

SpaceInvaders Sample Design

CSCI 134

This is one way to structure the program, but certainly not the only way. It covers the main properties of the game, but is not a complete implementation --- there may be additional useful methods, instance variables, constants, etc. to include in the code.

For instance, note that this design assumes projectiles are displayed as Lines. You might opt for FilledRects. If so, some other aspects of our design would need to be changed.

Feel free to use ideas from this design in addition to your original designs while working on the code.

class SpaceShip

Instance Vars:

- `static final int STEP_SIZE = 5` how much to move left/right
- `static final int WIDTH = 35` ship width
- `static final int HEIGHT = 15` ship height

- `FilledRect ship` ship body
- `FilledRect shooter` shooter part
- `Invaders invaders` the invading aliens
- `ScoreKeeper scoreKeeper` need to pass to defensive missiles
- `DrawingCanvas canvas` needed for new launching missiles

Constructor:

```
public SpaceShip(Location upperLeft, DrawingCanvas aCanvas,  
                  ScoreKeeper aScoreKeeper)
```

Create a ship at the given point with the given picture. Remember the canvas and scorekeeper for later.

Methods:

- `public void setTarget(Invaders someInvaders)`
Remember which invaders we're shooting at.

- `public void moveLeft()`

- `public void moveRight()`
move to the left/right by `STEP_SIZE`.
- `public void shoot()`
launch a new missile.
- `public void explode()`
end the game --- invoke when the ship is hit by a missile.
- `public boolean contains(Location point)`
return true if the ship body contains the point

class Alien

Instance Vars:

- | | |
|--|---|
| • <code>VisibleImage alien</code> | the alien image on the canvas |
| • <code>boolean isAlive</code> | has it been hit by a defensive missile |
| • <code>SpaceShip ship</code> | the ship it's trying to hit with missiles |
| • <code>ScoreKeeper scoreKeeper</code> | notify when when we're hit |
| • <code>DrawingCanvas canvas</code> | needed for new launching missiles |

Constructor:

```
public Alien(Image pic, double x, double y,
             DrawingCanvas aCanvas, SpaceShip aShip,
             ScoreKeeper aScoreKeeper)
```

Create an alien at the given (x,y) location with the provided image. Remember the canvas, ship, and scorekeeper for later.

Methods:

- `public void move(double dx, double dy)`
Move alien by given offsets.
- `public boolean isAlive()`
Return true if the alien has not been hit yet.
- `public void explode()`
Invoke when this alien is hit by a missile. Hide its image, change `isAlive` instance variable, and notify score keeper.

- `public double getX()`
get x-coordinate of alien.
 - `public double getWidth()`
 - `public double getHeight()`
get width and height of alien
 - `public void shoot()`
create a new `Projectile`
 - `public boolean contains(Location point)`
return true if the alien contains the point
-

class Invaders extends ActiveObject

Instance vars:

- `double speed` Speed for moving aliens
- `Alien[][] invaders` 2d array of alien invaders
- `SpaceShip ship` The good-guy ship
- `DrawingCanvas canvas` Canvas for new missiles
- `ScoreKeeper scoreKeeper` Needed to keep track of end-of-game
- `boolean movingRight` Which way the aliens are moving
- `RandomIntGenerator xGenerator, yGenerator` For picking aliens to shoot

Constructor:

```
public Invaders(Image alienPic, Location loc, int cols, int rows,
               int spacing, DrawingCanvas aCanvas, double aSpeed,
               SpaceShip aShip, ScoreKeeper aScoreKeeper)
```

Create the aliens array and all the aliens so that they appear in a grid at the given location. The array should be cols wide and rows high. Initialize the other instance variables with the values passed to the constructor. Create new generators for picking shooters. Call `start()`.

Methods:

- `public void run()`
while (the game is not over) {
 If the aliens are as far left/right as they can go, change direction (`movingRight`).
 Move each invader left or right, depending on the `movingRight` instance variable
 Pause

If (all aliens are dead) tell the scorekeeper the game is over
}

Use helper methods to move every alien one step, and to count how many aliens are still alive. (And any others that may be useful.)

- `boolean tryToHitAlien(Location point)`
If point is within a living alien, kill that alien, and return true. Remember: don't remove the segment from the array --- just tell it to hide its image. Return false if no aliens are hit.

class Projectile extends ActiveObject

Instance vars:

- `Line missile` missile on the canvas
- `double bottom` Stop when we go past the bottom
- `SpaceShip ship` The good guy

Constructor:

```
public Projectile(double x, double y, double height,  
                  DrawingCanvas canvas, SpaceShip aShip)
```

Create the missile line at the given location with the given height, and set up instance variables. Call `start()`.

Methods:

- `public void run()`
Move the line down the screen until it goes off the bottom or hits the ship
Then remove line from screen, and make the ship explode if appropriate.

class DefensiveProjectile extends ActiveObject

Instance vars:

- `Line missile` missile on the canvas
- `Invaders invaders` The bad guys

Constructor:

```
DefenseProjectile(double x, double y, double height,
```

```
DrawingCanvas canvas, Invaders invaders)
```

Create the missile line at the given location with the given height, and set up instance variables. Call start().

Methods:

- `public void run()`
Move the line up the screen until it goes off the top or hits an alien
Then remove line from screen.

class ScoreKeeper

Instance vars:

- `int score` current score
- `JLabel message` label to display on bottom of screen
- `boolean gameOver` Is the game over?

Constructor:

```
public ScoreKeeper(JLabel theMessage)  
    Save the label we modify to change the score / message at the bottom.
```

Methods:

- `public void addPoints()`
Add points for hitting alien.
- `public void gameOver()`
End the game.
- `public boolean isGameOver()`
Has the game ended?

class SpaceInvaders extends WindowController implements KeyListener

Instance vars:

- Constants for game setup --- change as you like :

- `Location ALIEN_LOC = new Location(10,10)`
- `int ALIEN_ROWS = 4;`
- `int ALIENS_PER_ROW = 8;`
- `int ALIEN_SPACING = 20;`
- `double ALIEN_SPEED = 5;`
- `ScoreKeeper scoreKeeper .`
- `SpaceShip ship`
- `Invaders invaders`

Methods:

- `public void begin()`
Create the black background and the objects stored in the instance variables. Also create the JLabel for the south part of the screen that will be used by the scorekeeper.
- `public void keyTyped(KeyEvent event)`
Tell the ship to move/shoot in response to the various keystrokes.
If the scorekeeper says that game is over, do nothing.