

## MonoGame – Storlek & Rotation

Den här guiden förutsätter att man kan lägga in en bild i programmet.

Ett grundläggande sätt att ändra en bilds storlek är med rektanglar.

1. Deklarera en ny variabel av typen **Rectangle**.

Den ska användas som positionsangivelse när bilden ritas ut.

```
2 references
public class Game1 : Game
{
    private GraphicsDeviceManager _graphics;
    private SpriteBatch _spriteBatch;

    Texture2D bild;
    Vector2 position;

    Rectangle bildRektangel;
}
```

Figur 1

2. Tilldela rektangeln ett värde genom **new Rectangle()**.

Med argumenten:  
(x, y, bredd, höjd).

```
0 references
protected override void LoadContent()
{
    _spriteBatch = new SpriteBatch(GraphicsDevice);

    bild = Content.Load<Texture2D>("happy200");
    position = new Vector2(100, 100);

    bildRektangel = new Rectangle(100, 100, 450, 450);
}
```

Figur 2

3. Ändra så att bilden istället ritas ut med hjälp av rektangeln.

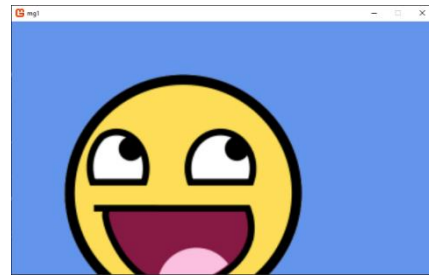
```
0 references
protected override void Draw(GameTime gameTime)
{
    GraphicsDevice.Clear(Color.CornflowerBlue);

    _spriteBatch.Begin();
    _spriteBatch.Draw(bild, bildRektangel, Color.White);
    _spriteBatch.End();

    base.Draw(gameTime);
}
```

Figur 3

4. Bilden ritas ut, men är större.



Figur 4

Ett annat sätt att ändra en bilds storlek, men också ändra dess rotation och andra egenskaper är genom att använda fler argument till Draw()-metoden än de tre som tidigare använts.

5. Förberedelse, deklarera nya variabler. En Vector2 som namnges **origin**, en float som namnges **rotation** och en float som namnges **scale**.

*Gör även en insats för att organisera variablerna efter vilken typ de har.*

```
2 references
public class Game1 : Game
{
    private GraphicsDeviceManager _graphics;
    private SpriteBatch _spriteBatch;

    Texture2D bild;

    Vector2 position;
    Vector2 origin;

    Rectangle bildRektangel;

    float rotation;
    float scale;
}
```

Figur 5

6. Tilldela värden till variablerna.

**Rotation** sätts till 90°. Detta sker med hjälp av **ToRadians()** som omvandlar grader till radianer.

Skalan (**scale**) sätts till 1,5, motsvarande 150 %.

```
0 references
protected override void LoadContent()
{
    _spriteBatch = new SpriteBatch(GraphicsDevice);

    bild = Content.Load<Texture2D>("happy200");

    position = new Vector2(100, 100);
    origin = new Vector2(0, 0);

    bildRektangel = new Rectangle(100, 100, 450, 450);

    rotation = MathHelper.ToRadians(90);
    scale = 1.5f;
}
```

Figur 6

7. Avslutningsvis används de nya variablerna tillsammans med lite annan information i Draw()-metoden.

*Notera att detta kan skrivas antingen som en lång rad, eller så kan man göra ny rad för varje argument inne i Visual Studio. Se figur 9 respektive figur 8.*

Variablerna **rotation** och **scale** kan sedan ändras för att rita ut bilden i annan vinkel och storlek.

```
0 references
protected override void Draw(GameTime gameTime)
{
    GraphicsDevice.Clear(Color.CornflowerBlue);

    _spriteBatch.Begin();
    _spriteBatch.Draw(
        bild,
        position,
        null,
        Color.White,
        rotation,
        origin,
        scale,
        SpriteEffects.None,
        0);
    _spriteBatch.End();

    base.Draw(gameTime);
}
```

Figur 8

```
0 references
protected override void Draw(GameTime gameTime)
{
    GraphicsDevice.Clear(Color.CornflowerBlue);

    _spriteBatch.Begin();
    _spriteBatch.Draw(bild, position, null, Color.White, rotation, origin, scale, SpriteEffects.None, 0);
    _spriteBatch.End();

    base.Draw(gameTime);
}
```

Figur 7