

Muilla kielillä:

- [English](#)
- [Suomi](#)

## Jypelin käyttöohjeet » Miten olion ulkonäköä voi muuttaa?

Luodaan ensin olio, jonka ulkonäköä esimerkeissä muutetaan:

```
PhysicsObject olio = new PhysicsObject(100, 50);
```

Luomisen yhteydessä oliolle sanotaan vain sen leveys ja korkeus. Ryhdytään nyt muokkaamaan olion ominaisuuksia haluamiksemme. Kaikkia olion ominaisuuksia, myös leveyttä ja korkeutta, voi muuttaa olion luomisen jälkeenkin.

### Tekstuuri

Tekstuurikuvat kannattaa tallentaa png-muodossa, jolloin kuvaan voi tallentaa myös läpinäkyviä osia. Kuva täytyy liittää mukaan projektiin, jotta siihen voidaan viitata pelin koodissa. Katso ohjeet [sisällön tuomisesta projektiin](#).

Sen jälkeen kun kuva on lisätty projektiin, ladataan kuva ensin muuttujan luokan alussa seuraavasti.

```
public class Peli : PhysicsGame
{
    Image olionKuva = LoadImage("kuvanNimi");

    public override void Begin()
    {
        //...
    }
}
```

Huomaa, että png-tunnistetta *ei* tarvitse laittaa kuvan nimen perään.

- Kohta kuvanNimi on Content-projektiin siirretyn kuvan nimi esimerkiksi, jos kuva on kissa.png, niin kuvan nimi on silloin pelkkä kissa.
- Miten kuvaan tehdään läpinäkyviä osia? [Lue ohje tästä](#).

Tämän jälkeen kuvan voi asettaa oliolle seuraavalla tavalla:

```
olio.Image = olionKuva;
```

Samaa kuvaa voi nyt käyttää helposti halutessaan monelle eri oliolle.

### Muoto

Joskus olion muodon voi antaa jo oliota luotaessa. Muotoa voi kuitenkin myös jälkikäteen muuttaa.

Esimerkiksi:

```
olio.Shape = Shape.Circle;
```

Tämä tekee oliostamme ympyrän muotoisen. Muita mahdollisia muotoja on esimerkiksi nelikulmio, `Shape.Rectangle`.

## Väri

Värin voi vaihtaa seuraavalla tavalla:

```
olio.Color = Color.Gray;
```

Esimerkissä siis oliosta tehtiin harmaa. Värejä on valmiina paljon ja niistä voi valita haluamansa. [Lista Jypelin sisäänrakennetuista väreistä](#).

Omia värejä voi myös tehdä seuraavasti:

```
olio.Color = new Color( 0, 0, 0 );
```

Oliosta tuli musta. Ensimmäinen arvo kertoo punaisen värin määrän, toinen arvo vihreän värin määrän ja kolmas sinisen värin määrän. "Värimaailman" lyhenne RGB (Red Green Blue) tulee tästä. Lyhenteestä on helppo muistaa missä järjestyksessä värit tulevat. Määrät ovat välillä 0-255.

Värille voi antaa myös läpinäkyvyyden (ns. alfa-arvo) neljäntenä parametrina:

```
olio.Color = new Color( 255, 0, 0, 100 );
```

Ylläoleva esimerkki tekee oliosta punaisen ja puoliksi läpinäkyvän. Nolla on täysin läpinäkyvä ja 255 täysin näkyvä. Vastaavan voi tehdä myös

```
olio.Color = new Color( Color.Red, 100 );
```

Muitakin tapoja värien asettamiseen on olemassa, mutta näillä pärjää jo hyvin.

## Koko

Kokoa voi vaihtaa seuraavasti:

```
olio.Width = leveys;  
olio.Height = korkeus;
```

Samantapainen asia voi tehdä myös vektorin avulla yhdellä rivillä:

```
olio.Size = new Vector( leveys, korkeus );
```

# Kulma

Oliota voi pyörittää asettamalla olion `Angle` ominaisuuteen jokin kulma:

```
olio.Angle = Angle.FromDegrees(kulma);
```

missä `kulma` tilalla on muuttuja tai luku, joka kuvaa montako astetta oliota käännetään. Oletuksena olion kulma on nolla astetta.

Kulman kasvaessa olio kääntyy vastapäivään päin:

Kulman voi antaa myös radiaaneina, esimerkiksi jos oliota halutaan pyörittää 90 astetta, eli  $0.5 * \text{PI}$  radiaania, niin se onnistuu näin:

```
olio.Angle = Angle.FromRadians(0.5 * Math.PI);
```

## Loppuhuomautus

Kaikissa esimerkeissä on käytetty olion nimeä `olio`, esim. `olio.Color`. Huomaan, että olion tilalla voi olla mikä vaan muu nimi. Jos on luotu `PhysicsObject` `kissa`, niin tällöin kutsutaankin esim. `kissa.Color`.