PyPlay Docentenhandleiding

Auteurs: Marten Postma en Renske Weeda

# Inleiding

Welkom bij de module **PyPlay**. Deze module is onderdeel van de **Python track**, waarin leerlingen stap voor stap leren programmeren aan de hand van motiverende opdrachten. De reeks onderwerpen is zorgvuldig op elkaar afgestemd, en het lesmateriaal wordt aangeboden op het moment dat het het meest relevant is voor de voortgang van de leerling.

Na het aanleren van de basisprincipes van Python, maken leerlingen in de PyPlay-module kennis met het programmeren van een eenvoudige game. Dit gebeurt via tutorials die toewerken naar het maken van het spel **Pong**. Ze leren onder andere hoe ze een bal en een batje tekenen, deze laten bewegen en op elkaar laten reageren.

Gedurende het traject worden nieuwe Python-onderwerpen geïntroduceerd waar dat nodig is – zoals bijvoorbeeld lijsten – en wordt het programmeren met de Play-engine hiermee afgewisseld. Daarnaast maken leerlingen kennis met **gametheorie** en leren ze hoe ze een **toestandsdiagram** kunnen ontwerpen.

Om de status van een game of de inventory van een speler op te kunnen slaan, worden ook **(geneste) dictionaries** en **JSON-bestanden** behandeld. Deze kennis is essentieel voor het eindproject, waarin leerlingen een eigen game ontwerpen en bouwen. Daarin verwerken ze op een betekenisvolle manier alle opgedane kennis in een eigen creatie.

**Didactische uitgangspunten:**

* We werken volgens het **PRIMM-model** (Predict, Run, Investigate, Modify, Make).
* Lesstof wordt aangeboden **op het moment dat het relevant is** voor de leerling.
* De onderwerpen keren op meerdere momenten terug in de module. Dankzij deze **circulaire didactische aanpak** kunnen leerlingen eerder geleerde concepten herhalen, verdiepen en toepassen in nieuwe contexten. Deze herhaling zorgt voor beter begrip en duurzame kennisopbouw.

**Vaardigheden:**  
Leerlingen ontwikkelen naast hun programmeervaardigheden ook hun **zelfstandigheid** en leren hoe ze effectief informatie kunnen opzoeken in onder andere officiële documentatie, tutorials en online bronnen. Denk aan het zoeken van methoden via een cheatsheet of w3schools, of het vinden van kleurcodes voor hun game.

Alle feedback is van harte welkom. Zo zorgen we er samen voor dat we het beste lesmateriaal en dus leerervaring voor onze leerlingen kunnen bieden!

Opmerkingen of vragen? Neem contact op met [renskeweeda@informatica-actief.nl](mailto:renskeweeda@informatica-actief.nl)

Mocht je een fout tegen komen in Play, dien een issue report in: <https://github.com/Corderius-College-Amersfoort/play/issues>

We wensen jou en de leerlingen ontzettend veel plezier!

## Onderwerpen

In deze module behandelen we de volgende onderwerpen aan bod:

* Introductie van de basisprincipes van Python (variabelen, functies, loops, etc.)
* Introductie tot de Play-engine en het tekenen van objecten (zoals een bal of batje)
* Objecten laten bewegen en reageren op elkaar
* Afwisselend: aanvullende Python-onderwerpen zoals lijsten en conditionele logica
* Kennismaking met gamedesign en eenvoudige spelregels
* Ontwerp van een toestandsdiagram voor de game
* Werken met dictionaries en geneste datastructuren
* Opslaan en laden van gegevens met JSON
* Eindproject: het ontwerpen en ontwikkelen van een eigen spel waarin alle onderdelen samenkomen
* *Python H1 t/m H7: Variabelen, keuzes, functies, while loop (geen for loop, geen plannen) (D1,D2)*
* *P: intro gamedesign. Delen van Cursus UX,gaming basis bol 1 (1.1 t/m 1.8)*
* *Pygame tutorials (t/m paddle en bal laten stuiteren)*
* *Python H9 Lijsten (NIET: H10 plannen met lijsten) (B2)*
* *Vervolg Pygame tutorials*
* *Python H11 Dictionaries (C1,C2,C3, B2)*
  + *B2: afweging lijsten vs dictionaries: 11.10*
* *H16 Plannen met dictionaries (B1) (variant op H10 plannen met lijsten)* 
  + *B1: (16.1 t/m 16.10 en 16.13-16.15)*
* *Python: JSON bestanden (C5) – dictionaries opslaan en uitlezen*
* *Toestandsdiagrammen (B3)*
* *Delen van Cursus UX,gaming basis bol 2 t/m 5 (P)*
* *Toestandsdiagrammen: Bij UX,gaming bolletje 4 states (*[*https://moodle.informatica-actief.nl/mod/page/view.php?id=64859*](https://moodle.informatica-actief.nl/mod/page/view.php?id=64859)*) theorie over toestanden opnemen, incl.* [*https://moodle.informatica-actief.nl/mod/page/view.php?id=55131*](https://moodle.informatica-actief.nl/mod/page/view.php?id=55131)*+ vervolgpagina (B3)*

