**MaffiaTrap (Technisch ontwerp)**



**Naam:** Vu Le  
**Klas:** OOSE R-B-s (vorig jaar I1DB-N)  
**Groepsnummer:** ? **Docentnaam:** Michel Koolwaaij **Datum:** 4-4-2019 **Course/semestergegevens:** OOPD, semester 3 (propedeuse-fase) **Versienummer:** 1

**Inhoudsopgave**

Inhoud

[1. Inleiding 3](#_Toc3888142)

[2. Klassendiagram 3](#_Toc3888143)

[2.1. Toelichting 3](#_Toc3888144)

[3. Klassen uitgelegd 4](#_Toc3888145)

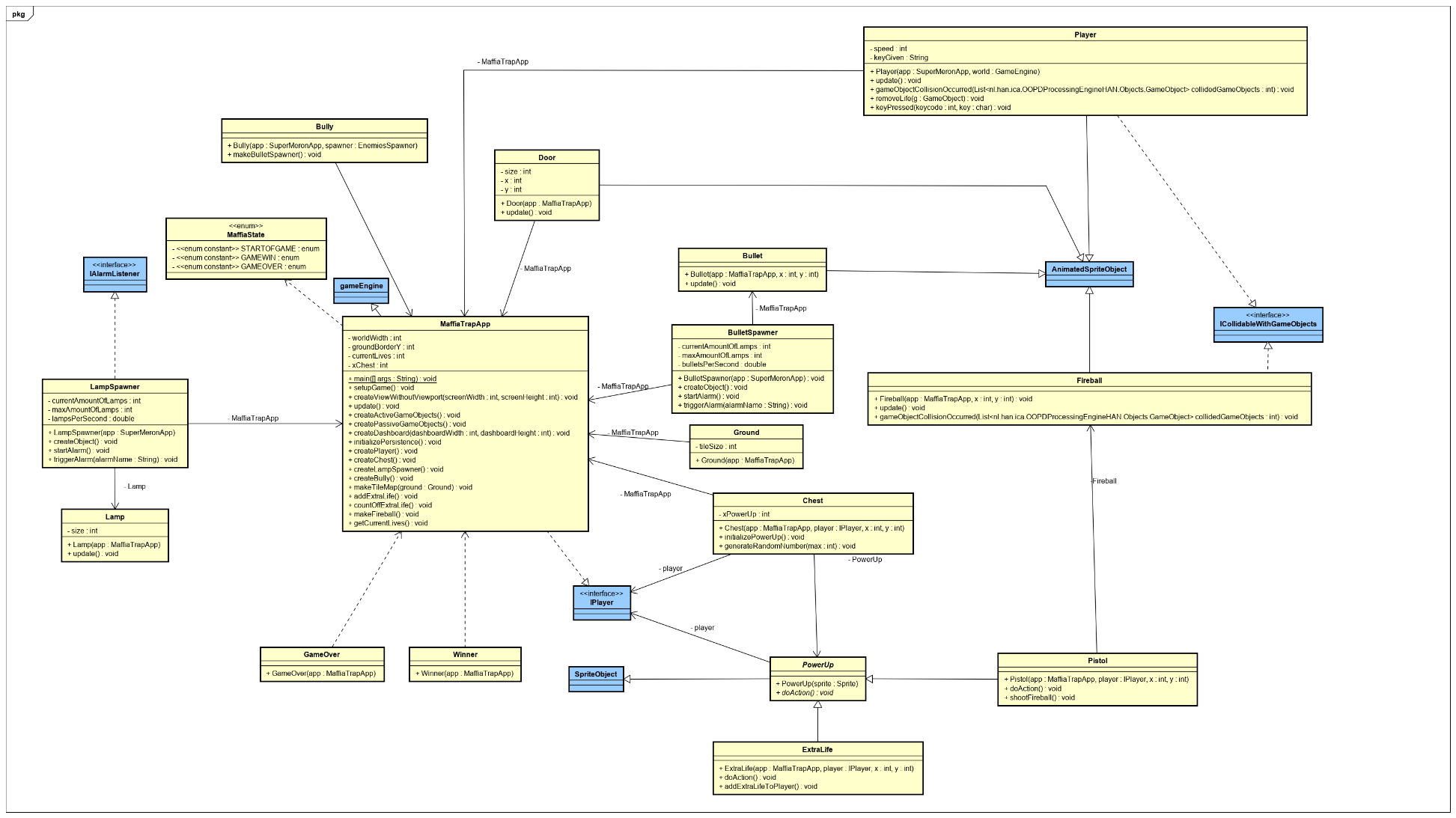
[4. Zelfgemaakte methodes in de klassen uitgelegd 7](#_Toc3888146)

