

# Programmeerproject 1: Verslag 2<sup>e</sup> Fase “Space Invaders”

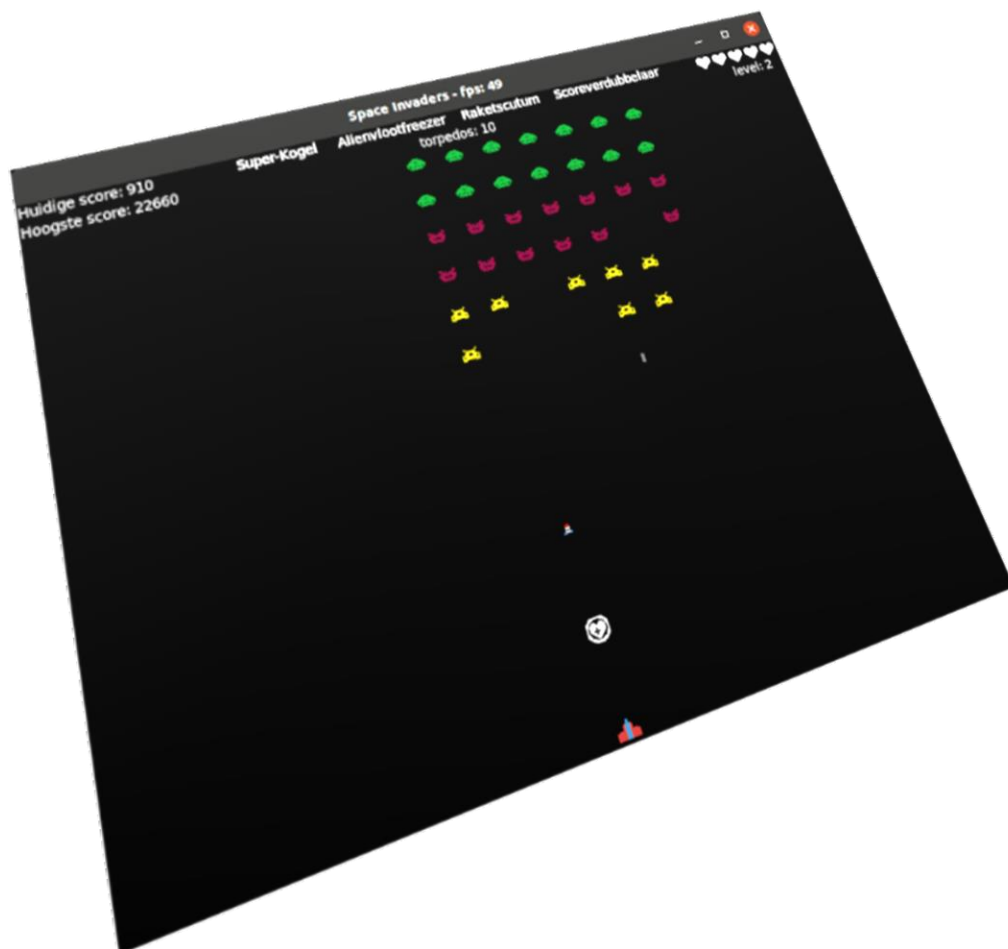
Abdullah Sabaa Allil

Rolnummer: 0575278

[abdullah.sabaa.allil@vub.be](mailto:abdullah.sabaa.allil@vub.be)

Academiejaar 2020-2021

Vrije Universiteit Brussel



## Inhoudsopgave

1. Introductie .....	2
2. Spel opstarten .....	2
3. ADT's .....	3
3.1. Positie ADT .....	3
3.2. Teken ADT .....	3
3.3. Kogel ADT .....	4
3.4. Botsingstoestand ADT .....	5
3.5. Levens ADT .....	6
3.6. Raket ADT .....	6
3.7. Alien ADT .....	7
3.8. Explosie ADT .....	8
3.9. Alienvloot ADT .....	8
3.10. Scorebord ADT .....	10
3.11. Powerup ADT .....	10
3.12. Parent Actieve Powerup ADT .....	11
3.13. Actieve Super-kogel ADT .....	12
3.14. Actieve Alienvlootfreezer ADT .....	12
3.15. Actieve Raketscutum ADT .....	12
3.16. Actieve Scoreverdubelaar ADT .....	13
3.17. Actieve Raketlevenshersteller ADT .....	13
3.18. Actieve Torpedo ADT .....	13
3.19. Actieve powerups ADT .....	14
3.20. Level ADT .....	14
3.21. Spel ADT .....	16
4. Afhankelijkheidsdiagram .....	17
5. Logboek .....	19

## 1. Introductie

Dit is het verslag van de tweede fase van mijn implementatie van het computerspel Space Invaders in de programmeertaal Scheme en in het bijzonder R5RS. Dit project maakt deel uit van het opleidingsonderdeel “Programmeerproject 1”. In deze tweede fase werd het spel uitgebreid met de resterende functionaliteit. Het spel ondersteunt de volgende functionaliteiten:

- Een spelwereld met een raket die door de speler bestuurd kan worden.
- Een vloot van alienschepen die bewegen over het scherm.
- Kogels die door de speler (de raket) kunnen afgevoerd worden en die de alienschepen kunnen raken.
- Alienschepen die zelf ook kogels kunnen schieten op de raket die de speler bestuurt.
- Het bijhouden van de score, de hoogste score en het aantal levens.
- Een mechanisme om naar andere levels te gaan en om het spel te herstarten wanneer de alienschepen de onderkant bereiken.
- Power-ups die bepaalde effecten, ten voordele van de speler, hebben op het spel.

Voor de implementatie van het spel Space Invaders wordt er gebruik gemaakt van de vierde versie van de grafische bibliotheek die op Canvas te vinden is.

In deze fase worden de volgende ADT's geïmplementeerd of verder uitgebreid; een gedetailleerde beschrijving voor elke ADT kan in Sectie 3 ADTs teruggevonden worden:

Positie ADT, teken ADT, kogel ADT, botsingstoestand ADT, levens ADT, raket ADT, alien ADT, alienvloot ADT, explosie ADT, scorebord ADT, powerup ADT, parent actieve powerup ADT, actieve super-kogel ADT, actieve Alienvlootfreezer ADT, actieve Raketscutum ADT, actieve Scoreverdubbelaar ADT, actieve raketlevenshersteller ADT, actieve Torpedo ADT, actieve powerups ADT, level ADT en spel ADT.

## 2. Spel opstarten

Het spel opstarten gebeurt door het bestand spel.rkt in DrRacket te runnen. Alle ADT's worden dan ingeladen en start! uit het spel ADT wordt opgeroepen.

De raket kan bestuurd worden door op:

- de linker en rechterpijl van het toetsenbord te drukken om de raket te laten bewegen naar links of naar rechts.
- de spatietoets van het toetsenbord te drukken om kogels af te vuren.
- de linker shift-toets te drukken om torpedo's af te vuren als de speler torpedo's heeft.

Wanneer de speler verliest, dan kan het spel herstart worden door op de entertoets te drukken.

