Mazur Szymon, Gl05, 134129

Saper

Temat projektu

Tematem projektu jest gra Saper („Minesweeper”), klasyczna gra jednoosobowa, polegająca na odkrywaniu poszczególnych pól na planszy w taki sposób, aby nie natrafić na ukrytą minę. Na każdym z odkrytych pól napisana zostaje liczba min na polach sąsiadujących z danym polem. Nieodkryte pola można również oznaczyć flagą, dzięki czemu jest zabezpieczone przed odsłonięciem, dzięki czemu można oznaczać miejsca, w których wie się o minie.

Funkcjonalność

1. Gra rozpoczyna się od wpisania przez gracza wymiarów planszy oraz ilości min, które mają zostać na niej ukryte oraz kliknięcia przycisku Enter bądź „Start!”
2. W prawym górnym rogu menu znajduje się licznik oznaczonych pól oraz ilość min, które są w grze. Dodatkowo gracz może włączyć możliwość czyszczenia pól wejściowych po rozpoczęciu rozgrywki.
3. Lewym przyciskiem gracz odkrywa nieodkryte dotychczas pole.
   Prawym przyciskiem gracz zmienia nieodkryte pole w oznaczone flagą i zablokowane przed przypadkowym kliknięciem, sugerujące, iż „tu jest mina”.
   Po ponownym przyciśnięciu prawym kliku w pole z flagą, zmienia się ono w pole z pytajnikiem, oznaczające „tu może być mina”. Kolejny klik resetuje wygląd pola.
4. Gra kończy się, gdy wszystkie miny zostały oznaczone flagą (i żadne dodatkowe pole) lub wszystkie pola nieposiadające miny zostały odkryte.

Moduł main

Rdzeń programu , w którym zawarte są niżej wymienione funkcje.
Zawiera inicjalizację modułów help.py, minesweeperExceptions.py oraz pngs.py.
Na początku odbywa się przygotowanie Menu oraz przycisku „Start!” oraz rozmieszczenie ich w odpowiednich ramkach (tkinter.Frame).

Każde wywołanie pętli zawiera:

- wykrycie wciśnięcia przycisku „Start!” lub klawisza Enter oraz związaną z tym funkcję rozpoczynającą nową grę

- wykrycie wciśnięcia litery oraz sprawdzenie, czy użytkownik wpisuje kod ‘xyzzy’

- wykrycie kliknięcia lewym bądź prawym przyciskiem myszy w jedno z pól

Funkcja newGame

Wywoływana przez naciśnięcie przycisku „Start!” lub klawisza Enter. Zaczyna się od walidacji danych wejściowych pobranych z tkinter.Entry, takich jak n, m oraz ilość bomb.
Występują próby wyłapania jednego z wyjątków z własnej klasy wyjątku, z modułu minesweeperExceptions.py. Sprawdzane są wszystkie oczekiwane przypadki błędów (któraś z współrzędnych mniejsza od 2 lub większa od 15, ilość min spoza dostępnego zakresu dla danych wymiarów pola, któreś pole pozostawione puste). Następnie przygotowuje nową planszę do gry, przy wykorzystaniu np. list comprehension.

Funkcja clickedField

Reaguje na wciśnięcie danego pola przez użytkownika lewym przyciskiem myszy, sprawdzając, czy zgodnie z obecnym stanem pola ma zajść jakaś akcja oraz czy przypadkiem nie kliknięto w pole z miną. Np. klikając w nieodkryte pole wywoływana zostaje funkcja checkForBombs dla danego pola. Klikając zaś np. w pole z flagą nie występuje żadna reakcja. Trafiając na bombę funkcja ta może wywołać funkcję lostGame

Funkcja rightClick

Reaguje na wciśnięcie danego pola prawym przyciskiem myszy, sprawdzając, czy zgodnie z ocenym stanem pola ma zajść jakaś akcja. Np. dla nieodkrytego pola następuje jego oflagowanie. Funkcja ta zlicza również liczbę oflagowanych pól oraz prawidłowo oflagowanych pól i może wywołać funkcję wonGame.

