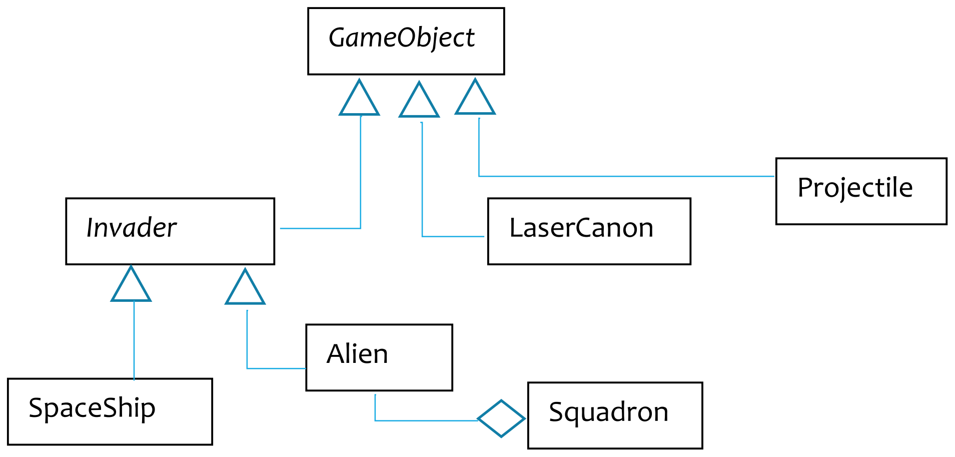
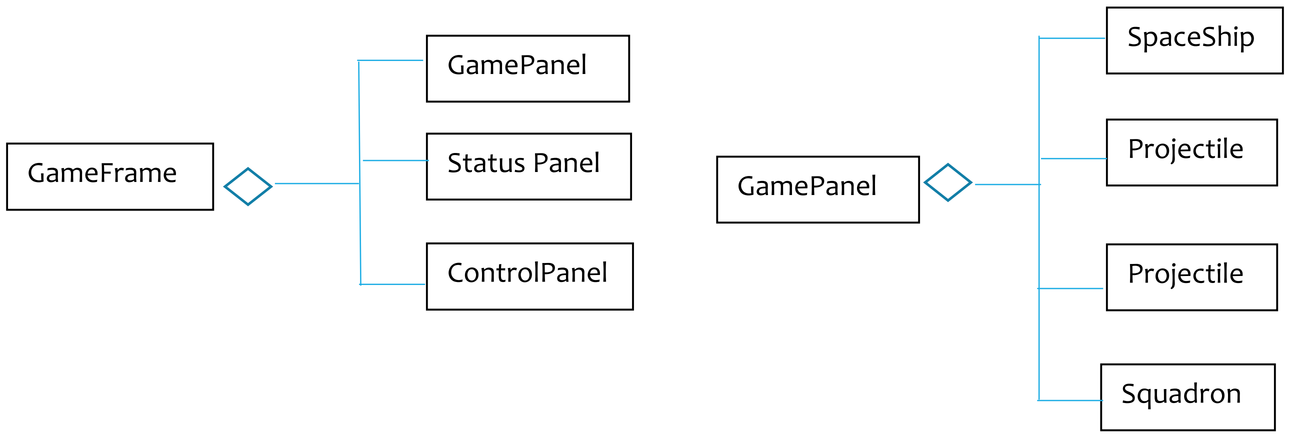
GEEN165 - Major Programming Assignment 3 – Fall 2015

1 Introduction

This assignment requires you to create a Java game application similar to the video classic named Space Invaders. You are required to use my design or you will not receive credit for the program. Some deviation is okay but discuss it with me first.





2 Classes

You are required to implement the following classes at a minimum. You may add other classes (and methods) if you need them.

2.1 GameObject Class (Abstract)

GameObject	
-visible : boolean -location : Point -color : Color	Is the object currently visible? Current location of the objects reference point Current color of the object
+GameObject () +GameObject(//all Engine properties) +//Accessor and Mutators *+draw(g : Graphics) : void* +toString():String	Abstract method implemented in each concrete subclass

2.2 Invader Class (Abstract)

Serves as a superclass for all Alien Invaders.

Invader	
-pointValue : int -speed : int -direction : Direction -height : int -width : int -rand : Random	Points awarded when hit with projectile Pixels moved when associated timer expires Direction of movement Height in pixels Width in pixels Used in various places to generate random values.

+Invader()
+//Getters and setters
+toString()	

2.3 SpaceShip Class

This class defines the ship the moves across the top of the screen at random intervals. The Spaceship will have two Timer properties. One timer to control launching the ship a random intervals between 5 and 25 seconds and another Timer to control the Spaceship animation. The Spaceship can be launched moving east to west or moving west to east.

