**--Instrukcja użytkownika--**

**Sposób gry:**

Grę uruchamiamy w terminalu przez Ubuntu komendą „python3 run\_snake\_game.py”. Następnie wybieramy czy chcemy zmienić ustawienia gry. Klikając Enter bez wpisywania ‘y’ lub ‘n’ grę uruchomimy bez zmiany ustawień.

Sterujemy strzałkami, po przytrzymaniu strzałek w danym kierunku wąż chwilowo przyspiesza do momentu w którym puścimy strzałkę. Klikając strzałkę w kierunku przeciwnym niż się poruszamy, przegrywamy, bo wąż zjada sam siebie. Aby wyjść z gry należy kliknąć „Q”.

Wąż może być w zwykłym trybie w którym ginie po wejściu na pole „#”, może być również w trybie ghost, w którym przenika przez te pole ale tylko w zadanym w ustawieniach czasie, później znów staje się śmiertelny.

W dolnym oknie odliczane są ostatnie 3 sekundy trybu Ghost. Obiekt Ghost pojawia się losowo i nie znika, czas trybu Ghost zmienić można w ustawieniach.

Zwykłe jedzenie pojawia się w losowym miejscu, zawsze jest jedno na całej mapie – po zjedzeniu jednego pojawia się nowe. Super fruits i poisoned fruits pojawiają się i znikają w losowych momentach, prawdopodobieństwo pojawienia ustawić można na początku gry.

**Mapy:**

Mapy możemy obejrzeć w folderze „maps”. Można tworzyć mapę – w folderze maps znajduje się plik ‘snake\_map\_custom.txt”, który możemy modyfikować dodając ‘#’ – ściany i ustalając pozycje węża -stawiając jeden „+” na całej mapie.

Należy pamiętać że bariery mapy muszą pozostać niezmienne, więc dodając jakiś znak musimy korygować pozycję bariery usuwając znak pusty w danej linii. Jeśli bariery zostaną jednak zmienione, prawidłowy ich układ można sprawdzić w pliku z inną mapą.

Aby wybrać mapę, na której chcemy grać należy podczas uruchomienia gry wybrać zmianę opcji i wpisać numer mapy od 1 do 6 lub „custom” jeśli chcemy uruchomić własną mapę.

Jeśli w mapie wystąpi błąd np. niepoprawny znak lub przesunięta granica, wyświetlony zostanie komunikat z koordynatami miejsca zauważenia błędu. Jeśli będzie więcej niż jeden „+” na mapie, również wyświetli się komunikat.

**--Kod—**

Sam wąż zrealizowany jest jako lista krotek, z koordynatami odpowiadającymi każdej „kondygnacji” węża. Przesuwając się usuwana jest krotka odpowiadająca ogonowi i dodawana jest nowa krotka (nowa głowa) w zadanym kierunku. Jako listy krotek zrealizowane są również inne obiekty „specjalne”. Funkcja move() w wężu oprócz poruszania, sprawdza czy krotka nowej głowy nie zawiera się w liście obiektów specjalnych. Jeśli tak jest, odpowiednio na to reaguje. Mapa ładowana jest poprzez funkcje w swojej klasie Map. Ustawienia ładowane i zapisywane są poprzez funkcje w klasie Settings. Obiekty wyświetlane są poprzez funkcję display\_object\_list() w klasie Game.

**Plik run\_snake\_game.py:**

Plik z dwoma pętlami – pętlą gry i pętlą „game over”. Są tutaj również zdefiniowane funkcje, które odpowiadają za zmianę ustawień wyświetlania terminalu oraz tworzenie okien gry:

- okno pierwsze (win1) – wyświetlana jest plansza gry

- okno drugie (win2) – wyświetlane są informację o grze – opis obiektów, które mogą pojawić się na mapie; odliczanie ostatnich 3 sekund trybu ghost\_mode

**Plik game.py:**

Zdefiniowana klasa Game() wraz z jej atrybutami oraz funkcje:

- display\_object\_list() – wyświetla zadany znak dla każdej krotki w liście określającej pozycję znaku - format (x, y)

-spawn\_object() – losowo wybiera wolny punkt na mapie, można podać prawdopodobieństwo w jakim obiekt się pojawi np. podając liczbę 400 w każdej pętli gry mamy szansę 1 do 400

**Plik snake.py:**

Zdefiniowana klasa Snake() wraz z jej atrybutami oraz funkcje:

- choose\_sign() – wybiera znak węża w zależności od tego czy ghost\_mode jest włączony

- new\_head() – oblicza koordynaty nowej głowy w zależności od wciśniętego przycisku

- count\_score() – liczy punkty węża

- move() – funkcja odpowiadająca za przemieszczanie węża, dodaje nową głowę, usuwa ogon.

Sprawdza czy nie wchodzimy na pole „specjalne” jeśli tak odpowiednio reaguje. Wejście:

+ na pole fruit sprawia że ogon węża nie zostaje usuwany – zwiększa się długość o 1;

+ na pole super\_fruit, że ogon nie zostaje usunięty w dwóch kolejnych pętlach;

+ na pole poisoned\_fruit, że ogon tacimy podwójnie – zamiast urosnąć wąż maleje;

+ na pole ghost\_coords, że uruchamia się ghost\_mode;

+ na pole snake.coords lub walls, że gra kończy się

**Plik settings.py:**

Zawiera klasę Settings() i jej atrybuty oraz funkcje:

- read\_from\_json() – odpowiada za odczytywanie ustawień z pliku settings.json

- load() – odpowiada za aktualizowanie atrybutów zgodnie z wartościami z pliku settings.json

- write\_to\_json() – odpowiada za zapis podanych ustawień do settings.json

**Plik load\_map.py:**

Zawiera klasę Map() i jej atrybuty oraz funkcje:

- path\_extension() - sprawdzającą ścieżkę pliku

- load() – „ładuje mapę” poprzez ustalenie pozycji węża oraz ścian

**Pozostałe pliki i foldery:**

- maps – folder z mapami

- test\_maps - folder z mapami używanymi w „test\_load\_map.py”

- test\_settings – folder z ustawieniami używanymi w „test\_load\_settings.py”

- settings.json – plik z ustawieniami gry

- test\_game.py; test\_settings.py; test\_load\_map.py; test\_snake.py – testy jednostkowe