



Specyfikacja wymagań aplikacji „The Game of Life”

Kacper Piwiński

Wydział Fizyki Technicznej i Matematyki Stosowanej

Spis treści

1. Streszczenie.....	3
2. Ogólny opis.....	3
2.1 Relacje do bieżących projektów.....	3
2.2 Relacje do wcześniejszych i następnych projektów.....	3
2.3 Funkcje i cele.....	3
2.4 Ustalenia dotyczące środowiska.....	3
2.5 Relacje do innych systemów.....	3
2.6 Ogólne ograniczenia.....	3
2.7 Opis modelu.....	3
3. Specyficzne wymagania.....	4
3.1 Wymagania dotyczące funkcji systemu.....	4
3.2 Wymagania dotyczące wydajności systemu.....	4
3.3 Wymagania dotyczące zewnętrznych interfejsów.....	4
3.4 Wymagania dotyczące wymaganych operacji.....	4
3.5 Wymagania dotyczące wymaganych zasobów.....	4
3.6 Wymagania dotyczące sposobów weryfikacji.....	4
3.7 Wymagania dotyczące sposobów testowania.....	4
3.8 Wymagania dotyczące dokumentacji.....	4
3.9 Wymagania dotyczące ochrony.....	4
3.10 Wymagania dotyczące przenośności.....	5
3.11 Wymagania dotyczące jakości.....	5
3.12 Wymagania dotyczące niezawodności.....	5
3.13 Wymagania dotyczące pielęgnacyjności.....	5
3.14 Wymagania dotyczące bezpieczeństwa.....	5
4. Harmonogram prac nad projektem.....	6

1. Streszczenie

„The Game of Life” (inaczej: „Gra w życie”) to symulacja automatu komórkowego, zaproponowanego przez Johna Conwaya w 1970 roku.

Jest to gra typu „zero”, gdzie na dalszy przebieg gry wpływ ma tylko stan początkowy, a działanie użytkownika nie jest potrzebne – można obserwować na planszy ewolucję „życia” lub tworzyć takie plansze początkowe, by w późniejszym rozwoju gry można było otrzymywać konkretne wzory ułożenia komórek.

2. Ogólny opis

2.1 Relacje do bieżących projektów

Nie dotyczy.

2.2 Relacje do wcześniejszych i następnych projektów.

Nie dotyczy.

2.3 Funkcje i cele

Celem programu jest zasymulowanie działania zasad przedstawionych przez Conwaya z wykorzystaniem dostępnych narzędzi współczesnego programowania. Jedną z jego głównych funkcji jest pokazanie, jak z użyciem prostych reguł początkowych można otrzymać skomplikowane wyniki, a także – jako druga funkcja – dostarczenie użytkownikowi rozrywki poprzez możliwość obserwowania ewolucji świata gry na ekranie monitora.

2.4 Ustalenia dotyczące środowiska

Podstawowym systemem operacyjnym, na którym będą przeprowadzane testy jest środowisko GNU/Linux, jednak założeniem projektu jest, by wszystkie komponenty wykorzystywane w programie były multiplatformowe.

2.5 Relacje do innych systemów

Program może być napisany z powodu łatwości tworzenia jakie dają nam biblioteki takie jak SDL i Allegro.

2.6 Ogólne ograniczenia

Podstawowym ograniczeniem ma być wydajność systemu komputerowego na którym działać będzie symulacja.

2.7 Opis modelu

Oprogramowanie jest tworzone zgodnie z modelem ewolucyjnym programowania.

3. Specyficzne wymagania

Pojęcia:

- Komórka - podstawowy budulec przestrzeni. Może być albo żywa albo martwa.
- Śmierć - przejście komórki ze stanu żywego do stanu martwego.
- Narodziny - przejście komórki ze stanu martwego do żywego.
- Czas - podzielony na pojedyncze momenty w których odbywa się symulacja zasad rządzących grą.