SpaceShip	
-moveTimer : Timer -launchTimer : Timer -gamePanel : GamePanel -rand : Random	Controls the ship movement Controls when the SpaceShip is launched Needed to get squadron movement boundaries and repaint after squadron updates. Used to launch ship at random times, give the spaceship a random point value between 50 and 300 (50 point increments) and randomly select ship direction.

+SpaceShip()
+startMoveTimer()
+stopMoveTimer()
+startLaunchTimer()
+stopLaunchTimer()
+draw(g : Graphics) : void
+toString():String	

2.4 Projectile Class

Projectiles are fired by the LaserCanon. Only one projectile will be active

Projectile	
-speed : int	Pixels moved when associated timer expires
+Projectile() +//Accessor and Mutators +draw(g : Graphics) : void +toString():String	

2.5 LaserCanon Class

Base station for firing projectiles. Only one projectile can be active at a time. The station is moved using the keyboard arrow keys and the projectile is fired using the spacebar.

LaserCanon	
-pnlWidth : int -pnlHeight : int -moveInc : int	GamePanel width GamePanel height Number of pixels moved when arrow key is pressed
+LaserCanon(pnlWidth : int, pnlHeight:int)	Includes the gamePanel dimensions as parameters

+//Accessor and Mutators +draw(g : Graphics) : void +toString():String	

2.6 Alien Class

One of the invaders in the large 2-D array of the Squadron

Alien	
-alienType : int	Determines which alien is displayed.
Alien() +//Accessor and Mutators +draw(g : Graphics) : void +toString():String	

2.7 Squadron Class

2-D array of Alien objects.

Squadron	
-aliens : Alien[5][12] -direction : Direction -moveTimer : Timer -numVisible : int -gamePanel : GamePanel	2-D array of invading aliens Current direction of the squadron of aliens Controls movement of squadron Current number of aliens that have not been destroyed Needed to get squadron movement boundaries and repaint after squadron updates.
+Squadron() +draw(g : Graphics) : void +//Accessor and Mutators +toString():String	Put each property on a separate line.

2.8 GamePanel

This is the heart of your application. It will include all the major game objects and the keyboard listener.

GamePanel	
-gameObjects : ArrayList<GameObject> -spaceShip : SpaceShip -laserCanon : LaserCanon -projectile : Projectile -projectileTimer : Timer -squadron : Squadron	Game object that need displaying in paintComponent

+GamePanel()
+addGameObject(go : GameObject)
	
	Add a GameObject to the gameObjects arrayList

+//Accessor and Mutators #paintComponent(g : Graphics) : void +toString():String	 Call draw() for each object in gameObjects that is visible.

Note: Your toString() methods can simply list the properties of the class on a single line separated by a comma.

3 Grading

If your project does not compile, it receives a grade of zero. If you do not document your program according to the documentation guidelines, the graders have been instructed to deduct **up to 25%.**

Level 1 (30%): implement the GameObject and LaserCanon classes. You should be able to move the canon laterally across the screen to the bounds of the GamePanel using the left and right arrow keys.

Level 2 (40%): Implement the Projectile class along with its needed Timer and ActionListener. You should now be able to move the LaserCanon and fire projectiles.

Level 3 (55%): Implement the Invader and SpaceShip classes. Your projectile should be able to detect when it makes contact with the spaceship. When there is contact, the spaceship and projectile will disappear.

Level 4 (60%): Add an additional Timer such that pointsValue of the spaceship appears for a couple of seconds after a projectile hits the spaceship.

Level 5 (80%): Add the StatusPanel and the ControlPanel to the GameFrame. The StatusPanel should display the current number of points earned and other desired status information. The ConrolPanel should allow the user to use the mouse to start a game, restart a game and exit the application.

Level 6 (100%): Implement the Alien and Squadron classes. You should create three different types of Aliens by using three different images. The bottom two rows of Aliens have the same image and are worth 10pts each. The next two rows of Aliens are worth 20pts and the final row of Aliens are worth 40pts. Your projectile can only destroy the lowest Alien in each column. The Squadron should move faster after every 4 calls to the movement timer and drop down lower once it reaches the boundary of the GamePanel. Once all the Aliens have been destroyed, the Squadron should be reset; however it should start at one vertical level lower than the previous level. You have to determine the lowest level that the Squadron can start based on the height of your LaserCanon. The game is over once any Alien reaches a position below the height of the LaserCanon.

Extra Credit:

(10%) Implement projectiles originating from the Aliens toward the ground. The game will start with 3 lives (LaserCanons) . Once the 3 LaserCanons have been destroyed or the Aliens have reached the ground, the game is over. Show some indication of the remaining lives in your StatusPanel.

Submission

Create a file named Readme in the root directory of your project that describes what grade level you fill you completed. If there is any other information the grader should know about your assignment, you should include it in this file. Archive your entire NetBeans project with a zip tool and uploaded it using the assignment upload link.