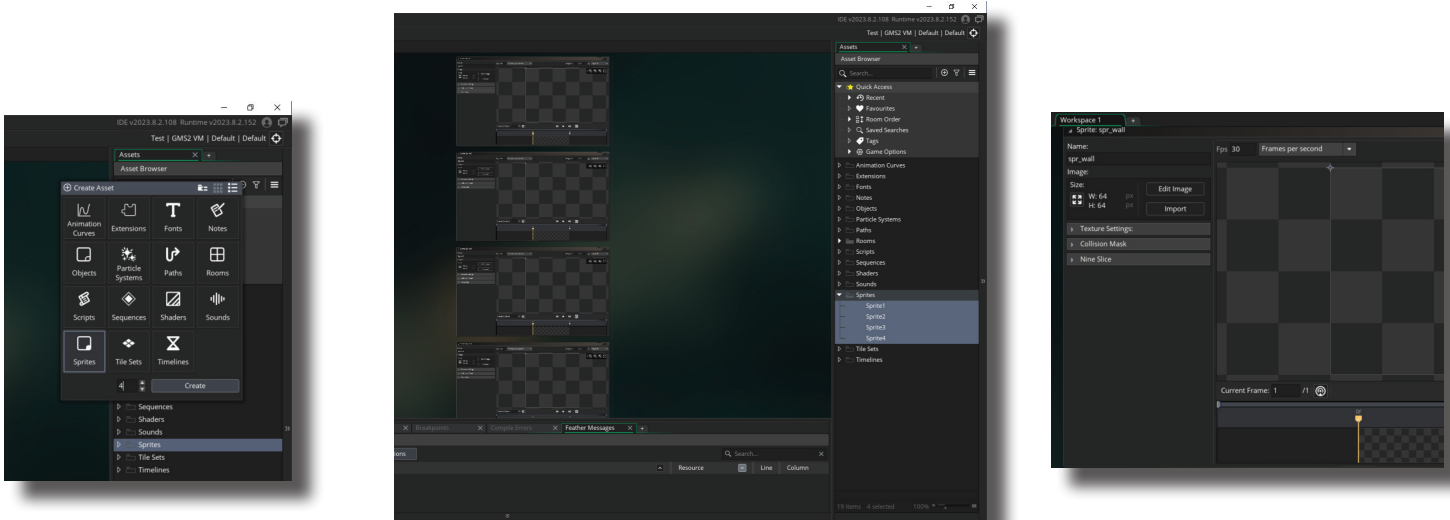
Let's get to it!

1. Launch GameMaker Studio.

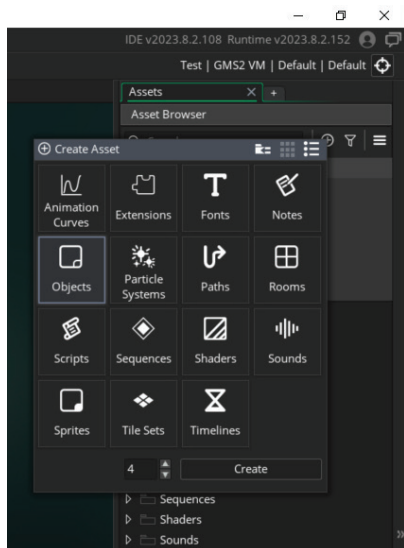
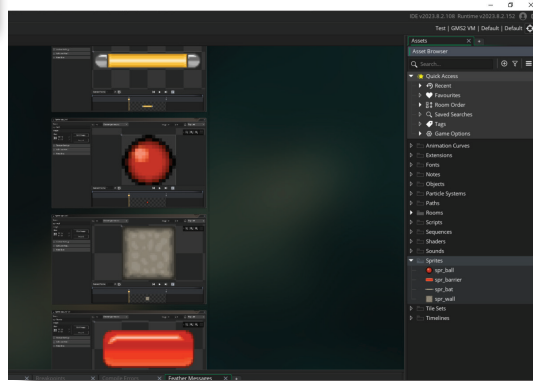
2. On the Welcome Screen, Click the New button and choose Blank Game. Then just to the right, in the Project Name field, type Your Name Breakout Part 1. Next choose a location to save your game on the computer. Finally click Let's Go! at the bottom right. In the window there will now be a asset browser along the right side, a work area in the center where the code or blocks will be assembled.



