

Programmeerproject 1: Voorstudie “Space Invaders” (Fase 2)

Maxim Brabants

0576581

Maxim.Lino.Brabants@vub.be

Academiejaar 2020-2021



VRIJE
UNIVERSITEIT
BRUSSEL

Inhoudsopgave

1	Verwachte aanpassingen.....	3
1.1	Schietende alienschepen	3
1.2	Scores.....	4
1.3	Andere levels.....	4
1.4	Power-ups	5
2	Nieuwe ADT's.....	5
2.1	Score-ADT.....	6
2.2	Power-up-ADT.....	6
3	Werktijd.....	7

1 Verwachte aanpassingen

Met fase 1 afgerond te hebben beschikken we nu over een eerste speelbare versie van Space Invaders, maar nog niet de echte Space Invaders zoals de meeste die kennen. Voor de eerste fase moesten we namelijk enkel de basiscomponenten voorzien zodat we daarna over kunnen gaan naar een volledig werkend spel met min of meer alles erop en eraan.

Dit zijn de onderdelen die we al hebben:

- Een raket die van links naar rechts over het scherm kan bewegen. (beperkt tot de schermranden)
- Een bewegend vloot van 5 rijen op 11 kolommen met daarin verschillende soorten aliens (verschillende kleur + meer/minder levens)
- De raket kan schieten en welbepaald meerdere kogels tegelijk. Als een kogel een alienschip raakt dan verdwijnt deze in kwestie.

Dit zijn de ADT's die we reeds hebben: raket-ADT, alienschip-ADT, alienvloot-ADT, kogel-ADT, kogels-ADT, positie-ADT, spel-ADT, level-ADT en teken-ADT.

Dit zijn de cruciale dingen die het spel zeker moet bevatten en waar we vervolgens op moeten verder bouwen als we tot een mooi afgewerkt spel willen geraken. De dingen die we hiervoor zeker nog nodig zullen hebben sommen we hier even kort op zodat we een overzichtje hebben van wat er verwacht wordt.

Dat zijn de volgende elementen:

- De alienschepen moeten afzonderlijk ook kogels kunnen afschieten richting de raket.
- We moeten op één of andere manier zowel de huidige als de beste score kunnen bijhouden samen met de levens van de raket.
- Als de raket dan geen levens meer zou hebben of het vloot zou de onderkant bereiken, dan zou het spel opnieuw opgestart moeten worden. In het geval dat alle aliens toch vernietigd zouden zijn, zou er naar een volgend level moeten gegaan worden.
- Ten slotte moeten we ook power-ups voorzien.

We zullen per puntje bespreken welke aanpassingen/vernieuwingen we gaan doen om dit te verwezenlijken.

1.1 Schietende alienschepen

Als eerste hebben we dus onze alienschepen die apart kogels moeten kunnen afvuren met als doel de raket te raken. Dit is een interessante uitbreiding die eenvoudig kan zijn als het eerste ontwerp goed zit.

In ons geval hoeft het niet perse moeilijk te zijn en denk ik ook niet dat we voor dit onderdeel nood gaan hebben aan een nieuw ADT of iets dergelijks. Dit veronderstel ik omdat we enerzijds ons vloot hebben met daarin aliens die apart gemanipuleerd kunnen worden en anderzijds de kogels die al afgevuurd kunnen worden door onze raket en die kunnen raken. Het gaat eerder een kwestie zijn van

nieuwe procedures bijschrijven in bestaande ADT's die ons dan de nieuwe functionaliteit gaan toelaten. Ook kan het mogelijk zijn dat we hier en daar iets gaan moeten aanpassen, maar dat zien we dan ook wel op de moment zelf.

Hier volgt een compacte beschrijving van de procedures die we denken nodig te zullen hebben om de aliens te kunnen laten schieten:

- Er moet sowieso een procedure zijn die een alien triggert om een kogel af te vuren. Op een bepaald moment in ons programma moeten we dus bijvoorbeeld kunnen zeggen: 'Als een bepaalde conditie waar is, dan vuur je voor die alien een kogel af'.
- We moeten een soort van procedure hebben die nakijkt of de raket geraakt is.

