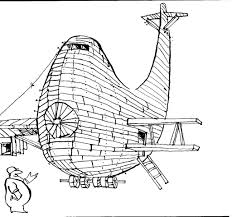
Technisch ontwerp

Door: Marijn Degen (582464) en Sven Molhuijsen(582660)

Onderdeel: Object oriented programming development (OOPD)

Klas: I1DD-N

# Inhoud

[**Inhoud**](#_6q6v1dotil1i) **2**

[**Inleiding**](#_czoalu26uviw) **4**

[**Klassendiagram**](#_uvs954bonpvl) **5**

[**Omschrijving klassen**](#_myblivi814i3) **6**

[**HeistWorld**](#_cqq0m6dxw6df) **6**

[Inhoud](#_dbsdaxhtlvm8) 6

[Werking](#_6ub8gbhqr86j) 6

[**VerkeersSpawner**](#_8xgandgxt7zn) **7**

[Inhoud](#_aew374pddv3j) 7

[Werking](#_4wuopklia3rn) 7

[**Missie**](#_oluvprz7dhq) **8**

[Inhoud](#_s9vtjwtzihfh) 8

[Werking](#_upcjyut5amt1) 8

[**Gebruiker**](#_ai1wfywyajsp) **9**

[Inhoud](#_otr440b63hx9) 9

[Werking](#_fz16d9prti9c) 9

[**VluchtAuto**](#_311j8oqy0iui) **10**

[Inhoud](#_dqstgcwysffu) 10

[Werking](#_ae8qx6z1zsm5) 10

[**Auto**](#_vievmgqlx5rb) **11**

[Inhoud](#_6i8p2ltgikt5) 11

[Werking](#_hgn8avfdbpec) 11

[**BurgerAuto**](#_stfyav3xpg51) **12**

[Inhoud](#_lvmd9396ssmc) 12

[Werking](#_oj2y9m6korgz) 12

