Technisch ontwerp

Pouncy Paws

A Duck Hunt reboot

Versie: 1.0

Auteur: Mikaela Monsma

Datum: 20/03/2025

Inhoudsopgave

**Overzicht3**

**Benodigdheden3**

Hardware en software

Keuze toelichting

**Doel4**

**Projectregels5**

Folderstructuur

Bestandtypes6

Benaming

Versiebeheer

Andere regels

**Technische keuzes7**

Rendering en View

Fysica

Scene Management

Manager Objecten

Interface8

Spelers

Dataopslag

Audio

Performance Optimalisatie9

**Development Roadmap7**

Overzicht

Dit project betreft de ontwikkeling van een 2D-game geïnspireerd op het klassieke Duck Hunt. De speler bestuurt een kat die muizen en vogels moet vangen om punten te scoren. De game bevat 3 levels en 2 verschillende vijanden die de gebruiker moet vangen. De game wordt ontwikkeld met behulp van Python en het Pygame framework.

Benodigdheden

**Hardware -**

**Ontwikkeling**:

Laptop of desktop met minimaal:

* + Processor: Dual-core (Intel i3 of AMD Ryzen 3 of beter).
  + Geheugen: 8GB RAM (minimaal), 16GB aanbevolen.
  + Grafische kaart: Basisondersteuning voor 2D-rendering (Intel UHD Graphics of vergelijkbaar).
  + Opslag: Minimaal 500MB vrije schijfruimte voor projectbestanden en assets

**Testen**:

Voor het testen van de game zal gebruik worden gemaakt van dezelfde test omgeving. Hier is voor gekozen omdat de game uiteindelijk niet gepubliceerd/ commercieel uitgebracht zal worden.

**Software -**

Ontwikkelomgeving - Visual Studio Code met de Python extensie.

* Python extensie v2025.6.1
* Pygame 2.6.1
* Python Debugger extensie v2025.8.0

Versiebeheer - Dit wordt gedaan met behulp van Github, in een speciale repository voor dit project.

**Waarom deze keuzes?**

Python is relatief makkelijk te leren en gebruiken, waardoor je snel een prototype kunt maken. Daarnaast biedt dit een mogelijkheid om op spelende wijze een nieuwe taal te beheersen.

Pygame is een lichte, open-source gamebibliotheek met goede ondersteuning voor 2D-games, geluid en animaties.

VS Code biedt veel handige extensies voor Python-ontwikkeling en debugging.

GitHub maakt versiebeheer en back-ups eenvoudig.

Doel

Dit spel is ontworpen voor desktopgebruik. Het wordt uitgevoerd vanuit de ontwikkelomgeving door op de run-knop te drukken. De game wordt niet commercieel uitgebracht, dus distributie vindt alleen plaats via GitHub of lokaal.

|  |  |  |
| --- | --- | --- |
| **Specificatie** | **Minimum** | **Aanbevolen** |
| OS | Windows 10 / 11 | Windows 10 / 11 |
| Processor | Dual-core, 1.5 GHz | Quad-core, 2.5 GHz |
| RAM | 8GB | 16GB |
| Grafische kaart | Onboard (Intel UHD) | Dedicated (Nvidia MX of beter) |
| Opslag | 500MB vrij | 1GB vrij |

Projectregels (Werkwijze)

**Folderstructuur**

DuckHunt/

├── main.py

├── game.py

├── menu.py

├── timer.py

├── useri.py

├── LICENSE

├── .gitignore

├── Assets/

│ ├── Animals/

│ │ ├── Images/

│ │ │ └──images\_spritesheet.png

│ │ ├── animal.py

│ │ ├── bird.py

│ │ ├── mouse.py

│ │ └── animal\_manager.py

│ │

│ ├── Character/

│ │ ├── Images/

│ │ │ └──player.png

│ │ └── player.py

│ │

│ ├── Audio/

│ │ ├── Music/

│ │ │ └──music.mp3

│ │ ├── Soundeffects/

│ │ │ └──effects.mp3

│ │ └── audio\_manager.py

│ │

│ ├── Levels/

│ │ └── level.py

│ │

│ ├── bgs/

│ │ └── Images\_backgrounds.png

│ │

│ └── Upgrades/

│ └── upgrade\_manager.py

│

└── \_\_pycache\_\_/

**Bestandstypes**

* **.py:** Python broncode bestanden.
* **.png, .jpg:** Afbeeldingsbestanden voor sprites en achtergronden.
* .**wav, .mp3:** Voor eventuele geluidsbestanden.
* **.ttf:** Voor eventuele lettertypebestanden.