Procedure	Signatuur
schiet	/ → /
raket-geraakt?	/ → /

Waar we ook over kunnen nadenken is hoe en waar we de kogels gaan bewaren die de alienschepen zullen afvuren. We zouden deze kunnen bijhouden in onze kogelslijst maar dan hebben we het probleem dat ze gemengd zitten met de kogels die de raket afvuurt en zouden we ze van elkaar moeten onderscheiden. Een andere oplossing waar ik aan denk is om voor elke soort kogel (afgevuurd door raket of alienschip) een aparte lijst bij te houden.

1.2 Scores

Bij het raken van een alien zouden er punten opgeteld moeten worden bij een globale score. Ook moet deze gereset worden bij een nieuw spel en uiteindelijk ook 'opgeslagen' indien het de hoogste is. Het aftrekken van scores moet niet en lijkt me niet echt logisch om te doen in Space Invaders.

De meest elegante optie lijkt me om het hele idee van een score onder te brengen in een ADT zodat we steeds werken met een compact score-objectje dat alles van de score bevat. Zo kunnen we via een dispatchprocedure gemakkelijk de bijhorende manipulatieprocedures aanroepen. De effectieve score moeten we dan bijhouden als lokale toestand in een let-expressie. Als we de score dan nodig zouden hebben dan kunnen we deze gemakkelijk aanroepen als we deze vrijgeven in onze dispatch. Voor het ADT verwijs ik door naar paragraaf 2.1 waar het Score-ADT beschreven staat.

Onze huidige score en hoogste score zouden we eventueel in het level-ADT kunnen bijhouden. We maken er dan een instantie van aan het begin van het spel en veranderen de staat ervan doorheen het spel. Het aantal levens van de raket kunnen we gewoon bijhouden in het raket-ADT zelf en deze ook aanpassen als de raket geraakt zou worden. Deze displayen we best op het scherm zodat de gebruiker zich bewust is van het resterende aantal levens.

1.3 Andere levels

Als ik denk aan andere levels, dan denk ik aan nieuwe spelsituaties waarbij het niveau moeilijker wordt. Voorbeelden daarvan zouden kunnen zijn: de alienschepen die sneller over het scherm bewegen zodat ze vlugger onderaan het scherm zijn of de aliens die frequenter kogels afvuren. We moeten zoeken naar een mechanisme dat ons toelaat, een nieuwe spelsituatie te genereren. Dat wilt zeggen dat alle

objecten opnieuw op hun oorspronkelijke positie moeten gegenereerd worden indien nodig en dat er dan een kleine verandering zit in het niveau.

We hebben dus een procedure nodig die dat voor ons zal doen (nieuwe situatie genereren). Daarnaast kunnen we misschien best een nummer bijhouden dat ons level identificeert. Zo kunnen we elk level(nummer) associëren met bepaalde functionaliteit. Op die manier is het ook makkelijker om achteraf nieuwe levels toe te voegen.

Procedure	Signatuur
verander-level!	/ → /

Er is niets dat we moeten meegeven aan deze operatie aangezien die er gewoon voor zal zorgen dat er kan overgegaan worden naar een nieuw level (logischerwijs 1 hoger). Dit houdt in: alles opnieuw tekenen, nieuwe instanties maken, etc. .

1.4 Power-ups

Wat er voor dit onderdeel van ons wordt verwacht is dat we minstens 5 verschillende soorten van power-ups ontwikkelen en implementeren in ons spel. Deze kunnen door de speler verkregen worden op de volgende manier: wanneer de raket sommige schepen raakt dan zal na elk geraakt schot van een willekeurig schip een bolletje vrijkomen en dat bolletje stelt de effectieve power-up voor. De speler krijgt dan de kans om deze power-up te bemachtigen door met de raket het bolletje te raken. Dit zal hij namelijk kunnen doen, aangezien het bolletje langzaam naar beneden zakt. Het is dus de bedoeling dat de speler zelf het initiatief neemt om de power-up op te halen want het risico bestaat dat de speler de power-up niet op tijd raakt en dan is hij die power-up kwijt.

