

Peliohjelmoinnin alkeet Unitylla

3. Taso

Sisällysluettelo

- Konsoliin tulostaminen
- Pelilooppi
- Unityn pelilooppi
- Start ja Awake
- Update ja FixedUpdate
- Rigidbody
- GetComponent
- Vector3
- Quaternion
- Raycast
- Trigger
- InputManager

Konsoliin tulostaminen

- Unityn konsoliin voi tulostaa omia viestejä komennoilla:

```
Debug.Log("Viesti");  
  
print("Viesti");
```

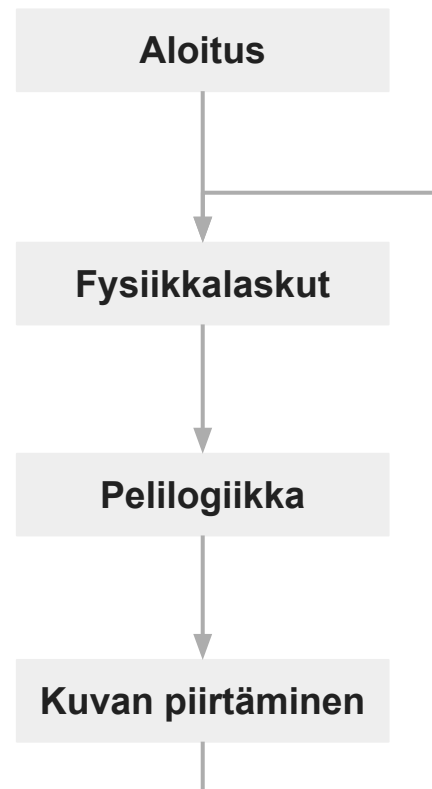
- Kappaleen sijainnin saa tulostettua esimerkiksi näin:

```
Debug.Log("Sijainti: " + transform.position);
```

Pelilooppi yleisesti

- Peliloopilla tarkoitetaan sitä, mitä asioita pelimoottori suorittaa yhden kierroksen aikana
 - Yhden kierroksen aikana pelimoottori piirtää ruudulle yhden kuvan
- Yksinkertainen pelimoottori voi sisältää esimerkiksi viereisen peliloopin

Aloituksessa alustetaan asioita ennen looppia



Unityn pelilooppi

- Unityssa pelilogiikka suoritetaan aina ennen kuin pelimoottori piirtää kuvan
- Sivussa on yksinkertaistettu malli Unityn pelimoottorin loopista
- Tarkka kuvaus Unityn peliloopista löytyy sivulta:

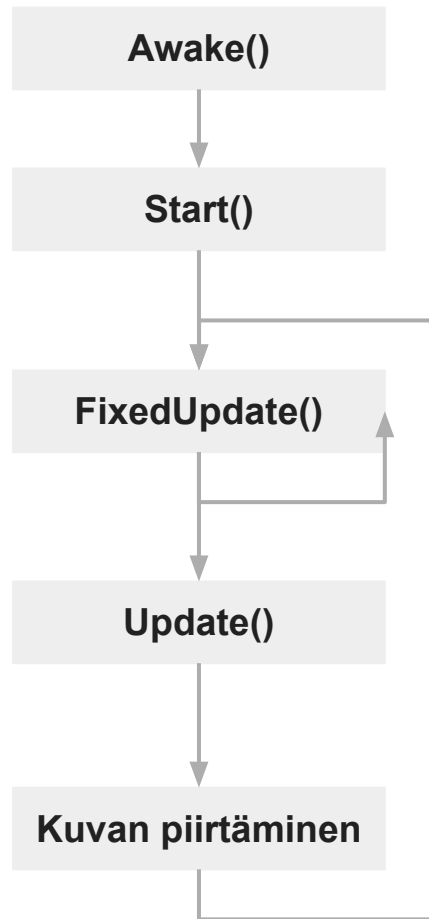
<https://docs.unity3d.com/Manual/ExecutionOrder.html>

Käyttäjän skriptien alustukset tapahtuvat ennen loopin alkua

Täällä lasketaan fysiikkalaskut. Niitä ajetaan kierroksen aikana vaihteleva määrä, välillä useampi, välillä ei kertaakaan

Käyttäjän tekemät skriptit ajetaan täällä, eli ns. pelilogiikka

Näytönohjain piirtää kuvan ja näyttää sen näytöllä



Start() ja Awake()

