

Ontwerpdocument

Custom rewards demonstrator

REINFORCEMENT LEARNING DEMONSTRATOR - PACMAN

Finn Alberts, Laurent Dassen en Noud Wijngaards
ZUYD HOGESCHOOL | HBO ICT



Inhoud

1 Inleiding.....	3
2 Werking.....	3
2.1 Variabelen	3
2.2 Main-loop.....	3

1 Inleiding

In de custom rewards applicatie wordt een simulatie gedraaid, waarin een vierkantje (“een auto”) over een grid (“de weg”) beweegt. Het doel is dat de auto bij het groene vierkant (“de goedkope parkeerplaats”) of bij het rode vierkant (“de dure parkeerplaats”).

De simulatie werkt aan de hand van de optimale policies. Deze zijn van tevoren bepaald (en worden dus niet berekend in de applicatie). Het bereiken van de dure parkeerplaats geeft een straf van 1 punt en het bereiken van de goedkope parkeerplaats een beloning van 1 punt. De straf voor het verplaatsen van één hokje kan door de gebruiker worden gekozen.

In iedere zet wordt de richting bepaald aan de hand van de gekozen straf voor het verplaatsen. Er is een 20 procent kans dat de auto echter een andere richting op zal gaan. Deze richting is nooit tegengesteld aan de gekozen richting.

Bijvoorbeeld, wanneer naar rechts wordt gekozen, zal de kans 10 procent zijn dat de auto toch naar boven gaat en 10 procent voor naar onder. De auto zal nooit naar links gaan.

2 Werking

De applicatie werkt middels de Python-library pygame.

2.1 Variabelen

Bovenin het script worden aantal functies en variabelen gedeclareerd. De eerste variabele is de *grid_size*. Deze variabele geeft de grootte (in pixels) van één hokje aan.

Een andere variabele is de *grid*. Deze variabele geeft het speelveld aan en wordt gerepresenteerd door een tweedimensionale list. Hierin staat 0 voor een lege ruimte, 1 voor de goedkope parkeerplaats, -1 voor de dure parkeerplaats en 2 voor een muur.

Ook staat bovenin de *policies* variabele. Deze variabele bewaard de optimale policies voor de verschillende straffen (voor het verplaatsen van één hokje). Deze hebben eenzelfde opbouw als de *grid*, echter geven de waardes hier de richting aan. 1 staat hier voor boven, 2 voor rechts, drie voor beneden en 4 voor links. In hokjes waar geen richting kan worden gekozen (parkeerplaats of muur) staat een 0.

De laatste variabele is de *position*-variabele. Deze geeft de huidige positie aan middels een x en y. Linksboven is (0, 0).

2.2 Main-loop

De algemene doorloop van het script begint pas op regel 159. Hier wordt pygame geïnitieerd en wordt de gebruiker gevraagd (middels een verhaal) welke policy hij wil gebruiken. Deze policy wordt opgeslagen in *chosen_policy*. Vervolgens worden achtereenvolgens de volgende stappen continu herhaald:

1. Teken de grid, inclusief de huidige positie;
2. Bepaal de te maken zet, inclusief de 20% kans dat de zet niet één-op-één wordt overgenomen vanuit de policy. Visualiseer deze zet middels pijlen;
3. Update de positie aan de hand van de te maken zet;
4. Controleer of dit de eindpositie is;
5. Wacht één seconde