Mediacollege Amsterdam

Game Development Verdieping

Les 1 : Introductie Physics Project – Wat gaan we doen?

SD1 – Periode 2

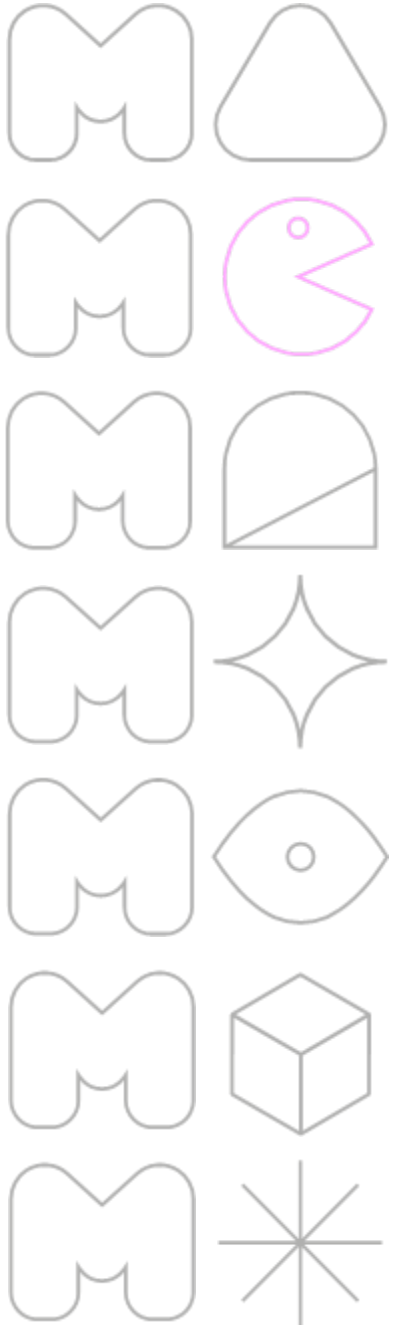
Datum:

Project:

Inhoud

1. Terugblik vorige periode & toets
2. Deze periode
3. Planning
4. Uitleg & theorie
5. Zelfstandig werken
6. Presenteren werk
7. Afsluiten

Mediacollege
Amsterdam



Vorige periode

Veel gemaakte fouten in toets van vorige periode.

- Time.deltaTime
- Update() methode
- Variabelen
- Verschil tussen if / switch

Mediacollege
Amsterdam

Toets – Time.deltaTime

Waarom gebruiken we Time.deltaTime?

Zonder Time.deltaTime:

- Je code draait één keer per frame.
- Snelle pc → meer frames → snellere beweging
- Langzame pc → minder frames → langzamere beweging
  De snelheid hangt af van de computer!

```
transform.Translate(Vector3.forward * speed);
```

Mediacollege
Amsterdam

Toets – Time.deltaTime

Waarom gebruiken we Time.deltaTime?

Met Time.deltaTime:

- Houdt rekening met de tijd tussen frames.
- Beweging wordt omgerekend naar seconden, niet naar frames.
- De beweging is even snel op elke computer.

```
transform.Translate(Vector3.forward * speed * Time.deltaTime);
```

Mediacollege
Amsterdam

Toets – Time.deltaTime

Waarom gebruiken we Time.deltaTime?

- Stel je twee auto’s voor die 1 seconde rijden:
  - Auto A heeft 60 frames
  - Auto B heeft 300 frames

Zonder deltaTime: Auto B rijdt 5x verder

Met deltaTime: beide rijden even ver

Mediacollege Amsterdam

Toets – Update();

Denk aan een film: elk frame is een nieuw plaatje. Update() voert jouw code uit bij elk plaatje.

- Update() draait elke frame van de game
- 60 FPS = 60 keer per seconde
- Gebruik voor: beweging, input, camera volgen
- Perfect voor alles wat constant moet gebeuren

```
// Update is called once per frame
Unity Message | 0 references
void Update()
{
    CameraPosition.position = transform.position + offSetCamera;
}
```

Mediacollege Amsterdam

Toets – Variabelen();

Een variabele is een doosje met een label waar je iets in bewaart.
Je kiest wat erin zit (getal, tekst, positie) en hoe het heet.