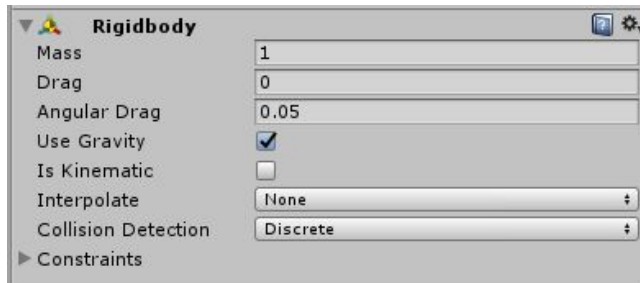
- Peliloopin alussa suoritettavia funktioita
- Awake() kutsutaan ennen Start()-funktiota ja sitä kutsutaan huolimatta siitä onko peliobjekti päällä (enabled) tai pois päältä (disabled)
- Start() kutsutaan Awake() jälkeen ja kutsutaan vain kun peliobjekti on päällä
 - Eli jos peliobjekti on pois päältä ja se laitetaan päälle, niin Start()-funktio suoritetaan
 - Jos peliobjekti on päällä, niin Start() suoritetaan kun kaikki Awake()-funktiot on suoritettu
- Kummatkin suoritetaan ennen Update()-funktiota

Update() ja FixedUpdate()

- Update() sisältää koodia, joka suoritetaan jokaisella kierroksella ennen kuvan piirtämistä ja näyttämistä pelaajalle
 - Sisältää käytännössä pelilogiikkaa
- FixedUpdate() suoritetaan ennen fysiikkalaskuja
 - Fysiikkalaskut eivät ole riippuvaisia siitä kuinka usein Update() suoritetaan, vaan ne lasketaan oletuksena aina 50 kertaa sekunnissa
 - Tätä lukua voi muuttaa projektin asetuksista

Rigidbody

- Rigidbody-komponentti lisää kappaleeseen fysiikat
 - Toisin sanottuna Unityn pelimoottori ottaa kappaleen huomioon fysiikkalaskuissa
 - Eli painovoima alkaa vaikuttamaan kappaleeseen, ja jos kappaleessa on Collider, niin se pystyy törmäämään toisten kappaleiden kanssa, eli muut kappaleet voivat vaikuttaa siihen voimilla
- “Is Kinematic” tekee kappaleesta kinemaattisen, jolloin voimat eivät enää vaikuta siihen, mutta se pystyy vaikuttamaan muihin kappaleisiin voimilla



GetComponent<>()

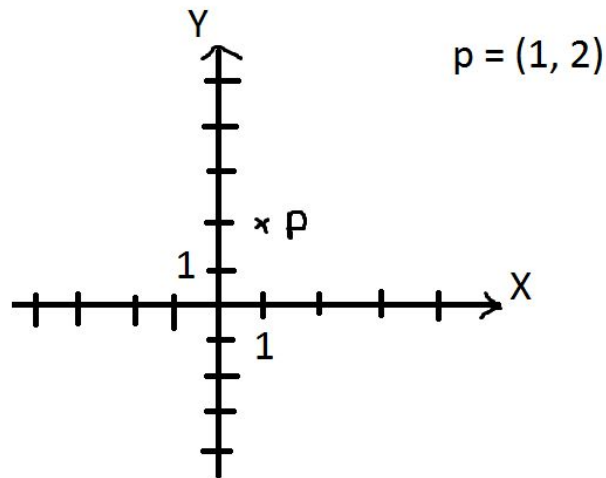
- Komento joka hakee peliobjektista halutun komponentin
- Palauttaa null jos haluttua komponenttia ei löydy
- <> merkkien väliin tulee halutun komponentin tyyppi

```
tyyppi muuttuja = GetComponent<tyyppi>();
```

```
Rigidbody rb = GetComponent<Rigidbody>();
```

Vector3

- Vector3 on tapa esittää (esimerkiksi) sijaintia kolmiulotteisessa koordinaatistossa
 - Sisältää kolme liukulukua (desimaalilukua) (x, y, z)
- Vertaa koulusta tuttuun kaksiulotteiseen koordinaatistoon (kuva), mutta vain kolmiulotteisena
- Käytetään myös asennon ja koon ilmaisemiseen
 - Kulma jokaisen akselin ympäri
 - Koko jokaisen akselin suuntaan
- Voidaan käyttää myös suunnan ilmaisemiseen
 - Tai minkä tahansa kolmiulotteisen vektorin ilmaisemiseen