Daarnaast worden meerdere cheat-toetsen geïntroduceerd om bepaalde functionaliteiten van het spel snel te kunnen testen:

- De x-toets zorgt ervoor dat het spel naar het volgende level gaat.
- De getallen van 1 tot en met 6 in een numeriek toetsenbord zorgen voor het activeren van een powerup. Indien de speler niet beschikt over een numeriek toetsenbord, dan kunnen de F1 tot en met F6 toetsen gebruikt worden als alternatief:

- 1 of F1 activeert de Super-kogel powerup.
- 2 of F2 activeert de Alienvlootfreezer powerup.
- 3 of F3 activeert de Raketscutum powerup.
- 4 of F4 activeert de Levenshersteller powerup.
- 5 of F5 activeert de Scoreverdubbelaar powerup.
- 6 of F6 activeert de Torpedo powerup.

### 3. ADT's

Deze sectie beschrijft de ADT's van het spel.

#### 3.1. Positie ADT

Het positie ADT is een abstractie voor het bijhouden van een positie in een tweedimensionale wereld. Dit gebeurt door de encapsulatie van twee Scheme numbers, die de x- en de y-waarde voorstellen.

Naam	Signatuur
maak-positie	(number number $\rightarrow$ positie)
get-x	( $\emptyset \rightarrow$ number)
get-y	( $\emptyset \rightarrow$ number)
set-x!	(number $\rightarrow \emptyset$ )
set-y!	(number $\rightarrow \emptyset$ )
beweeg!	(symbol $\rightarrow \emptyset$ )
vergelijk?	(positie $\rightarrow$ boolean)

Tabel 1: operaties van het positie ADT.

Het positie ADT bevat de volgende procedures:

- maak-positie is een aanmaakoperatie van het positie ADT. Deze procedure maakt een nieuw object voor het positie ADT aan door gebruik te maken van de twee getallen die als argumenten aan de procedure doorgegeven worden.
- get-x en get-y zijn procedures die resp. de x- en y-waarde van een positie object kunnen opvragen.
- set-x! en set-y! zijn operaties die resp. de x- en y-waarde van een positie object kunnen aanpassen. Beide procedures verwachten een argument, namelijk een waarde voor resp. de nieuwe x- en y-positie.
- beweeg! is een operatie die de positie verandert aan de hand van richtingen (rechts, links, omhoog, omlaag) die meegegeven worden als symbool bij het oproepen van de procedure.
- vergelijk? is een predikaat waarmee twee posities vergeleken kunnen worden. Deze procedure verwacht 1 argument, namelijk een positie object, en zal dit positie object vergelijken met het positie object waarnaar het bericht gestuurd wordt. Dit predikaat slaagt beide positie objecten dezelfde x- en y-waarden hebben.

#### 3.2. Teken ADT

Het teken ADT zorgt ervoor dat alle elementen op het scherm getekend worden. Het teken ADT zorgt dus voor de tekenlogica.

Naam	Signatuur
maak-teken	(number number $\rightarrow$ teken)
set-toets-functie!	((symbol any $\rightarrow \emptyset$ ) $\rightarrow \emptyset$ )
set-spel-lus-functie!	((number $\rightarrow \emptyset$ ) $\rightarrow \emptyset$ )
teken-spel!	(spel-ADT $\rightarrow \emptyset$ )
teken-gameover!	( $\emptyset \rightarrow \emptyset$ )
verwijder-alle-aliens!	( $\emptyset \rightarrow \emptyset$ )
herteken!	( $\emptyset \rightarrow \emptyset$ )

Tabel 2: operaties van het teken ADT.

Het teken ADT bevat de volgende procedures:

- maak-teken is een aanmaakoperatie die een nieuw tekenobject aanmaakt. Gegeven twee Scheme numbers, namelijk de dimensies van het venster waarin het spel getekend zal worden, zal deze procedure een nieuw tekenobject aanmaken.
- set-toets-functie! is een procedure die een andere procedure als argument verwacht, namelijk een procedure die verantwoordelijk is voor het afhandelen van de toetsenbordinput. set-toets-functie! zal set-key-callback! uit het Window ADT van de grafische bibliotheek oproepen en het meegegeven argument aan set-key-callback! meegeven. Meer informatie over set-key-callback! kan in de documentatie van de grafische bibliotheek teruggevonden worden.
- set-spel-lus-functie! is een procedure die een andere procedure verwacht als argument, namelijk een procedure die verantwoordelijk is voor het verzorgen van de spellogica. set-spel-lus-functie! zal set-update-callback! uit het Window ADT van de grafische bibliotheek oproepen en het argumenten daaraan meegeven. Meer informatie over de werking van set-update-callback! kan in de documentatie van de grafische bibliotheek teruggevonden worden.
- teken-spel! zorgt voor het tekenen van alle spelelementen op het scherm. Deze procedure neemt een spel ADT als argument aan, en zal vervolgens aan het spelobject het levelobject opvragen, dat op zijn beurt alle spelobjecten gaat doorgeven aan de procedures die verantwoordelijk zijn voor het tekenen en verwijderen voor het specifieke spelobject.
- teken-gameover! is een procedure waarmee de boodschap "Game Over" wordt afgegeven. Druk op Enter om het spel te herstarten." op het scherm getekend wordt wanneer de speler het spel verliest.
- verwijder-alle-aliens! is een procedure waarmee alle aliens uit het scherm verwijderd worden. Deze procedure wordt gebruikt in het geval dat de cheatknop x ingedrukt is, zodat de aliens uit het vorige level niet op het scherm getekend blijven.
- herteken! is een procedure waarmee alles in het teken ADT terug geïnitieerd wordt, zoals het eindscherm leegmaken en de verschillende lagen opnieuw initialiseren. Deze procedure wordt gebruikt wanneer het spel herstart wordt.

### 3.3. Kogel ADT

Het kogel ADT stelt een kogel voor die afgevuurd kan worden door een object, zoals een raket en een alien, en die andere objecten kan raken.

Het kogel ADT bevat de volgende procedures:

Naam	Signatuur
maak-kogel	(positie symbol $\rightarrow$ kogel)
positie	( $\emptyset \rightarrow$ positie)
beweeg!	(number $\rightarrow \emptyset$ )
richting	( $\emptyset \rightarrow$ symbol)
verwijder?	( $\emptyset \rightarrow$ boolean)
verwijder!	( $\emptyset \rightarrow \emptyset$ )

Tabel 3: operaties van het kogel ADT.

- maak-kogel is de aanmaakprocedure voor het kogel ADT. Gegeven een positie en een symbool zal er een kogel object aangemaakt worden. De positie is de positie van het object waarvan de kogel vertrokken is. Het symbool is de richting van de kogel. Als de richting 'omhoog is, dan zal de kogel omhoog bewegen, en als het 'omlaag is, dan zal de kogel omlaag bewegen.
- positie is een procedure waarmee de positie van het kogelobject opgevraagd kan worden.
- beweeg! zorgt voor de beweeglogica van de kogel. Deze procedure verwacht een getal, namelijk de delta-tijd, en zal de kogel laten bewegen afhankelijk van zijn snelheid. beweeg! zal ook nagaan of de kogel buiten het scherm zit.
- richting is een procedure waarmee de richting van de kogel opgevraagd kan worden.
- verwijder? is een predikaat dat aangeeft of de kogel verwijderd moet worden.
- verwijder! is een procedure waarmee de predikaat verwijder? op #t gezet kan worden.