[**PolitieAuto**](#_qf423sgx7hbh) **13**

[Inhoud](#_dcsj39vg7dk6) 13

[Werking](#_kh3dheo6k5wq) 13

[**Toets**](#_f8x3xxx15ss0) **13**

[Inhoud](#_bp2lcihb15tn) 14

[Werking](#_fhy6a2pdgjqq) 14

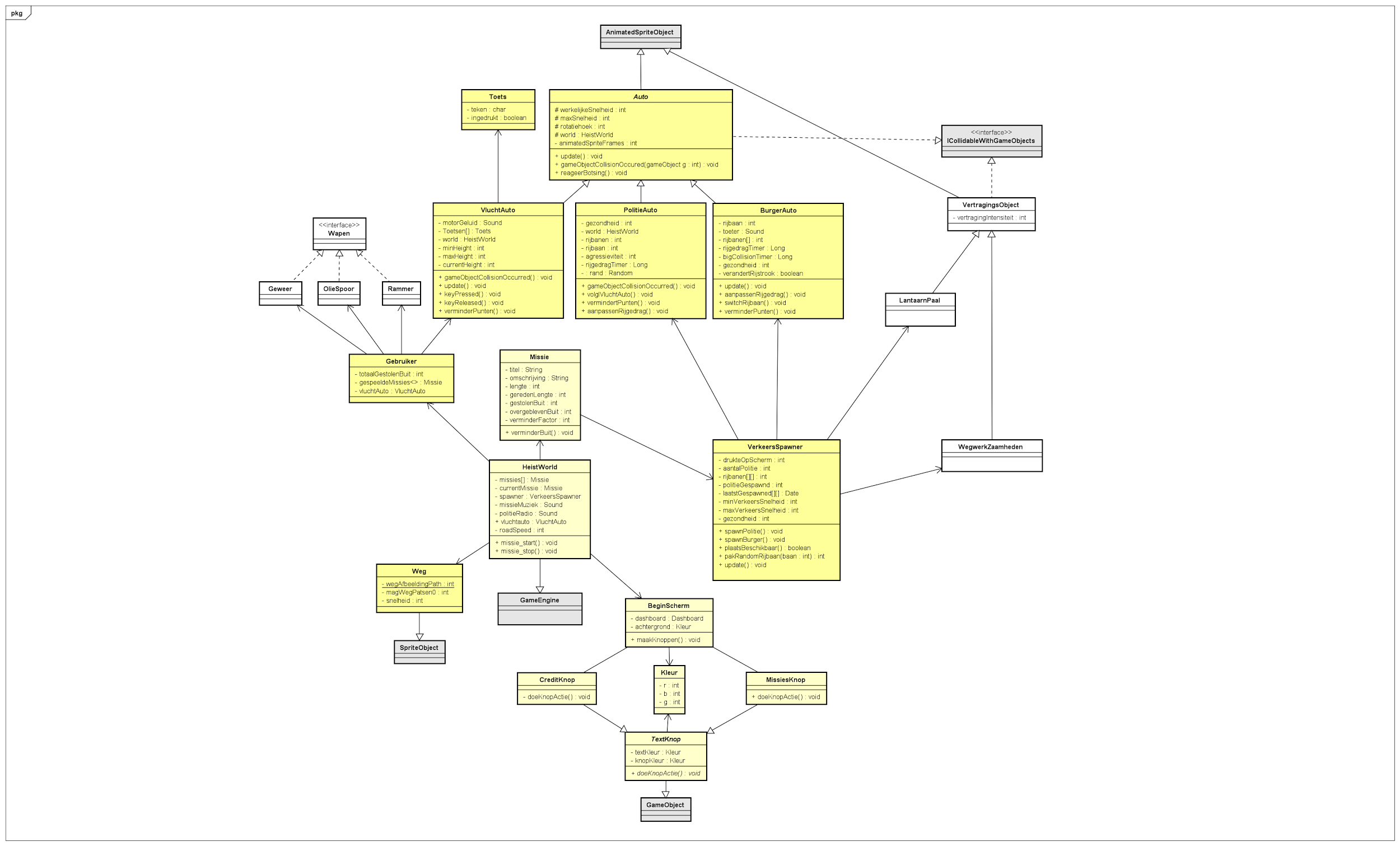
[**Conclusie**](#_f5t1m1enlmz0) **15**

# Inleiding

In dit functioneel ontwerp gaan we in op de technishe aspecten van ons project, eerst zullen we het klassendiagram laten zien wat we hebben otworpen. Daarna zullen we de klasses een voor een uitleggen, deze uitleg zal ook terug te vinden zijn in de javadocs

# Klassendiagram

Dit klassendiagram is ook te bekijken via het bijgeleverde astah bestand.

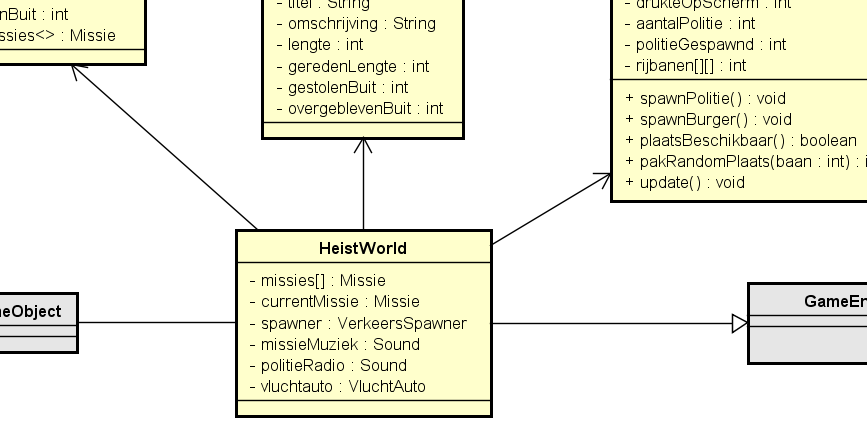


# Omschrijving klassen

In onderstaande kopjes gaan we de werking van iedere klasse uitleggen en hoe ze met elkaar samenhangen, eerst noemen we de naam, daarna leggen we de klasse uit, en eventueel de methodes binnen die klassen. Voor de prioriteit van de implementatie van de klassen, zie de MosCow toelichting van het functioneel ontwerp.

