

Smalltalk Programming

Lesson 31

In the last lesson, you added the ability to use realistic images for your ship and enemies. Everything appears to be complete with the game design. However, the enemies are always starting at the same location and are very predictable. Today, you will make changes so the enemies start at random positions on the ShooterGame screen. With this change, you will never know where they will start!

1. Make the following changes to ShooterGame>>initializeEnemies and then save them.

ShooterGame>>**initializeEnemies**

```
3 timesRepeat: [
  | enemy enemyHeight enemyX enemyY |
  enemy := Enemy new.
  enemyHeight := enemy height.
  enemyX := (self left to: self right - enemy width) atRandom.
  enemyY := (self top + enemyHeight to: self bottom - (enemyHeight *
3)) atRandom.
  enemy position: enemyX @ enemyY.
  self addMorph: enemy]
```

2. Test your changes. Does it work?

3. How cool is the *timesRepeat:* method? It lets you repeat the same action multiple times. This code creates 3 enemies that will appear in random positions on the ShooterGame screen. Each enemy's position is chosen randomly within the allowed area of the screen. Then, the enemies are added to ShooterGame for you to see.

4. *enemyX* calculates a random horizontal (X) position for the enemy. It picks a position between the left and right edges of the game screen, but subtracts the width of the enemy from the right edge, so the enemy stays fully on the screen. (Remember, *position* and *position:* refer to the top left side of a morph.)

5. *enemyY* calculates a random vertical (Y) position for the enemy. It picks a position between the top and bottom edges of the game screen. The range starts below the height of one enemy from the top edge (to leave room for the score display, though this is not absolutely necessary) and ends at the height of three enemies above the bottom edge, so that enemies do not appear too close to the bottom.

6. The *to:* method is also pretty cool. Here, it provides a range of numbers, including both the start and end numbers. For example, “1 to: 5” provides the range of numbers 1, 2, 3, 4, and 5. It increments by 1. But the *to:* method can be used with more things than just numbers! For example, “\$a to: \$z” (the \$ before *a* refers to the character “a”) provides the range of characters from “a” to “z”.

7. Try shooting some enemies. Remember, you can also press the “r” key to create even more than 3 enemies! You can have a screen full of them, and with aimbot, you are the boss!

8. Great job on finishing your shooter game! You have added some awesome features, and your game is really coming together. But this is just the beginning. There is always more you can do – new ideas, new challenges, and new ways to make your game even better. With the coding that you have learned in these lessons, you can create so much more. What will you build next? Keep experimenting, keep learning, and most importantly, have fun with it!

9. Save and Quit your Smalltalk image.