```
using System;
using System.Collections.Generic;
using System.Linq;
using Microsoft.Xna.Framework;
using Microsoft.Xna.Framework.Audio;
using Microsoft.Xna.Framework.Content;
using Microsoft.Xna.Framework.GamerServices;
using Microsoft.Xna.Framework.Graphics;
using Microsoft.Xna.Framework.Input;
using Microsoft.Xna.Framework.Media;

namespace SpaceInvadersN
{
    /// <summary>
    /// This is the main type for your game
    /// </summary>
    public class Game1 : Microsoft.Xna.Framework.Game
    {
        GraphicsDeviceManager graphics;
        SpriteBatch spriteBatch;
        Rectangle[] prviTalas;
        Rectangle[,] drugiTalas, treciTalas;
        Texture2D alienOne, alienTwo, alienThree, bullet, ship;
        Rectangle rectAlienOne, rectAlienTwo, rectAlienThree, rectBullet, rectShip;
        int aliens = 10;
        int red = 2, kolona = 10;


        Display hud;

        public Game1()
        {
            graphics = new GraphicsDeviceManager(this);
```

```
    Content.RootDirectory = "Content";
}

/// <summary>
/// Allows the game to perform any initialization it needs to before starting to run.
/// This is where it can query for any required services and load any non-graphic
/// related content.  Calling base.Initialize will enumerate through any components
/// and initialize them as well.
/// </summary>
protected override void Initialize()
{
    // TODO: Add your initialization logic here

    base.Initialize();
}

/// <summary>
/// LoadContent will be called once per game and is the place to load
/// all of your content.
/// </summary>
protected override void LoadContent()
{
    // Create a new SpriteBatch, which can be used to draw textures.
    spriteBatch = new SpriteBatch(GraphicsDevice);

    alienOne = Content.Load<Texture2D>("alienOne");
    prviTalas = new Rectangle[aliens];
    for (int i = 0; i < aliens; i++)
    {
        prviTalas[i].Width = alienOne.Width;
        prviTalas[i].Height = alienOne.Height;
        prviTalas[i].X = 60 * i;
```

```
        }
        alienTwo = Content.Load<Texture2D>("alienTwo");
        drugiTalas = new Rectangle[red, kolona];
        for (int r = 0; r < red; r++)
            for (int k = 0; k < kolona; k++)
            {
                drugiTalas[r, k].Width = alienTwo.Width;
                drugiTalas[r, k].Height = alienTwo.Height;
                drugiTalas[r, k].X = 60 * k;
                drugiTalas[r, k].Y = (60 * r) + 60;
            }
        alienThree = Content.Load<Texture2D>("alienThree");
        treciTalas = new Rectangle[red, kolona];
        for (int r = 0; r < red; r++)
            for (int k = 0; k < kolona; k++)
            {
                treciTalas[r, k].Width = alienThree.Width;
                treciTalas[r, k].Height = alienThree.Height;
                treciTalas[r, k].X = 60 * k;
                treciTalas[r, k].Y = (60 * r) + 180;
            }
        ship = Content.Load<Texture2D>("ship");
        rectShip.Width = ship.Width;
        rectShip.Height = ship.Height;
        rectShip.X = GraphicsDevice.Viewport.Width / 2 - 35;
        rectShip.Y = 442;
    }

    /// <summary>
    /// UnloadContent will be called once per game and is the place to unload
    /// all content.
    /// </summary>
    protected override void UnloadContent()
```

```
{
    // TODO: Unload any non ContentManager content here
}

/// <summary>
/// Allows the game to run logic such as updating the world,
/// checking for collisions, gathering input, and playing audio.
/// </summary>
/// <param name="gameTime">Provides a snapshot of timing values.</param>
protected override void Update(GameTime gameTime)
{
    // Allows the game to exit
     KeyboardState keys = Keyboard.GetState();

    if (keys.IsKeyDown(Keys.Escape))
        this.Exit();

    // TODO: Add your update logic here

    base.Update(gameTime);
}

/// <summary>
/// This is called when the game should draw itself.
/// </summary>
/// <param name="gameTime">Provides a snapshot of timing values.</param>
protected override void Draw(GameTime gameTime)
{
    GraphicsDevice.Clear(Color.Black);

    spriteBatch.Begin();
    for (int i = 0; i < aliens; i++)
    {
```

```
                spriteBatch.Draw(alienOne, prviTalas[i], Color.White);
            }
            for (int r = 0; r < red; r++)
                for (int k = 0; k < kolona; k++)
                {
                    spriteBatch.Draw(alienTwo, drugiTalas[r, k], Color.White);
                }
            for (int r = 0; r < red; r++)
                for (int k = 0; k < kolona; k++)
                {
                    spriteBatch.Draw(alienThree, treciTalas[r, k], Color.White);
                }
            spriteBatch.Draw(ship, rectShip, Color.White);
            spriteBatch.End();

            base.Draw(gameTime);
        }
    }
}
```