

# Ohjelmointikurssi – Jakso 1: Godotin perusteet ja ohjelmoinnin alkeet

## 1. Jakson tiedot

### Tavoitteet:

- Ymmärtää, mitä ohjelmointi ja pelinkehitys tarkoittavat.
  - Tutustua Godotin käyttöliittymään ja peruskäsitteisiin.
  - Oppia ohjelmoinnin perustanastoa ja koodaamisen alkeet.
  - Tehdä ensimmäinen oma yksinkertainen peli tai sovellus.
- 

## 2. Ohjelmointikurssi – Syventyminen aiheeseen

### Miksi oppia ohjelmointia?

Ohjelmointi on luova tapa ratkaista ongelmia ja rakentaa asioita, kuten pelejä, sovelluksia ja verkkosivustoja. Pelinkehityksessä yhdistyvät koodaus, suunnittelu ja tarinankerronta.

### Godotin vahvuudet:

- Helppo oppia aloittelijalle.
- Ilmainen ja avoimen lähdekoodin.
- Tukee 2D- ja 3D-pelejä.
- Nopea koodata **GDScriptin** avulla.

### Pelinkehityksen perusvaiheet:

1. **Suunnittelu:** Määrittele, millainen peli on ja miten sitä pelataan.
  2. **Rakentaminen:** Luo hahmot, kohtaukset ja logiikka.
  3. **Testaus:** Pelaa ja korjaa mahdolliset virheet.
- 

## 3. Tutustuminen Godotiin

### Godotin osat:

- **Scene (Kohtaus):** Pelin perusyksikkö, kuten huone, esine tai hahmo.
- **Node (Solmu):** Rakennuspalikka, jolla luodaan kohtaus (esim. Sprite, Kamera).
- **Script (Skripti):** Koodi, joka ohjaa solmujen toimintaa.

#### Godotin käyttöliittymän osat:

- **Scene-välilehti:** Näyttää pelin rakenteen.
- **Inspector:** Muokkaa valitun solmun ominaisuuksia.
- **FileSystem:** Kaikki projektin tiedostot.
- **Play-painike:** Testaa pelisi.

#### Harjoitus: Godotin käyttöliittymän tutkiminen

1. Avaa uusi projekti.
2. Tunnista Godotin eri osat (Scene, Inspector, FileSystem).
3. Lisää uusi kohtaus ja tutki, miten solmuja lisätään (esim. Sprite).

## 4. Sanasto ja harjoitukset

### Sanasto (laajennettu):

Sana	Selitys
<b>Node</b>	Solmu, joka edustaa yksittäistä elementtiä kohtauksessa.
<b>Scene</b>	Kohtaus, joka koostuu yhdestä tai useammasta solmusta.
<b>Script</b>	Kooditiedosto, joka määrittää solmujen toiminnan.
<b>Variable (Var)</b>	Muuttuja, joka tallentaa tietoa.
<b>Function (Func)</b>	Koodipätkä, joka suorittaa tietyn tehtävän.

<b>If-else</b>	Ehtolause: tarkistaa, mikä toimintatapa valitaan.
<b>Array</b>	Lista, joka sisältää useita arvoja.
<b>Position</b>	Sijainti kohtauksessa (x- ja y-koordinaatit).
<b>Sprite</b>	2D-kuva tai hahmo.
<b>CollisionShape</b>	Määrittelee alueen, jossa törmäys tunnustetaan.
<b>Input</b>	Käyttäjän syöte, kuten näppäimistön painallus.
<b>Process</b>	Godotin funktio, joka päivittää joka ruudun (frame).

---

## Sanaston tehtävät

### 1. Täydennä lauseet:

- Solmu on pelin \_\_\_\_\_.
- Sijainti määritetään muuttujalla \_\_\_\_\_.
- Koodi, joka tekee tarkistuksen, on nimeltään \_\_\_\_\_.

### 2. Paritehtävä:

Yhdistä oikeat parit numeroimalla:

Termi	Selitys
1 Node	___Lista useista arvoista.
2 Variable	___Kooditiedosto, joka ohjaa toimintaa.
-	
3 Array	___Pelin rakennuspalikka.
4 Script	___Tallentaa tietoa.

### 3. Kirjoita koodi:

Kirjoita koodi, joka:

- Luo muuttujan nimeltä "pisteet".
  - Tulostaa konsoliin tekstin: *"Pelaajan pisteet: 10"*.
- 

## 5. Harjoituksia Godotissa

### Harjoitus 1: Kuution luominen (3D)

1. Avaa uusi projekti ja valitse **3D Scene**.
2. Lisää kohtaukseen **MeshInstance** ja valitse Mesh-kohdasta **CubeMesh**.
3. Lisää kuutioon väri:
  - **Material** → **New Spatial Material** → **Albedo** ja valitse väri.

### Harjoitus 2: 2D-hahmon luominen ja liikuttaminen

1. Avaa uusi **2D Scene**.
2. Lisää **KinematicBody2D** ja sen alle **Sprite** sekä **CollisionShape2D**.
3. Lisää hahmolle koodi:

gdscript

Copy code

```
extends KinematicBody2D
```

```
var speed = 200
```

```
func _process(delta):
```

```
    var direction = Vector2()
```

```
    if Input.is_action_pressed("ui_right"):
```

```
        direction.x += 1
```

```
    if Input.is_action_pressed("ui_left"):
```

```
        direction.x -= 1
```

```
    move_and_slide(direction * speed)
```

4. Lisää Input Mapissa näppäimet "ui\_right" ja "ui\_left".

### **Harjoitus 3: Hahmon törmäys esineen kanssa**

1. Lisää kohtaukseen toinen **StaticBody2D** ja törmäysmuoto.
  2. Koodaa hahmo niin, että se pysähtyy törmätessään esineeseen.
-