3. Create a new sprite. At the top of the resource asset browser the circle with the plus in it. Choose the Sprite icon and then type 4 into the box to the left of the create button. The Sprites folder in the resource tree will open, revealing 4 Sprites and the Workspace in the center will also show the properties for each Sprite. In the workspace, change each Sprites name to spr_ and then bat, ball, wall and barrier.



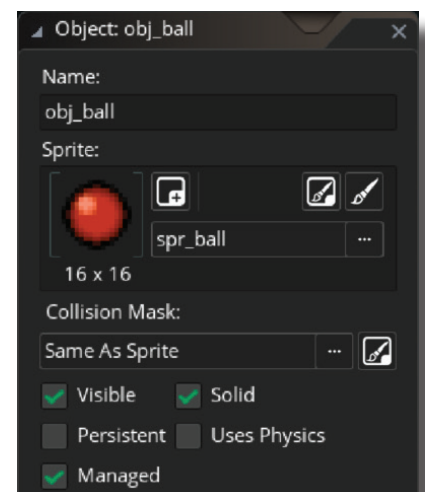
4. Next, load the appropriate Sprite for each. In each properties box, click the Import Button, navigate to the location of that the Sprite was saved, select and upload it. If a message about not being able to undo appears, proceed anyway. Repeat this process for the other three Sprites. Always name 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.



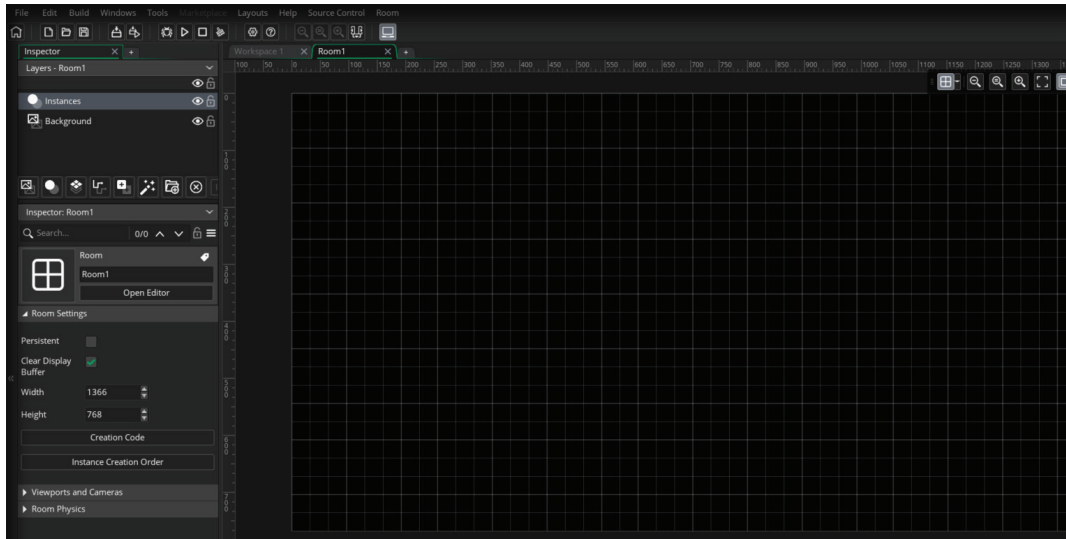
5. Now, create an object for each Sprite. To do this click on the circle with the plus in it at the top of the asset browser. On the new panel that opens, choose Object and set the count (beside the Create button at the bottom) to 4. Now click create. Four new objects will appear in the Asset Browser and also in the workspace at the middle of the screen.

6. Rename each object as obj_ball by changing the name of the first in the Workspace. Below the name, change the box that says No Sprite to obj_ball. Finally check visible and solid to make the ball show up in the game and solid so that things can actually collide with it. Leave the Collision Mask at Same as Sprite.

