

GameAI - Dokumentacja

1. Opis projektu:

GameAI to aplikacja która powstała z myślą o sztucznej inteligencji ale może też służyć jako gra zręcznościowa. Została ona dostosowana pod tym kątem, aby działać w różnych konfiguracjach map.

Mapa gry jest generowana w sposób losowy z plików tekstowych i składana jak klocki z warunkiem na dopasowanie wysokości wyjścia bloczka poprzedzającego i bloczka kolejnego. Długość mapy jest zależna od gracza za wyjątkiem trybu sztucznej inteligencji, w którym to długość mapy jest narzucana odgórnie (ale ciągle mapa jest generowana losowo). Pomimo losowego generowania mapy gracz może po porażce zagrać ponownie na poprzedniej mapie - mapa jest losowana tylko na początku gry, kolejne przejścia gry mogą być tworzone z tą samą mapą.

Gracz przedstawiony jako zielony kwadracik porusza się po mapie ze stałym wektorem prędkości oraz wpływa na niego stały podczas całej rozgrywki wektor grawitacji - wektory dwuwymiarowe. Gracz może za pomocą skoku i hover'a manipulować wektorem ruchu bloczka co w efekcie jest widoczne jako skok bądź unoszenie się.

Graficzny interfejs jest generowany w 60 klatkach na sekundę przy użyciu grida, który jest rozszerzoną wersją mapy i jest używany do wygładzenia ruchów bloczka (w gridzie każdy blok mapy jest dzielony siatką 120 na 120)

W trybie sztucznej inteligencji został użyty algorytm ewolucyjny. AI wykonuje kilka niezależnych przejść mapy z których wybierane są 3 najlepsze (kryterium wyboru jest długość mapy jaką udało się przejść AI) i na ich podstawie tworzone są kolejne generacje. Przez odsiewanie najlepszych jednostek kolejne generacje stają się coraz lepsze w przechodzeniu mapy. Dzięki zastosowaniu tego algorytmu AI po czasie nauczy się przechodzić daną mapę o dowolnej długości. Warto zaznaczyć że zmiana mapy wymaga rozpoczęcia procesu uczenia na nowo ponieważ proces uczenia jest ściśle powiązany z zadaną mu mapą.

2. Struktura projektu:

Projekt został podzielony na pakiety, aby poprawić czytelność i strukturę projektu:

- pakiet AI - zawiera wszystkie klasy biorące udział w implementacji algorytmu ewolucyjnego wykorzystywanego w trybie AI
- pakiet Map - zawiera wszystkie klasy związane bezpośrednio lub pośrednio z mapą lub gridem oraz pliki tekstowe które są używane do losowego generowania mapy
- pakiet Gui - zawiera wszystko związane z interfejsem graficznym użytkownika

3. Podział pracy:

Cyprian Neugebauer

- algorytm ewolucyjny,
- klasa określająca położenie na mapie,
- graficzny interfejs użytkownika gry,
- klasa definiująca gracza

Jakub Strojewski

- implementacja przycisków pojedynczej reakcji,
- gui menu startowego,
- mechanika hover,
- losowy generator mapy z plików,
- logika fizyki w poruszaniu oparta na wektorach

4. Co można zmienić/ulepszyć:

- zmiana - mechanika poruszania się gracza z wykorzystaniem float'ów - typ int wymaga użycia bardzo dużej ziarnistości aby delikatnie skorygować ruch bloczka, gra jest płynna ale przez to czas ładowania grida się wydłuża
- uzupełnienie - typing we wszystkich klasach w celu poprawy jakości i czytelności kodu
- ulepszenie - zmiana struktury, w której przetrzymywana jest mapa z listy pythonowej na macierz rzadką z scipy (csc_matrix lub dok_matrix), gdyż w aktualnej mapie przetrzymywana jest duża ilość zbędnych zer oznaczających brak przeszkody - powietrze
- zmiana - wykorzystanie dataclasses do przetrzymywania np. ustawień mapy, gracza
- ulepszenie - zmiana grafik na własne zamiast używania bibliotecznych kształtów
- ulepszenie - ewaluowanie kolejnych generacji w algorytmie ewolucyjnym na osobnych wątkach lub procesach, w celu zmniejszenia czasu oczekiwania