# Jumper oefening VR:

## Doel:

Het trainen van een model om een agent over een reeks blokken te laten springen die ieder een random snelheid krijgen.

## Setup:

### Onderdelen in het systeem:

|  |  |
| --- | --- |
|  | |
|  | Agent, het getrainde object dat zal proberen om over de blokken te springen. De agent gebruikt ook enkel rays om op te vangen waar hij en de blokken zich bevinden.  De agent kan enkel over de y as bewegen. |
|  | Block, de hindernis waarover de agent moet springen.  Blokken worden om de 4 seconden ingeladen en bewegen automatisch tegen een random snelheid tussen de 2 en 3.4. |
|  | Speelgebied. In de grijze zonnen zullen de blokken bewegen en zal de agent proberen te springen. De groene zone is de eindzone. |

### Het beloningssysteem:

Wanneer de agent niet juist springt, zal het blok de agent raken. In dit geval wordt er een negatieve beloning van -1 gegeven en zal de scene opnieuw gestart worden. Als de agent juist springt zal het blok verder gaan en de eindzone bereiken. Als een blok de eindzone haalt, word er een positieve beloning van 0.1 gegeven.

Het succesvol springen over een blok heeft dus een kleinere impact dan het raken van het blok, zodat het model sneller leert dat het springen zeer belangrijk is.

## Guide:

### Scripts

In deze oplossing zijn er 3 scripts nodig geweest:

1. Spawner
2. Move
3. Jumper\_agent

1.Spawner

Dit script zorgt ervoor dat de blokken steeds op dezelfde start positie spawnen. Het script roept elke 4 seconden een spawn functie op zodat er steeds enige plaats tussen de verschillende objecten ontstaat.

Dit script moet op het speelgebied gehangen worden.

Dit script verwacht 2 input waardes:

1. Block (het game object dat gespawnt moet worden)
2. Spawn (de spawn plaats van de objecten)

2.Move

Dit script zorgt ervoor dat de blokken automatisch naar voor bewegen. De blokken krijgen in hun start functie een random snelheid toegewezen die tussen de 2 en 3.5 ligt. Deze snelheden zijn gekozen om zo veel mogelijk groepering van de blokken te verminderen.

Ook dit script wordt aan het block object gehangen.

3.Jumper\_agent

Dit is het script dat uitgevoerd zal worden om het model te trainen. Dit agent script gebruikt volgende belangrijke functies:

1. Initialize
2. OnActionReceived
3. Heuristic
4. onEpisodeBegin
5. ResetPlayer
6. OnCollisionEnter
7. OnTriggerEnter
8. Thrust

En heeft 3 input waardes:

1. Force (de grootte van de krachten in het systeem)
2. Orig (de originele spawn positie van de agent)
3. Jump strength (de grootte van de spring kracht)

1.Initialize

In de initialize functie definiëren we de agent, in ons geval de blauwe bal dus. Ook limiteren we enkele assen van dit object zodat de bal enkel omhoog kan springen.

2.OnActionRecieved

Aangezien de agent enkel omhoog kan bewegen is er maar 1 action. Deze action zal de thrust functie aanroepen.

3.Heuristic

Met deze functie kunnen we de scene zelf testen. Deze functie controleert of de speler op de “up arrow” klikt. Indien de speler deze knop induwt, wordt de thrust functie opnieuw aangeroepen.

4.OnEpisodeBegin

Deze functie wordt in het begin van elke episode aangeroepen en gaat de resetPlayer functie aanroepen.

5.ResetPlayer

Deze functie verplaatste de agent terug naar de originele positie en verwijdert elke kracht die mogelijks op het object uitgevoerd word.

6.OnCollisionEnter

Deze functie checkt of de agent het block raakt en zal ook een negatieve beloning geven indien dit het geval is.

Er is ook een extra veiligheid ingebouwd, het gebeurde soms dat de agent te hoog sprong. Daarom is er een jump limiet object voorzien, dit is een block object dat boven de agent geplaatst is op de gewenste jump limiet. De functie checkt of de agent dit object raakt, is dit het geval dan zal er weer een negatieve beloning gegeven worden.

7.OnTriggerEnter

Deze functie checkt of de blokken de eindzone bereiken en geeft ook een positieve beloning indien dit het geval is.

8.Thrust

Deze functie zorgt ervoor dat de agent kan springen. Er wordt gecontroleerd of de agent nog niet aan het springen is. De agent kan enkel springen als hij nog niet aan het springen is.