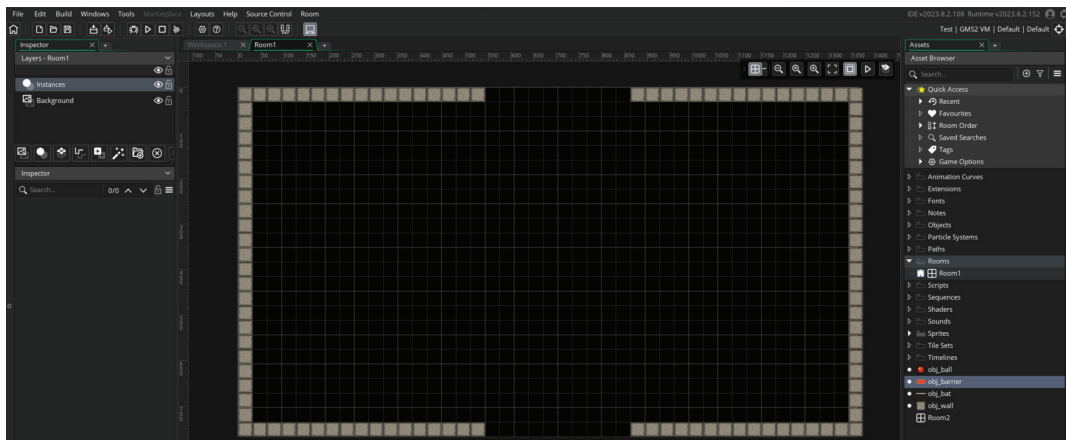
7. Now do the same thing for obj_wall, obj_bat and obj_barrier using the other three objects that have been created.



8. All the resources needed are now in the game, they just need a room to reside in. In the Asset Browser, there should be a section labeled Rooms. Open the folder and double click on Room 1. The room will open in a new tab where the Workspace with all the Objects were. The grid shows the entire area that the game will occupy. This is known as a Room.



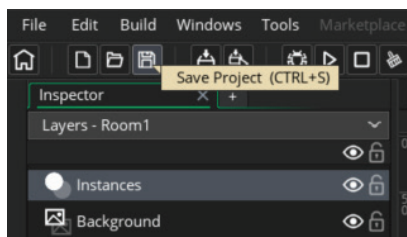
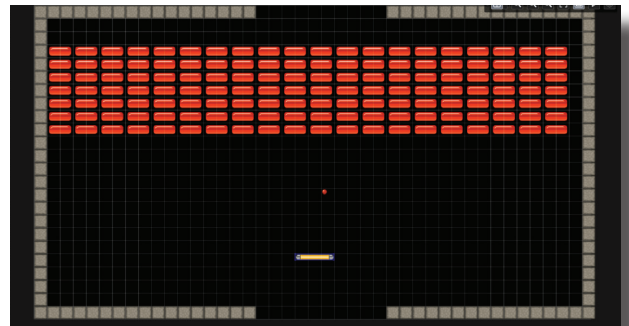
9. From the Asset Browser, select the obj_wall. This should appear in the panel along the left edge. Make sure that the Instances Layer is selected and then drag one instance of obj_wall from the Asset Browser into the room. Position it around the perimeter. Repeat this until the entire room is closed by obj_wall instances except for a 10 block opening at the top center and a 10 block opening at the bottom center.



10. Select obj_barrier and drag it into the room. Set instances in place so there are numerous rows of barriers.

11. Now change the object to add to obj_bat and place one bat in the room approximately 3/4 the way down.

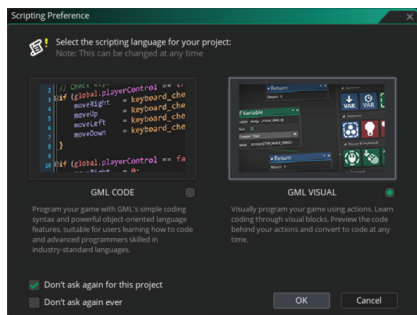
12. There is one final object to add to the room. Change the object to add to the obj_ball and place one ball in the room. Now close the room by clicking on the x on the tab at the top of the workspace.



