Brick Breaker -peli Pythonilla

Tämän ohjeen avulla rakennat Brick Breaker -pelin Pythonilla käyttäen turtle-kirjastoa. Peli tehdään kolmessa vaiheessa, ja jokaisen vaiheen askeleissa peli rakentuu pala palalta. Etene askel kerrallaan ja kokeile pelin toimivuutta aina askeleen lopussa ennen kuin jatkat seuraavaan.

Käytetään tällä kertaa Thonny-ympäristöä. Tee pelille oma tiedosto (esim. breakout.py). Tallenna tiedosto OneDriveen.

Vinkki: *Tallenna ohjelma usein Ctrl+S-näppäinyhdistelmällä. OneDrive synkronoi tiedoston pilveen, jolloin työ ei jää vain koulun tietokoneelle.*

Vaihe 1: Peli-ikkuna ja pelaajan maila

Aloitetaan pelin rungosta: kasvava ikkuna, pelaajan maila ja pallo. Tässä vaiheessa pallo ei vielä liiku, mutta liike tulee mukaan vaiheessa 2.

Askel 1: Peli-ikkuna

Tuo turtle-kirjasto ja luo peli-ikkuna. Muuttujat leveys ja korkeus määrittävät ikkunan koon. Ikkunan kokoa kasvatetaan seuraavassa askeleessa.

```
import turtle

leveys = 10
korkeus = 10

ikkuna = turtle.Screen()
ikkuna.title("Brick Breaker")
ikkuna.bgcolor("black")
ikkuna.setup(leveys, korkeus)
```

Askel 2: Kasvava ikkuna

Lisätään pelin pääsilmukka, joka laittaa pelin näkymään. Silmukka kasvattaa ikkunaa 5 pikseliä kerrallaan, kunnes se saavuttaa 600x600 koon. Tämän jälkeen kasvu pysähtyy.

```
dx = 5
dy = 5

while True:
    leveys += dx
    korkeus += dy
    if leveys > 600 and korkeus > 600:
        dx = 0
        dy = 0

    ikkuna.setup(leveys, korkeus)
    ikkuna.update()
```

Kokeile: Aja koodi jo tässä vaiheessa, ja myöhempienkin askelien jälkeen. Mustan ikkunan pitäisi kasvaa vähitellen 600x600 kokoiseksi. Kaikki seuraavat lisäykset kirjoitetaan tämän silmukan yläpuolelle, jotta silmukka pysyy koodin viimeisenä osana.

Askel 3: Pelaajan maila

Lisää seuraavien askeleiden koodit ennen while-silmukkaa. Komento penup() nostaa kynän ylös, jotta olio ei piirrä viivaa liikkuessaan, ja shapesize muokkaa mailan suorakulmion muotoiseksi.

```
px = 0
py = -250

maila = turtle.Turtle()
maila.penup()
maila.speed(0)
maila.color("blue")
maila.shape("square")
maila.shapesize(0.5, 5, 5)
maila.setpos(px, py)
```

Askel 4: Pallo (ei vielä liiku)

Lisää pallo samalla tavalla, edelleen ennen while-silmukkaa.

```
x = 0
y = -200

pallo = turtle.Turtle()
pallo.penup()
pallo.speed(0)
pallo.color("yellow")
pallo.shape("circle")
pallo.setpos(x, y)
```

Askel 5: Mailan liikutusfunktiot

Luodaan funktiot mailan liikuttamista varten sivuttain. Komento xcor() palauttaa olion x-koordinaatin, ja setpos() asettaa uuden sijainnin.

```
def oikealle():
    px = maila.xcor()
    px += 50
    maila.setpos(px, py)

def vasemmalle():
    px = maila.xcor()
    px -= 50
    maila.setpos(px, py)
```

Askel 6: Näppäinohjaus

Komento ikkuna.listen() käskee ikkunaa kuuntelemaan näppäinpainalluksia. Komento onkeypress(funktio, näppäin) yhdistää näppäimen ja funktion.

```
ikkuna.listen()
ikkuna.onkeypress(vasemmalle, "Left")
ikkuna.onkeypress(oikealle, "Right")
```

Kokeile: Aja koodi. Ikkunan pitäisi kasvaa mustana, ja näkyviin tulee sininen maila alareunassa ja keltainen pallo sen yläpuolella. Nuolinäppäimet ←/→ liikuttavat mailaa sivuttain.

Kustomoi oma pelisi: Voit muokata peliä omannäköiseksesi:

- Taustaväri: ikkuna.bgcolor("darkblue") - kokeile eri värien nimiä tai hex-koodia (esim. "#1a1a2e")
- Mailan väri: maila.color("red") - kokeile myös pallon väriä pallo.color(...)
- Pallon muoto: pallo.shape("triangle") tai pallo.shape("square") toimivat "circle"-muodon sijaan
- Koko: shapesize()-arvoja suurentamalla tai pienentämällä, esim. pallo.shapesize(2) saa isomman pallon