|  |
| --- |
| OOpd Game technisch ontwerp\_  Ping (Pong) |

Tim Wouters en Lars van Duijnhoven

29 maart 2023

INHOUDSOPGAVE

[1 Analyse/ functioneel ontwerp 4](#_Toc129618442)

[1.1 Naam: 4](#_Toc129618443)

[1.2 Naam bestaand/ originele spel: 4](#_Toc129618444)

[1.3 Doel van het spel: 4](#_Toc129618445)

[1.4 Omschrijving van de wereld: 4](#_Toc129618446)

[1.5 Het perspectief waarin de speler de wereld ziet: 4](#_Toc129618447)

[1.6 Een beschrijving van de acties die de speler kan doen: 4](#_Toc129618448)

[1.7 Een beschrijving van objecten: 5](#_Toc129618449)

[1.8 Een beschrijving van start/eind: 6](#_Toc129618450)

[1.9 Een beschrijving van overige elementen: 6](#_Toc129618451)

[2 Schermschetsen: 7](#_Toc129618452)

[2.1 Welke delen van het spel maak je: 8](#_Toc129618453)

[9](#_Toc129618454)

# Technisch ontwerp

## Klassendiagram

Diagram

Description automatically generated

Indien dit niet te lezen is, is het astah bestand toegevoegd in deze folder.

## Toelichting Klassendiagram

* Ping:

Dit is de main klasse van het spel.

main: De main functie die het programma start.

setupGame(): Dit is de functie die de settings aanpast van het spel, zoals de size en titel.

setupScenes(): Zoals de naam al zegt, worden hier de scenes toegevoegd.

* gameSchermMultiplayer:

De klasse voor het gameScherm van multiplayer

setupScene(): De achtergrond en achtergrondmuziek aanpassen van deze scene.

setupEntities(): De entities die in deze scene voorkomen aanmaken en toevoegen aan het scherm.

setupEntitySpawner(): Deze spawnt de power-ups en extra ballen.

* gameOverSchermMultiplayer:

De klasse voor het gameOverScherm van multiplayer

setupScene(): De achtergrond en achtergrondmuziek aanpassen van deze scene.

setupEntities(): De entities die in deze scene voorkomen aanmaken en toevoegen aan het scherm.

setGewonnenSpeler(): Dit veranderd de tekst die gedisplayed wordt.

* gameSchermSingleplayer:

De klasse voor het gameScherm van singleplayer

setupScene(): De achtergrond en achtergrondmuziek aanpassen van deze scene.

setupEntities(): De entities die in deze scene voorkomen aanmaken en toevoegen aan het scherm.

* gameOverSchermSingleplayer:

De klasse voor het gameOverScherm van singleplayer

setupScene(): De achtergrond en achtergrondmuziek aanpassen van deze scene.

setupEntities(): De entities die in deze scene voorkomen aanmaken en toevoegen aan het scherm.

* titelScherm:

Dit is de klasse voor het titelscherm/ beginscherm.

titelScherm(): Dit is de constructor van titelscherm waar ping aan wordt meegegeven.

setupScene(): De achtergrond en achtergrondmuziek aanpassen van deze scene.

setupEntities(): De entities die in deze scene voorkomen aanmaken en toevoegen aan het scherm.

* playAgainKnop:

De knop die verwijst naar de voorheen gespeelde game scene.

onMouseButtonPressed(): De functie die aangeroepen wordt op het moment dat er op de knop geklikt wordt. Hier moet je dus in stoppen wat er gebeurd als er op de knop is gedrukt.

onMouseEntered(): De functie om te vertellen wat er moet gebeuren als er met de muis over de knop wordt gehovered.

onMouseExited(): De functie die wordt aangeroepen als de muis weer van de knop af is. Bijvoorbeeld om de onMouseEntered te counteren, zodat de knop weer naar zijn originele kleur gaat.

* exitKnop:

De knop die het spel afsluit.

onMouseButtonPressed(): De functie die aangeroepen wordt op het moment dat er op de knop geklikt wordt. Hier moet je dus in stoppen wat er gebeurd als er op de knop is gedrukt.

onMouseEntered(): De functie om te vertellen wat er moet gebeuren als er met de muis over de knop wordt gehovered.

onMouseExited(): De functie die wordt aangeroepen als de muis weer van de knop af is. Bijvoorbeeld om de onMouseEntered te counteren, zodat de knop weer naar zijn originele kleur gaat.

* multiPlayerKnop:

De knop die verwijst naar de multiplayer game scene.

onMouseButtonPressed(): De functie die aangeroepen wordt op het moment dat er op de knop geklikt wordt. Hier moet je dus in stoppen wat er gebeurd als er op de knop is gedrukt.

onMouseEntered(): De functie om te vertellen wat er moet gebeuren als er met de muis over de knop wordt gehovered.

onMouseExited(): De functie die wordt aangeroepen als de muis weer van de knop af is. Bijvoorbeeld om de onMouseEntered te counteren, zodat de knop weer naar zijn originele kleur gaat.

