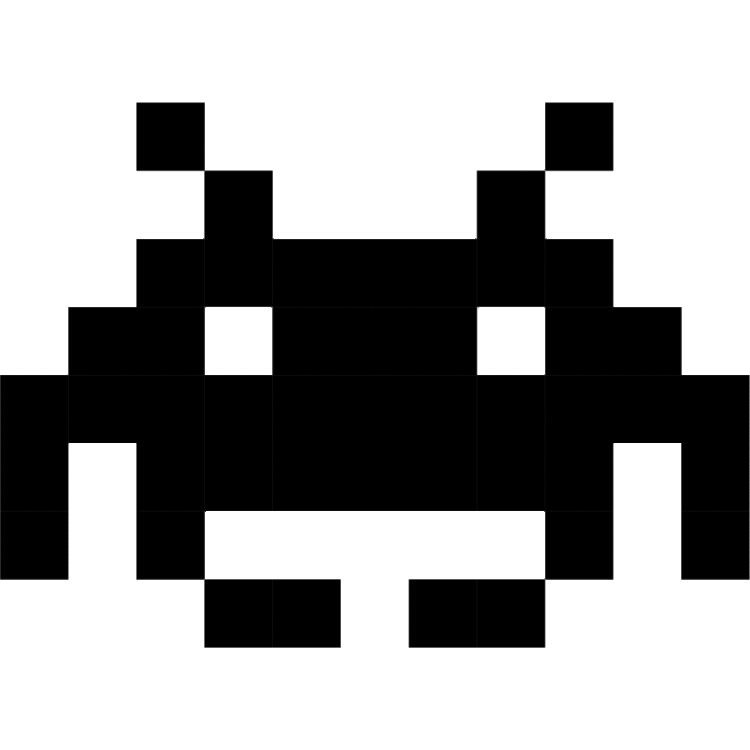
Rapport Space Invader Agent

Nima Agha Ghafar Hamedani (22085947)

Tommi Lander

Isa Dijkstra

Vince Ammerlaan

6-5-2025

# Inhoudsopgave

[**Inhoudsopgave**](#_8mylnyhrae4h) **2**

[**Inleiding**](#_vbfemkw2ouk1) **3**

[**Probleemanalyse: Waarom is Reinforcement Learning (RL) geschikt?**](#_lpbkog8jcvdx) **3**

[Kenmerken van het probleem:](#_osodvs5c6xr) 3

[Waarom past RL goed bij dit probleem?](#_mnzqxo9rfb2w) 3

[Probleem en doel van dit project](#_wssr3z72l1i2) 4

[**Verschil met supervised learning en niet-AI-oplossingen**](#_p27j97anpbca) **4**

# Inleiding

Bij deze opdracht richten we ons op het klassieke computerspel Space Invaders. In dit spel bestuur je een kanon dat onderaan het scherm heen en weer beweegt en op inkomende aliens schiet. Ons doel is om een agent te ontwikkelen die zelfstandig leert spelen en een zo hoog mogelijke score behaalt. Dit doen we met een Reinforcement Learning algoritme, waarbij de agent via rewards (de punten die je krijgt als je een alien raakt) leert welke acties in welke situaties het beste zijn. Space Invaders start met meerdere rijen aliens die langzaam over het scherm bewegen. De speler bestuurt een kanon onderaan en kan naar links of rechts bewegen, en vuur afvuren. Tussen het kanon en de aliens staan bunkers die bijhoudend schieten door vijanden geleidelijk kapot gaan. Het spel eindigt op twee manieren: of alle aliens zijn verslagen en je gaat door naar het volgende, moeilijkere level, of je laatste levenspunt is opgebruikt omdat je geraakt bent door een vijandelijk projectiel of doordat een alien het onderste verdedigingsniveau bereikt heeft. In dat laatste geval is het spel afgelopen en krijg je je eindscore te zien.

# Probleemanalyse: Waarom is Reinforcement Learning (RL) geschikt?

### Kenmerken van het probleem:

* **Veranderende omgeving:** In het spel SpaceInvaders bewegen de vijanden steeds anders. Soms zijn ze sneller of gebruiken ze andere patronen. Hierdoor verandert het spel constant.
* **Beslissingen in stappen:** Je moet meerdere acties achter elkaar goed uitvoeren, zoals schieten en opzij bewegen, om een goede score te halen.
* **Beloningen komen later:** Je krijgt niet meteen punten voor elke actie. Soms weet je pas later of iets goed was, bijvoorbeeld als je een vijand raakt.

