

CISC 110 Lab 4

General Instructions for All Labs/Assignments: Same instructions as Lab 2!

Specific Instructions for Lab 4

In this lab, you will complete the first portion of Assignment 4. In your assignment, you will create a game in which the player gains points by hitting an outer space monster with a spaceship via arrow keys. The monster gains points during the time gaps when the player is not hitting it.

In the lab portion, you will create the first step, a spaceship controlled by arrow keys and a score only for the player, not the monster; therefore the player always wins and the game is boring! Create a very simple version of a spaceship during the lab; you can improve it while working on the assignment. In the assignment portion, you will finish the game. For a similar example, look at the Wk5Pacman examples on the CISC 110 web page in the Schedule table under *ActionScript and Flash Examples*.

There are many useful ActionScript methods, too many to cover all of the ones you will use during the lectures, so some will be introduced in the labs and assignments. One that you will use in this lab is a collision-detection method, used to check when two objects hit each other (overlap on the stage), called `hitTestObject`. To check if two MovieClips, with instance names of `obj1` and `obj2` have collided, use the following test:

```
if ( obj1.hitTestObject( obj2 ) )
{    // Code saying what to do when they hit goes here
}
```

For an example in your text, see *5.5 Case Study 1*, page 245. The test is on page 252.

Here are the required functions:

Function Name	Description
<code>Assign4</code>	The main function as well as the class constructor. This function is responsible for adding a listener for <code>KEY_DOWN</code> events and creating a custom score property for the spaceship (player), which it initializes to 0.
<code>moveShip</code>	The event handler for a <code>KEY_DOWN</code> event. This function rotates and moves the ship according to the arrow key pressed, and it updates the score

Here are the required display objects:

Object Identifier	Description	Type of Object
ship	MovieClip instance: This is the spaceship instance controlled by arrow keys.	MovieClip
monster	MovieClip instance: This symbol is a static image that just sits there waiting to be hit. In the assignment portion, it will be moved randomly to make it harder to hit and will also have its own custom score property.	MovieClip
scoreBox	TextField instance: A dynamic text field in which the player's score is displayed	TextField

Here are the required properties for the MovieClip instances:

MovieClip Instance	Description
ship	score: the current score of the spaceship (player)

Here are the steps for you to complete:

1. Create a new folder called `Assign4` in your CISC 110 folder. Then create a new ActionScript 3.0 file called `Assign4.fla` and save it in your `Assign4` folder (Create New | ActionScript 3.0). Create your display objects (Steps 2-4).
2. Insert a new MovieClip symbol and create a simple spaceship, which you can improve upon later, for instance just a triangle. Drag an instance of it onto the stage and give it the instance name `ship`.
3. Insert a new MovieClip symbol and create a simple monster, which you can improve upon later, for instance just a circle. Drag an instance of it onto the stage and give it the instance name `monster`.
4. Insert a new dynamic TextField and give it an instance name of `scoreBox`.
5. Specify your document class to be `Assign4`. This tells your Flash file (`.fla`) to link to your ActionScript file (`.as`). Next write your ActionScript code (Steps 6-9).
6. Download the file `Assign4.as` from the CISC 110 website under Lab 4. This contains a starting template for your program script. Notice that an additional package is imported to give you access to the Keyboard class, which has the definitions of the constants `LEFT`, `RIGHT`, `UP`, and `DOWN`.
7. Make the changes below to your `Assign4.as` file. After each change, save your `.as` file and run your movie. Use `trace` statements wherever useful to check the values of variables.

8. Within your constructor function, `Assign4`, complete the tasks specified in the comments:
 - // TASK 1: Create a custom score property for the ship and initialize it to 0
 - // TASK 2: Add a `KEY_DOWN` event listener to allow the player to move the ship
9. Within your `KEY_DOWN` handler function, `moveShip`, complete the tasks specified in the comments:
 - // TASK 1: Check which key has been pressed. If it's an arrow key, rotate and move the ship in that direction
 - // TASK 2: Check if player has hit target. If so, add 1 to the player's score and display it in the text field.
10. Run your movie.

Lab 4 Marking Scheme (1% of final mark)

Marked out of 5:

1 mark - Two `MovieClip`, named `ship` and `monster`, and a dynamic `TextField` called `scoreBox` created.

1 mark - A custom property for `ship` called `score` is created and initialized to 0.

2 marks - `KEY_DOWN` handler written that checks key that has been pressed and rotates and moves the ship in the appropriate direction if the key is an arrow key

1 mark - `KEY_DOWN` handler also checks if the ship has collided with the monster, and if so adds 1 to the ship's (player's) score.