

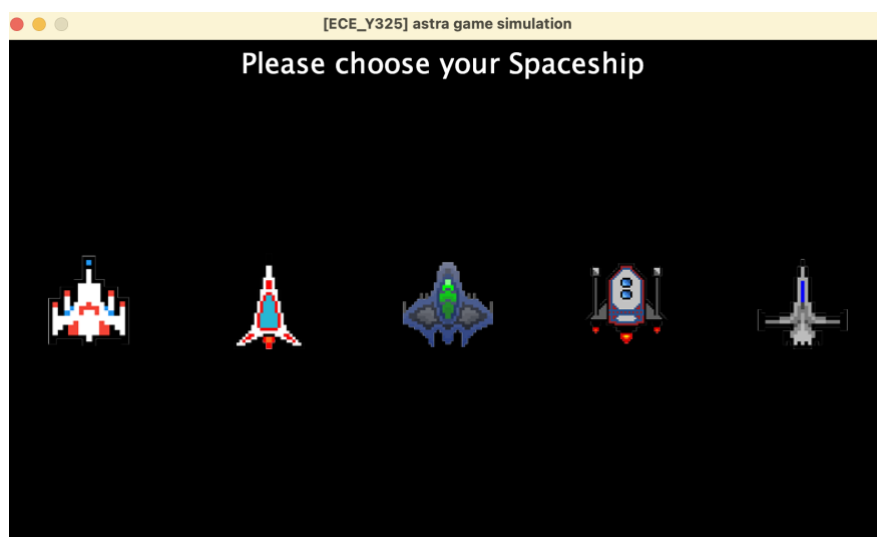


Άσκηση 7η

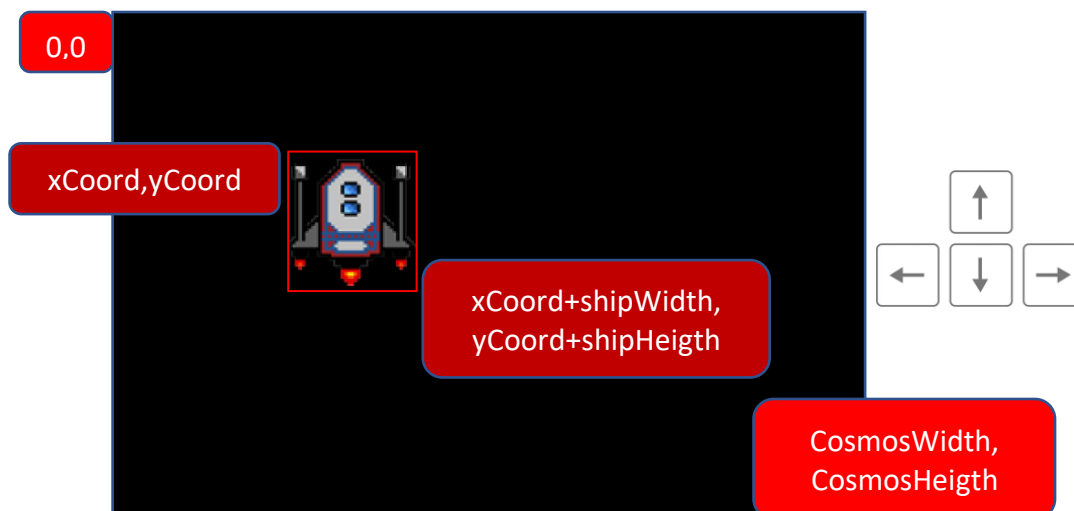
Ο σκοπός της 7^{ης} άσκησης είναι η απόκτηση εμπειρικής γνώσης σχετικά με τον χειρισμό γεγονότων (*Event Handling*) και την δημιουργία γραφικών διεπαφών χρήσης. Χρησιμοποιήσετε κλάσεις όπως: *JFrame*, *JPanel*, *JButton*, *JLabel*, *Image*, *ImageIcon*. Υπενθυμίζουμε ότι η 7^η άσκηση είναι η δεύτερη από μία σύνθετη εργασία τριών ασκήσεων (6^η, 7^η και 8^η) που έχουν ως στόχο το σχεδιασμό και την υλοποίηση ενός παιχνιδιού τύπου *arcade spaceship shooting game*. Σημειώστε ότι ενδεικτικός κώδικας για την άσκηση υπάρχει στις διαφάνειες της εβδομάδας.

Ζητούμενα της 7^{ης} Άσκησης:

A. Δημιουργήστε το πρώτο *panel* επιλογής του διαστημοπλοίου. Εμφανίζονται τα πέντε διαστημόπλοια και ο χρήστης πρέπει να επιλέξει με αριστερό κλικ στο ποντίκι του υπολογιστή πάνω στο αντίστοιχο εικονίδιο το διαστημόπλοιο της επιλογής του.



B. Αφού ο χρήστης επιλέξει το διαστημόπλοιο εμφανίζεται η παρακάτω οθόνη στην οποία ο χρήστης μπορεί να μετακινεί το διαστημόπλοιο με χρήση των πλήκτρων (up/down/left/right *arrows*).