### Agent setup

|  |  |
| --- | --- |
|  | Voor de rays van de agent is enige setup vereist. We geven op welke objecten er allemaal gedetecteerd moeten worden en enkel basis settingen zoals de lengte en het aantal rays. |
|  | Het behavior parameter script zorgt ervoor dat de agent bepaalde beslissingen gaat maken tijdens de training. |
|  | De decision requester bepaalt het aantal beslissingen dat gemaakt zal worden. |

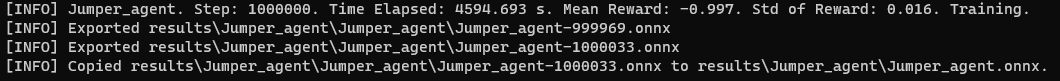
### Configuratie van de training:

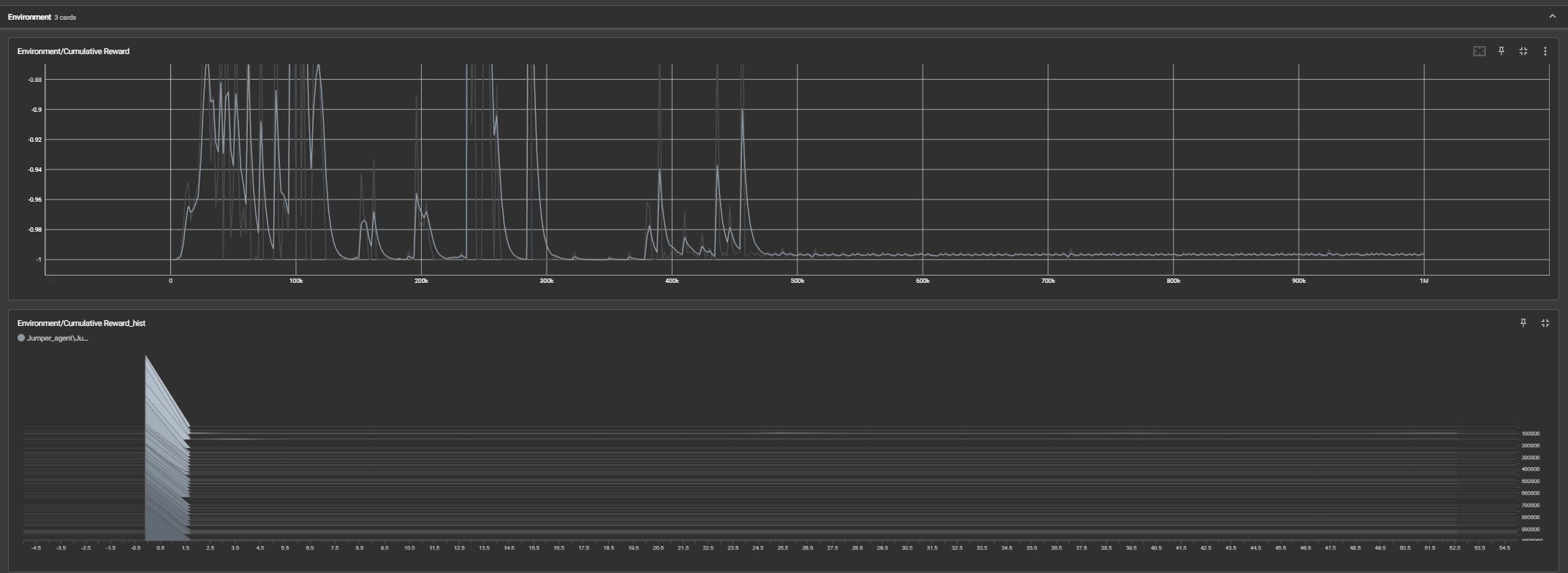
|  |  |
| --- | --- |
|  | Voor de setup van de training heb ik deze setup gebruikt. Dit is dezelfde setup van de vorige machine learning oefening.  Ik heb enkel het maximum aantal stappen verhoogt zodat het model langer traint bij elke training. |

TensorFlow resulaten van de training:

Ik heb de training uitgevoerd op 11/04/2023 en dit leverde volgende resultaten op na

1.000.000 stappen.





De eerste 500 duizend stappen liggen de resultaten zeer sterk uit elkaar. Tussen de iteraties liggen er redelijk grote verschillen. De agent haal redelijk hoge cijfers maar vaak ook redelijk lage cijfers.

Uit de eerste grafiek kunnen we zien dat er rond de 500 duizendste stap een grote verandering plaats vind. De resultaten stabiliseren en stabiel tussen de -0.9 en -1 liggen.

Op de onderste grafiek kunnen we zien dat alle waardes tussen de -1 en 1.5 liggen. Omdat onze “straf” veel hoger is dan de “beloning” zien we ook dat de waardes meer naar de -1 neigen. Omdat er toch hoger scores bereikt worden, kunnen we afleiden dat het model toch enige training ondervind en er soms toch in slaagt om de scene succesvol af te ronden.

## Conlusie:

Het trainen op zich is goed gelukt, maar ik denk dat het resultaat nog beter kan door bepaalde zaken nog meer op punt te zetten

Ik denk dan aan het eenvoudiger maken voor de agent om resultaten te halen

* Het springen verbeteren zodat de agent meer tijd heeft om te springen. Nu is deze timing zeer precies. De kans op te laat springen is groot.
* Het springen sterker maken zodat de agent over 2 blokjes kan springen. Soms komt een snel en een traag blokje samen en geraakt de agent er niet over

De resultaten zijn nog niet altijd even consequent maar meer training met die scenario had niet veel zin want na stap 500.000 bleven de resultaten gelijk.