[5. Bibliografie 11](#_Toc3888147)

# Inleiding

Voor het vak Object-Oriented Program Development (OOPD) wordt een game gemaakt door middel van een game-engine, die ontwikkeld is door medestudenten van dezelfde opleiding.   
In dit verslag wordt eerst een klassendiagram laten zien, die gemaakt moet worden voordat de game gerealiseerd mag worden. Want dan is er een ontwerp waar de game-ontwikkelaars zich vast aan kunnen houden.   
Na het lezen van dit document moet de lezer een idee kunnen krijgen van hoe de game eruit moet zien en ook de game kunnen bouwen.

# Klassendiagram



Figuur : Klassendiagram OOPG-MaffiaTrap (Astah)

### Toelichting

Dit is de eerste, tussentijdse versie van het klassendiagram.

# Klassen uitgelegd

*Dikgedrukte woorden (Calibri 14) zijn tevens de namen van de packages waar de klassen per package worden opgeslagen.*

**main**

* **MaffiaTrapApp**

De *main* class (wereld) van deze game, hierin worden de belangrijke onderdelen (methoden) van de game aangeroepen en gerealiseerd.  
Hier wordt dus de setup van de game geinitialiseerd, bijvoorbeeld het dashboard, het scherm, de actieve gameobjecten en de passieve gameobjecten (objecten die ‘stilstaan’, geen beweging uitvoeren) zoals Door en Ground. Zie 3. Zelfgemaakte methodes in de klassen uitgelegd voor verduidelijking.

**Gamestates**

* **MaffiaState**

Een enum die overal toegankelijk is. In deze enum zitten verschillende gamestatussen, die aangegeven in welke fase de game zich bevindt: Begin van de game, gewonnen en game over.

In de main-klasse(MaffiaTrapApp) wordt de enum toegepast, per gamestatus wordt beschreven wat er wordt gedaan.

* **Winner**

In deze klasse maakt het scherm wanneer de speler gewonnen heeft.

* **GameOver**

In deze klasse maakt het scherm wanneer de speler verloren heeft.

**entities**

* **IPlayer**

Een interface waarin de functies voor een willekeurige power-up zitten die de speler van een power-up voorziet. Die functies worden in MaffiaTrapApp gevuld en gebruikt in de child-klassen (ExtraLife en Pistol) van de abstracte klasse PowerUp (wordt later uitgelegd, zie powerUps) gebruikt.

* **Player**

De klasse waarin de speler wordt geinitialiseerd. In de Player-klasse wordt onder anderen de sprite van de speler aangemaakt, aangegeven hoe een speler kan bewegen, de toetsenbordinstellingen aangegeven (dus hoe je de speler kan besturen) en wat er gebeurt als de speler een bepaald object raakt.

* **Chest**

De klasse waarin de schatkist wordt aangemaakt, die meteen bij de setup van de game (in MaffiaTrapApp) wordt gemaakt. Ook is in deze klasse een functie waarin een willekeurige power-up uit de schatkist wordt gekozen.

**powerUps**

* **PowerUp**

Een abstracte klasse die functies initialiseert die in de klassen ExtraLife en Pistol worden gebruikt. Deze klasse heeft als extensie de klasse SpriteObject van de game-engine.  
Deze klasse bevat een constructor waarin de sprite van een power-up geinitialiseerd kan worden en bevat een abstracte functie genaamd DoAction(), waarin die actie van een power-up wordt uitgevoerd.

* **ExtraLife**

Een klasse die een extra leven voor de speler aanmaakt. Deze klasse is een child van de abstracte klasse PowerUp.

* **Pistol**

Een klasse die een pistool voor de speler aanmaakt. Deze klasse is een child van de abstracte klasse PowerUp.

**Traps**

* **Bully**

Deze klasseinitialiseert de sprite van de gangster, de vijand van de speler, die gemaakt kan worden. Ook maakt deze klasse de spawner van de schietkogels die de gangster kan afvuren (klasse BulletSpawner).

* **Lamp**

Deze klasse initialiseert de beweging en de sprite van de vallende lampen. In de Player-klasse wordt aangegeven wat er gebeurt als je een lamp raakt.

