

Dokumentacja projektu

Cel i opis projektu

W ramach projektu miałem zaprojektować program umożliwiający symulację rozprzestrzeniania się choroby w przestrzeni dwuwymiarowej. Osoba jest reprezentowana jako koło poruszające się z losowymi wektorami ruchu.

Główne założenia

1. Tylko jeden obiekt jest zarażony na początku symulacji,
2. Zarażony po pewnym czasie przechodzi w stan choroby kiedy już nie może zarażać kolejnych osób,
3. Kule poruszają się z losowym wektorem prędkości początkowej,
4. Gdy zdrowa kula oraz zarażona zderzą się dochodzi do transmisji choroby
5. Zderzenia kul między sobą oraz z granicami przestrzeni są elastyczne,
6. Część kul na początku symulacji jest nieruchoma
7. Ilość wszystkich kul, kul nieruchomych, prędkość początkowa oraz okres pomiędzy zakażeniem się a okresem gdy już nie można zarażać innych podawane są przez użytkownika.
8. Sposób rozprzestrzeniania się choroby powinien zostać zwizualizowany na wykresie.

Struktura projektu

person.py

Zawiera klasę Person, klasa ta definiuje pojedynczy obiekt podczas symulacji reprezentowany wizualnie przez kulę. Zawiera on informacje na temat położenia obiektu w przestrzeni dwuwymiarowej, stan jego zdrowia czy wektor po jakim porusza się on we wspomnianej przestrzeni.

population.py

Zawiera klasę Population. Obiekt tej klasy reprezentuje zbiór wszystkich obiektów testowych podczas symulacji. Ponadto, klasa ta umożliwia wylosowanie takich pozycji w przestrzeni dwuwymiarowej by uniknąć nakładania się na siebie kul.

Klasa ta zawiera metody umożliwiające obsługę wszystkich zdarzeń do jakich może dochodzić pomiędzy obiektami klasy Person jak na przykład transmisja choroby czy zderzenia między sobą.

main.py

Główny trzon projektu, zawiera funkcje tworzące wizualną reprezentację symulacji za pomocą pygame. Ponadto wyposażona jest funkcje walidującą argumenty podane przez użytkownika oraz dzięki skorzystaniu z modułu matplotlib generuje wykres przebiegu choroby.

test_simulation.py

Plik python zawierający test wykorzystane podczas tworzenia projektu. Plik ten posiada zbiór testów ze wszystkich trzech plików pythona

icon.png

Zdjęcie wykorzystane w aplikacji, pojawia się ono przy odpaleniu programu w lewym górnym rogu. Samo zdjęcie pochodzi ze strony www.flaticon.com i wykonana została przez [Eucalyp](#).

Obsługa projektu

Wymagane moduły

Do uruchomienia projektu konieczne jest zainstalowanie kilku różnych modułów, które były konieczne przy tworzeniu projektu.

1. pygame - moduł ten posłużył mi do utworzenia wizualnej reprezentacji symulacji, to dzięki temu modułowi powstało menu końcowe, wizualizacja ruchu kul.
2. argparse oraz sys - moduł umożliwiający przekazywanie argumentów programowi z poziomu terminala oraz do pewnego stopnia ich walidację.
3. matplotlib - moduł ten posłużył mi do stworzenia wykresu reprezentującego rozprzestrzenianie się choroby podczas symulacji.
4. numpy - moduł ten posłużył mi do obliczenia wektorów obiektów po ich zderzeniu.

Instrukcja uruchomienia programu

Program uruchamia się poprzez podanie plikowi czterech parametrów. Wywołanie wygląda w następujący sposób:

```
main.py NO_OF_PPL NO_OF_IMMOB INC_PERIOD SPEED  
np.
```

```
main.py 20 10 6 3
```

Przy wywołaniu programu należy pamiętać o tym, że:

NO_OF_PPL reprezentuje ilość wszystkich obiektów klasy Person, powinno być wartością całkowitą z zakresu $<2 ; 100>$

NO_OF_IMMOB reprezentuje liczbę nieruchomych obiektów klasy Person podczas startu symulacji, powinno być wartością całkowitą z zakresu $<0 ; (NO_OF_PPL - 1)>$

INC_PERIOD - okres podany w sekundach reprezentujący czas po jakim, od momentu zakażenia, zarażony przejdzie w stan kiedy już nie rozprzestrzenia choroby.

SPEED - wartość całkowita z przedziału $<1;10>$, reprezentuje ona szybkość z jaką poruszają się kule po planszy, gdzie 1 oznacza bardzo wolno a 10 bardzo szybko.

Przebieg symulacji

Po uruchomieniu aplikacja przeprowadzi symulację, losowo wybrane kule zaczną się poruszać po planszy aż do momentu gdy ostatnia zarażona kula przejdzie w stan, w którym nie rozprzestrzenia już choroby. Wtedy symulacja kończy się a na ekranie aplikacji pojawiają się dwa przyciski. Jeden służący do opuszczenia programu oraz drugi umożliwiający pokazanie wykresu przedstawiającego rozprzestrzenianie się choroby.

Wnioski oraz przemyślenia

Problemy i przemyślenia

Projekt wykonuje wszystkie z zadanych w projekcie poleceń, niestety nie udało mi się stworzyć odpowiedniego menu umożliwiającego interaktywne wprowadzanie przez użytkownika danych wykorzystywanych przez program. Niestety gdy wybrałem moduł pygame do tworzenia gui nie byłem świadomy kilku z jego ograniczeń:

1. Brak możliwości stworzenia typowych dla na przykład html input boxów. Brak takowych postawił mnie przed ciężkim zadaniem stworzenia czegoś takiego na własną rękę co ostatecznie mnie przerosło przez co ostateczny projekt przyjmuje argumenty z terminala zamiast z GUI.
2. Problematiczna optymalizacja, pygame w ciągu każdego wywołania pętli rysuje ekran aplikacji, o ile stworzenie wizualizacji dzięki pygame okazało się dość proste, całkowicie ograniczyło mi to możliwość rozszerzenia projektu o dodatkowe menu gdyż przyciski również są tworzone w każdej pętli dlatego wraz z ich pojawieniem się drastycznie spada płynność symulacji (co jest widoczne na koniec działania programu gdy pojawiają się tylko dwa przyciski a już widoczny jest spadek klatek)
3. Podczas symulacji pojawia się nieprzyjemny błąd polegający na łączeniu się ze sobą kul, niestety nie udało mi się go naprawić a tylko ograniczyć jego występowanie poprzez zmniejszenie prędkości poruszania się kul co widocznie wpłynęło na częstotliwość tego problemu

Oprócz problemów z pygame pojawił się błąd związany z matplotlibem, przez co musiałem ograniczyć funkcjonalność ostatniego menu do jedynie wyświetlania wykresu a usunąć możliwość jego zapisywania, gdyż istnienie obu tych możliwości jednocześnie powodowało, że wykres pokazywał podwójną ilość kluczy w legendzie a przy późniejszych odpaleniach przycisku całkowicie wymazywał swoją zawartość.

Wnioski

Ostatecznie, gdybym miał podejść do projektu jeszcze raz, prawdopodobnie spróbowałbym najpierw poszukać innego rozwiązania niż pygame, który mimo wygody w implementacji, okazał się dość problematyczny przy dużej ilości poleceń na każdy tik czasu. Wydaje mi się jednak, że ostateczny efekt jest naprawdę dobry i przyjemny dla oka.

.