### 3.4. Botsingstoestand ADT

Het botsingstoestand ADT zorgt voor het bijhouden van de botsingstoestand van een bepaald object, zoals de raket. Het zorgt ook voor het nagaan of het object met een ander object geraakt heeft, en voor het de nodige aanpassingen te brengen indien het object gebotst heeft.

Naam	Signatuur
maak-botsingstoestand	(positie $\rightarrow$ botsingstoestand-ADT)
is-geraakt?	(positie $\rightarrow$ boolean)
pas-aan-indien-geraakt!	(( $\emptyset \rightarrow$ any) $\rightarrow \emptyset$ )

Tabel 4: operaties van het botsingstoestand ADT

- maak-botsingstoestand is de aanmaakprocedure voor het botsingstoestand ADT. Gegeven een positie van een object zal deze procedure een botsingstoestandsobject aanmaken.
- is-geraakt? is een procedure waarmee nagegaan kan worden of de positie een andere positie geraakt heeft.
- pas-aan-indien-geraakt! is een procedure waarmee het object dat het botsingstoestandsobject aangemaakt heeft, aangepast kan worden. Deze procedure verwacht 1 argument, namelijk een lambda. Doorgaans is dat een actie die uitgevoerd moet worden wanneer het object geraakt is, zoals het aantal levens van de raket verminderen. Deze procedure zal daarbij ook de toestand van de botsing op false zetten.

### 3.5. Levens ADT

Het levens ADT is een abstractie die verantwoordelijk is voor het bijhouden van het aantal levens van een bepaald object. Het levens ADT bevat de volgende operaties:

Naam	Signatuur
maak-levens-ADT	(number $\rightarrow$ levens-ADT)
get-levens	( $\emptyset \rightarrow$ number)
set-levens!	(number $\rightarrow \emptyset$ )
verminder-levens!	( $\emptyset \rightarrow \emptyset$ )
nul-levens?	( $\emptyset \rightarrow$ boolean)

Tabel 5: operaties van het levens ADT.

- maak-levens-ADT is een aanmaakprocedure voor het levens ADT. Gegeven een getal dat het aantal initiële levens voorstelt, wordt een nieuw levens ADT gemaakt.
- get-levens zal het huidig aantal levens opvragen.
- set-levens! zal, gegeven een getal, het aantal levens aanpassen naar het meegegeven getal.
- verminder-levens! zal het huidig aantal levens met 1 verminderen.
- nul-levens? is een predikaat dat aangeeft of het object nog levens heeft of niet.

### 3.6. Raket ADT

Het raket ADT is een abstractie voor de raket die door de speler bestuurd wordt. Het raket ADT houdt daarbij de positie van de raket bij, en zorgt voor het afvuren van kogels.

Naam	Signatuur
maak-raket	(positie $\rightarrow$ raket)
positie	( $\emptyset \rightarrow$ positie)
beweeg-horizontaal!	(symbol $\rightarrow \emptyset$ )
schiet!	( $\emptyset \rightarrow$ kogel)
get-levensobject	( $\emptyset \rightarrow$ leven-ADT)
geraakt?	(positie $\rightarrow \emptyset$ )
pas-aan-indien-geraakt!	( $\emptyset \rightarrow \emptyset$ )

Tabel 6: operaties van het raket ADT.

- maak-raket is een aanmaakoperatie van het raket ADT. Gegeven een positie zal deze procedure een object aanmaken voor het raket ADT.
- positie is een operatie waarmee de positie van de raket opgevraagd kan worden.
- beweeg-horizontaal! is verantwoordelijk voor de raket horizontaal te laten bewegen naar links of naar rechts. Gegeven een symbool, 'right' of 'left', zal deze procedure de raket laten bewegen naar rechts of naar links. De raket mag naar rechts blijven bewegen zolang de x-waarde van de raketpositie de rechterzijkant van het venster niet geraakt heeft, en mag naar links blijven bewegen zolang de x-waarde van de positie de linkerzijkant van het venster niet geraakt heeft.
- schiet! is een procedure die zorgt voor het afvuren van een kogel. Deze procedure zal een nieuw kogelobject aanmaken en teruggeven.
- get-levensobject is een procedure waarmee het levensobject van de raket opgevraagd kan worden.

- geraakt? is een predikaat dat nagaat of de raket een positie geraakt heeft of niet.
- pas-aan-indien-geraakt! is een procedure die het aantal levens van de raket zal verminderen indien de raket geraakt wordt door een kogel. Zowel geraakt? als pas-aan-indien-geraakt! maken gebruik van het botsingstoestand ADT.

### 3.7. Alien ADT

Het alien ADT stelt een alienschip voor dat een bepaalde kleur heeft en een bepaald aantal levens. Dit ADT zorgt daarnaast ook voor het schieten van kogels en een powerup achterlaten wanneer het alienschip ontploft.

Naam	Signatuur
maak-alien	(positie symbol boolean boolean $\rightarrow$ alien)
positie	( $\emptyset \rightarrow$ positie)
beweeg!	(symbol $\rightarrow \emptyset$ )
eenheid-omlaag!	( $\emptyset \rightarrow \emptyset$ )
get-type	( $\emptyset \rightarrow$ symbol)
nul-levens?	( $\emptyset \rightarrow$ boolean)
zijkant-geraakt?	( $\emptyset \rightarrow$ boolean)
bodem-geraakt?	( $\emptyset \rightarrow$ boolean)
is-geraakt?	(positie $\rightarrow$ boolean)
pas-aan-indien-geraakt!!	( $\emptyset \rightarrow \emptyset$ )
schiet!	( $\emptyset \rightarrow$ kogel)
gooi-powerup?	( $\emptyset \rightarrow$ boolean)

Tabel 7: operaties van het alien ADT.

- maak-alien is de aanmaakprocedure die een alienschip ADT aanmaakt. Gegeven een positie, een symbool en twee booleans, zal deze procedure een nieuw object voor het alien ADT aanmaken. Wanneer het symbool gelijk is aan geel, wordt een geel alienschip aangemaakt dat één leven telt. Als het symbool gelijk is aan paars, wordt een paars alienschip aangemaakt dat 2 levens heeft. Wanneer het symbool gelijk is aan groen wordt een alienschip aangemaakt dat 3 levens heeft. De eerste boolean geeft aan of het alienschip een schietende alien is. De tweede boolean geeft aan of het alienschip wel of geen powerup moet achterlaten wanneer het ontploft.
- positie is een operatie waarmee de positie van het alienschip opgevraagd kan worden.
- beweeg! is een procedure die de alien zal laten bewegen volgens een bepaalde richting. Deze richting wordt meegegeven aan de procedure als symbool.
- eenheid-omlaag! zal het alienschip met 1 eenheid naar beneden laten bewegen.
- get-type zal het type (de kleur) van het alienschip opvragen.
- nul-levens? is een predikaatprocedure die aangeeft of het alienschip nog levens heeft of niet.
- zijkant-geraakt? is een predikaatprocedure die aangeeft of het alienschip een zijkant van het venster geraakt heeft of niet. Dit gebeurt door na te gaan of de x-waarde van de positie van het alienschip gelijk is aan de rechterzijkant of de linkerzijkant van het venster.