Ik maak graag de redenering dat wanneer we een nieuw concept willen toevoegen aan ons spel en dat concept houdt in dat er een nieuw object of nieuwe objecten in onze spelwereld aanwezig gaan zijn, dat ik dan in de meeste gevallen opteer om daarvan een ADT'je te maken. Dit om de simpele redenen dat het me logisch lijkt en alle objecten eigenlijk één voor één geëncapsuleerd zitten in een objectje. Dat maakt je spel in zijn geheel overzichtelijk.

Voor het idee om power-ups te implementeren dacht ik in eerste instantie aan het uitbreiden van het level-adt met procedures die dan communiceren met het teken-adt en zo op het juiste moment de power-ups inschakelen, maar eigenlijk is het misschien beter dat ik dan mijn redenering volg en het probleem oplos met de aanmaak van een nieuw ADT. Op deze manier moeten we dit allemaal niet gaan bijhouden in een ander ADT en wordt het weer een stuk overzichtelijker. Ik verwijs door naar de paragraaf 2.2 voor een kleine beschrijving van dit ADT.

2 Nieuwe ADT's

In deze sectie beschrijven we de nieuwe ADT's die we gaan nodig hebben of die handig zijn voor de implementatie van fase 2. Veel nieuwe ADT's hebben we niet echt nodig want we hebben eigenlijk al bijna alle cruciale ADT's geïmplementeerd in fase 1.

2.1 Score-ADT

Dit is een klein ADT'tje met een beperkt aantal operaties maar het is handig omdat alles dan overzichtelijk blijft en dat je er gemakkelijker mee kunt werken.

Volgende operaties horen bij dit type:

Procedure	Signatuur
maak-score	/ → /
verhoog!	hoeveelheid → /
herstel!	/ → /

- De **maak-score** verwacht geen argumenten want het is gewoon een éénmalig score-object met lokale toestand.
- De **verhoog!** zal dan de globale score in kwestie verhogen met een meegegeven hoeveelheid. Deze hoeveelheid kan gebaseerd worden op het soort alien dat geraakt werd.
- **herstel!** gaat de score terug naar zijn initiële toestand zetten (niet zeker of deze noodzakelijk is).

2.2 Power-up-ADT

Het abstracte datatype voor de 5 power-ups die we gaan bedenken. Hieronder staan de operaties waarvan ik denk dat het ADT deze zal omvatten. Het zal ook weer een redelijk klein ADT zijn zoals het Score-ADT maar het is denk ik wel elegant om te hebben en het op de juiste manier te implementeren.

Procedure	Signatuur
maak-power-up	symbol, positie → /
beweeg!	/ → /

- We zouden aan de **maak-power-up**-constructor een symbol kunnen meegeven dat een soort power-up voorstelt of gewoon een kernwoord dat de power-up beschrijft. Dit met als doel om later alle verschillende power-ups te kunnen onderscheiden en ook de juiste activeringsprocedures aan te roepen. Het zou ook handig zijn om een positie mee te kunnen geven zodat er wordt geweten op welke positie de power-up zou moeten aangemaakt en getekend worden.
- De **beweeg!** is een simpele operatie die de power-up langzaam naar beneden zal doen bewegen tot hij ofwel door de raket is opgenomen of uit het scherm is verdwenen. We hoeven

hier geen richtingsargument aan mee te geven, aangezien de enige toegelaten richting voor een power-up naar onder is volgens de projectbeschrijving.

3 Werktijd

Dit verwachten we te doen de komende weken:

Week	Taak
Week 27	Ervoor zorgen dat de aliens kogels kunnen afvuren.
Week 28	De kogels de raket laten raken en zien dat de aliens elkaar niet raken.
Week 29	Het Score-ADT implementeren
Week 30	Proberen de score te tonen tijdens het spel en ook punten bij de score tellen wanneer aliens worden geraakt.
Week 31	De hoogste score laten bijhouden.
Week 32	Een manier vinden om alle spelelementen te herinitialiseren zodat er aan een nieuw level kan begonnen worden.
Week 33	Bij elk nieuw level een moeilijkheid voorzien door bijvoorbeeld het alienvloot sneller te laten bewegen.
Week 34	Nadenken over mogelijke power-ups + Power-up-ADT implementeren.
Week 35	De power-ups in de vorm van een bolletje naar de onderkant van het scherm laten bewegen.
Week 36	De power-ups activeren voor de raket.