Spis treści

1 Dostępne funkcje

```
Initiator().create() -> SimFrame

Simulator().simulate(arg: SimFrame) -> SimFrame

SimFrame::get_particles() -> List[Particle]

Particle::get_position() -> (float, float)

Particle::get_velocity() -> (float, float)

ConfigMainpulator().read(arg:ConfigFields) -> str
```

2 Użycie

2.1 SimFrame

Na obiekcie typu SimFrame można użyć metody get.particles, która zwróci tablicę obiektów typu Particle, które będą odpowiadać wszystkim cząsteczkom i ich stanom w danym momencie czasu opisywanym przez SimFrame

2.2 Particle

2.2.1 get.position

Tą metode można wywołać na obiekcie typu Particle i zwróci pozycję cząsteczki jako dwójkę, gdzie pierwszy element to koordynat x a drugi y

2.2.2 get.velocity

Tą metode można wywołać na obiekcie typu Particle i zwróci prędkość cząsteczki jako dwójkę, gdzie pierwszy element to składowa x a druga y

2.3 ConfigManipulator

Służy do czytania z configu, posiada jedna publczną metodę - read. Przyjmuje ona jedno z mozliwości:

```
size # wielkosc boku pudelka w ktorym jest animacja
maxSpeed # maksymalna wypadkowa predkosc czasteczki
particleAmount # ilosc czasteczek
boxSize # wielkosc pojedynczej komorki - do liczenia mikrostanow, entropi
time # calkowita ilosc kalatek do wyrenderowania
timeDelta # dyskretny krok czasu
init_state_file # plik  z ustawieniami poczatkowymi
particleSize # wielkosc czasteczki
maximalDistanceAsCollision # do ustawien silnika
maximalTimeDeltaAsColliding # jw
```

Używa się tego tak:

```
ConfigManipulator().read(ConfigFields.cos_z_powyzszych)
```

2.4 Plik simulation.sim

Jest to zapicklowana tablica z kolejnymi SimFrame'ami, jest ich tam tyle na ile został ustawiony time w pliku *config.ini*. Aby ją odpicklować można użyć metody *pickle.load*(*nazwapliku*)

3 Testowanie

3.1 Uruchamianie

```
python3 -s # tworzy plik simulation.sim
python3 -r # odtwarza plik konfiguracyjny
python3 -f # wczytuje poczatkowe polozenie czasteczek z pliku
python3 -p # tworzy wykres do zadania 1
python3 -v # zadanie 2
python3 -t # tworzy wykres do prawdopodobienstwa termodynamicznego
python3 -e # tworzy wykres z entropia
python3 -h # ogolny help w bashu
```

3.2 config.ini

Można sobie w nim wszystko pozmieniać, a jak coś pójdzie nie tak to można użyć flagi r i wszystko się przywróci do stanu początkowego

4 TODO

Wszystkie klasy po wywołaniu funkcji evaluate() muszą zczytać z pliku simulate.sim dane i stworzyć animacje

DrawVelocity Tworzy animację położenia cząsteczek w przestrzeni VxVy, gdzies na wykresie powinna być informacja jaki numer klatki się aktualnie wyświetla i zapisuje ją do pliku .mp4

DrawTHPrb Tworzy wykres zmian prawdopodobienstwa i zapisuje go do pliku

DrawEntropy Tworzy wykres zmian entropii i zapisuje go do pliku

5 Przyblizony sposób używania

1. Uzytkownik tworzy symulacje flagą -s
2. Uruchamia do woli różne inne dostępne flagi
3. Tworzy inna symulację flagą -s