Vector3 matematiikka

- Yhteenlasku

- Kaksi Vector3:a voidaan laskea yhteen
- Tällöin kummankin vektorin jokainen komponentti lasketaan erikseen yhteen
- Esim. $v1(2, 4, 1) + v2(0, 2, 2) = v3(2, 6, 3)$

- Vähennyslasku

- Sama kuin yhteenlasku, mutta summauksen sijaan komponentit vähennetään toisistaan
- Esim. $v1(2, 4, 1) - v2(0, 2, 2) = v3(2, 2, -1)$

- Kertolasku

- Vector3 voidaan kertoa yksittäisellä luvulla, mutta ei toisella Vector3:lla
- Tällöin jokainen komponentti kerrotaan annetulla luvulla
- Esim. $v1(2, 4, 1) * 2 = v2(4, 8, 2)$

- Jakolasku

- Sama kuin kertolasku, mutta kertomisen sijaan komponenttien arvot jaetaan
- Esim. $v1(2, 4, 1) / 2 = v2(1, 2, 0.5)$

Quaternion (Kvaternio)

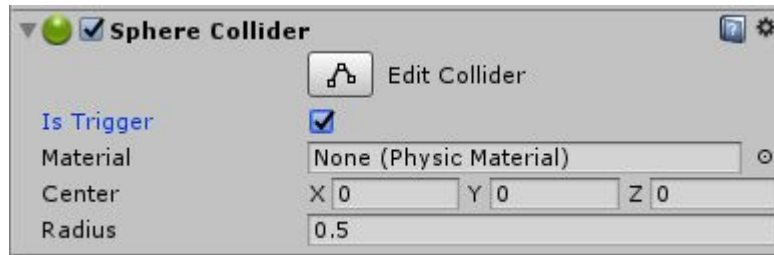
- Käytetään asentojen ilmaisemiseen kolmiulotteisessa tilassa
- Sisältää neljä liukulukua (x, y, z, w)
- Monimutkaista matriisilaskentaa, ei ymmärrettävää tavallisille ihmisille (minut mukaan lukien)
- Ei toimi samalla tavalla kuin Euler-kulmat, vaan jonkin asteen taikuudella
- Oleellista näistä tietää on se, että Unity laskee asentoja sisäisesti kvaternioilla
 - Esimerkiksi skriptissä `transform.rotation` on kvaternio
- Kvaternioille löytyy funktioita, jotka muuttavat ne ymmärrettäviksi Euler-kulmiksi
- Esimerkiksi Wikipediasta ja Unityn dokumentaatiosta löytyy lisää tietoa kiinnostuneille

Physics.Raycast

- Raycast on pelimoottoreissa yleinen tapa tarkistaa asioiden sijaintia tai niiden näkyvyyttä
- Raycast voidaan kuvitella siten, että ammutaan lasersäde tietyistä pisteistä tiettyyn suuntaan, ja se palauttaa tiedon siitä mihin se osui
 - Eli se on vain täysin suora säde joka lentää annettuun suuntaan
- Tankkipelin tapauksessa sillä katsotaan mihin pisteeseen näytöllä oleva kursori osuu pelimaailmassa
- Voidaan käyttää myös esimerkiksi sen tarkistamiseen, että näkeekö vihollinen pelaajan, vai onko sen edessä jokin este

Collider ja Trigger

- Collider on kappaleen fyysinen muoto, jota käytetään törmäysten laskemiseen
- Collider voidaan muuttaa Triggeriksi, jolloin se ei enää törmää muihin kappaleisiin, vaan ne voivat kulkea sen läpi ilman mitään estettä
- Trigger kuitenkin tunnistaa, jos se sisällä on kappale
 - Tätä voidaan käyttää alueena, joka tunnistaa jos pelaaja menee sen sisään
 - Esimerkiksi pelaaja kulkee oven läpi



InputManager

- Unityn sisäinen järjestelmä, joka vastaa käyttäjän syötteestä
- Käytännössä mahdollistaa nappuloiden, ohjainten tai muun käyttäjän tekemän toiminnon käyttämisen skripteissä
- Nappuloita voi muokata, lisätä ja poistaa Unityn sisäisen ikkunan kautta
 - Edit -> Project Settings -> Input

