

README.md

# Symulacja Życia

## Budowa projektu

```
mkdir symulacjazycia  
cd symulacjazycia  
cmake ..  
cmake --build . --config Release
```

## Wykonywanie testów (po budowie)

```
ctest -C Release
```

## Uruchomianie Projektu

### Losowe rozłożenie ekosystemu

```
./SymulacjaZycia
```

### Rozłożenie ekosystemu pobrane z "plik.txt"

```
./SymulacjaZycia plik.txt
```

# Struktury danych

## Rows - tablica dwuwymiarowa

**plik: Rows.h**

Własna implementacja tablicy "dwuwymiarowej".

Rows przechowuje dane w tablicy jednowymiarowej i eksponuje funkcje i przeciążenia operatorów, które pomagają użytkownikowi traktować tą klasę jak tablicę dwuwymiarową zachowując cache locality.

Aby uzyskać jeden rząd tablicy "dwuwymiarowej" należy ponownie użyć operatora[].

## Row - jeden rząd w tablicy dwuwymiarowej

**plik: Rows.h**

Rząd uzyskuje się po użyciu operatora[] na klasie Rows<T>. Klasa ta eksponuje wycinek jednowymiarowej tablicy (rząd), który ma wielkość RowLen.

Aby uzyskać wartość T należy ponownie użyć operatora[]

## Vicinity - Sąsiedztwo

**plik: Vicinity.h**

Sąsiedztwo pobiera nisze otaczające wskazaną niszę.

Klasa zwraca szczególną uwagę aby nie pobrać informacji nie należących do instancji Rows<T>.

Próba dostępu do pozycji spoza Rows<T> skutkuje w std::range\_error.

**std::optional**

**plik: <optional>**

Klasa `optional` może zawierać `T` lub `std::nullopt`. Używana jest w `Cell` ponieważ jej konstrukcja nie wymaga kolejnej alokacji na stacku. Dodatkowo lepiej wskazuje intencję parametru `m_contents` niż wskaźnik.

## Klasy

### RandomGenerator - Generator liczb pseudolosowych

**plik: RandomGenerator.h**

Generator to klasa singleton, która posiada tylko i wyłącznie jedną instancję klasy `std::random_device` jako parametr. `RandomGenerator` eksponuje funkcje związane z generacją funkcji losowych.

### Simulation - Symulacja

**plik: SymulacjaZycia.cpp**

Symulacja zawiera `Grid` i zarządza działaniem symulacji. Szczególnie rozpoczęciem i zakończeniem symulacji. Dodatkowo przekazuje stan ekosystemu użytkownikowi programu.

### Grid - Siatka

**plik: Grid.h**

Siatka zawiera tablicę dwuwymiarową `Rows`, która zawiera instancje `Cell`. Siatka przekazuje do `Cell` jej `Vicinity`.

## Cell - Nisza

plik: Grid.h

Nisza zawiera `Organism` lub `nic(std::nullopt_t)`. Nisza jest główną klasą która w pełni zarządza jej instancją `Organism`. Na podstawie podanego `Vicinity Cell` wykonuje krok symulacji.

## Organism - Organizm

Organizm to klasa, która zawiera dane organizmu i eksponuje je w sposób przystosowany do działania Symulacji. Modyfikacja i przeglądanie danych organizmu jest ograniczone za pomocą funkcji getter i setter np. organizm może zostać postarzony o tylko jeden krok symulacji i nakarmiony tylko jednym pokarmem.