Project Bounce & Collect

In dit project gaan jullie een **interactief Pygame-spel** maken waarin een **bal stuitert** en **munten verzamelt**. Dit spel combineert beweging, botsingsdetectie, geluiden en een gebruikersinterface.

**Stap 1: Voorbereiding**

* **Installeer Pygame** (indien nog niet geïnstalleerd)
* **Maak een projectmap aan**  
  Maak een map aan genaamd **BounceAndCollect** en zorg dat je de volgende structuur gebruikt:

📂 BounceAndCollect  
   📄 main.py *(hoofdscript, start het spel)*  
   📂 images *(map voor spelafbeeldingen: bal, munt, achtergrond)*  
   📂 sounds *(map voor geluidseffecten en muziek)*

* **Download of maak afbeeldingen en geluiden**  
  Gebruik de bestaande afbeeldingen of maak je eigen ball.png, coin.png en geluiden zoals collect.wav en bounce.wav.

**Stap 2: Code Implementeren**

De volgende functionaliteiten **moeten** aanwezig zijn:

* Een **bal** die door het scherm stuitert en **munten verzamelt**.
* Een **spelgebied** waarin de bal wordt **bestuurd met de pijltjestoetsen**.
* **Muren** waaraan de bal **terugkaatst**.
* **Munten** die op willekeurige plekken verschijnen en verdwijnen bij aanraking.
* Een **scoreteller** die het aantal verzamelde munten bijhoudt.
* **Geluidseffecten** bij het verzamelen van een munt en botsing met een muur.
* **Achtergrondmuziek**.

Hieronder staan elementen die in de uitbreidingsopdracht aanwezig moeten zijn:

* **Muur-obstakels** toevoegen die de bal moeten ontwijken.
* Een **startscherm** en **eindscherm** bevatten.
* **Tijdslimiet** toevoegen (bijvoorbeeld 60 seconden om zoveel mogelijk munten te verzamelen).
* **Toetsenbordbesturing** toevoegen zodat de bal naast stuiteren ook gestuurd kan worden.

**Stap 3: Testen en verbeteren**

1. **Run je script** (python main.py).
2. Controleer of de **bal beweegt en stuitert**.
3. Zorg dat **munten verdwijnen bij aanraking** en de **score stijgt**.
4. Luister of de **geluiden correct werken**.
5. **Verbeter en breid uit** met obstakels, tijdslimiet of extra levels.

**Stap 4: Inlevering**

**Wat lever je in?**

* Je **volledige projectmap** inclusief main.py, afbeeldingen en geluiden.
* Een **korte documentatie** waarin je uitlegt hoe je code werkt.

**Deadline:** zie Planner  
**Inleverwijze:** Uploaden in de uploadzone op Smartschool.

# Evaluatiecriteria

**1. Werking van het spel**

| **Score** | **Beschrijving** |
| --- | --- |
| **10 punten (Uitmuntend)** | Het spel werkt foutloos. De bal beweegt realistisch, stuitert correct, en munten worden zonder vertraging verzameld. Alle interacties functioneren zoals bedoeld. |
| **7-9 punten (Goed)** | Het spel werkt grotendeels correct, maar bevat kleine bugs zoals lichte vertraging bij het verzamelen van munten of kleine visuele foutjes. |
| **4-6 punten (Voldoende)** | Het spel werkt, maar bevat merkbare fouten, zoals een niet-werkende score, onrealistisch stuiteren of een munt die niet verdwijnt. |
| **0-3 punten (Onvoldoende)** | Het spel functioneert niet of vertoont ernstige fouten (bv. crasht, bal beweegt niet, munten worden niet herkend). |

**2. Codekwaliteit**

| **Score** | **Beschrijving** |
| --- | --- |
| **10 punten (Uitmuntend)** | Code is **zeer goed gestructureerd**, overzichtelijk en bevat **duidelijke commentaar**. Variabelen en functies hebben **logische namen** en de code is modulair. |
| **7-9 punten (Goed)** | Code is grotendeels gestructureerd, maar mist hier en daar commentaar of bevat enkele onlogische variabelen en functies. |
| **4-6 punten (Voldoende)** | Code werkt, maar is **onoverzichtelijk** en mist grotendeels commentaar, waardoor het lastig te begrijpen is. |
| **0-3 punten (Onvoldoende)** | Code is chaotisch, bevat veel herhaling en mist commentaar. Het is moeilijk te begrijpen of aan te passen. |

**3. Gebruik van Pygame-functies**

| **Score** | **Beschrijving** |
| --- | --- |
| **10 punten (Uitmuntend)** | Efficiënt en correct gebruik van **rects, botsingsdetectie, geluid, toetsenbordinvoer en blit()**. Code is flexibel en makkelijk uitbreidbaar. |
| **7-9 punten (Goed)** | Pygame-functies worden correct gebruikt, maar sommige implementaties zijn minder efficiënt of minder flexibel. |
| **4-6 punten (Voldoende)** | Pygame-functies worden gebruikt, maar op een inefficiënte manier of met verkeerde logica. Sommige belangrijke functies ontbreken. |
| **0-3 punten (Onvoldoende)** | Pygame-functies zijn nauwelijks of incorrect gebruikt. Botsingen, beweging of geluid ontbreken of werken niet correct. |

**4. Interactie & UI (gebruikersinterface)**

| **Score** | **Beschrijving** |
| --- | --- |
| **10 punten (Uitmuntend)** | Het spel bevat een **startscherm, scoreweergave en een eindscherm**. UI is **duidelijk, goed ontworpen en gebruiksvriendelijk**. |
| **7-9 punten (Goed)** | UI is functioneel en bevat scoreweergave, maar mist extra afwerking (bv. duidelijke meldingen, visuele feedback of een eindscherm). |
| **4-6 punten (Voldoende)** | UI is minimaal: score wordt getoond, maar er is geen startscherm of extra visuele feedback voor de gebruiker. |
| **0-3 punten (Onvoldoende)** | UI ontbreekt grotendeels, of de weergave van score en meldingen werkt niet correct. |

**5. Creativiteit & Extra’s**

| **Score** | **Beschrijving** |
| --- | --- |
| **10 punten (Uitmuntend)** | Het spel bevat **extra features**, zoals obstakels, power-ups, levels of een timer. Er zijn originele toevoegingen die de speelervaring verrijken. |
| **7-9 punten (Goed)** | Er zijn enkele extra toevoegingen, zoals een timer of een obstakel, maar er is geen uitgebreide gameplay-uitbreiding. |
| **4-6 punten (Voldoende)** | Het spel bevat geen extra’s, maar voldoet aan de basisvereisten zonder extra uitdaging. |
| **0-3 punten (Onvoldoende)** | Geen extra's en het spel werkt niet volledig volgens de basisvereisten. |