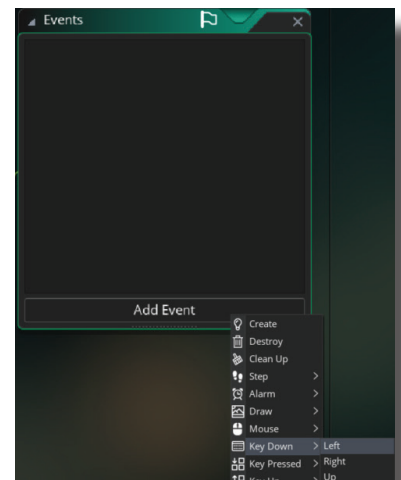
13. With the room and all objects set, this would be a great time to save. Either click on the disk icon at the top left of the program, press Ctrl S, or choose File>Save. Once it's saved, it would be a good idea to back it up on a thumbdrive, Dropbox, Google Drive or some other location too.

14. It's time to program how all of the actions and events will occur and take place.

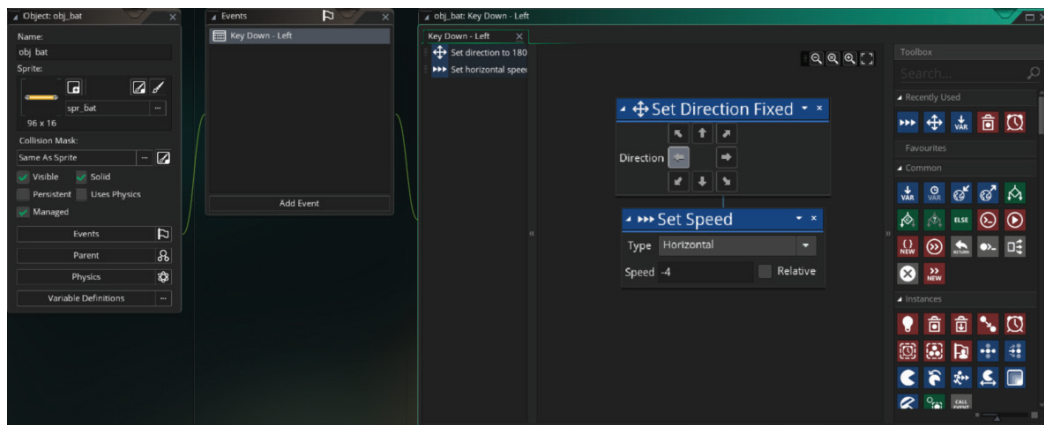
In the Workspace all the objects should still be open. If not, double click OBJ_Bat. This will open a dialogue box for the bat that includes its sprite, actions and events in the Workspace.



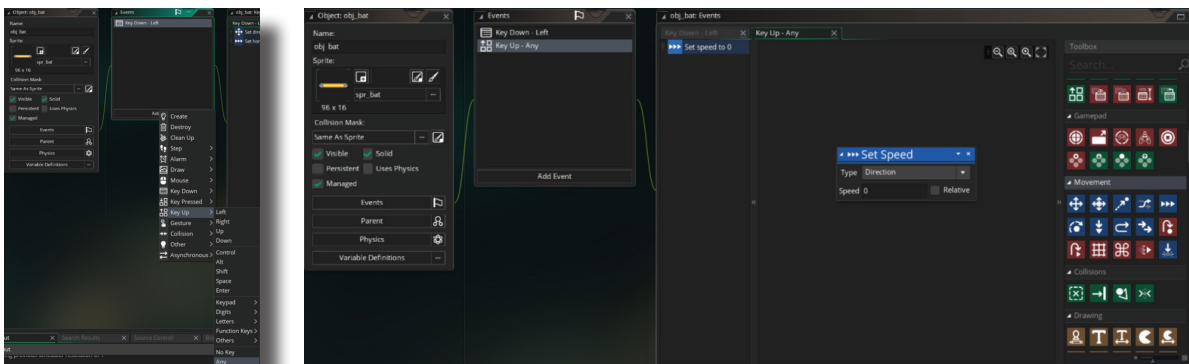
15. Click on the Add Event button, key down, and finally, left. A new dialogue box will appear asking if the software should run in visual or code mode. Here select visual to build with blocks and check Don't Ask Again for this Project. There is now an event for when the left key is down but no action takes place when this happens... the event needs an action on the event will do nothing.