## Verwerkingstijd

Doorlooptijd van +/- 3 perioden (driekwart jaar)

Python: 40 u

Game: 20 u

PO: 20 u

# Relatie met het examenprogramma

**A - Vaardigheden:**

* **A1**: Ontwerpen en ontwikkelen
* **A5**: Informaticainstrumentarium hanteren
* **A6**: Werken in contexten

**B - Grondslagen:**

* **B1**: Algoritmen (standaardalgortimen komen aan bod bij geneste dictionaries)
* **B2**: Datastructuren (lijsten, dictionary’s, geneste dictionary’s)
* **B3**: Automaten (ontwerp van een game mbv. toestandsdiagram)

**C - Informatie:**

* **C1**: Doelstellingen (zoeken en bewerken van gegevens in (geneste) dictionary’s en JSON bestanden)
* **C2**: Identificeren (voor eigen game benodigde gegevens identificeren)
* **C3**: Representeren (overweging maken tussen lijsten, dictionary’s, geneste dictionary’s, JSON)
* **C5**: Gestructureerde data (zoeken in geneste dictionary’s)

**D - Programmeren:**

* **D1**: Ontwikkelen (eigen game in Python maken)
* **D2:** Inspecteren en aanpassen (code lezen, aanpassen, uitbreiden – zowel in Python lesmateriaal als Play tutorials)

**P- User experience (keuzedomein):**

* **P1**: Analyse (bestaande game analyseren adhv 4 pijlers)
* **P2**: Ontwerp (eigen game ontwerpen adhv 4 pijlers)

Afsluiting en toetsing  
De module wordt afgesloten met een **praktische opdracht (PO)** waarin leerlingen een **eigen game ontwerpen en ontwikkelen**. Deze opdracht bestaat uit de volgende onderdelen:

* **Ontwerp:**  
  In de presentatie (zie sjabloon: PO\_EigenGame.pptx) beschrijft de leerling het spelconcept aan de hand van de vier pijlers van spelontwerp: esthetiek, mechanics, verhaal en technologie. Daarnaast maken ze een **toestandsdiagram**, bijvoorbeeld van de speler (power-up, health) of van het spelverloop (startscherm, levels, game over).
* **Ontwikkeling:**  
  De leerling vertaalt het ontwerp naar een werkend spel met behulp van de Play-engine. In de game moeten minimaal de volgende elementen terugkomen:
  + ten minste één bewegend object;
  + ten minste één timer (bijv. voor een power-up of spelduur);
  + gebruik van een dictionary in combinatie met een JSON-bestand (bijv. voor highscore, inventory of level-design);
  + ten minste één grafisch object (zoals een sprite of achtergrondafbeelding).
* **Evaluatie en reflectie:**  
  De leerling evalueert het eindproduct en verzamelt feedback door een ander persoon het spel te laten testen. Ze beschrijven de spelervaring van de gebruiker én reflecteren op hun eigen spel en proces.  
  Daarnaast maken ze een **korte demofilm** waarin ze hun spel demonstreren en toelichten wat ze hebben gemaakt en waar ze trots op zijn.
* **Inleverproducten:**
  + De ingevulde presentatie met ontwerp en evaluatie
  + Een link naar de demofilm
  + Een zip-bestand met de broncode en alle benodigde bestanden (zoals sprites)

**Geschatte verwerkingstijd:** 10 uur