CISC 110 Assignment 4

In this 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 moves randomly and gains points during the time gaps when the player is not hitting it. You stop whenever you feel like it and compare the scores, both of which are displayed in a text field, increasing as the game is played. The game is played on a space scene background that you create, which includes composite movie clips with their own timelines, so that parts of them move, but that can be done just via tweens on their timelines, rather than via ActionScript.

In order to move the monster randomly, you will use the `Math.random` method twice for each move, once to generate a random x position on the screen and once to generate a random y position on the screen. In order for these positions to be on the screen, you need to generate one integer between 0 and the stage width and one integer between 0 and the stage height. To do this, use the predefined properties for the stage: `stage.stageWidth` and `stage.stageHeight`. Multiply the result from `Math.random` by the one of these you want and then round it with `Math.round` to get an integer in the correct range.

Below are the main steps for your assignment.

1. In the lab, create a simplified version in which only the player's spaceship moves and only the player scores points. For details, see the `Lab 4` instructions.

2. Design your space game.

3. Modify your work from your lab by improving the images you use for your spaceship and monster. Also, create a space scene background.

4. Add one or more composite MovieClips to your scene that have their own timeline, so that parts or all of them move when placed on the single frame in the main timeline. For instance, you might have an asteroid that moves in a rotating path and gets larger when it's "closer" to the viewer or you might have another space monster that has appendages or eyes that are moving as it hovers in one spot. You do not need to control these timelines via ActionScript, but you could if you want to.

5. Add a custom score property to your monster and initialize it to 0.

6. Change your if statement in your `KEY_DOWN` handler by adding an else clause, in which you add one to the monster's score every time the handler is called and the player has NOT hit the monster. Update the score on the screen each time the handler is called. You also might want to add 10 or more points to the player's score for each hit, instead of only one, to give the player a greater chance of winning.

7. Make your monster move randomly. Add a frame loop listener and a frame loop handler function. Within your handler function, randomly generate an integer between 0 and `stage.stageWidth` and assign it to the x position of your monster. Also randomly generate an integer between 0 and `stage.stageHeight` and assign it to the y position of your monster.

8. Run your movie and notice that the monster moves too fast. In order to slow down your monster so the player has a chance to hit it, only move it once every 50 or so frames - you decide how often. To do this, create a count variable that you use to count the number of times the frame loop handler is executed and use an if statement to move the monster only if the count is greater than 50. When it is, set the count back to 0. You will need to create a global count variable, a variable that exists throughout your program. To do so, add the following line immediately BEFORE the header line of your constructor function: before `public function Assign4( )` but after `public class Assign4 extends MovieClip {,` so that it is inside the class but before the constructor.

```
    var count = 0;
```

Then inside your frame loop handler function, add an if statement that checks if the count is greater than 50. If it is, move the monster and set the count back to 0. If it is not, add 1 to the count.

9. Adjust the scoring rate for the player and monster, the frame rate of your movie and the count value at which you move the monster to get a game to your liking that moves and scores the spaceship and monster at fun, challenging rates.

10. Once your space game is working the way you'd like, publish it and upload it to the CISC 110 web space. You need to upload two files: your .swf and .html files. Do not upload your .fla or .as files.

OPTIONAL VERY FUN EXTRA that will make your game much better and way more fun!

Notice that the spaceship reacts slowly and moves jerkily and slowly. Also, it would be great if your spaceship could move diagonally.

Change your spaceship to be able to react faster and move more smoothly by using four global Boolean variables to keep track of which arrow key has been pressed down: leftArrow, rightArrow, upArrow, or downArrow. Use the KEY_DOWN handler only to determine a switch in direction, at which point you will set one of the Boolean variables to true. Use a KEY_UP handler to determine when the player stops pressing an arrow key, at which point you set one of the Boolean variable to false. Use a frame loop handler to actually move the spaceship and update the scores. Modify the if-else statement to be four if statements with no else clauses. Then if the user holds down both the left arrow and up arrow key, both the leftArrow and upArrow Boolean variables will be set to true, so the spaceship will move diagonally up and to the left. If the user holds down both the left and right arrow keys, the spaceship won't move, since it's being told to move in two opposite directions.

Follow the Wk5EnhancedPacman example on the CISC 110 web page in the Schedule table to do all of this. No extra marks for adding this, just LOTS of satisfaction!

Assignment 4 Marking Scheme (2% of final mark)

Marked out of 10:

2 marks: One or more composite movie clips that have their own timelines so that parts or all of them move when placed on the single frame in the main timeline.

1 mark: Nice background space scene and spaceship and monster graphics.

1 mark: Custom score property added for monster and increased during periods when player doesn't make a hit.

3 marks: Monster moves randomly via a frame loop and use of Math.random( )

2 marks: Monster moves only once every 50 frames or so via use of a count variable

1 mark: Assignment 3 is published and uploaded on the CISC 110 website (`.swf` and `.html` files).