# HeistWorld

### Inhoud

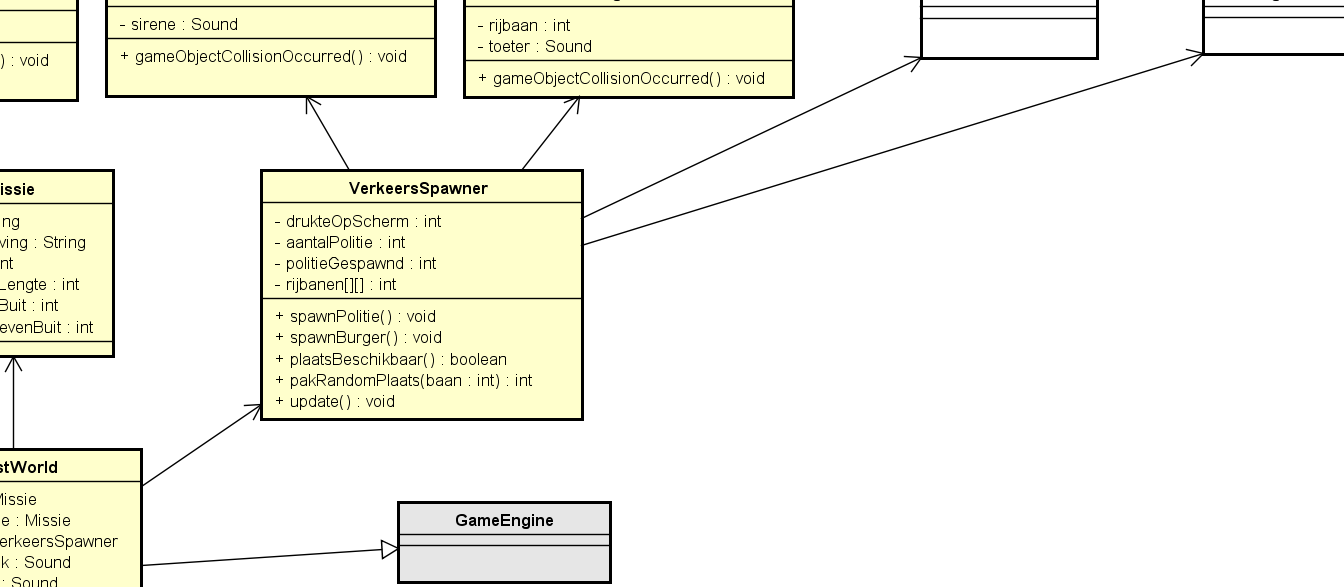


### Werking

Deze klasse is onze hoofdklasse, in deze klasse zitten alle belangrijke elementen,die door de klasse in harmonie met elkaar kunnen werken. De klasse heeft een aantal missies, daarbij wordt iedere missie die op dit moment wordt gespeeld in de currentMissie variabele gezet. ook staat de vluchtauto hier gedefinieerd, die tijdens de start van een missiezal beginnen met rijden.

# VerkeersSpawner

### Inhoud



### Werking

De verkeersSpawner heeft een aantal variabelen die kunnen bijhouden hoeveel verkeer er op de weg is. met de functie update wordt de verkeersintensiteit in stand gehouden, wanneer er te weinig auto’s op het scherm zijn zal hij met de functie spawnBurger een burger spawnen, die vervolgens een array met zijn vervolgens zal hij met in de araryllist laatstGespawned de spawntijd voor een rijbaan invullen.