- bodem-geraakt? is een predikaat dat nagaat of het alienschip de bodem van het venster geraakt heeft. Dit gebeurt door na te gaan of de y-waarde van de positie van het alienschip gelijk is aan de bodem van het venster.
- is-geraakt? is een predikaat dat nagaat of de alien een object met een positie geraakt heeft of niet.
- pas-aan-indien-geraakt! is een procedure die het aantal levens van de alien zal verminderen indien de alien geraakt wordt door een kogel. Zowel geraakt? als pas-aan-indien-geraakt! maken gebruik van het botsingstoestand ADT.
- schiet! zal, indien het alienschip een schietend alienschip is, een kogel afvuren die de raket kan raken.
- gooi-powerup? is een predikaat dat nagaat of het alienschip een powerup moet achterlaten wanneer het ontploft of niet.

### 3.8. Explosie ADT

Het explosie ADT zorgt voor het tonen van een explosie wanneer een alien ontploft.

Het explosie ADT bevat de volgende procedures:

Naam	Signatuur
maak-explosie	(positie $\rightarrow$ explosie-ADT)
positie	( $\emptyset \rightarrow$ positie)
verwijder?	( $\emptyset \rightarrow$ boolean)
toon-explosie!	(number $\rightarrow \emptyset$ )

Tabel 8: operaties van het explosie ADT.

- maak-explosie is een aanmaakoperatie van het explosie ADT. Gegeven een positie, de positie van het alienschip dat ontploft is, zal deze procedure een object aanmaken voor het explosie ADT.
- positie is een procedure waarmee de positie van het explosie ADT opgevraagd kan worden.
- verwijder? is een predikaat dat nagaat wanneer de explosie uit het scherm verwijderd moet worden.
- toon-explosie! is de procedure die ervoor zorgt dat de explosie voor een bepaalde duur op het scherm getekend wordt. Deze procedure verwacht 1 argument, namelijk de delta-tijd, en zal dus aan de hand daarvan weten wanneer de explosie gedaan moet zijn.

### 3.9. Alienvloot ADT

Het alienvloot ADT is een abstractie die de alienvloot voorstelt. Het is een abstractie bovenop het alienschip ADT en het positie ADT.

Naam	Signatuur
maak-alienvloot	(number $\rightarrow$ alienvloot)
voor-alle-aliens	((alien $\rightarrow$ any) $\rightarrow \emptyset$ )
map-over-aliens	((alien $\rightarrow$ any) $\rightarrow$ vector)

beweeg!	$(\emptyset \rightarrow \emptyset)$
beweeg-eenheid-omlaag!	$(\emptyset \rightarrow \emptyset)$
set-tegengestelde-richting!	$(\emptyset \rightarrow \emptyset)$
schiet!	$(\text{number} \rightarrow \text{kogel})$
bodem-geraakt?	$(\emptyset \rightarrow \text{boolean})$
aantal-aliens	$(\emptyset \rightarrow \text{number})$
kleur-laatste-geraakte-alien	$(\emptyset \rightarrow \text{symbol})$
kleur-aliens-geraakt-door-torpedo	$(\emptyset \rightarrow \text{pair})$
level-up!	$(\emptyset \rightarrow \emptyset)$
gegooide-powerups	$(\emptyset \rightarrow \text{pair})$
geraakt?	$(\text{Positie} \rightarrow \text{boolean})$
door-torpedo-geraakt?	$(\emptyset \rightarrow \emptyset)$
pas-aan-indien-geraakt!	$(\emptyset \rightarrow \emptyset)$

Tabel 9: operaties van het alienvloot ADT.

- maak-alienvloot is de aanmaakprocedure die een alienvloot ADT aanmaakt. Gegeven een Scheme number dat het level aangeeft, zal deze procedure een object maken voor de alienvloot. Dit nummer kan verschillende levels voorstellen. In het eerste level wordt een vloot van vier rijen en vijf kolommen aangemaakt.
- voor-alle-aliens is een procedure die een andere procedure die meegegeven wordt als argument, gaat toepassen op alle aliens in de vloot.
- map-over-aliens is een procedure die een andere procedure die meegegeven wordt als argument, gaat toepassen op alle aliens in de vloot en vervolgens een vector van vectoren (een matrix) gaat teruggeven met de resultaten van de procedure-oproep van de meegegeven procedure op de alienschepen.
- beweeg! zal alle aliens laten bewegen volgens de richting die in de variabele richting opgeslagen zit.
- beweeg-eenheid-omlaag! zal alle aliens met 1 eenheid naar beneden laten bewegen.
- set-tegengestelde-richting! zal de richting van de vloot naar de tegengestelde richting zetten. Als de richting bijvoorbeeld links was, dan wordt hij rechts.
- schiet! is een procedure die een willekeurig alienschip uit de alienvloot een kogel laat schieten. Gegeven een getal, dat de delta-tijd voorstelt, zal deze procedure na een bepaald tijdsinterval een willekeurig alienschip selecteren en het laten schieten, indien het een schietend alienschip is.
- bodem-geraakt? is een predikaat dat nagaat of een alienschip uit de vloot de bodem van het venster geraakt heeft.
- aantal-aliens is een procedure waarmee het aantal aanwezige aliens in de vloot opgevraagd kan worden.
- kleur-laatste-geraakte-alien is een procedure waarmee de kleur van het laatste geraakt alienschip opgevraagd kan worden. Dit is handig wanneer de score berekend moet worden.
- kleur-aliens-geraakt-door-torpedo is een procedure waarmee de lijst van de kleuren die door een torpedo geraakt worden opgevraagd kan worden zodat de score berekend kan worden.

- level-up! is een procedure die zorgt voor het overgaan naar het volgende level van de vloot. Dit zorgt voor een nieuwe vloot aanmaken voor het nieuwe level die groter is en die meer schietende alienschepen bevat.
- gegooide-powerups is een procedure waarmee de lijst van de gegooide powerups opgevraagd kan worden.
- geraakt? is een procedure die, gegeven een positie, nagaat of een alien uit de vloot de positie geraakt heeft.
- door-torpedo-geraakt? is een procedure die, gegeven de positie van een torpedo, nagaat of een alien uit de vloot een torpedo geraakt heeft. Deze procedure zal vervolgens naar de omliggende alienschepen van het geraakte alienschip de boodschap sturen om zich aan te passen en hun levens te verminderen.
- pas-aan-indien-geraakt! is een procedure waarmee de alienvloot aangepast wordt indien hij geraakt is. Als een alienschip geraakt wordt en geen levens meer heeft, dan wordt het uit de vloot verwijderd, en wordt een powerup gegooit indien dat alienschip een powerup moest gooien.