* **LampSpawner**

Dit is de bel-klasse voor de klasse Lamp. Deze klasse implementeert de interface IAlarmListener. Deze klasse zorgt ervoor dat vallende lampen door middel van een timer tevoorschijn komen.

**shootEffects**

* **Bullet**

Deze klasse initialiseert de sprite van een schietkogel en in de methode update(), die uit SpriteObject (extensie van Bullet) wordt overgeërfd, worden de eigenschappen (richting + snelheid van de kogel) aangegeven.

* **BulletSpawner**

Dit is de bel-klasse voor de klasse Bullet. Deze klasse implementeert de interface IAlarmListener. Deze klasse zorgt ervoor dat de schietkogels die de gangster op de speler kan schieten door middel van een timer tevoorschijn komen.

* **Fireball**

Deze klasse initialiseert de sprite voor een vuurbal. Net als in de klasse Bullet worden de eigenschappen (richting + snelheid van de vuurbal) aangegeven in de methode update(), die wederom uit SpriteObject wordt overgeërfd.

**standardGameObjects**

* **Door**

Deze klasse initialiseert de sprite van de deur. In de Player-klasse wordt aangegeven wat er gebeurt als je een deur ‘opent’.

* **Ground**

Deze klasse maakt de ondergrond waar de speler op staat door middel van een sprite en een array genaamd TileMap.

# Zelfgemaakte methodes in de klassen uitgelegd

|  |  |  |
| --- | --- | --- |
| **Klasse:** MaffiaTrapApp |  |  |
| **Type** | **Naam** | **Beschrijving** |
| void | createActiveGameObjects() | Deze functie maakt de ‘actieve’ (objecten die beweging uitvoeren) gameobjecten aan die in het begin van de game al staan. |
| void | createPassiveGameObjects() | Deze functie maakt de ‘stilstaande’ (niet-bewegende) gameobjecten aan die in het begin van de game al staan. |
| void | createPlayer() | Deze functie maakt de speler, deze speler komt van de klasse **Player.** Deze functie is tevens onderdeel van de functie createActiveGameObjects(). |
| void | createChest() | Deze functie maakt de schatkist aan, de schatkist komt van de klasse Chest. Deze functie is tevens onderdeel van de functie createActiveGameObjects(). |
| void | createLampSpawner() | Deze functie maakt de lampspawner. Deze functie is tevens onderdeel van de functie createActiveGameObjects(). |

|  |  |  |
| --- | --- | --- |
| void | createBully() | Deze functie maakt de gangster (vijand) aan, de gangster komt van de klasse Bully. Deze functie is tevens onderdeel van de functie createActiveGameObjects(). |
| void | makeTileMap() | Deze functie maakt de ondergrond waar de speler op loopt. Deze functie is een onderdeel van de functie createPassiveGameObjects(). |
| void | addExtraLife() | Deze functie zorgt dat de teller van het aantal levens dat een speler op dat moment heeft (bij de methode initializePersistence uit de game-engine-klasse wordt het aantal levens bij de start gemaakt) opgehoogd met 1. Deze functie komt uit de interface *IPlayer*, die interface wordt geïmplementeerd door MaffiaTrapApp. |
| void | countoffExtraLife() | Deze functie zorgt dat de teller van het aantal levens dat een speler op dat moment heeft (bij initializePersistence wordt het aantal levens bij de start gemaakt) verlaagd wordt met 1. Het gevolg kan zijn dat in de klasse Player een functie checkt hoeveel levens er nog over zijn, en als er 0 levens over zijn, is de speler dood en is het spel voorbij. Deze functie komt uit de interface *IPlayer*, die interface wordt geïmplementeerd door MaffiaTrapApp. |
| void | makeFireball() | Deze functie initialiseert en voegt de vuurbal toe, die uit het pistool van de speler geschoten kan worden, als dat de power-up is die gekozen is. Deze functie komt uit de interface *IPlayer*, die interface wordt geïmplementeerd door MaffiaTrapApp. |
| int | getCurrentLives() | Deze functie verkrijgt het huidige aantal levens, die in de klasse Player wordt aangeroepen om te checken of de speler genoeg levens heeft. Zo niet, wordt de gamestate (enum) GAMEOVER geactiveerd en wordt het gameoverscherm geladen. Als er wel genoeg levens over is, wordt de functie countOffExtraLife() aangeroepen die het aantal levens – 1 doet. |

