Versió 1

Primera fase

1. Representació del model NPC dintre de Unity

El NPC l’he representat com un objecte que com a fills te *AnchorLeft, AnchorRight and Agent*.

Els Anchors son objectes buits que tenen un component de detecció de Collisions. El seus comportaments no son de collisió típica ja que realmente l’Agent mai farà collisió real sobre els Anchors, però tindràn la propietat de ser triggers d´events.

Unity té una API que permet detectar aquests events a nivel de C#.

L’Agent es un objecte que també t’un detector de collisions, i com a fills uns objectes que permeten projectar rajos per detectar on esta la collisió més próxima. La collisió més próxima dependrà de com s’han configurats els rajos. En aquest cas surt un raig per l’esquerra i un per la dreta.

Es important notar que, l’objecte NPC no es visible en la imatge seguent, pero fa de contenidor. Això es important perque l’unic objecte que es mou es el fill Agent del NPC, que es qui al final tindrà el model de Neuronal Network.

Escala de tiempo

Descripción generada automáticamente con confianza media

Texto

Descripción generada automáticamente

1. Com modelitzar un problema fent ús de Unity MLAgents

Primer de tot ens em de instal·lar el paquet ML Agents dintre d’un projecte de Unity. Per instalar un paquet s’ha d’anar al *Package Manager*.  
  
Jo estic treballant amb la versió d’Unity 2021.2.0f1 I amb la versió 2.0.0 de ML Agents. Es important les versions amb les que treballem perquè pot provocar problemas de compatibilitat. Al repository de [ML Agents](https://github.com/Unity-Technologies/ml-agents) d’Unity en l’apartat de *Releases and Documentation* hi ha la taula de compatibilitats entre Unity i la dependencia.

La resta d’eines que s’utilitzen per poder conectar Unity amb el procés de Deep Learning esta especificat en la seguent [documentació](https://github.com/Unity-Technologies/ml-agents/blob/release_17_docs/docs/Installation.md).

1. Modelització de moure l´Agent d’esquerra a dreta ciclicament.

Per poder modelitzar el problema es necesari que creen una clase que contindrà la part lógica de la modelització i que l’extenguem de la clase *Agent* de ML Agent.

La clase *Agent* conte varis metodes que poden ser sobrescrits en la nostra clase principal.

Metodes que s’han sobrescrit de la clase *Agents*

* Initialize
* OnEpisodeBegin
* OnActionReceived
* CollectObservations
* Heuristic

A continuació explicaré per quina funcionalitat están pensades aquests methodes sobrescrits.

**Initialize**

Aquest método es crida un cop únicament en el cicle de vida del aprenentatge.

Esta pensat per fer *caching* d’objectes de la escena o inicialitzar valors.

Texto

Descripción generada automáticamente

**OnEpisodeBegin**

Aquest método es crida cada cop que un episodi d’aprenentatge ha acabat, forçadament o perque el màxim nombre de pasos del episodi s’han completat.

S’ha de ser consistent amb aquest método, es a dir, en el meu cas l’Agent sempre sortira del centre del NPC, i jo forço la seva recolocacio al centre, també s’ha de tenir en compte els problemas d’inercia que porta l’objecte, o la recalculació de target o metas pel nou episodi.

Texto

Descripción generada automáticamente

**OnActionReceived**

Aquest método porta un parametre d’entrada, anometat *ActionBuffer*, l’*ActionBuffer,* es un objecte que te dos arrays, un per poder representar valors continos i altres per representar valors discrets.

Pel meu problema necesitava emular el moviment lateral. Els valors que hi ha dintre de cada posicio dels arrays es simbolic, i ho has de saber tu com a desarollador del model.

Com he dit, només estava interest en la emulacio del moviment lateral. El rang de valors que m’espero va de -1 a 1, -1 representa moviments cap a l’esquerra i 1 cap a la dreta.

Amb aquesta informació jo puc canviar la dirección de la velocitat del Agent.

Aquesta funció pot ser cridada pel motor de Deep Learning o per interacció directa amb un jugador. En cas que el programa estigui corrent en mode d’aprenentatge, será el motor de DeepLearning que lid oni valors a *actions.ContinuousActions[0]*. En cas que no estigui aprenent, es a dir corrent en mode *Heuristic*, els valors serán donats gracias a l’interacció amb la funcio *Heuristic* que veurem més endavant.

En cas que estigui corrent en mode d’aprenentatge, el nombre d’elements dels arrays de *ActionsBuffer* s’han de configurar per fora, ho veurem més tard quan parlem sobre la configuració en l’entorn d’Unity.

Texto

Descripción generada automáticamente

**CollectObservations**

*CollectObservations* es un metode que s’executa cada x temps, aquest temps d’execució es configura amb l’eina de Unity.

Fixat que igual que en el método *OnActionReceived* aquest te un parametre. En aquest cas no esta pensat per consumir les dades d’aquest parametres sinó per donar informació al motor de DeepLearning.

Fent ús de la referencia sensor podem fer que cada x temps, el motor de DeepLearning sapiga dades relevants per que ell pugui resoldre el problema.

Lo que vaig pensar que seria interesant que observes, es, la horientació del Agent respecte el target, la distancia del Agent al target, la posición del target i la velocitat amb la que el Agent es mou.

Es important notar que encara que enregistre una observacio d’un objecte de tipus Vector2D, no es guarda com una observació sinó com a dos. Això es important perque de la mateixa manera que he explicat abans del *ActionBuffer* s’aplica al *VectorSensor.* Hem de saber quantes observacions es fan realmente perque desprès amb Unity em de configuar l’allargada del vector d’observacions.

**Texto

Descripción generada automáticamente**

**Heuristic**

El metode d’*Heuristic* permet interactuar amb el jugador. Fixat que te la mateixa estructura de dades com a parametre, el *ActionBuffers*.  
Com he explicat previament, el método *OnActionReceived* interactúa tant amb el motor de DeepLearning com amb el jugador, i amb el jugador s’interactua mitjant aquest método.

Les modificacions sobre l’estructura *ActionsBuffers* de Heuristics serán rebudes en el método *OnActionReceived.*

Com només em fixo en els moviments laterals, en la primera posició del continuos actions ha de a ver-hi un valor entre -1 a 1, representant els moviments laterals.

**Texto

Descripción generada automáticamente**