### Waarom past RL goed bij dit probleem?

* **Leren door te doen:** Een RL-agent leert van het proberen en krijgt feedback als iets goed of fout gaat. Hij leert dus zonder dat je hem eerst voorbeelden hoeft te geven.
* **Geen gelabelde data nodig:** In tegenstelling tot andere methodes (zoals supervised learning), heeft RL geen juiste antwoorden vooraf nodig.
* **Kijkt naar de toekomst:** De agent maakt niet alleen keuzes die nu goed zijn, maar kijkt ook wat op de lange termijn het meeste oplevert.
* **Kan zich aanpassen:** Omdat het spel verandert, is het belangrijk dat de agent flexibel is. RL kan zich goed aanpassen aan nieuwe situaties.

### Probleem en doel van dit project

**Wat is het probleem?** We willen een slimme agent maken die zelf leert om het spel SpaceInvaders goed te spelen. Deze agent moet:

* Leren hoe de vijanden bewegen,
* Goed kunnen bewegen en op het juiste moment schieten,
* Zich aanpassen aan veranderingen in het spel.

**Wat is het doel?** De agent moet:

* Zelf leren van zijn ervaringen en een goede strategie ontwikkelen,
* Beter presteren dan een agent die zomaar wat doet (bijvoorbeeld een agent die willekeurig beweegt),
* Slim reageren op patronen in het spel, zodat hij langer overleeft en een hoge score haalt.

# Verschil met supervised learning en niet-AI-oplossingen

Bij supervised learning train je een model op voorbeelddata met gelabelde uitkomsten. Voor Space Invaders zou dat betekenen dat je voor elke schermsituatie handmatig moet opschrijven wat de beste zet is, iets wat in de praktijk onmogelijk is omdat er duizenden verschillende situaties zijn. Eenvoudige scripts werken met voorgeprogrammeerde regels, bijvoorbeeld telkens schieten zodra een alien recht boven je staat. Zulke regels zijn niet adaptief; ze houden geen rekening met onverwachte patronen of verrassende vijandbewegingen. RL daarentegen zoekt voortdurend naar de strategie die de hoogste totale score oplevert, waarbij hij zowel directe als toekomstige beloningen meeneemt in zijn beslissingen.

# Methodologie

We hebben gekozen voor **Q-learning**, omdat het model hiermee zelf leert door beloningen te krijgen wanneer het iets goed doet. In tegenstelling tot technieken zoals supervised learning, hebben we hier geen gelabelde data nodig. De agent leert alles zelf, puur door te spelen en te experimenteren.

Omdat de observaties in het spel bestaan uit veel getallen (bijvoorbeeld de positie van vijanden, projectielen en het ruimteschip), hebben we deze gegevens eenvoudiger gemaakt door de waarden op te delen in zogeheten "bins". Dit betekent dat we de omgeving in vakjes hebben verdeeld, zodat het model makkelijker kan inschatten in welke situatie het zich bevindt.

De agent kan verschillende acties uitvoeren, zoals:

* Verplaatsen naar links of rechts
* Schieten
* Niets doen

Welke actie wordt gekozen, hangt af van een strategie genaamd **epsilon-greedy**. Dit betekent dat de agent soms willekeurig iets probeert (exploratie), en soms kiest wat tot nu toe het beste werkte (exploitatie). Hoe vaak hij iets willekeurigs doet, wordt bepaald door de waarde van epsilon (ε).

Tijdens het trainen van de agent hebben we meerdere keren geëxperimenteerd met verschillende instellingen:

* **Alpha (α)**: bepaalt hoe snel het model leert van nieuwe informatie.
* **Gamma (γ)**: bepaalt hoe belangrijk toekomstige beloningen zijn.
* **Epsilon (ε)**: bepaalt hoe vaak het model iets nieuws uitprobeert.

Voor elk van deze drie parameters hebben we een aantal waarden getest. De agent werd getraind over 100 episodes per instelling. De behaalde scores per episode werden opgeslagen, zodat we later konden analyseren wat het effect van elke parameter was op de prestaties van de agent.

Na afloop hebben we drie aparte grafieken gemaakt waarin we per parameter (alpha, gamma, epsilon) kunnen zien hoe de leerprestaties zich ontwikkelen. Zo kunnen we bepalen welke instellingen het beste werken voor het spel *SpaceInvaders*.

# Resultaten