|  |  |  |
| --- | --- | --- |
| **Klasse:** Chest |  |  |
| **Type** | **Naam** | **Beschrijving** |
| void | initializePowerUp() | Deze functie genereert door een willekeurig nummer, die de functie generateRandomNumber wordt gekozen, een power-up die uit de schatkist moet komen. |
| int | generateRandomNumber() | Deze functie genereert een willekeurig getal. |

|  |  |  |
| --- | --- | --- |
| **Klasse:** Player |  |  |
| **Type** | **Naam** | **Beschrijving** |
| void | removeLife() | Deze functie verlaagt het aantal levens van de speler met 1 en checkt via een if-statement en een functie uit de MaffiaTrapApp (getCurrentLives) of het aantal levens 0 is. Als dat zo is, is het game-over. |

|  |  |  |
| --- | --- | --- |
| **Abstracte klasse:** PowerUp |  |  |
| **Type** | **Naam** | **Beschrijving** |
| void | doAction() | Deze functie voert de actie van een power-up uit (klasse ExtraLife/Pistol). |

*Notitie bij doAction(): DoAction() kan dus veel betekenen, het kan een extra leven toevoegen of een vuurbal schieten uit het pistool. Dit is polymorfie, want in de functie gameObjectCollisionOccured (uit de interface ICollidableWithGameObjects) wordt aangegeven dat als de speler een power-up aanraakt (je weet niet welke), dat DoAction() uitgevoerd kan worden. Je weet van tevoren niet wat DoAction() doet bij de speler, totdat je de schatkist opent en dat je weet welke power-up de speler op dat moment krijgt.*

|  |  |  |
| --- | --- | --- |
| **Enum:** MaffiaState |  |  |
| **Type** | **Naam** | **Beschrijving** |
| enum constant | *START\_OF\_GAME* | Deze enum-constante wordt in MaffiaTrapApp gebruikt voor het oproepen van methodes in die fase: in deze fase worden alle objecten die nodig zijn bij de setup van de game aangemaakt. |
| enum constant | *GAMEWIN* | In deze fase wordt het winnaarsscherm gemaakt wanneer een speler heeft gewonnen. |
| enum constant | *GAMEOVER* | In deze fase wordt het verliezersscherm gemaakt wanneer een speler heeft dood is gegaan. |

|  |  |  |
| --- | --- | --- |
| **Klasse:** ExtraLife |  |  |
| **Type** | **Naam** | **Beschrijving** |
| void | addExtraLifeToPlayer() | De functie die een extra leven aan de speler toevoegt. Dit wordt gedaan door een functie genaamd addExtraLife uit de main (MaffiaTrapApp) aan te roepen. Dan wordt deze functie in doAction() aangeroepen in die klasse. |

|  |  |  |
| --- | --- | --- |
| **Klasse:** Pistol |  |  |
| **Type** | **Naam** | **Beschrijving** |
| void | createFireball() | De functie die een vuurbal maakt. Dit wordt gedaan door een functie genaamd makeFireball uit de main (MaffiaTrapApp) aan te roepen. Dan wordt deze functie in doAction() aangeroepen in die klasse. |

|  |  |  |
| --- | --- | --- |
| **Klasse:** BulletSpawner |  |  |
| **Type** | **Naam** | **Beschrijving** |
| void | createObject() | Deze klasse initialiseert de schietkogel die afgevuurd kan worden, uit de klasse Bullet. |
| void | startAlarm() | Deze functie start de timer die de schietkogels van de gangster kan afvuren door middel van het aantal kogels per seconde die ingesteld kan worden (bulletsPerSecond). |

|  |  |  |
| --- | --- | --- |
| **Klasse:** Bully |  |  |
| **Type** | **Naam** | **Beschrijving** |
| void | makeBulletSpawner() | Deze klasse maakt de bulletspawner aan die de gangster kan afvuren. |

|  |  |  |
| --- | --- | --- |
| **Klasse:** LampSpawner |  |  |
| **Type** | **Naam** | **Beschrijving** |
| void | createObject() | Deze klasse initialiseert de vallende lamp(en), uit de klasse Lamp. |
| void | startAlarm() | Deze functie start de timer van de vallende lampen, door middel van het aantal lampen per seconde die ingesteld kan worden (lampsPerSecond). |

# Bibliografie

*I-OOPD-17/18-S2*. (sd). Opgehaald van OnderwijsOnline: https://onderwijsonline.han.nl/elearning/content/dNwkM7NR