

MonoGame – Animering

Den här guiden förutsätter att man kan lägga in en bild och få den att röra sig med hjälp av tangentbordet.

Ett sätt att animera utritade objekt i MonoGame är att byta textur baserat på vad som händer med objektet, exempelvis byta håll på bilden om man trycker på en viss tangent.

1. Deklarerar variabler. Det enda nya är att det finns **tre olika variabler** av typen Texture2D.

Planen är att rita ut bilden via **currentTexture**, och sen ändra den texturen baserat på vad som händer i metoden **Update()**.

I planen ingår även att spegelvända bilden beroende på om man trycker på A eller D, detta uppnås via variabeln **happyEffects**.

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

    Texture2D happy;
    Texture2D happyCrouch;
    Texture2D currentTexture;

    Vector2 position;
    Vector2 origin;

    SpriteEffects happyEffects;
    float rotation;
    float scale;
}
```

Figur 1

2. Tilldelar värden till variablerna. Notera att värdet på **currentTexture** sätts till samma som den "vanliga" texturen på bilden, så att bilden från början är som vanligt.

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

    happy = Content.Load<Texture2D>("happy200");
    happyCrouch = Content.Load<Texture2D>("happy200Crouch");
    currentTexture = happy;

    happyEffects = SpriteEffects.None;
    rotation = MathHelper.ToRadians(0);
    origin = new Vector2(0, 0);
    scale = 1.0f;
}
```

Figur 2

3. Ritar ut bilden, använder versionen med många argument, för att kunna använda variabeln happyEffects. Notera att metoden här är skriven på två rader för att göra bilden mindre avlång.

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

    _spriteBatch.Begin();
    _spriteBatch.Draw(currentTexture, position, null, Color.White,
        rotation, origin, scale, happyEffects, 0);
    _spriteBatch.End();
}
```

Figur 3

4. Lägger till rörelse för bilden. Se guide om **rörelse**.

5. För att byta textur vid behov används logik i metoden **Update()**.

Tolkning: När tangenten S är nedtryckt ska bildens position öka i y-led, och så ska bildens textur bytas till den "duckande" texturen. Annars ska texturen bytas tillbaka till den vanliga.

Notera: Variabeln keyboard används för tangentbordets nuvarande state. Se tidigare guider.

```
if (keyboard.IsKeyDown(Keys.S))
{
    position.Y += speed;
    currentTexture = happyCrouch;
}
else
{
    currentTexture = happy;
}
```

Figur 4

6. På motsvarande sätt spegelvänds bilden när man trycker på tangenten D.

```
if (keyboard.IsKeyDown(Keys.D))
{
    position.X += speed;
    happyEffects = SpriteEffects.FlipHorizontally;
}
else
{
    happyEffects = SpriteEffects.None;
}
```

Figur 5