### 3.10. Scorebord ADT

Het scorebord ADT zorgt voor het bijhouden van de huidige score en voor het aanpassen en het teruggeven van de hoogst behaalde score.

Naam	Signatuur
maak-scorebord-adt	$(\emptyset \rightarrow \text{scorebord-adt})$
huidige-score	$(\emptyset \rightarrow \text{number})$
pas-huidige-score-aan!	$(\text{symbol} \rightarrow \emptyset)$
hoogste-score	$(\emptyset \rightarrow \text{number})$
verdubbel-aan!	$(\emptyset \rightarrow \emptyset)$
verdubbel-uit!	$(\emptyset \rightarrow \emptyset)$

Tabel 10: operaties van het scorebord ADT.

- maak-scorebord-adt zal een nieuw scorebord object aanmaken.
- huidige-score is een procedure waarmee de huidige score opgevraagd kan worden.
- pas-huidige-score-aan! zal de huidige score verhogen volgens het argument. Het argument is een symbool dat de kleur van een alienschip voorstelt. Deze procedure zal bovendien ook nagaan of er een nieuwe hoogste score is, en de hoogste score vervolgens aanpassen.
- hoogste-score zal de hoogst behaalde score opvragen en teruggeven.
- verdubbel-aan! is een procedure die zorgt dat de score verdubbeld wordt. Dit wordt later gebruikt in een powerup.
- verdubbel-uit! is een procedure die zorgt voor het uitzetten van de scoreverdubbeling.

### 3.11. Powerup ADT

Het powerup ADT is een abstractie voor de powerup bolletjes die achtergelaten worden door sommige alienschepen wanneer ze ontploffen.

Naam	Signatuur
maak-power-up	(positie $\rightarrow$ power-up)
get-type	( $\emptyset \rightarrow$ number)
positie	( $\emptyset \rightarrow$ positie)
richting	( $\emptyset \rightarrow$ symbol)
beweeg!	(number $\rightarrow \emptyset$ )
verwijder?	( $\emptyset \rightarrow$ boolean)
verwijder!	( $\emptyset \rightarrow$ boolean)

Tabel 11: operaties van het powerup ADT.

- maak-power-up is de aanmaakprocedure voor het powerup ADT. Gegeven een positie, de positie van het ontplofte alienschip, wordt een nieuw powerup object aangemaakt.
- get-type is een procedure waarmee het type powerup opgevraagd kan worden.
- positie is een procedure waarmee de positie van de powerup opgevraagd kan worden.
- richting is een procedure waarmee de richting, doorgaans is dat omlaag, van de powerup opgevraagd kan worden.
- beweeg! zorgt voor de beweeglogica van de powerup. Deze procedure verwacht een getal, namelijk de delta-tijd, en zal de powerup laten bewegen afhankelijk van de ingestelde snelheid.
- verwijder? is een predikaat dat aangeeft of de powerup verwijderd moet worden.
- verwijder! is een procedure waarmee de predikaat verwijder? op true gezet kan worden.

### 3.12. Parent Actieve Powerup ADT

Het parent actieve powerup ADT is een abstractie die verantwoordelijk is voor het bijhouden en aanpassen voor de toestand en het type actieve powerup. Dit ADT wordt dus gebruikt in de andere actieve powerup ADT's aan de hand van de programmeertechniek **compositie**.

Naam	Signatuur
maak-parent-actieve-powerup	(number $\rightarrow$ parent-actieve-powerup-ADT)
verwijder?	( $\emptyset \rightarrow$ boolean)
get-type	( $\emptyset \rightarrow$ number)
activeer!	( $\emptyset \rightarrow \emptyset$ )
deactiveer!	( $\emptyset \rightarrow \emptyset$ )

Tabel 12: operaties van het parent actieve powerup ADT.

- maak-parent-actieve-powerup is een aanmaakprocedure voor het parent actieve powerup ADT. Gegeven een getal, dat het type actieve powerup voorstelt, wordt een nieuw parent actieve powerup ADT aangemaakt.
- verwijder? is een predikaat dat aangeeft wanneer de actieve powerup gedaan is en verwijderd moet worden.
- get-type is een procedure waarmee het type actieve powerup opgevraagd kan worden.
- activeer! is een procedure die zorgt voor het activeren van de actieve powerup.
- deactiveer! is een procedure die zorgt voor het deactiveren van de actieve powerup.

### 3.13. Actieve Super-kogel ADT

Het actieve super-kogel ADT zorgt voor het bijhouden van de toestand en de actie van de super-kogel powerup. De super-kogel powerup zal alle kogels voor een bepaalde periode de aliens die ze raken direct laten ontploffen, en de kogels verdwijnen alleen wanneer ze het venster verlaten. Dit ADT begrijpt dezelfde operaties als het parent actieve powerup ADT buiten de aanmaakoperatie aangezien dat het actieve super-kogel ADT intern een parent actieve powerup ADT aanmaakt. Daarnaast begrijpt dit ADT de volgende operaties:

Naam	Signatuur
maak-actieve-super-kogel	$(\emptyset \rightarrow \text{actieve super-kogel ADT})$
voer-actie-uit!	$(\text{number} \rightarrow \emptyset)$

Tabel 13: operaties van het actieve super-kogel ADT.

- maak-actieve-super-kogel zal een nieuw actieve super-kogel object aanmaken.
- voer-actie-uit! zal de actie van de super-kogel uitvoeren en de kogels in de super-kogel toestand voor een bepaalde periode. Aangezien dat dit een tijdgebaseerde powerup is, wordt delta-tijd meegegeven bij de procedure-oproep zodat het weet wanneer de tijd van de powerup gedaan is.

### 3.14. Actieve Alienvlootfreezer ADT

Het actieve Alienvlootfreezer ADT zorgt voor het bijhouden van de toestand en de actie van de Alienvlootfreezer powerup. De Alienvlootfreezer powerup zal de alienvloot voor een bepaalde periode bevroren, zodat die niet meer beweegt. Dit ADT begrijpt dezelfde operaties als het parent actieve powerup ADT buiten de aanmaakoperatie aangezien dat het actieve Alienvlootfreezer ADT intern een parent actieve powerup ADT aanmaakt. Daarnaast begrijpt dit ADT de volgende operaties:

Naam	Signatuur
maak-actieve-alienvlootfreezer	$(\emptyset \rightarrow \text{actieve Alienvlootfreezer ADT})$
voer-actie-uit!	$(\text{number} \rightarrow \emptyset)$

Tabel 14: operaties van het actieve Alienvlootfreezer ADT.

- maak-actieve-alienvlootfreezer zal een nieuw actieve Alienvlootfreezer object aanmaken.
- voer-actie-uit! zal de actie van de Alienvlootfreezer uitvoeren en de alienvloot voor een bepaalde periode bevroren. Aangezien dat dit een tijdgebaseerde powerup is, wordt delta-tijd meegegeven bij de procedure-oproep zodat die weet wanneer de tijd van de powerup gedaan is.

