Smalltalk Programming
Lesson 11

Today you will add the ability for your ship to receive input from the user. Before you can code how your ship will move, you first need to add code that can read keyboard events. In Smalltalk, input is handled by *events*. Your ship will be moved by using keyboard keys. You will first need to know which keyboard key was pressed before you can write the code that goes with that key. Today you will make changes to your `ShooterGame` and `Ship` classes.

1. Your `ShooterGame` class is your main game class. It should have the code for handling events, or input, even though your ship is what will be moving. There are reasons for this that will become clear as you type the code to handle events.

2. In your class, type and save the following code.

```
handlesMouseDown: anEvent

    ^ false
```

2. This code will tell the Morphic framework (the Smalltalk code that handles graphic images and user input, etc.) that you do not want to handle pressed mouse button events.

3. Type and save the following code.

```
handlesMouseOver: anEvent

    ^ true
```

4. This code tells the Morphic framework to handle events when the mouse cursor is over your game screen. If the mouse cursor is not over the game screen, the game will not receive events.

5. Type and save the following code.

```
mouseEnter: anEvent

    anEvent hand newKeyboardFocus: self
```

6. This code tells Morphic what to do when the mouse is over your game screen. Here you are telling Morphic that you want `ShooterGame` to handle the keyboard events when the mouse cursor is over the game screen.

7. Type and save the following code.

```
mouseLeave: anEvent

    anEvent hand releaseKeyboardFocus: self
```

8. This code tells Morphic that you want to stop handling keyboard events when the mouse cursor is not over the game screen.

9. Type and save the following code.

```
handleKeystroke: anEvent

    | keyString |
    keyString := anEvent keyString asLowercase.
    ship keystroke: keyString
```

10. This code receives the keyboard event from Morphic. Your code then takes that keyboard event and then uses `keyString` to find out what key character (“a”, “b”, “c”, “1”, “2”, etc.) was pressed on the keyboard. You also

sent the message `asLowercase` so that you do not have to look for both lowercase and uppercase key characters (“a”, “A”, “b”, “B”. etc.). Note that, because of the mouse event code above that you created, this method will only be executed when the mouse cursor is over the game screen.

11. The last line in your code is sending the keystring event to the ship instance. Your `ShooterGame` class is not doing anything with keyboard events, so the keystrings are only being sent to your ship instance to handle, for now.

12. Now it is time to add a code change to your `Ship` class. Type and save the following code.

keystroke: keyString

```
keyString = '<left>' ifTrue: [self moveLeft].
keyString = '<right>' ifTrue: [self moveRight].
keyString = '<up>' ifTrue: [self moveUp].
keyString = '<down>' ifTrue: [self moveDown]
```

13. Look at the code that you entered. Can you guess what each line is doing? Note that Squeak uses “<left>” for the left arrow key since there is no key string character like “a”, “b”, etc. to use. The other arrow keys are named similarly.

14. You could have handled these keyboard events in your `ShooterGame` class, but it would be good for the ship to handle the keystrokes that it needs, instead of another object.

15. Save and Quit your Smalltalk image.