

Projekt z Języków Symbolicznych

Dokumentacja

Temat projektu

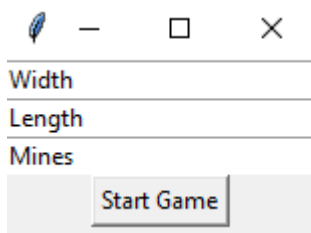
Projekt jest znaną grą Saper (Minesweeper) znana jako podstawowy dodatek do Windowsa już od wersji 7. Celem gry jest oflagowanie wszystkich pól zawierających miny lub kliknięcie wszystkich pól nie zawierających min. Ułatwieniem w osiągnięciu celu jest oznaczenie pola liczbą min sąsiadujących z nim.

Sterowanie odbywa się za pomocą kliknięcia prawym lub lewym przyciskiem myszy na pole:

- prawy przycisk - jedno kliknięcie - oflagowanie, kolejne - oznakowanie pyłajnikiem, kolejne - zdjęcie oznakowania z pola
- lewy przycisk - jeśli jest pod nim mina - przegrywamy grę, jeśli nie to pole i każde sąsiadujące nie posiadające miny oznaczone jest jako kliknięte

Funkcjonalność:

1. Pole zdefiniowania wielkości planszy i ilości min wraz z walidacją wprowadzonych danych. Po wprowadzeniu poprawnych danych okno zamyka się i uruchamia plansza gry



tk — □ ×

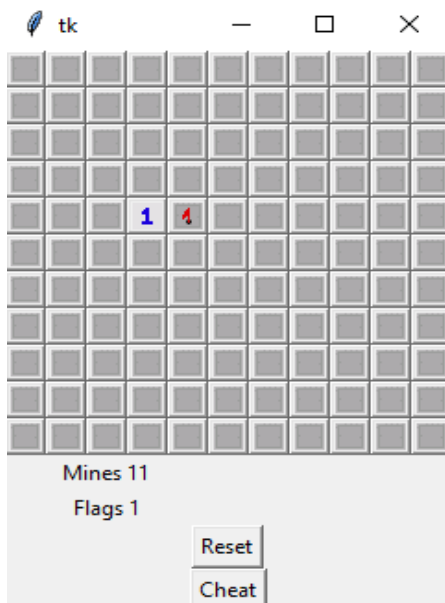
Width

Length

Mines

Start Game

2. Oznaczenie pola jako flaga z dodaniem ilości postawionych flag



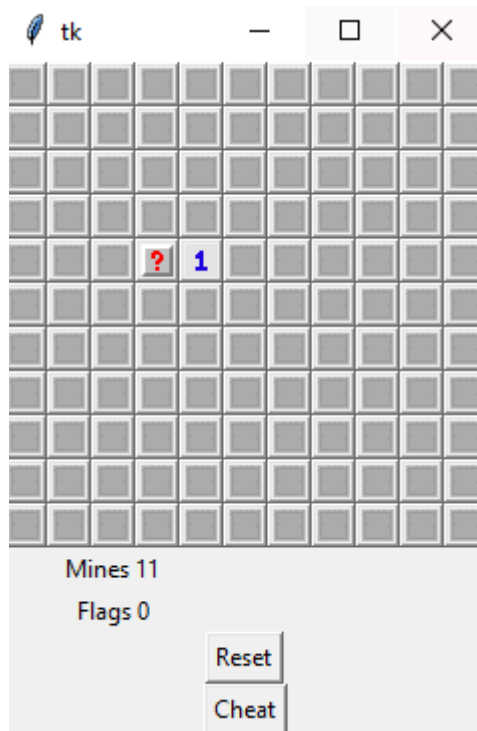
tk — □ ×

			1	4					

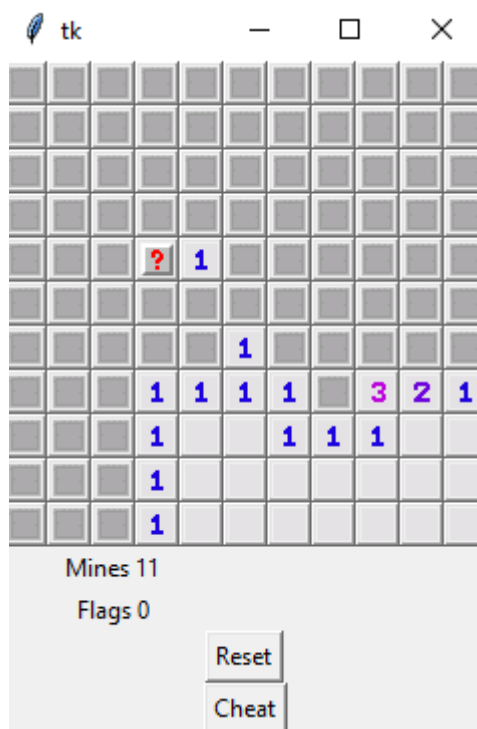
Mines 11
Flags 1

Reset
Cheat

3. Oznaczenie pola jako pytajnik bez dodania ilości postawionych flag

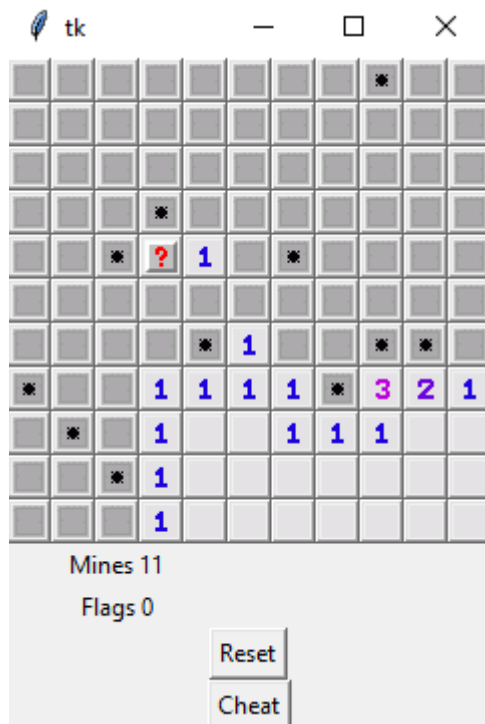


4. Kliknięcie pola i jedna z dwóch reakcji - przegranie gry jeśli jest to pole z miną, oznaczenie pola jako kliknięte i odkrycie wszystkich pól bez miny jako sąsiada



