

MonoGame – Rörelse

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

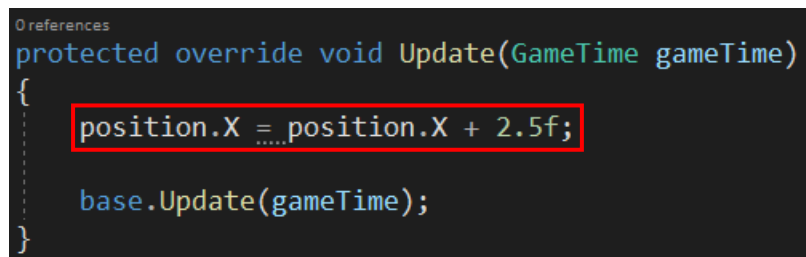
Förberedelser

Ladda in din bild och rita ut dess position med hjälp av en variabel, t.ex. av typen **Vector2** eller **Rectangle**. I det här exemplet är variabeln namngiven till **position**, från tidigare guider.

1. För att få bilden att röra på sig behöver dess position ändras kontinuerligt. Detta sker i metoden **Update()**.

För att få bilden att röra sig åt höger kan man använda följande kod, se figur 1. En tolkning kan vara att

"Bildens nya position ska bli lika med var den var förut plus 2,5".



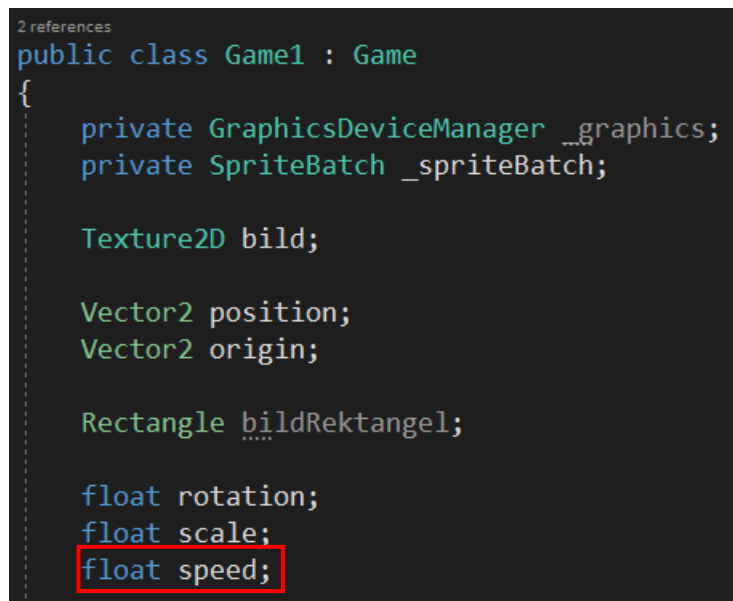
```
0 references
protected override void Update(GameTime gameTime)
{
    position.X = position.X + 2.5f;

    base.Update(gameTime);
}
```

Figur 1

2. Det går också bra att istället för 2,5 använda en variabel.

Deklarerar en variabel av typen float, namnger den **speed**.



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

    Texture2D bild;

    Vector2 position;
    Vector2 origin;

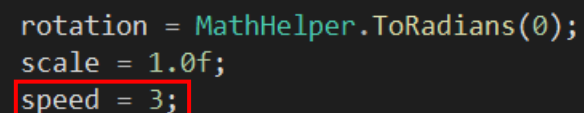
    Rectangle bildRektangel;

    float rotation;
    float scale;
    float speed;
}
```

Figur 2

3. Tilldelar **speed** ett värde, i metoden **LoadContent()**.

Byt sedan ut 2.5f mot speed i metoden **Update()**, se figur 1.



```
rotation = MathHelper.ToRadians(0);
scale = 1.0f;
speed = 3;
```

Figur 3

4. För att påverka hur bilden rör sig används logik inne i metoden **Update()**.

För att få bilden att stanna vid kanten av fönstret kan man skriva en if-sats som kontrollerar om bildens position är vid kanten, och i så fall tilldela speed värdet noll.

```
0 references
protected override void Update(GameTime gameTime)
{
    if (position.X >= (800-bild.Width))
    {
        speed = 0;
    }

    position.X = position.X + speed;

    base.Update(gameTime);
}
```

Figur 4

Tips: Man kan tänka sig

Update()-metoden tillsammans med Draw()-metoden som en stor loop, som går 60 gånger per sekund.

5. Alternativt kan man få bilden att byta håll när den når kanten.

Kan uppnås genom att byta tecken på variabeln **speed**. Alltså från 3 till (-3), detta uppnås genom att multiplicera **speed** med (-1) när bilden når kanten.

```
ences
protected override void Update(GameTime gameTime)
{
    if (position.X >= (800-bild.Width) || position.X <= 0)
    {
        speed = speed * (-1);
    }

    position.X = position.X + speed;

    base.Update(gameTime);
}
```

Figur 5

Tips: Ändrade också något i if-satsen villkor, vad tror du det innebär?

6. För att uppnå rörelse i y-led används samma metod, med enda skillnad att man istället ändrar **position.Y**.