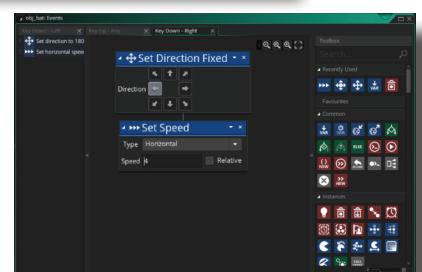
16. Find the Movement Section from on the right side of the actions panel. From this section, click and drag the Set Direction Fixed icon to the actions list. This will make a new dialogue box appear. In this new box, select only the left arrow (making it turn gray) and then drag Set Speed (also in the Movement Section) set the speed to -4. This should move the bat to the left at a speed of 4 when the left key is pressed. Save the game (Ctrl S) and Run in by pressing the play button in the top left. The bat should move when the left key is pressed but it never stops. Close the window running the game.



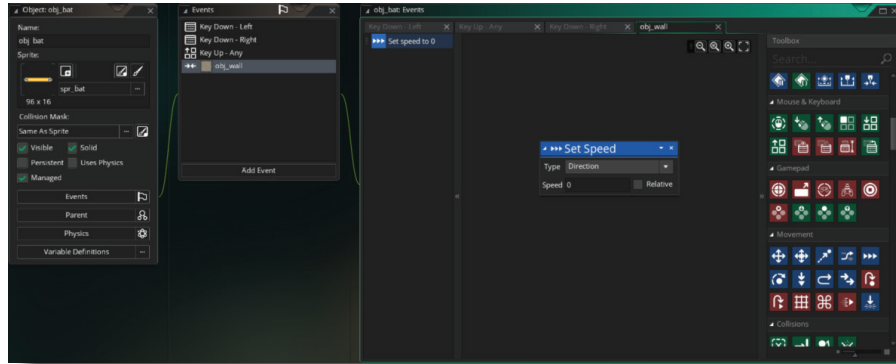
17. Add an event to obj_bat by pressing the Add Event button and choosing Key Up - Any. This will happen when any key is released. From the Movement Section, drag Set Speed into the Actions Area. Change the Speed to 0. Save and run the game and the bat should stop moving when any key is released.



18. Repeat steps 15-17 to make an Event that will move obj_bat to the right but instead of using -4 for the speed, just use 4. Once set, save the game and run it to test the bats movement.

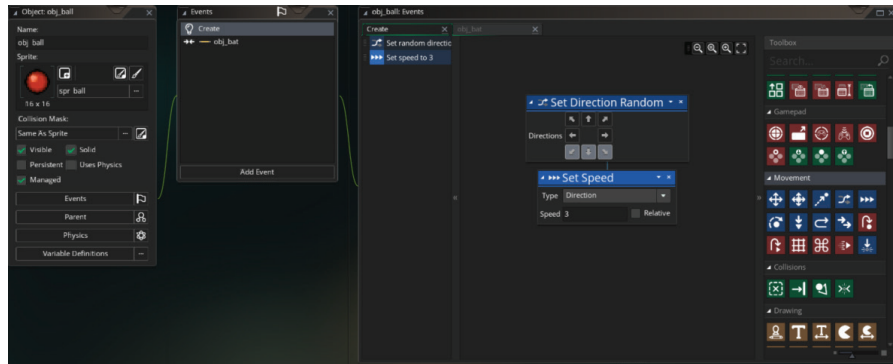


19. There's a serious issue... the bat continues moving and actually leaves the room. To fix it, open obj_bat if it's not already open. Next click Add Event>Collision>obj_wall. With this event, hitting the wall should stop the bat. To make it stop the bat, from the Movement Section, drag Set Speed into the actions area. Change the Speed to 0.