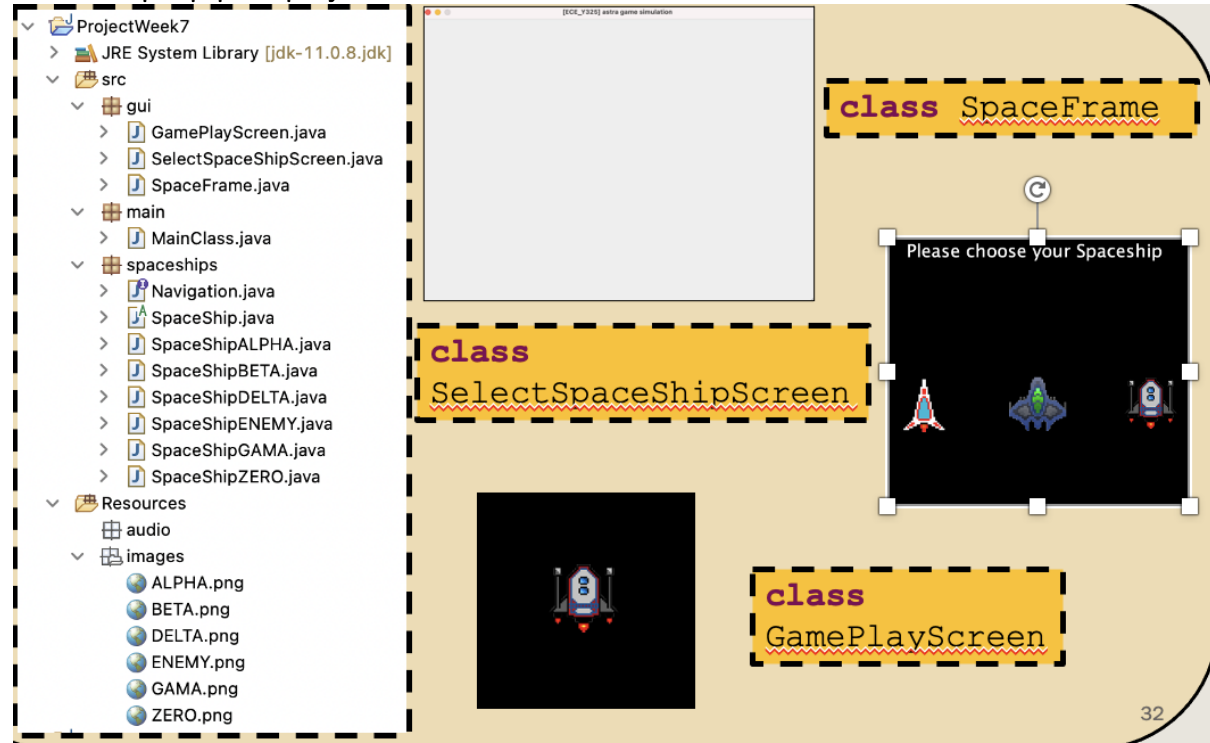




ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΣΕ Υ325: ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗΣ ΤΕΧΝΟΛΟΓΙΑ

Μπορείτε να βρείτε στο εclass τον κατάλογο πολυμεσικού υλικού τον οποίο μπορείτε να χρησιμοποιήσετε για τις εικόνες των διαστημοπλοίων.

Ενδεικτική δομή του project:



Ενδεικτικός κώδικας της GameplayScreen:

```
public class GameplayScreen extends JPanel implements KeyListener{
    private static final long serialVersionUID = 1L;

    private SpaceShip userSpaceShip;
    GameplayScreen(){
        addKeyListener(this);
        this.setVisible(true);
        this.setBackground( Color.BLACK );
    }
    @Override
    public void paintComponent (Graphics g) {
        super.paintComponent(g);
        userSpaceShip.getIcon().paintIcon(this, g, userSpaceShip.getX(), userSpaceShip.getY());
    }
    @Override
    public void keyPressed (KeyEvent e) {
        if (e.getKeyCode() == KeyEvent.VK_UP) userSpaceShip.moveUP();
        if (e.getKeyCode() == KeyEvent.VK_DOWN) userSpaceShip.moveDOWN();
        if (e.getKeyCode() == KeyEvent.VK_LEFT) userSpaceShip.moveLEFT();
        if (e.getKeyCode() == KeyEvent.VK_RIGHT) userSpaceShip.moveRIGHT();
        this.repaint();
    }
    @Override
    public void keyReleased (KeyEvent e) {}
    @Override
    public void keyTyped (KeyEvent e){}
    void setUserSpaceShip(SpaceShip usel) {
        userSpaceShip=usel;
    }
}
```



Ενδεικτικός κώδικας της Main:

```
package main;
import gui.SpaceFrame;
public class MainClass {
    public static int cosmosWidth=800;
    public static int cosmosHeight=600;
    public static int spaceShipWidth=100;
    public static int spaceShipHeight=100;
    public static int yOffset=30;
    public static SpaceFrame myShootingGame;
    public static void main(String[] args) {
        myShootingGame=new SpaceFrame(cosmosWidth,cosmosHeight);
    }
}
```

Ενδεικτικός κώδικας της SpaceFrame:

```
public class SpaceFrame extends JFrame{
    private static final long serialVersionUID = 1L;
    static int width;
    static int height;
    static SelectSpaceShipScreen selectSpaceShipScreen=new SelectSpaceShipScreen();
    static GamePlayScreen gamePlayScreen=new GamePlayScreen();
    static CardLayout cardLayout=new CardLayout();
    static JPanel spaceFramePanel=new JPanel();
    public SpaceFrame(int width, int height) {
        SpaceFrame.width=width;
        SpaceFrame.height=height;
        this.setSize(width, height);
        this.setTitle("[ECE_Y325] astra game simulation");
        this.setDefaultCloseOperation(EXIT_ON_CLOSE);
        this.setLocationRelativeTo(null);
        this.setFocusable(true);
        setupMasterPanel();
        setResizable(false);
        this.setVisible(true);
    }

    void setupMasterPanel() {
        spaceFramePanel.setLayout(cardLayout);
        spaceFramePanel.add(selectSpaceShipScreen);
        spaceFramePanel.add(gamePlayScreen);
        this.add(spaceFramePanel);
    }
}
```