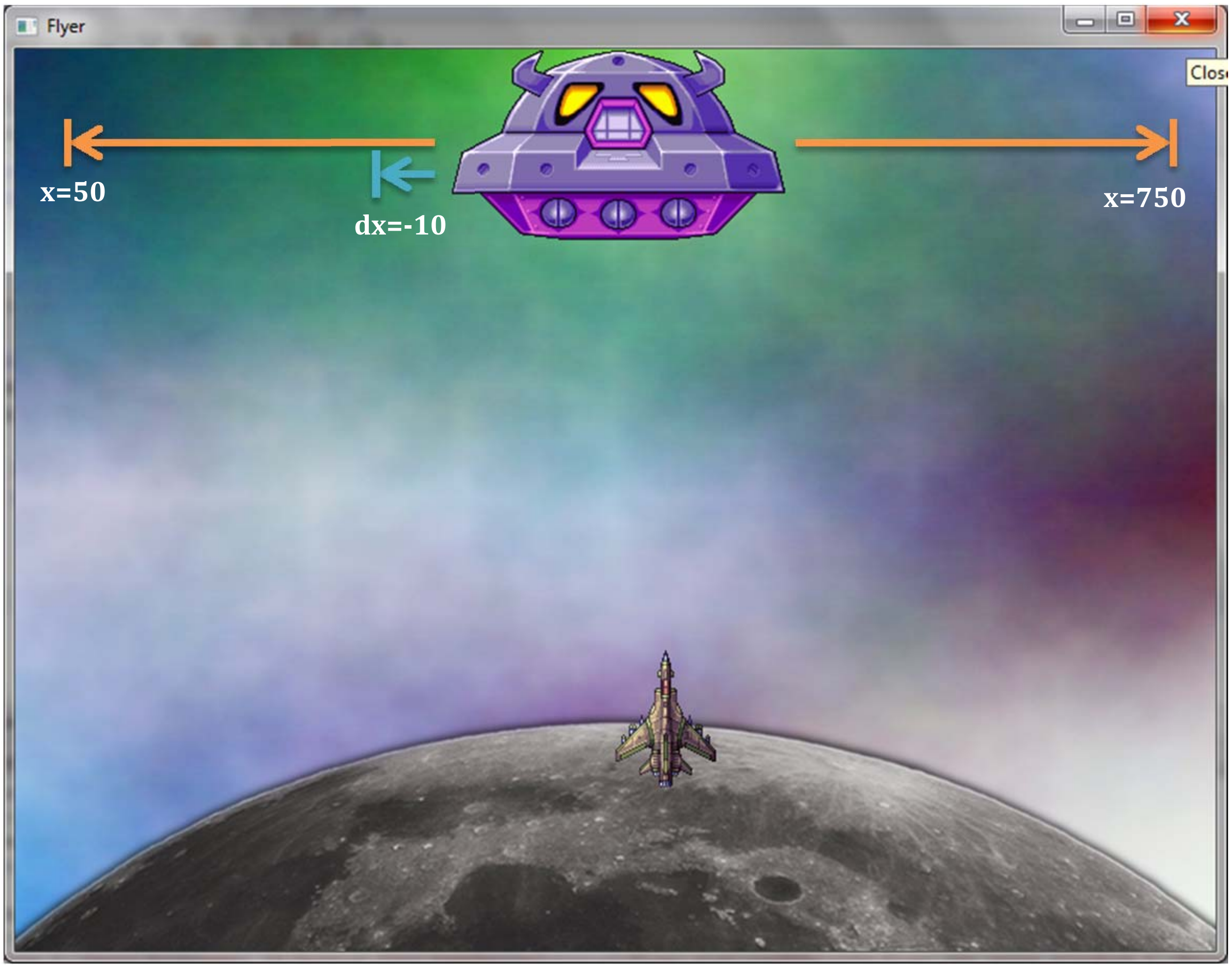
PART TWO – USING A TIMER

Many applications need to make use of a timer. JavaFX provides a powerful timer that can be used for a range of purposes in a class called AnimationTimer.

We will be using an AnimationTimer in our main class to control the actions of the Baddie, and later on the missile.

This section will focus on automating the enemy so it move from side to side as indicated below.



MOVING THE BADDIE

Add a move() method to your Baddie class. Looking at the diagram above, we want the sprite to move until the left edge passes x=50, then move in the following direction.

```
public void move(){
    if (x<50){
        dx=-dx;
    }
    if (x>700){
        dx=-dx;
    }
    x= x-dx;
    ivBad.setLayoutX(x);
}
```

This works in a similar manner to the moveLeft() and moveRight() methods for the rocket. The if statements check the current value of x. If the left edge is has gone passed the boundary then the sign of dx is reversed.

dx is then subtracted from x and the x-position of the sprite is set.

The method is now ready to be called from a timer to automate its movement.

CREATING AND USING ANIMATIONTIMER

The timer will be setup in the main class for your game. To use an AnimationTimer one must import the following package:

```
import javafx.animation.AnimationTimer;
```

The code below will add a timer to your code. It creates an anonymous inner class to handle the call which is executed on each frame.

```
    new AnimationTimer(){
            @Override
            public void handle(long now){
                //actions go here
            }
    }.start();

}
```

The commented line indicates where the programmer can add actions to be carried out each time the call is generated. You can add a reference to the move method of your baddie.

```
baddie.move();
```

You can fine-tune the movement of the baddie by adjusting the boundaries of movement and dx (to alter the speed).