Łukasz Topolski

Snake game

Oficjalna dokumentacja projektu

Spis treści

[2 Opis Projektu 3](#_Toc30706389)

[2.1 Cel projektu 3](#_Toc30706390)

[2.2 Zasady gry 3](#_Toc30706391)

[2.3 Wymagania projektu 3](#_Toc30706392)

[3 Struktura projektu 4](#_Toc30706393)

[3.1 Podział ogólny 4](#_Toc30706394)

[3.2 Struktura modułów i klas 4](#_Toc30706395)

[3.3 Schemat działania 4](#_Toc30706396)

[3.4 Użyte biblioteki 4](#_Toc30706397)

[3.5 Najistotniejsze funkcje 5](#_Toc30706398)

[3.6 Algorytmika 5](#_Toc30706399)

[3.6.1 Faza wstępna 5](#_Toc30706400)

[3.6.2 Faza gry 5](#_Toc30706401)

[4 Instrukcja użytkowa 7](#_Toc30706402)

[4.1 Wstęp 7](#_Toc30706403)

[4.2 Wprowadzenie danych 7](#_Toc30706404)

[4.3 Poprawność danych 7](#_Toc30706405)

[4.3.1 Plansza 7](#_Toc30706406)

[4.3.2 Prędkość 7](#_Toc30706407)

[4.3.3 Prawdopodobieństwo 7](#_Toc30706408)

[4.3.4 Tryb gry 7](#_Toc30706409)

[4.4 Gra 8](#_Toc30706410)

[4.5 Koniec gry 8](#_Toc30706411)

[5 Testy Jednostkowe 9](#_Toc30706412)

[6 Podsumowanie 9](#_Toc30706413)

# Opis Projektu

## Cel projektu

Projekt polega na realizacji w języku Python popularnej gry zwanej The Snake Game.

## Zasady gry

* Gra odbywa się na prostokątnej planszy o z góry ustalonych wymiarach
* Na planszy znajdują się przeszkody (ściany, oraz obiekty wewnątrz planszy), wąż i jedzenie
* Wąż jest reprezentowany na planszy za pomocą ciągu odp. znaków
* Wąż porusza się z ustalonym przez użytkownika interwałem
* Wąż może poruszać się w czterech kierunkach, jest on sterowany przez użytkownika, poprzez zmianę kierunku głowy
* Wąż zjada losowo generowane obiekty jedzenia, przez co staje się większy
* Gra kończy się jeśli gracz uderzy głową w ciało węża, lub przeszkodę
* W przypadku braku przeszkody na krawędzi planszy wąż się zawija – wychodzi z przeciwległego końca planszy (plansza musi być wtedy symetryczna wzdłuż odp. osi)

## Wymagania projektu

* Gra ma interfejs konsolowy (2D w terminalu)
* Użytkownik powinien mieć możliwość:
  + wczytania planszy z pliku
  + ustawienia szybkości poruszania się węża
  + ustawienia prawdopodobieństwa generacji jedzenia
  + ew. przedmioty pozwalające na przenikanie przez przeszkody albo samego siebie

# Struktura projektu

## Podział ogólny

Projekt został podzielony na kilka części:

1. Obsługa planszy i wężyka: *board.py, snake.py*
2. Przedstawienie planszy na ekranie termianala: *screen.py*
3. Przeprowadzenie gry: *game.py*
4. Uruchomienie gry: *play\_game.py*
5. Testy jednostkowe: *test\_game.py. test\_snake.py, test.board.py*

Również w folderze projektowym znajdują się przykładowe plansze do gry (*Example.txt* – prawidłowo sformatowana oraz *WrongBoard1.txt, WrongBoard2.txt, WrongBoard3.txt, WrongBoard4.txt* – źle sformatowane)

## Struktura modułów i klas

## Schemat działania

1. Uruchom plik *play\_game.py*
2. Wprowadź potrzebne argumenty
3. Gra się rozpoczyna
4. Koniec gry

## Użyte biblioteki

* curses
* random
* re
* time

## Najistotniejsze funkcje

1. **board.generate\_food(**prob=”1”**)** – w zależności od wprowadzonego prawdopodobieństwa albo zwraca współrzędne jedzenia albo zwraca None
2. **snake.move(**direction**)** – w zależności od podanego kierunku wstawia do listy koordynatów wężyka współrzędne nowej głowy
3. **game.play\_game()** – przebieg samej gry (czyt. Algorytmika, Faza gry)

## Algorytmika

### Faza wstępna

Użytkownik po uruchomieniu pliku *play\_game.py* zostanie poproszony o wpisanie danych.

