Animation (Sprite Sheet) in MonoGame:

New Project --→ MonoGame --→ Windows --→Game --→ ok

Right click --→ add --→new class --→call it (Animation):

In the class add the code below:

```
Texture2D texture;
Rectangle rectangle;
Vector2 position;
Vector2 origin;
Vector2 velocity;

int currentFrame;
int frameHeight;
int frameWidth;

float timer;
float interval=75;

public Animation(Tecture2D newTexture, Vector2 newPosition, int
newFrameWidth, int newFrameHeight)
{
texture= newTexture;
position=newPosition;
frameHeight= newFrameHeight;
frameWidth=newFrameWidth;

}
Public void Draw (SpriteBatch spriteBatch)
{
// zero rotation
spriteBatch.Draw(texture,position,rectangle,Color.White,0f,origin,1.0f
,spriteEffects.None,0);
}

Public void Update (GameTime gameTime)
{
//set a rectangle
rectangle=new Rectangle(currentFrame*frameWidth,
0,frameWidth,FrameHeight)

// origin is the center of each image
```

```
      origin= new Vector2 (rectangle.Width/2, rectangle.Height/2);
position=position+velocity;
}

Public void AnimateRight (GameTime gameTime)
{

Timer+=(float)gameTime.ElapsedGameTime.TotalMilliseconds/2;
If (timer>interval);
{
// current frame goes up
currentFrame++;
//timer gets reset to zero
Timer=0;
// reaches three resets back to zero (loop) for animation
If(currentFrame>3)
currentFrame=0;
}
}

Public void AnimateLeft (GameTime gameTime)
{

Timer+=(float)gameTime.ElapsedGameTime.TotalMilliseconds/2;
If (timer>interval);
{
// current frame goes up
currentFrame++;
//timer gets reset to zero
Timer=0;
// reaches three resets back to zero (loop) for animation
If(currentFrame>7 || currentFrame<4)
currentFrame=4;
} }
```

In Update:

```
      if (Keyboard.GetState().IsKeyDown(Keys.Right))
{
AnimateRight(gameTime)
Velocity.X=3;

}
else if (Keyboard.GetState().IsKeyDown(Keys.Left))
{
AnimateLeft(gameTime)
```

```
Velocity.X=-3;
}
else velocity=Vector2.Zero;
}
```

Go back to Game1 clas and add the following code:

```
  Animation player;
```

In initialization:

```
    player = new Animation(Content.Load<Texture2D>("run"), new
Vector2(100,100), 47,44);

player.Draw(spriteBatch);
```

Completed code:

Animation class:

```
using Microsoft.Xna.Framework;
using Microsoft.Xna.Framework.Graphics;
using Microsoft.Xna.Framework.Input;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace SpriteSheet
{
   class Animation
   {
      Texture2D texture;
      Rectangle rectangle;
      Vector2 velocity;
      Vector2 position;
      Vector2 origin;

      float Interval = 75;
      float timer;

      int currentFrame;
```

```
      int frameHeight;
      int frameWidth;

      public Animation(Texture2D newTexture, Vector2 newPosition, int newFrameHeight, int
newFrameWidth)
      {
         texture = newTexture;
         position = newPosition;
         frameWidth = newFrameWidth;
         frameHeight = newFrameHeight;


      }


public void AnimateRight (GameTime gameTime)
         {
            timer+=(float)gameTime.ElapsedGameTime.TotalMilliseconds/2;
            if (timer>Interval)           {
// current frame goes up
               currentFrame++;
//timer gets reset to zero
               timer=0;
// reaches three resets back to zero (loop) for animatio
               if(currentFrame>3)
               currentFrame=0;
            }
         }



public void Update(GameTime gameTime)
{
   rectangle = new Rectangle(currentFrame * frameWidth, 0, frameWidth, frameHeight);
   origin = new Vector2(frameWidth / 2, frameHeight / 2);
   position = position + velocity;

   if (Keyboard.GetState().IsKeyDown(Keys.Right))
   {
      animateRight(gameTime);
      velocity.X = 3;
   }
   else if
      (Keyboard.GetState().IsKeyDown(Keys.Left))
   {
      animateLeft(gameTime);
      velocity.X = -3;
   }
   else velocity=Vector2.Zero;
}
public void animateRight(GameTime gameTime)
{
   timer += (float)gameTime.ElapsedGameTime.TotalMilliseconds / 2;
   if (timer > Interval)
   {
      // current frame goes up
      currentFrame++;
      //timer gets reset to zero
```

```
      timer = 0;
      // reaches three resets back to zero (loop) for animation
      if (currentFrame > 3)
         currentFrame = 0;
   }
}

         public void animateLeft (GameTime gameTime)
         {
            timer+=(float)gameTime.ElapsedGameTime.TotalMilliseconds/2;
            if (timer>Interval)
            {
// current frame goes up
               currentFrame++;
//timer gets reset to zero
               timer=0;
// reaches three resets back to zero (loop) for animation
               if(currentFrame> 7 || currentFrame < 4)
                  currentFrame=4;
            }
         }
         public void AnimateLeft(GameTime gameTime)
         {

            timer += (float)gameTime.ElapsedGameTime.TotalMilliseconds / 2;
            if (timer > Interval)
            {
               // current frame goes up
               currentFrame++;
               //timer gets reset to zero
               timer = 0;
               // reaches three resets back to zero (loop) for animation
               if (currentFrame > 7 || currentFrame < 4)
                  currentFrame = 4;
            }
         }

         public void Draw(SpriteBatch spriteBatch)
         {
            spriteBatch.Draw(texture, position, rectangle, Color.White, 0.0f, origin,
1.0f, SpriteEffects.None, 0);

         }

      }
   }
```

Game 1 class:

```
using Microsoft.Xna.Framework;
using Microsoft.Xna.Framework.Graphics;
using Microsoft.Xna.Framework.Input;

namespace SpriteSheet
{
```

```
    /// <summary>
    /// This is the main type for your game.
    /// </summary>
    public class Game1 : Game
    {
        GraphicsDeviceManager graphics;
        SpriteBatch spriteBatch;
        Animation player;
        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()

        {
            // locate the image at 100, 100 and frame height is 47, frame width is 44
            player = new Animation(Content.Load<Texture2D>("run"), new Vector2(100,100),
47,44);
            // 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);

            // TODO: use this.Content to load your game content here
        }

        /// <summary>
        /// UnloadContent will be called once per game and is the place to unload
        /// game-specific 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)
      {
         if (GamePad.GetState(PlayerIndex.One).Buttons.Back == ButtonState.Pressed ||
Keyboard.GetState().IsKeyDown(Keys.Escape))
            Exit();

         player.Update(gameTime);

         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();
         player.Draw(spriteBatch);
         spriteBatch.End();

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