Zasady

Świat gry składa się z dwuwymiarowej, przestrzeni, podzielonej na „komórki”. Każda komórka wchodzi w interakcję ze swoimi ośmioma sąsiadami – czyli komórkami, które stykają się z daną komórką pionowo, poziomo i na skos. W czasie gry, świat podlega zmianom zgodnie z poniższymi zasadami.

- Każda żyjąca komórka, która ma mniej niż dwóch sąsiadów, umiera z „samotności”.
- Każda żyjąca komórka, która ma dwóch lub trzech sąsiadów, przeżywa.
- Każda żyjąca komórka, która ma więcej niż trzech sąsiadów, umiera z „przeludnienia”.
- Każda umarła komórka, która ma równo trzech sąsiadów, „rodzi się” i staje się żywa.

3.1 Wymagania dotyczące funkcji systemu

Jedną z rzeczy którą chce osiągnąć jest maksymalny widoczny rozmiar symulacji tj. jedna komórka przestrzeni to jeden piksel na monitorze. Da to największą możliwą przestrzeń dla symulacji.

Drugą rzeczą jest zakrzywienie przestrzeni w taki sposób by życie na komórkach nie natrafiało na pustą przestrzeń przy krawędziach okna (monitora), tylko by przestrzeń była zakrzywiona w taki sposób by życie mogło przejść z górnej krawędzi do dolnej i odwrotnie. I analogicznie przy lewej i prawej. Bez żadnych widocznych nieścisłości.

3.2 Wymagania dotyczące wydajności systemu

Aplikacja ma wykorzystywać cały pojedynczy rdzeń procesora

3.3 Wymagania dotyczące zewnętrznych interfejsów

Wymagana klawiatura lub mysz do wyboru początkowego układu przestrzeni.

3.4 Wymagania dotyczące wymaganych operacji

Użytkownik ma za zadanie wybrać początkowy stan przestrzeni na której ma odbyć się symulacja.

3.5 Wymagania dotyczące wymaganych zasobów

Nie dotyczy.

3.6 Wymagania dotyczące sposobów weryfikacji

Komputer na którym zainstalowany jest system GNU/Linux z jądrem w wersji >4.0.

3.7 Wymagania dotyczące sposobów testowania

Zakresem działania programu jest implementacja zasad gry The Game of Life stworzonej przez Johna Conwaya, oraz pokazywanie przebiegu symulacji.

3.8 Wymagania dotyczące dokumentacji

Nie dotyczy.

3.9 Wymagania dotyczące ochrony

Nie dotyczy.

3.10 Wymagania dotyczące przenośności

Nie dotyczy.

3.11 Wymagania dotyczące jakości

Założenia:

- Wykorzystanie w programie tylko komponentów które nie ograniczają programu do uruchomienia tylko na jednej platformie.
- Sprowadzenie symulacji do poziomu jedna komórka przestrzeni- jeden piksel na monitorze.
- Zakrzywienie przestrzeni, brak krawędzi przestrzeni.

3.12 Wymagania dotyczące niezawodności

Program powinien wyświetlać stosowne komunikaty.

3.13 Wymagania dotyczące pielęgnacyjności

Nie dotyczy.

3.14 Wymagania dotyczące bezpieczeństwa

Nie dotyczy.

4. Harmonogram prac nad projektem

- 29.11.2016 – stworzenie podstawowej funkcji „life”, dot. liczenia życia względem czasu;
- 06.12.2016 – wprowadzenie obsługi wejścia (input), wykorzystujący nowy wątek (thread);
- 13.12.2016 – pierwsze testy z wykorzystaniem losowych, początkowych stanów przestrzeni;
- 20.12.2016 – wyszukiwanie błędów i ich poprawa;
- 10.01.2017 – zaimplementowanie „zakrzywienia przestrzeni”;
- 17.01.2017 – obsługa argumentów podawanych z linii poleceń podczas uruchamiania programu;
- 24.01.2017 – zaimplementowanie predefiniowanych stanów przestrzeni;