### 3.15. Actieve Raketscutum ADT

Het actieve Raketscutum ADT zorgt voor het bijhouden van de toestand en de actie van de Raketscutum powerup. Deze powerup zorgt ervoor dat de raket geen levens verliest wanneer hij geraakt wordt door een kogel die geschoten is door een alienschip. Dit ADT begrijpt dezelfde operaties als het parent actieve powerup ADT buiten de aanmaakoperatie aangezien dat het actieve Raketscutum ADT intern een parent actieve powerup ADT aanmaakt. Daarnaast begrijpt dit ADT de volgende operaties:

Naam	Signatuur
maak-actieve-raketscutum	$(\emptyset \rightarrow \text{actieve Raketscutum ADT})$
voer-actie-uit!	$(\text{number} \rightarrow \emptyset)$

Tabel 15: operaties van het actieve Raketscutum ADT.

- maak-actieve-raketscutum zal een nieuw actieve Raketscutum object aanmaken.
- voer-actie-uit! zal de actie van de Raketscutum uitvoeren en de raket voor een bepaalde periode beschermen. Aangezien dat dit een tijdgebaseerde powerup is, wordt delta-tijd aan de procedure meegegeven zodat die weet wanneer de Raketscutum gedaan is.

### 3.16. Actieve Scoreverdubbelaar ADT

Het actieve Scoreverdubbelaar ADT zorgt voor het bijhouden van de toestand en de actie van de Scoreverdubbelaar powerup. Deze powerup zal voor een bepaalde periode de score die de speler verwerft verdubbelen. Dit ADT begrijpt dezelfde operaties als het parent actieve powerup ADT buiten de aanmaakoperatie aangezien dat het actieve Scoreverdubbelaar ADT intern een parent actieve powerup ADT aanmaakt. Daarnaast begrijpt dit ADT de volgende operaties:

Naam	Signatuur
maak-actieve-scoreverdubbelaar	(scorebord → actieve Scoreverdubbelaar ADT)
voer-actie-uit!	(number → $\emptyset$ )

Tabel 16: operaties van het actieve Scoreverdubbelaar ADT.

- maak-actieve-scoreverdubbelaar is de aanmaakprocedure voor het actieve Scoreverdubbelaar ADT. Gegeven een scorebord object, wordt een actieve Scoreverdubbelaarobject aangemaakt.
- voer-actie-uit! zal de actie van de Scoreverdubbelaar powerup uitvoeren en de verworven score verdubbelen voor een bepaalde periode. Aangezien dat dit een tijdgebaseerde powerup is, wordt delta-tijd aan de procedure meegegeven om te weten wanneer de powerup gedeactiveerd moet worden.

### 3.17. Actieve Raketlevenshersteller ADT

Het actieve raketlevenshersteller ADT zorgt voor het bijhouden van de toestand en de actie van de Raketlevenshersteller powerup. Deze powerup zal de levens van de raket herstellen en de raket terug 5 levens geven. Dit ADT begrijpt dezelfde operaties als het parent actieve powerup ADT buiten de aanmaakoperatie aangezien dat het actieve Raketlevenshersteller ADT intern een parent actieve powerup ADT aanmaakt. Daarnaast begrijpt dit ADT de volgende operaties:

Naam	Signatuur
maak-actieve-raket-levenshersteller	(levens ADT → actieve Raketlevenshersteller ADT)
voer-actie-uit!	( $\emptyset \rightarrow \emptyset$ )

Tabel 17: operaties van het actieve Raketlevenshersteller ADT.

- maak-actieve-raket-levenshersteller is de aanmaakprocedure voor het actieve Raketlevenshersteller ADT. Gegeven een levensobject, dat het aantal levens van de raket voorstelt, wordt een nieuw object voor het actieve Raketlevenshersteller aangemaakt.
- voer-actie-uit! zal de actie van de Raketlevenshersteller powerup uitvoeren en de levens van de raket terug herstellen.

### 3.18. Actieve Torpedo ADT

Het actieve Torpedo ADT zorgt voor het bijhouden van de toestand en de actie van de Torpedo powerup. Deze powerup zal ervoor zorgen dat het aantal torpedos dat de speler heeft met 1

vermeerderd wordt. De torpedo zelf is een speciale soort kogel die de speler kan schieten door op shift te drukken indien de speler torpedos heeft. De torpedo zal dan het alienschip en de omliggende aliens beschadigen. Dit ADT begrijpt dezelfde operaties als het parent actieve powerup ADT buiten de aanmaakoperatie aangezien dat het actieve Torpedo ADT intern een parent actieve powerup ADT aanmaakt. Daarnaast begrijpt dit ADT de volgende operaties:

Naam	Signatuur
maak-actieve-torpedo	$((\emptyset \rightarrow \emptyset) \rightarrow \text{actieve Torpedo ADT})$
voer-actie-uit!	$(\emptyset \rightarrow \emptyset)$

Tabel 18: operaties van het actieve Torpedo ADT.

- maak-actieve-torpedo is de aanmaakprocedure voor het actieve Torpedo ADT. Gegeven een procedure, die verantwoordelijk is voor het aantal beschikbare torpedos te verhogen, wordt een nieuw object gemaakt voor het actieve Torpedo ADT.
- voer-actie-uit! zal de actie van de torpedo powerup uitvoeren en het aantal beschikbare torpedo's verhogen. Dit gebeurt door de procedure die meegegeven werd bij het aanmaken van het actieve Torpedo ADT op te roepen.

### 3.19. Actieve powerups ADT

Het actieve powerups ADT is een ADT dat zorgt voor het bijhouden, opzoeken en teruggeven van de verschillende powerups. Dit ADT houdt ook bij welke powerups niet tijdgebaseerd zijn.

Het actieve powerups ADT bevat de volgende procedures:

Naam	Signatuur
maak-actieve-powerups	$(\text{Level ADT} \rightarrow \text{actieve powerups ADT})$
geef-actieve-powerup-terug	$(\text{number} \rightarrow (\emptyset \rightarrow \text{actieve super-kogel ADT V actieve Alienvlootfreezer ADT V actieve Scoreverdubbelaar ADT V actieve Raketlevenshersteller ADT V actieve Torpedo ADT}))$
niet-tijdgebaseerde-powerup?	$(\text{number} \rightarrow \text{boolean})$

Tabel 19: operaties van het actieve powerups ADT.

- maak-actieve-powerups is de aanmaakprocedure van het actieve powerups ADT. Gegeven een level ADT, dat nodig is om aan de aanmaakprocedures van de verschillende powerups de nodige argumenten te geven, wordt een nieuw object gemaakt voor het actieve powerups ADT.
- geef-actieve-powerup-terug zal, gegeven een getal dat het type powerup voorstelt, een procedureobject dat geen argumenten aanneemt teruggeven. Wanneer dit procedureobject opgeroepen wordt, wordt een object gemaakt voor het gezochte actieve powerup.
- niet-tijdgebaseerde-powerup? is een procedure die, gegeven een getal dat het type actieve powerup voorstelt, nagaat of de powerup een niet-tijdgebaseerde powerup is of niet.

