

Dokumentacja projektu symulacji środowiska oraz jego agenta

Zespół projektowy nr. 2:

Członkowie:

Marcin Drzewiczak, Michał Kubiak, Julia Maria May, Przemysław Owczarczyk

Kierownik zespołu:

Adam Lewicki

1 Opis projektu

Nazwa kodowa projektu: UberKelner

Cel projektu: stworzenie statycznego środowiska dyskretnego (dalej zwanego symulacją) odpowiadającego w przybliżeniu realnej restauracji oraz agenta sztucznej inteligencji jako kelnera w restauracji.

Agent na podstawie dostępnych informacji o symulacji (zakładamy, że symulacja jest znana i agent ma dostęp do stanów wszystkich obiektów wewnątrz środowiska) wybiera akcje ruchu albo obsługi obiektów symulacji.

Na początku symulacji generowana jest restauracja w dowolnym momencie swojej aktywności - stoły i kuchnie mają losowo przydzielone stany, z którymi zaczynają symulację. Przy uruchomieniu gry część stołów będzie czekać na złożenie zamówień lub zabranie naczyń. Zadaniem kelnera będzie ustalenie strategii wyzerowania stanów stołów w jak najkrótszym czasie. Symulacja kończy się, gdy wszystkie stoły zostaną obsłużone do końca i nie będą miały naczyń do oddania, zamówień do złożenia ani posiłków do otrzymania.

Specyfikacja PEAS:

- miara wydajności: odwrotność czasu potrzebnego do wyzerowania symulacji
- środowisko: jeden stan restauracji wygenerowany pseudolosowo z ziarna lub parametrów pobranych z logu, zawierający puste kafelki, stoły, kuchnie i kelnera
- efektory: ruch kelnera o jedno pole w pionie lub poziomie, obsługa stolika, obsługa kuchni
- sensory: agent posiada tablet ze stanem restauracji.

Dokument przedstawia poziom zaawansowania projektu na dzień 26 marca 2019.

2 Układ projektu

Projekt składa się z następujących elementów:

1. folder **/images** zawiera wszystkie grafiki(sprite'y) użyte w projekcie;
2. folder **/scripts** zawiera wszystkie obiekty:
 - **/scripts/__init__.py**:
zawiera wszystkie globalne zmienne (parametry początkowe symulacji) i ustawienia graficzne "planszy".
 - **/scripts/dinning_table.py**:
ogólne informacje o obiekcie typu "stół", konstruktor oraz podłączenie odpowiedniej grafiki
 - **/scripts/furnance.py**:
ogólne informacje o obiekcie typu "kuchnia", konstruktor oraz podłączenie odpowiedniej grafiki. Klasa została wstępnie rozszerzona o listy z daniami (ten element może ulec zmianie podczas tworzenia zasad gry).

- **/scripts/matrix.py:**

Klasa przechowująca dane symulacji - rozumiemy przez to planszę gry. Jest to przestrzeń N^2 (N ustalone jako parametr początkowy symulacji). Każde miejsce (współrzędne) w macierzy odpowiada jednemu obiektowi (typów: kelner, kuchnia lub stół). Klasa ma metody wspomagające przetwarzanie zawartych w niej danych zgodnie z postulatami CRUD (wstawianie, odnajdywanie szukanych obiektów, modyfikację oraz usuwanie danych).

- **/scripts/waiter.py:**

Klasa kelnera posiada konstruktor i metody kontrolujące symulację. To w niej tworzymy stoły i piece oraz rozkładamy je po planszy.

Jest to znaczne ułatwienie, ponieważ główny blok gry za jednym wywołaniem tworzy wszystkie obiekty. Kelner posiada metody do poruszania się po planszy, czyli w naszym przypadku macierzy. Ponadto symulacja obsługuje podział na rundy - jedna runda to akcja kelnera i następujące po niej zmiany stanów obiektów w symulacji.

Jedyną dostępną akcją jest zmiana położenia kelnera w macierzy, a próba wejścia na stół lub kuchnię powoduje automatyczną obsługę obiektu według zadanych priorytetów czynności (zabranie naczyń, podanie posiłku, zebranie zamówienia). Priorytety są dostępne w dodatku do projektu.

3. **/main.py:**

Główny plik projektu odpowiedzialny za grafikę - okno graficzne Pygame i rysowanie sprite'ów. Zawiera procedurę początkową symulacji i zapis stanu symulacji do logu.

4. **/simulation_log.txt:**

Plik tekstowy zawierający informacje o stanie początkowym każdej utworzonej symulacji oraz czasie utworzenia. Służy do odtwarzania symulacji.

Projekt znajduje się pod adresem: <https://github.com/naapster/UberKelner>

3 Zasady obowiązujące symulację i agentów

Poniższy zestaw reguł jest przykładową propozycją, która ulegnie zmianie na dalszym etapie projektu:

1. Kelner:

- ma informacje o stanach wszystkich obiektów w symulacji;
- nie może poruszać się po stołach i kuchniach;
- musi wrócić do kuchni, aby złożyć zamówienie;
- wchodząc na stół lub kuchnię wyzwala automatyczną obsługę obiektu (zebranie naczyń, dań, złożenie zamówienia itp.) według zadanych priorytetów.

2. Stoły:

- zaczynają symulację z losowo wybranym stanem aktywności;
- mogą posiadać stany: oddania naczyń, chęci złożenia zamówienia lub oczekiwania na zamówienie;

3. Kuchnie:

- po zadanej liczbie rund od złożenia zamówienia produkują potrawę do dostarczenia na dany stolik;
- przechowują zamówienia z ich identyfikatorami, aby można było sprawdzić jakie zamówienie jest już na liście dań zrobionych;
- przygotowują zamówienia z listy dostarczonej przez kelnera;

Agent będzie agentem dążącym do minimalizacji czasu obsługi danego stanu symulacji.