Funkcja checkForBombs

Sprawdza obecność bomb w sąsiedztwie danego pola. Jeśli jest zerowa, wykonuje się dla każdego nieodkrytego pola w sąsiedztwie. Funkcja zlicza liczbę odkrytych pól, dzięki czemu może wywołać funkcję wonGame.

Funkcja xyzzyCheck

Wywołana zostaje przy wykryciu wciśnięcia klawisza z klawiatury. Po sprawdzeniu, czy jest to litera, sprawdzane jest, czy użytkownik wpisuje tzw. xyzzy code, który w moim przypadku zmienia wygląd pól z bombami na ciemniejsze.
Wywołana zostaje przy pomocy root.bind(„<Key>”, xyzzyCheck)

Funkcje wonGame oraz lostGame

Mają za zadanie zakończyć rozgrywkę, ustawiając zmienną gameOn na 0, która blokuje dalszą grę. Wyświetlają odpowiednią wiadomość użytkownikowi. Funkcja lostGame ma za zadanie także po porażce zmianę wyglądu pól na planszy, w celu pokazania użytkownikowi informację o bombach oraz źle oflagowanych polach.

Moduł help.py

Posiada kilka pomocniczych funkcji czyszczących dane pola lub kontenery danych.

Moduł pngs.py

Odpowiedzialny za pomoc w użyciu i przypisywaniu odpowiednich obrazów png obiektom z biblioteki tkinter.

Wnioski podsumowujące projekt

Problemy

Z powodu braku rozdzielenia części projektu odpowiadającej za logikę od części odpowiadającej za interfejs, nie udało mi się stworzyć testów jednostkowych dla mojego projektu (problemy z unittest z powodu mainloop). Po kilku wersjach i wielu poprawkach ostateczna wersja jest jednak w pełni grywalna i prawdopodobnie wszystkie podane testy spełniłaby. Na przyszłość będę oddzielał logikę od interfejsu przy korzystaniu z tego typu bibliotek.

Jednym z problemów na jakie natrafiłem był brak możliwości prawidłowego użycia wyrażenia lambda jako funkcji, która ma następować po naciśnięciu przycisku, który tworzony jest w pętli. Jako zamiennik dwa razy wykorzystałem funkcję partial, przez co w całym projekcie użyłem tylko jednego wyrażenia lambda.

Dodatkowe funkcjonalności i wyjaśnienia

Jeśli chodzi o zgodność programu z opisem, wszystkie wyznaczone podzadania zostały spełnione. Dodatkową funkcjonalnością jaką zamieściłem jest wybór przez użytkownika poprzez zaznaczenie opcji „erase” w menu, jeśli chce, aby po każdej rozpoczętej poprawnie rozgrywce pola na dane wejściowe były czyszczone. Poza przyciskiem rozpoczęcia nowej gry dodałem również opcję próby takowej dzięki klawiszowi ‘Enter’, co znacznie przyspiesza rozpoczęcie kolejnej gry, szczególnie jeśli użytkownik decyduje się grać cały czas przy tych samych danych wejściowych.

Nawiązując do klasycznego sapera, po kliknięciu w bombę, mój program (funkcja lostGame) podmienia teksturę danego przycisku na bombę na czerwonym tle, oraz zamienia pola oznaczone flagą na przekreślone bomby, aby pokazać użytkownikowi, gdzie się mylił co do oflagowywania.

Chciałbym również wytłumaczyć, że przycisku po kliknięciu danego pola nie zmieniają faktycznego stanu na tkinter.DISABLED. Grafika takiego przycisku wygląda znacznie gorzej, co utrudniało rozgrywkę oraz było mało estetyczne, więc zamiast tego napisałem własną imitację dezaktywacji odkrytego przycisku.

Ostatnią, małą, ale sprawiającą że gra się dużo przyjemniej rzeczą jaką zaimplementowałem jest ponowne rozstawianie bomb, dopóki w polu, gdzie użytkownik kliknął jako pierwsze nie będzie bomby.