**Benaming**

* **Bestanden**: snake\_case of snake (bijv. game\_manager.py, duck\_sprite.png)
* **Classes**: PascalCase (bijv. GameManager, DuckSprite)
* **Functies en variabelen**: snake\_case (bijv. update\_score(), player\_health)
* **Constanten**: UPPERCASE\_WITH\_UNDERSCORES (bijv. MAX\_DUCKS, DEFAULT\_SPEED)

**Versiebeheer**

* GitHub wordt gebruikt voor versiebeheer van deze game.
* Commit-berichten moeten duidelijk en beschrijvend zijn.

**Andere Regels**

* Houd een consistente code- en commentstijl aan.
* Zorg voor duidelijke comments.
* Gebruik OOP (Object Oriented Programming) voor beter onderhoud en hergebruik van code.

Technische Keuzes

**Rendering en View**

* 2D-rendering met Pygame's sprite en surface system.
* Zijwaarts perspectief, vergelijkbaar met het originele DuckHunt.
* Alleen een achtergrond, het is zijwaarts perspectief en de items komen aan de zijkanten tevoorschijn het scherm in.

**Fysica**

* Eenvoudige 2D-fysica voor beweging van objecten.
* Bepaalde patronen voor de beweging van de items.
* Zwaartekracht voor het springen van de speler.

**Scene Management**

* Eén scène per level zoals in het origineel.
* De achtergrond scene verandert alleen bij een nieuw level.
* Het pauze scherm en game-over scherm blijven hetzelfde tijdens de hele game.
* Progressieve moeilijkheidsgraad per niveau

**Manager Objecten**

De volgende managers worden gebruikt:

* GameManager: Centrale controle over de spelstatus
* Assets Map: Laden en cachen van afbeeldingen, geluiden. Ook komen hier de benodigde .py files in.
* Scene/ Level Manager: Beheer van verschillende spelschermen

**Interface**

Een simpele interface met de volgende onderdelen:

* Tijd-display in de hoek linksonder.
* Score komt naast de tijd-display.
* Rechtsonder komt de score die behaald moet worden in het level, of er komt een Progressbar die de voortgang duidelijk laat zien.
* Level-indicator linksboven.

**Spelers**

* Besturing voor de primaire game met W, A, D.
* Toetsenbordondersteuning voor menu-navigatie zijn de pijltjes toetsen.
* Pauze toets, backspace.
* Geen online functionaliteit.

**Data Opslag**

* Spelvoortgang wordt niet opgeslagen tussen sessies
* Tussentijds in dezelfde sessie wordt bepaalde informatie opgeslagen.

**Audio**

Kleine audio fragmenten zullen worden toegevoegd na het bekend worden van de minimale eisen voor het project.

**Performance Optimalisatie**

* Sprite sheets voor efficiënt geheugengebruik.
* Beperking van actieve objecten op het scherm
* Selectieve updates voor objecten buiten het zichtbare gebied

(Deze game wordt gemaakt met de volgende doelen:

* Python ontdekken
* Pygame ontdekken
* Het oefenen met OOP

Hierdoor is performance iets meer op de achtergrond geplaatst.)

Development Roadmap

1. Basis menu met opties.
2. Core game mechanics, input en collision.
3. UI, levels en score systeem.
4. Level progressie en behandeling moeilijkheidsgraad.
5. Visuele effecten (sprite sheets en audio).
6. Performance optimaliseren en bug fixing.
7. Polishing en game balancing, denk hierbij aan moeilijkheidsgraad afstellen.