Brick Breaker -peli Pythonilla

Jatketaan siitä mihin vaiheessa 2 jäätiin: pallo liikkuu ja kimpoaa, ja tiilet ovat paikallaan. Tässä vaiheessa tiilet tuhoutuvat palloon osuessaan, peliin tulevat pisteet ja elämät, ja peli päättyy kun elämät loppuvat.

Kaikki uusi koodi lisätään ennen while-silmukkaa, paitsi putoamisen ja osumien tarkistukset, jotka lisätään silmukan sisälle. Muista tallentaa usein ja kokeile peliä aina askeleen jälkeen.

Vaihe 3: Pisteet, elämät ja tiilten tuhoutuminen

Askel 1: Pisteet ja elämät

Pisteet ja elämät näytetään ruudun yläreunassa tekstinä. Tekstin kirjoittamiseen tehdään kaksi turtle-oliota, pisteliitu ja elamaliitu. Koska tekstiä päivitetään monta kertaa, kirjoittaminen tehdään omassa funktiossa: clear() pyyhkii vanhan tekstin ja write() kirjoittaa uuden.

Lisää nämä pallon luonnin jälkeen, ennen oikealle()-funktiota:

```
def kirjoita(liitu, teksti):
    liitu.clear()
    liitu.write(teksti, align="center", font=("Courier", 16))

pisteet = 0

pisteliitu = turtle.Turtle()
pisteliitu.penup()
pisteliitu.speed(0)
pisteliitu.color("white")
pisteliitu.hideturtle()
pisteliitu.setpos(-200, 260)
kirjoita(pisteliitu, f"PISTEET: {pisteet}")

elamat = 5

elamaliitu = turtle.Turtle()
elamaliitu.penup()
elamaliitu.speed(0)
elamaliitu.color("white")
elamaliitu.hideturtle()
elamaliitu.setpos(200, 260)
kirjoita(elamaliitu, f"ELÄMÄT: {elamat}")
```

hideturtle() piilottaa kynän, jolloin ruudulle jää vain teksti. Merkkijonon edessä oleva f tekee siitä f-merkkijonon: aaltosulkujen {} sisällä oleva muuttuja korvataan sen arvolla.

Kokeile: Aja koodi. Ruudun yläreunaan pitäisi ilmestyä teksti PISTEET: 0 vasemmalle ja ELÄMÄT: 5 oikealle.

Askel 2: Mailan liikerajat

Nyt maila voi vielä livahtaa ruudun ulkopuolelle. Muokkaa oikealle()- ja vasemmalle()-funktioita niin, että mailan x-koordinaatti ei voi ylittää rajoja. Ikkuna on 600 pikseliä leveä ja maila 100 pikseliä leveä, joten mailan keskipiste saa olla enintään 250 pikseliä keskeltä kumpaankin suuntaan.

```
def oikealle():
    px = maila.xcor()
    px += 50
    if px > 250:
        px = 250
    maila.setpos(px, py)

def vasemmalle():
    px = maila.xcor()
    px -= 50
    if px < -250:
        px = -250
    maila.setpos(px, py)
```

Kokeile: Aja koodi ja yritä viedä maila ruudun reunan yli. Sen pitäisi pysähtyä reunaan.

Askel 3: Pallo putoaa ja elämä menee

Tähän asti pallo on kimmonnut myös alareunasta. Nyt alareuna on vaarallinen: kun pallo putoaa ruudun alareunan ohi, pelaaja menettää yhden elämän ja pallo palaa keskelle uudella satunnaisella nopeudella.

Etsi pääsilmukasta y-suunnan reunatarkistus (if pallo.ycor() > 280 or pallo.ycor() < -280:) ja muuta se niin, että vain yläreuna kimmottaa palloa:

```
    if pallo.ycor() > 280:
        vy = -vy
```

Lisää sen perään uusi tarkistus, joka huolehtii putoamisesta:

```
    if pallo.ycor() < -280:
        x = 0
        y = 0
        vx = random.randint(3, 8)
        vy = random.randint(5, 8)
        pallo.setpos(x, y)
        elamat -= 1
        kirjoita(elamaliitu, f"ELÄMÄT: {elamat}")
        if elamat <= 0:
            break
```

Kun elämät loppuvat, break lopettaa silmukan ja peli päättyy.

Kokeile: Anna pallon pudota. ELÄMÄT-luvun pitäisi pienentyä yhdellä jokaisesta putoamisesta, ja pallon pitäisi palata keskelle. Kun elämiä ei ole enää jäljellä, ohjelma päättyy.

Askel 4: Tiilet tuhoutuvat

Viimeiseksi tiilet. Pallon osuminen tiileen tarkistetaan omassa funktiossa, jota kutsutaan jokaiselle tiilelle. Funktio laskee, kuinka kaukana pallo on tiilestä x- ja y-suunnassa. abs() eli itseisarvo muuttaa luvun positiiviseksi, joten etäisyys on sama riippumatta siitä, onko pallo tiilen vasemmalla vai oikealla puolella. Jos pallo on alle 20 pikselin päässä molemmissa suunnissa, se osuu tiileen.

Lisää funktio vasemmalle()-funktion jälkeen, ennen näppäinohjausta:

```
def osuu(pallo, tiili):
    ero_x = abs(pallo.xcor() - tiili.xcor())
    ero_y = abs(pallo.ycor() - tiili.ycor())
    return ero_x < 20 and ero_y < 20
```

Lisää sitten pääsilmukkaan mailan osumatarkistuksen jälkeen, ennen riviä x = pallo.xcor():

```
    for tiilet in tiilikasa:
        for tiili in tiilet:
            if tiili.isvisible() and osuu(pallo, tiili):
                tiili.hideturtle()
                pisteet += 1
                kirjoita(pisteliitu, f"PISTEET: {pisteet}")
                vy = -random.randint(5, 8)
```

Sisäkkäiset silmukat käyvät läpi jokaisen tiilen. isvisible() kertoo, onko tiili vielä näkyvissä, jotta jo tuhottuun tiileen ei osuta uudestaan. Osuman jälkeen tiili piilotetaan hideturtle()-komennolla, pisteet kasvavat yhdellä ja pallo lähtee alaspäin uudella satunnaisella nopeudella.

Kokeile: Pallon osuessa tiileen tiilen pitäisi hävitä ja PISTEET-luvun kasvaa yhdellä. Peli on nyt valmis!

Kustomoi oma pelisi: Kun peli toimii, voit kokeilla esimerkiksi:

- Elämien määrä: muuta rivi elamat = 5 toiseksi luvuksi
- Pelin loppu: ohjelma sulkeutuu heti elämien loppuessa. Lisää while-silmukan jälkeen (ilman sisennystä) viimeiseksi rivi ikkuna.mainloop(), niin ikkuna jää auki ja pisteet jäävät näkyviin
- Voitto: tiiliä on 69. Voit lopettaa pelin, kun pisteet ovat 69