# Missie

### Inhoud

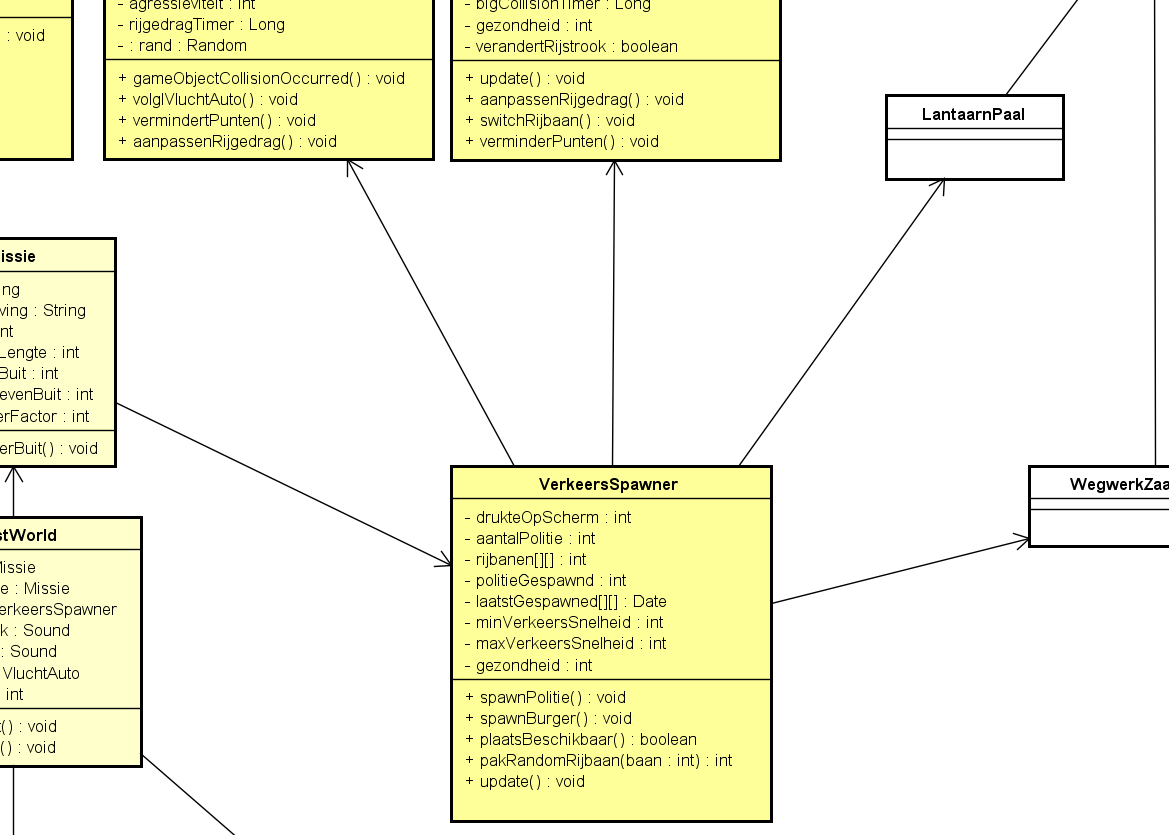
# 

### Werking

De missie bevat een omschrijving van de missie, een titel, de lengte die gereden moet worden, de buit die gestolen wordt en de overgebleven buit. de laatste twee worden verminderd door de functie verminderBuit, die de buit verminderd met de meegeleverde verminderingsfactor

# Gebruiker

### Inhoud

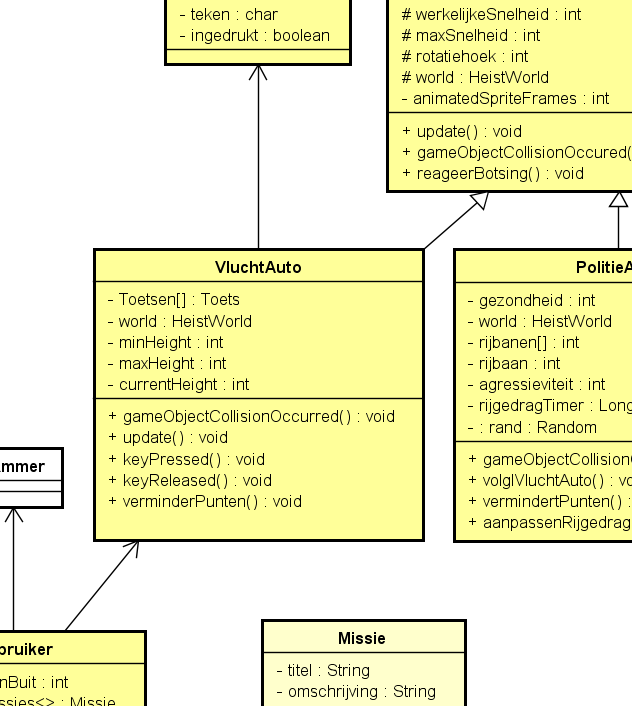


### Werking

De gebruiker heeft de variabele totaalGestolenBuit, een arrayList met gespeelde missies en een vluchtAuto, de bedoeling is dat de gebruiker uiteindelijk upgrades zou kunnen kopen met de totaalGestolenBuit.