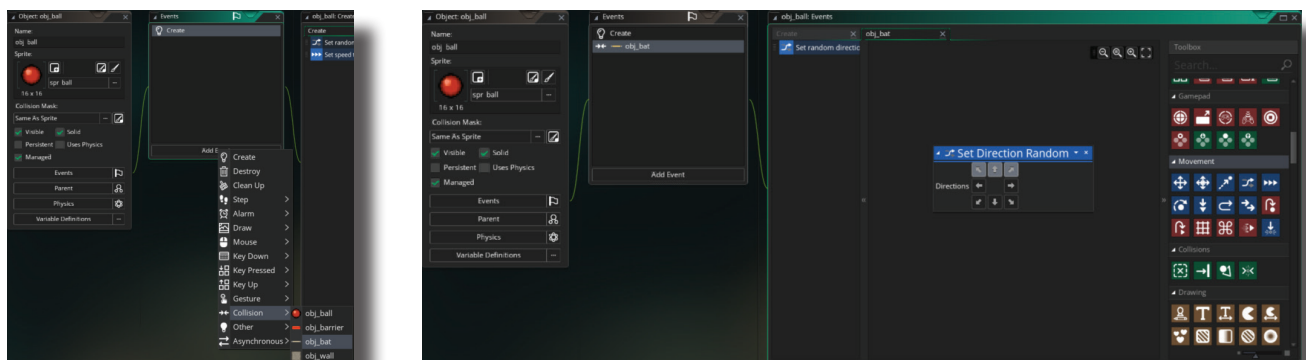


20. Test the new tweaks by clicking the Play button to see if the new event fixed the problem. Stop when it collides with the wall.

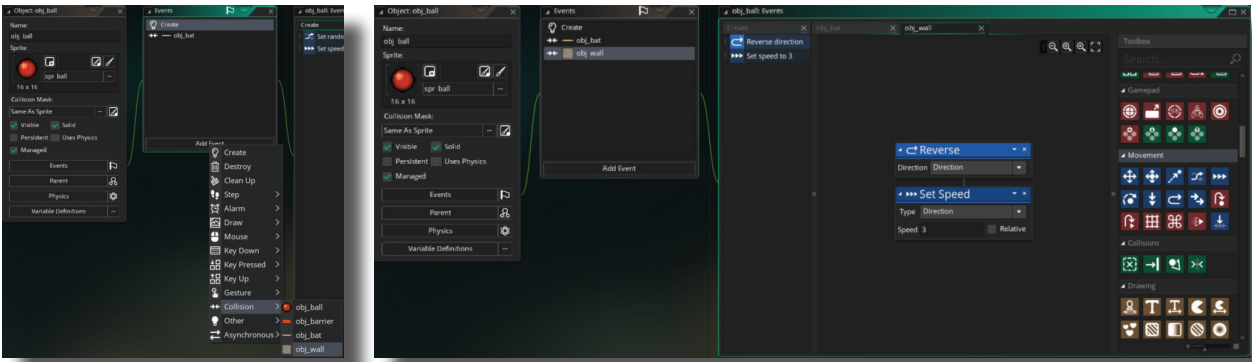
21. With the bat set, the ball needs to move down when the game starts. Add a Create Event and from the Movement Section and drag the Set Direction Random icon into the actions area. Choose all three of the down arrows to set the direction of movement. Now add the Set Speed icon just below the Set Direction Random block. The ball should be a little slower than the bat so plug in 3.