### 3.20. Level ADT

In het level ADT wordt de spellogica geïmplementeerd. In dit ADT kunnen de verschillende spelelementen met elkaar interageren.

Naam	Signatuur
maak-adt-level	(number number number $\rightarrow$ level ADT)
update!	(number $\rightarrow \emptyset$ )
toets!	(symbol $\rightarrow \emptyset$ )
raket	( $\emptyset \rightarrow$ raket)
raket-levens	( $\emptyset \rightarrow$ leven-ADT)
alienvloot	( $\emptyset \rightarrow$ alienvloot)
scorebord	( $\emptyset \rightarrow$ scorebord)
kogels	( $\emptyset \rightarrow$ pair)
torpedos	( $\emptyset \rightarrow$ pair)
power-ups	( $\emptyset \rightarrow$ pair)
actieve-powerups	( $\emptyset \rightarrow$ pair)
level-gewonnen?	( $\emptyset \rightarrow$ boolean)
spel-verloren?	( $\emptyset \rightarrow$ boolean)
level-up!	( $\emptyset \rightarrow \emptyset$ )
huidig-level	( $\emptyset \rightarrow$ number)
aantal-beschikbare-torpedos	( $\emptyset \rightarrow$ number)
incr-aantal-beschikbare-torpedos!	( $\emptyset \rightarrow \emptyset$ )
niet-tijdgebaseerde-powerup?	(number $\rightarrow$ boolean)

Tabel 20: operaties van het level ADT.

- maak-adt-level procedure zal, gegeven drie Scheme numbers, een nieuw object aanmaken voor het level ADT. Het eerste getal stelt het huidige level voor. De speler kan een hoger level bereiken wanneer hij of zij alle aliens uit het vorige level succesvol heeft vernietigd. Hoe hoger het level, hoe meer aliens er zullen zijn in de alienvloot, en hoe sneller de aliens zullen bewegen, en hoe meer schietende aliens aanwezig zijn in de vloot. De laatste twee numbers stellen resp. het aantal cellen in de breedte en in de hoogte voor.
- update! is een procedure die zorgt voor het in gang zetten van de spellogica, zoals alle elementen op het scherm laten bewegen. Deze procedure is een heel cruciaal deel van de spellus. Het verwacht 1 argument, namelijk de delta-tijd.
- toets! is de procedure die zorgt voor het afhandelen van toetseninvoer en bepaalde acties te doen bij bepaalde toetseninvoer (zoals raket besturen wanneer linker of rechterpijl ingedrukt is).
- raket, raket-levens, alienvloot en scorebord zijn procedures waarmee resp. het raketobject, het levensobject van de raket, het alienvloot-object, en het scorebord-object opgevraagd kunnen worden.



- kogels, torpedos, power-ups en actieve-powerups zijn procedures waarmee resp. de lijst van de kogels, de lijst van de torpedos, de lijst van de powerups en de lijst van de actieve-powerups opgevraagd kunnen worden.
- level-gewonnen? is een predikaat dat aangeeft of de speler het huidige level gewonnen heeft (als er geen alienschepen meer zijn in de vloot).
- spel-verloren? is een predikaat dat aangeeft of de speler het spel verloren heeft (als de raket geen levens meer heeft, of als een alienschip de bodem raakt).
- level-up! is een procedure die zorgt voor het overgaan naar het volgende level.
- huidig-level is een procedure waarmee het nummer van het huidige level opgevraagd kan worden.
- aantal-beschikbare-torpedos is een procedure waarmee het aantal beschikbare torpedos die de spelers heeft opgevraagd kan worden.
- incr-aantal-beschikbare-torpedos! is een procedure waarmee het aantal beschikbare torpedos met 1 verhoogd kan worden.
- niet-tijdgebaseerde-powerup? is een procedure die, gegeven een getal dat het type actieve powerup voorstelt aangeeft, nagaat of de actieve powerup een niet-tijd gebaseerde powerup is.

### 3.21. Spel ADT

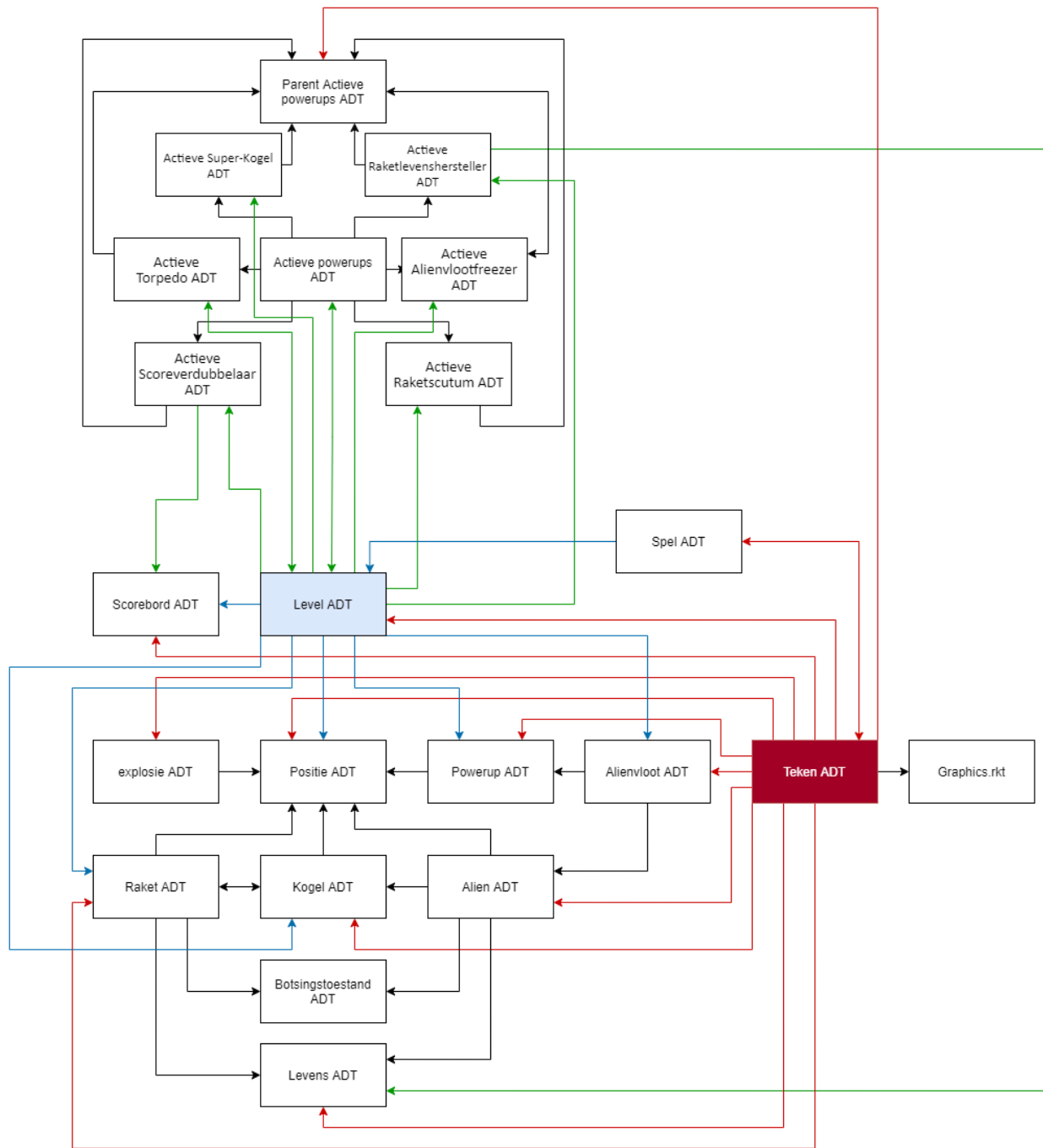
Het spel ADT is het centrale ADT van het spel. Het is verantwoordelijk voor het maken van een nieuw levelobject en een nieuw tekenobject. Het spel ADT zal het spel opstarten en de nodige procedures uit het gemaakte tekenobject en levelobject oproepen om de spelelementen te tekenen, naar toetsenbordinput te luisteren, en het spel op gang te zetten.