# VluchtAuto

### Inhoud



### Werking

De vluchtauto heeft al het gedrag van een normale auto overgenomen, het enige verschil is echter dat deze auto zichzelf kan besturen. bij een collision occured wordt verminderPunten uitgevoerd

# Auto

### Inhoud

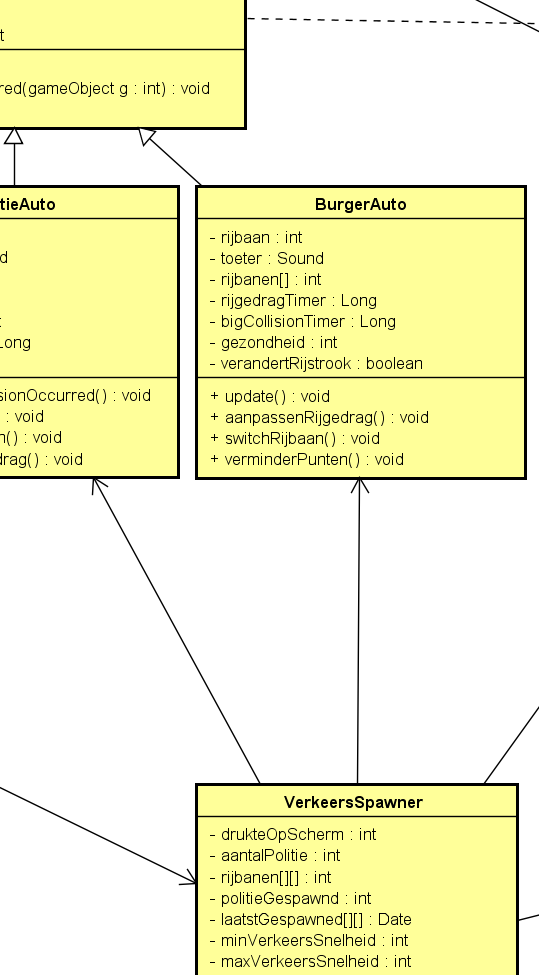
# 

### Werking

Een auto overerft alle waardes van spriteObject, en heeft zelf de waarde gezondheid, werkelijkeSnelheid en maxSpeed, die bepalen het rijgedrag van de auto.

# BurgerAuto

### Inhoud



### Werking

Een burgerauto heeft een meegegeven rijbaan, en een array met alle beschikbare rijbanen, uiteindelijk moet een auto kunnen switchen van rijbaan

# PolitieAuto

### Inhoud

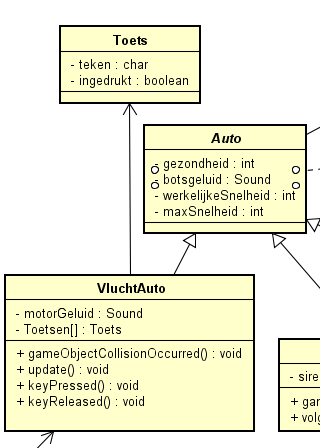


### Werking

Een politieAuto zal altijd de vluchtauto proberen te volgen doormiddel van de functie volgVluchtAuto.

# Toets

## Inhoud



### Werking

Omdat de keyreleased en keypressed methodes op zich zelf niet genoeg functionaliteit bieden in het controleren van de speler, worden deze methodes aangevuld met de class toets, een toets. De klass houdt bij welke toets op het huidige moment is ingedruikt.

# Gebruikte middelen

**Muziek/geluiden**

Gorillaz - Feel good

Politie sirenes - <https://www.youtube.com/watch?v=WuY2zOYFPbE>

Bots geluid - <http://www.freesfx.co.uk/soundeffects/impacts-crashes/>

politie radio - <https://www.youtube.com/watch?v=qbXwpa1mz6g>

**Afbeeldingen**

politieAuto <https://www.123rf.com/photo_60003142_stock-vector-police-and-taxi-cars-top-view-vector-set-car-for-taxi-and-police-illustration-transport-taxi-and-aut.html>

<https://s-media-cache-ak0.pinimg.com/originals/48/35/f5/4835f5bf1b927e21cbb980dd39b451d0.jpg>

gewone autos - <https://static2.bigstockphoto.com/8/9/2/large2/29861441.jpg>

Weg- paint.net

FO - <https://suneco.nl/blogs/het-functioneel-ontwerp-de-bouwtekening-van-de-website>

TO - <http://labyrinth.rienkjonker.nl/sites/default/files/5Technischontwerp.jpg>

# 

# Conclusie

Wij denken op basis van bovenstaande klassen en objecten het spel te kunnen bouwen,

# Bronnenlijst