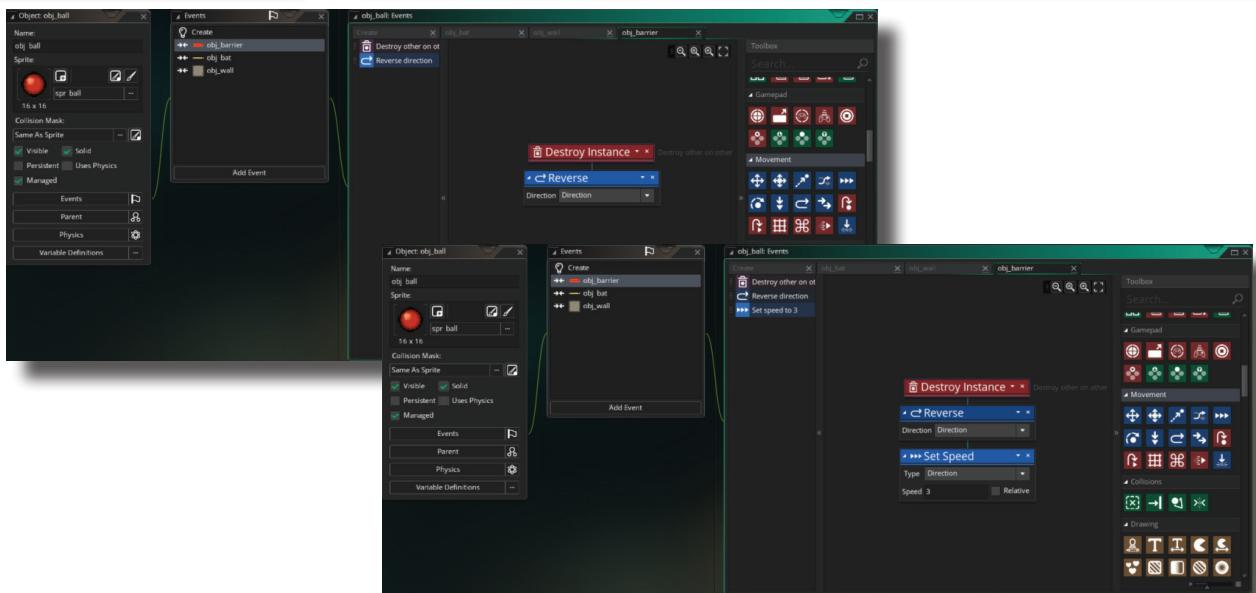
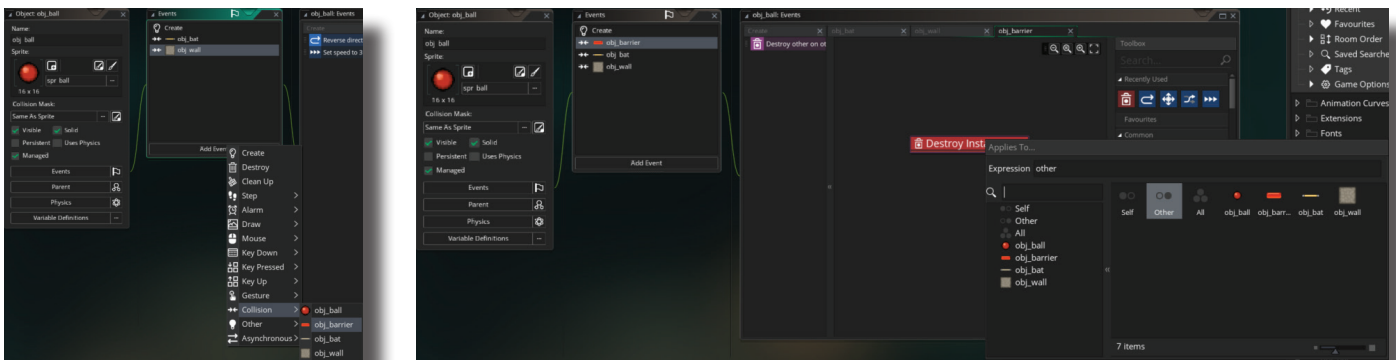
22. While still in obj_ball, make an event for when the ball hits the bat. To do this, press Add Event>Collision>obj_bat. Now when the ball collides with the bat it needs to bounce off and go upward toward the barriers. To make this happen, find the Movement Section and drag the Set Direction Random icon into the Action area. Choose all three of the upward facing arrows. Next attach a Set Speed Block to the bottom of the Set Direction Random block. Here, set the Speed to 3.



23. Now the collisions need to be added. Click on Add Event>Collision>OBJ_Wall as the object the ball will collide with. The ball needs to bounce off the wall when it collides with it, so to make this happen, find the Movement Section and drag the Reverse icon into the Action area. Also from the Movement Section, place a Set Speed action below Reverse and make the speed 3.



24. The final action for the ball is the collision with the obj_barrier. This is more complex than the other collisions because it will have to break the barrier and change direction. Choose Add Event>Collision>obj_barrier. From the Instances Section find and drag the Destroy Instance icon into the Actions area. Rather than destroying the ball, click downward facing arrow and change the expression to Other. This will remove the barrier that the ball collides with. Now back in the Movement Section find Reverse and add it to the Actions area. Next add Set Speed just below it and make the speed 3.



25. Now save the game and run it. When playing, the ball should bounce off the bat and move back up toward the barriers. The ball should collide with the barrier, break it and move back downward. When the ball hits the wall it should bounce off of it. This is the basic version of Breakout... the Alpha. It's a functioning program that the player can interact with, but not yet a game.