* singlePlayerKnop:

De knop die verwijst naar de singleplayer game scene.

onMouseButtonPressed(): De functie die aangeroepen wordt op het moment dat er op de knop geklikt wordt. Hier moet je dus in stoppen wat er gebeurd als er op de knop is gedrukt.

onMouseEntered(): De functie om te vertellen wat er moet gebeuren als er met de muis over de knop wordt gehovered.

onMouseExited(): De functie die wordt aangeroepen als de muis weer van de knop af is. Bijvoorbeeld om de onMouseEntered te counteren, zodat de knop weer naar zijn originele kleur gaat.

* spelerRechthoek:

Dit is de abstracte klasse waarmee spelers aangemaakt kunnen worden.

* speler1:

onPressedKeysChange(): De functie die wordt aangeroepen als er een knop is ingedrukt. Daarna kan er worden gekeken welke knoppen ingedrukt zijn en kan aan de hand daarvan het juiste gedaan worden. Bijvoorbeeld W en pijltjestoets omlaag.

onCollision(): Wat er gebeurd met de speler als deze ergens tegenaan komt. Hiermee kan je bijvoorbeeld powerups oppakken en niet verder dan de border gaan met je speler.

invert: Dit zorgt ervoor dat de controls inverted worden.

* speler2:

onPressedKeysChange(): De functie die wordt aangeroepen als er een knop is ingedrukt. Daarna kan er worden gekeken welke knoppen ingedrukt zijn en kan aan de hand daarvan het juiste gedaan worden. Bijvoorbeeld W en pijltjestoets omlaag.

onCollision(): Wat er gebeurd met de speler als deze ergens tegenaan komt. Hiermee kan je bijvoorbeeld powerups oppakken en niet verder dan de border gaan met je speler.

invert: Dit zorgt ervoor dat de controls inverted worden.

* scoreTekst:

Dit is de klasse voor het/de scorebord(en).

setPuntenText(): Dit is de functie die het aantal punten aanpast, zodat dit gedisplayed wordt.

* middenlijn:

Dit is de klasse om de middenlijn te maken.

middenlijnExpand(): de functie die ervoor zorgt dat de middenlijn steeds breder wordt.

* border:

Dit is de klasse om de borders te maken

* singlePlayerMuur:

Dit is de klasse om de muur van singleplayer te maken.

* eindScherm: Dit is de klasse voor het eindscherm.

eindScherm (): Dit is de constructor van eindscherm waar ping aan wordt meegegeven.

setupScene(): De achtergrond en achtergrondmuziek aanpassen van deze scene.

setupEntities(): De entities die in deze scene voorkomen aanmaken en toevoegen aan het scherm.

* bal:

Dit is de klasse waar de bal(len) mee aangemaakt worden.

onCollision(): Deze functie laat de bal weten dat de bal is geraakt door een border, muur of een speler.

getAfwijking(): Berekent een afwijking met maximaal de meegegeven waarde.

getStartRichting(): Dit berekent automatisch een startrichting voor de bal.

notifyBoundaryTouching(): Deze functie laat weten of er een punt is gescoord. onPressedKeysChange(): Met deze functie resetten wij de bal als deze vast komt te zitten of een ander onbekend probleem ondervindt.

checkGewonnen(): Deze functie checkt of de spelers voldoen aan de win conditie van 10 punten.

* powerUpSpawner:

Dit is de klasse die alle powerups en de extra ballen spawnt.

spawnEntities(): Dit is de functie waarin alle nieuwe power-ups/ ballen worden gegenereerd.

spawnNewBal(): Dit is de functie die laat weten of er een nieuwe bal gespawned moet worden of niet.

* powerUps:

Dit is de klasse voor de powerups.

onCollision(): Hier moet je zeggen dat de doPowerUp() uitgevoerd moet worden.

getRandom(): Dit returnt een random getal tussen 0 en de meegegeven waarde

doPowerUp(): Dit is een abstracte klasse.

setMotionPowerUp(): Dit is een abstracte klasse.

* snelheidAanpassen:

Dit is de klasse voor de powerup om de snelheid van de bal aan te passen.

doPowerUp(): Voert de code uit die bij de bijbehorende power-up hoort

setMotionPowerUp(): Dit zorgt ervoor dat de power-up ook echt een kant op beweegt i.p.v. stil staan.

* balToevoegen:

Dit is de klasse voor de powerup die een extra bal toevoegt.

doPowerUp(): Voert de code uit die bij de bijbehorende power-up hoort

setMotionPowerUp(): Dit zorgt ervoor dat de power-up ook echt een kant op beweegt i.p.v. stil staan.

* inverter:

Dit is de klasse voor de powerup die de controls van je tegenstander omdraait.

doPowerUp(): Voert de code uit die bij de bijbehorende power-up hoort

setMotionPowerUp(): Dit zorgt ervoor dat de power-up ook echt een kant op beweegt i.p.v. stil staan.

* grootteSpeler:

Dit is de klasse voor de powerup waarmee de grootte van de spelers kan worden aangepast.

doPowerUp(): Voert de code uit die bij de bijbehorende power-up hoort

setMotionPowerUp(): Dit zorgt ervoor dat de power-up ook echt een kant op beweegt i.p.v. stil staan.