5. Oznaczenie z iloma minami sąsiaduje pole oznaczone jako kliknięte
6. Przycisk Reset przywracający planszę do stanu początkowego i na nowo losujący położenie min

7. Przycisk Cheat pokazuje wszystkie znajdujące się na planszy miny



Klasy i funkcje zawarte w projekcie

Funkcja main

[link](#)

Zawiera rozpoczęcie działania interpretera tkintera stworzenie instancji klasy `get_data` która po zamknięciu okna zapisuje dane dotyczące wielkości okna i ilości min. Tworzy instancję klasy `Main_field` przekazując do niej dane w instancji klasy `get_data`

Klasa `get_data`

[link](#)

Zawiera trzy pola `Entry` służące do wprowadzenia wielkości planszy i sprawdzenia poprawności danych

Funkcja `validate`

[link](#)

Sprawdza poprawność wprowadzonych danych jeśli są niepoprawne wyświetla wiadomość dotyczącą błędu. Jeśli poprawne zapisuje dane i zamyka okno

Klasa `Main_field`

[link](#)

Zawiera definicje wszystkich zmiennych używanych przez klasę, zdefiniowanie słownika zawierającego wszystkie używane ikony. Zdefiniowanie miejsca wyświetlania ilości min i flag oraz dwóch przycisków wraz z ich funkcjami. Zdefiniowanie labeli. Zapisanie obiektu `get_data` do używania w obrębie klasy.

Funkcja start_game

[link](#)

Używana do ustawienia na nowo planszy. Tworzy słownik wszystkich przycisków wczytując do nich podstawowe dane jak stan, obraz położenie na planszy oraz przypisanie do nich przycisku

Funkcja onClick

[link](#)

Obsługuje lewy przycisk myszy i jego funkcjonalność. Sprawdza czy pod polem nie było miny i wywołuje funkcję zmieniającą stan sąsiadów na kliknięte jeśli nie była to mina. Aktualizuje również liczbę klikniętych pól

Funkcja onRightClick

[link](#)

Obsługuje prawy przycisk myszy. Sprawdza stan i aktualizuje go na kolejny - z nie klikniętego na oflagowany z niego na pytajnik, a z niego na stan nie kliknięty Aktualizuje również licznik flag

Funkcja gameOver

[link](#)

Kończy grę wybierając jaki załadować obrazek do ostatnio klikniętego pola. Wyświetla komunikat o wygranej lub przegranej

Funkcja clearSurroundingTiles

[link](#)

Używając [kolejki](#) dodaje x i y pola wymagającego zmiany na stan kliknięty i aktualizacji ikony

Funkcja clearTile

[link](#)

Zmienia status przycisku dodaje go do kolejki i zmienia obrazek przycisku zależnie od min które go otaczają. Zwiększa również licznik klikniętych pól i sprawdza czy nie są kliknięte wszystkie

Funkcja refreshLabels

[link](#)

Odświeża ilość oflagowanych pól

Funkcja reset

[link](#)

Umożliwia ponowne uruchomienie całej planszy z przywróceniem ikon w stan domyślny i ponownym losowaniem położenia min

Funkcja cheat

[link](#)

Odkrywa wszystkie znajdujące się na planszy miny

Klasa Minesweeper_test(unittest.TestCase)

[link](#)

testShouldValidateEntry

[link](#)

Testuje na zdefiniowanych w liście rozmiarach planszy walidację danych wywołując funkcję `data.validate`

`testShouldEndWhenClicked`

[link](#)

Testuje czy przy kliknięciu pola które jest miną gra zakończy się

`testShoulCountFlags`

[link](#)

Testuje czy licznik zaktualizuje się po oflagowaniu danego pola oraz czy stan pola zmieni się

`testShouldCountQuestion`

[link](#)

Testuje czy przy dwóch kliknięciach zaktualizuje się status pola i ilość oflagowanych pól najpierw z 0 na 1 potem z powrotem na 0 oraz czy stanem pola będzie pytajnik

`testShouldRefreshFlaggedCount`

[link](#)

Testuje czy przy dwóch kliknięciach zaktualizuje się status pola i ilość oflagowanych pól. Sprawdza ilość flag przed, po kliknięciu prawym przyciskiem oraz stan pola

`testShouldEndWhenAllClicked`

[link](#)

Testuje czy gra zakończy się po kliknięciu wszystkich pól bez miny

`testShouldEndWhenAllFlagged`

[link](#)

Testuje czy gra zakończy się po oznaczeniu wszystkich pól z minami

`testShouldNotCountFlags`

[link](#)

Testuje czy licznik oznaczeń nie wzrośnie po kliknięciu na kliknięte pole

`testShoulZerodFlagCounts`

[link](#)

Testuje czy po oznaczeniu flag i resecie liczniki i pola zresetują się

`testCheatResetWork`

[link](#)

Testuje czy po uruchomieniu cheatu i resecie pola wrócą do standardowego koloru