Następnie sprawdzana jest ilość argumentów oraz ich poprawność. Jeśli któraś z danych jest nieprawidłowa lub ilość tych danych jest zła, to wtedy zostanie wypisana wiadomość na temat tego błędu oraz użytkownik zostanie poproszony jeszcze raz o wprowadzenie argumentów.

Jeśli wszystko się zgadza, to gra się rozpoczyna

### Faza gry

W tym momencie na ekranie terminala pojawia się wgrana plansza z pliku tekstowego, a w niej wężyk (zawsze znajduje się on w drugim wierszu oraz w kolumnach 2, 3 i 4) oraz pierwsze wygenerowane jedzenie. Początkowo wężyk ma skierowaną głowę w prawą stronę.

Wężyk zaczyna się przemieszczać. Ruch wężyka polega na tym, że najpierw są wprowadzane do listy koordynatów wężyka współrzędne nowej głowy. Jeśli te współrzędne są takie same, jak współrzędne jedzenia, to wtedy (w zależności od wprowadzonego prawdopodobieństwa) jest (losowo) generowane nowe jedzenie. W przeciwnym wypadku ogon wężyka zostaje usunięty. Następnie zostają wyświetlone zmiany na planszy. W kolejnych przejściach pętli sprawdzane jest czy jedzenie zostało wygenerowane. Jeśli nie, to następuje kolejna próba wygenerowania jedzenia.

Regulacja prędkości wężyka następuje poprzez usypianie programu (w pętli) na okres czasu, który zależy od podanej prędkości (zależność 1/prędkość)

Gra trwa do momentu, w którym wężyk uderzy głową w przeszkodę albo w samego siebie. Jeśli gracz spróbuje obrócić głowę o 180°, wtedy gra również się kończy (głową uderzy w pozostałą część ciała).

Po porażce zostaje wyświetlony napis „Game over!” oraz wynik gracza, który jest liczony jako ilość zjedzonego jedzenia (inaczej: długość wężyka pomniejszona o 3). Po wciśnięciu dowolnego przycisku stan termiala przed rozpoczęciem gry zostaje przywrócony.

Jeśli gracz wybrał specjalny tryb gry, to po osiągnięciu wyniku, którego reszta z dzielenia przez 5 jest równa 4, zostaje wygenerowane specjalne jedzenie. Gdy wężyk je zje, to powiększy się i przejdzie w specjalny tryb, w którym może „przenikać” przez przeszkody i samego siebie. Ten tryb trwa przez kolejne 9 kroków wężyka. Potem użytkownik może „uciec od śmierci” (jeśli jakaś część ciała, oprócz głowy, jest w przeszkodzie, to gra się nie kończy) Kolejne specjalne jedzenia są generowane co piąte wydłużenie wężyka(reszta z dzielenia wyniku przez 4).

# Instrukcja użytkowa

## Wstęp

Aby uruchomić grę, należy w terminalu przejść do folderu z grą i wpisać *python play\_game.py*

## Wprowadzenie danych

W terminalu wyświetli się wiadomość „Enter path to board, snake’s speed, probability of generating food and game mode (normal or special)”. Należy teraz wprowadzić ścieżkę do pliku tekstowego z planszą (jeśli ten plik znajduje się w tym samym folderze, to wystarczy jego nazwa), prędkość gry, prawdopodobieństwo generowania jedzenia (co każdy krok jeśli nie zostało wcześniej wygenerowane) oraz tryb gry (special lub normal). Jeśli coś zostało źle wprowadzone (np. za mało danych), to program wypisze informację o błędzie i poprosi ponownie o wprowadzenie danych.