Je gebruikt variabelen om:

- Gegevens op te slaan
- Ze later te veranderen
- Ze te gebruiken in je code

```
int score = 0;          // getal
string naam = "Player"; // tekst
bool isAlive = true;    // ja/nee
```

Mediacollege
Amsterdam

Toets – if & Switch

- **If**

Gebruik je als er één of enkele voorwaarden zijn.

```
if (score > 10) {
    Debug.Log("Je wint!");
}
```

Controleert iets en voert alleen code uit als dat waar is.

- **Switch**

Gebruik je als er meerdere vaste opties zijn.

```
switch (color) {
    case "red":
        Debug.Log("Stop");
        break;
    case "green":
        Debug.Log("Go");
        break;
```

Handig bij meerdere keuzes in plaats van veel if-regels.

Kort gezegd:

- **If** = “Als dit, dan dat”
- **Switch** = “Kies wat past uit meerdere mogelijkheden”

Mediacollege Amsterdam

Thema van deze periode

Thema: Physics & Mechanics in Unity

Einddoel: Bouw je eigen *Peggle/pachinko* -spel!

- Je leert werken met Unity's physics-systeem
- Je voegt elke week nieuwe mechanics toe
- Je werkt aan één doorlopend Unity-project
- In week 7 lever je je spel in
- In week 8 maak je de theorietoets

Mediacollege
Amsterdam

Wat is een peggle/pachinko spel?

Een Peggle- of Pachinko-spel is een mix van skill en geluk, waarin je een bal afvuurt, physics zijn werk laat doen en probeert zoveel mogelijk punten te scoren door slimme schoten af te vuren.

Kenmerken:

- Je mikt en vuurt een bal af in een hoek.
- De bal stuitert via physics tegen pinnen of objecten.
- Je scoort punten of activeert effecten bij elke aanraking.
- Het doel is om bepaalde targets te raken (vrije interpetatie)
- Elk schot verloopt anders (skill én geluk spelen mee)

Mediacollege Amsterdam

Planning – overzicht periode

- Week 1 : Introductie & opzetten project
- Week 2 : Forces & Collision
- Week 3 : Triggers & Score
- Week 4 : Level Design & UI
- Week 5 : Power-ups & Gamefeel
- Week 6 : Animatie & freestylen (zelf feature toevoegen)
- Week 7 : Oefentoets, presenteren & inleveren
- Week 8 : Eindtoets

Mediacollege
Amsterdam

Werkwijze tijdens deze module

We werken aan één doorlopend Unity-project. Je bouwt elke week verder aan hetzelfde spel en scene.

Belangrijk:

- Werk in één scene (geen nieuwe projecten maken).
- Voeg iedere week één nieuwe feature toe.
  - Maak een GIF of korte video van je feature.
  - Schrijf een korte reflectie/toelichting in README.
- Push je werk naar Git als back-up.
- Lever aan het begin van de periode je Git-link in via Simulise.

Mediacollege
Amsterdam

Beoordeling en inlevermomenten

- **Week 7 - Inlevermoment & oefentoets:**
  - Je levert je game in via Simulise (Git-link repository). **(50%)**
  - Je spel bevat alle weekly features .
    - Korte reflectie en gif per feature op readme.
    - Commit per feature
  - Je presenteert je spel in de les.

Week 8 - Toets:

- Schriftelijke theorietoets over Unity Physics. **(50%)**
- Vragen over Rigidbody, Colliders, Forces, Triggers en UI.

Mediacollege
Amsterdam

Opdracht van deze week

Wat ga je vandaag doen

- Nieuw Unity-project opzetten
- Git koppelen en eerste push doen
- Inspiratie zoeken: bekijk een Peggle-achtig spel
- Concept vastleggen in je README
- Levelschets maken (in de README)
- Reflectie schrijven over wat je hebt gedaan

Alle informatie staat op git!

Mediacollege Amsterdam

Afsluiten

- Deel je concept ☺

- Volgende week
  - Opdracht afmaken!
  - Forces & Collisions

Mediacollege
Amsterdam

Bedankt!

Mediacollege
Amsterdam