Naam	Signatuur
maak-adt-spel	$(\emptyset \rightarrow \text{spel})$
start	$(\emptyset \rightarrow \emptyset)$
level	$(\emptyset \rightarrow \text{Level})$

Tabel 21: operaties van het spel ADT.

- maak-adt-spel zal een nieuw spel object aanmaken.
- start is een procedure die zorgt voor het opstarten van het spel. Deze procedure zal set-spel-lus-functie! uit het teken ADT oproepen op de spel-lus-procedure. De spel-lus-procedure zorgt voor het in gang zetten van de spellogica (door update! uit het level ADT op te roepen) alsook alle spelelementen op het scherm tekenen (door teken-spel! uit het teken ADT op te roepen.)
- level is een procedure waarmee het gemaakte level object opgevraagd kan worden.

## 4. Afhankelijkheidsdiagram



Figuur 1: Het afhankelijkheidsdiagram van Space Invaders.



Figuur 2: Legende van het afhankelijkheidsdiagram, ADT 1 is afhankelijk van ADT 2.

Figuur 1 toont het afhankelijkheidsdiagram van mijn implementatie van de tweede fase van het spel Space Invaders. In de tweede fase van het spel zijn er in totaal 21 ADT's. Er zijn dus 10 ADT's meer dan ik gepland heb in mijn voorstudie. 8 van deze 10 ADT's zijn ADT's voor de actieve powerups. De resterende 2 ADT's zijn het explosie ADT en het botsingstoestand ADT.

Figuur 2 toont de legende van het afhankelijkheidsdiagram. Een pijl tussen twee ADT's betekent dat het ADT waarvan de pijl vertrekt, afhankelijk is van het ADT waar de pijl naartoe gaat. Een dubbele pijl tussen twee ADT's betekent dat beide ADT's afhankelijk zijn van elkaar. Om verwarring te vermijden worden alle pijlen van het teken ADT in het **rood** aangeduid, alle pijlen van het level ADT in het **blauw** aangeduid en alle pijlen van de actieve powerups ADT's naar andere ADT's in het **groen** aangeduid.

De belangrijkste afhankelijkheden zijn:

- Het actieve super-kogel ADT, het actieve Alienvlootfreezer ADT, het actieve Raketscutum ADT, het actieve Scoreverdubbelaar ADT, het actieve raketlevenshersteller ADT en het actieve Torpedo ADT zijn afhankelijk van het parent actieve powerup ADT omdat het parent actieve powerup ADT het type en de toestand van de verschillende actieve powerups bijhoudt een aanpast.
- Het actieve powerups ADT is afhankelijk van de verschillende actieve powerups ADT's omdat het actieve powerups ADT een procedureobject dat een actief powerupobject aanmaakt teruggeeft.
- Het teken ADT is afhankelijk van de meeste ADT's die de spelelementen voorstellen omdat het teken ADT deze elementen op het scherm moet tekenen.
- Het level ADT is afhankelijk van de meeste ADT's die verschillende spelelementen voorstellen omdat de spelelementen met elkaar in het level ADT interageren. De spellogica is dus in het level ADT geïmplementeerd.
- Het alien ADT en het raket ADT zijn afhankelijk van het kogel ADT omdat ze kogels moeten schieten.
- Het level ADT is afhankelijk van het kogel ADT omdat de torpedo's die in het level ADT gemaakt worden kogels zijn met een andere actie bij de botsingsdetectie.
- Het alien ADT en het raket ADT zijn afhankelijk van het levens ADT omdat het levens ADT hun levens bijhoudt.
- Het alien ADT en het raket ADT zijn afhankelijk van het botsingstoestand ADT omdat dit ADT de staat van hun botsing bijhoudt en aanpast.

In de voorstudie had ik geen ADT's voorzien voor de actieve powerups. Hier heb ik meerdere ADT's daarvoor voorzien. Voor elke actieve powerup is er een ADT, aangezien dat elke powerup iets anders doet en in principe niets te maken heeft met de acties van de andere powerups (separation of concerns). Om codeduplicatie te vermijden, wordt er gebruikgemaakt van compositie. Er wordt in elk actieve powerup ADT een parent actieve powerup ADT aangemaakt dat de toestand en het type van de powerup bijhoudt. Het botsingstoestand ADT is ook een ADT dat dient om codeduplicatie te vermijden, aangezien dat de procedures en variabelen van dit ADT identiek waren in zowel het raket ADT als het alien ADT, worden ze in een ADT geëncapsuleerd.

## 5. Logboek

Onderstaande tabel geeft aan wanneer ik aan het project gewerkt heb in de voorbije weken.

Week	Actie
Week 27	Basis voorzien van het scorebord ADT en het levens ADT.
Week 28	Afwerken scorebord ADT en leven ADT, aanpassen alienvloot ADT.
Week 29	<b>Tussentijds indienmoment 3</b> – level ADT uitbreiden met nieuwe levels, schietende aliens.
Week 30	Aanpassingen aan bestaande ADT's brengen (zoals aliens die schieten, ...)
Week 31	Basis voorzien van de verschillende ADT's m.b.t. de powerups, nodige aanpassingen brengen in het level ADT.
Week 32	Even pauze :)
Week 33	<b>Tussentijds indienmoment 4</b>
Week 34	Bugs hersteld en powerups afgewerkt. Explosie ADT en botsingstoestand ADT geïmplementeerd.
Week 35	Verslag schrijven. Code verder verbeteren.
Week 36	Verslag schrijven. – <b>Indienen code en verslag.</b>

Tabel 22: Mijn planning om de tweede fase het computerspel Space Invaders te implementeren.

Mijn planning kwam voor het grootste deel overeen met die van de voorstudie. De grootste verschillen waren het implementeren van de powerups. Dit had meer tijd in beslag genomen dan gepland., wat mij logisch lijkt aangezien dat ik de implementatie van de actieve powerups niet had beschreven in mijn voorstudie.