Dane powinny być zapisane w cudzysłowie, np.:

„Example.txt 5 0.5 normal”

## Poprawność danych

### Plansza

Prawidłowo sformatowana plansza powinna mieć 10 wierszy i 25 kolumn. Jedynymi dozwolonymi znakami są „#” (hasz) oraz ‘’ ‘’ (spacja). Jeśli w planszy ma być pusty wiersz, to należy postawić znak spacji 25 razy. W sytuacji, gdy któreś z powyższych założeń nie jest spełnione, to program informuje, że jest jakaś nieprawidłowość z planszą. Przykładowa prawidłowa plansza znajduje się w pliku *Example.txt*

### Prędkość

Służy regulacji szybkości ruchu wężyka. Powinna to być liczba dodatnia zapisana w postaci dziesiętnej (oczywiście może też to być liczba naturalna).

### Prawdopodobieństwo

Służy regulacji szansy na wygenerowanie jedzenia po powiększeniu wężyka. Próba wygenerowania jedzenia następuje co krok wężyka (jeśli nie ma jedzenia na planszy). Ta dana powinna być liczbą zapisaną w postaci dziesiętnej z przedziału (0, 1] (jedynkę zapisujemy jako liczba naturalna)

### Tryb gry

Do wyboru są dwa tryby gry: *normal* lub *special.* W drugim z nich jest możliwość przenikania przez przeszkody, a w pierwszym gra przebiega normalnie.

## Gra

Następnie pojawi się w oknie terminala plansza wgrana z pliku tekstowego wraz z wężykiem(ciąg znaków S), który pojawia się **zawsze w wierszu pierwszym oraz w kolumnach: druga, trzecia i czwarta. Początkowo wężyk porusza się w prawą stronę** (Warto mieć to na uwadze przy tworzeniu planszy), oraz z losowo wygenerowanym jedzeniem (znak A) . Za pomocą strzałek sterujemy głową węża. Gdy wąż zje jedzenie, to powiększy się o jedną jednostkę. Gra kończy się, gdy wąż uderzy głową w przeszkodę lub w samego siebie. Jeśli na krawędziach planszy nie ma przeszkód, to wąż może przejść przez kraniec planszy i wyjść z przeciwległej krawędzi.

Jeśli został wybrany tryb specjalny, to po zjedzeniu 4 jedzonek zostaje wygenerowane specjalne jedzenie (znak E). Gdy wąż je zje, to powiększy się i będzie mógł przenikać przez wszelkie przeszkody przez kolejne 9 kroków. Następnie wężyk może uciec z przeszkody, jeśli nie przeszedł całym ciałem przez nią. Kolejne specjalne jedzenia są generowane co piąte powiększenie wężyka. W sytuacji, gdy nowe specjalne jedzenie powinno się pojawić, a wcześniejsze nie zostało zjedzone, to nowe nie pojawi się (utrata okazji do przejścia w tryb przenikania). Gra kończy się tak samo, jak w normalnym trybie.

## Koniec gry

Wtedy na ekranie wyświetla się komunikat „Game Over” oraz osiągnięty wynik. Aby wyjść z gry, należy wcisnąć dowolny przycisk.

# Testy Jednostkowe

Obraz zawierający tekst

Opis wygenerowany automatycznieProjekt zawiera testy jednostkowe sprawdzające atrybuty klas całego programu. Również sprawdzane są wszystkie możliwe wyjątki.

Rysunek: Przeprowadzone testy jednostkowe

# Podsumowanie

Podsumowując, udało mi się zrealizować wszystkie założenia początkowe projektu. Największą trudność miałem ze sprawieniem, żeby program nie czekał na wciśnięcie przez użytkownika klawisza. Jednak dzięki biblioteki curses rozwiązałem ten problem. Także dzięki niej mogłem wykorzystać klawisze strzałek do sterowania wężem.