

Projekt z Języków Symbolicznych

Dokumentacja

Mastermind

Gracz wpisuje "kod" (4 cyfry) składający się z liczb od 1 do 6 włącznie. Jego zadaniem będzie odgadnięcie tego kodu w skończonej liczbie prób (12 prób). Po każdej próbie gracz jest informowany jedynie ile z elementów, które wypróbował w danej kolejce znajduje się w ukrytym kodzie (wypisuje „0”) i ile z nich znajduje się już na swojej pozycji (wypisuje „X”). Zbierając informacje z listy w końcu dochodzi do ostatecznego rozwiązania. Jeśli gracz z 12 prób nie odgadł kodu, wyświetla się napis "Przegrana!" i kod który był zgenerowany. W przypadku zwycięstwa wyświetla się napis "Wygrana!". Także na początku każdej gry program wybiera losowe reguły. Przy niepoprawnych regułach program będzie wypisywać niepoprawne odpowiedzi. Jeśli gracz zauważy, że gra wypisuje dziwne odpowiedzi to on może to sprawdzić wciskając przycisk "Oszust!". Jeśli reguła gry będzie niepoprawna to wyświetla się napis "Złapałeś/łas mnie!". Jeśli poprawna to "Tere fere" i kod który był zgenerowany.

Okno gry zawiera:

- przycisk Sprawdź (Enter)
- przycisk Oszust
- przycisk Reset (Esc)
- okno z polem tekstowym (Tab)
- lista odpowiedzi

Klasy i funkcje zawarte w projekcie

Klasa

```
class RulesGame
```

Klasa programu, zawierająca funkcji logiki gry, restarta, sprawdzania reguł oraz rysowania okna. [link](#)

Klasa

```
TestProject
```

Klasa programu, zawierająca tesy. [link](#)

Funkcje

`main.py` [link](#)

- Zawiera inicjalizację modułu tkinter oraz głównego modułu programu `game`.

`game.py` [link](#) zawiera

- `class RulesGame` [link](#)
- `__init__` inicjalizacja zmiennych. [link](#)
- `resetBtn` funkcja restartuje nową grę. [link](#)
- `isScammer` sprawdzanie czy reguły gry są prawidłowe. [link](#)
- `mainLogic` główna logika gry. Liczy i daje odpowiedzi czy gracz poprawnie podał liczby. [link](#)
- `drawField` funkcja rysuje okno. [link](#)

`functions.py` [link](#) zawiera

- `isTrueInput` sprawdza czy prawidłowo podane dane [link](#)
- `generateSample` generuje losowe liczby do odgadnięcia. [link](#)
- `whatIsRule` program wybiera reguły gry. [link](#)
- `createFakeFeedback` generuje niepoprawne odpowiedzi przy niepoprawnych regułach. [link](#)

Testy

`test_0_checkIfNotGuessed` [link](#)

Wpisuje odpowiedzi z błędnymi cyframi.

`test_1_checkIfTwoNumbersAreCorrect` [link](#)

Wpisuje odpowiedzi z dwoma poprawnymi cyframi w złych miejscach.

`test_2_checkIfNumbersAreCorrect` [link](#)

Wpisuje odpowiedzi z poprawnymi cyframi w złych miejscach oraz dobrych miejscach.

`test_3_checkIfGuessed` [link](#)

Wpisuje poprawne odpowiedzi.

test_4_checkIfValidInput [link](#)

Wpisuje niepoprawny kod - 1231231.

test_5_checkIfValidInput [link](#)

Wpisuje niepoprawny kod - 8779.

test_6_checkIfScammer [link](#)

Wciśnięcie przycisku "Oszust!" przy niepoprawnych zasadach gry.

test_7_checkIfScammer [link](#)

Wciśnięcie przycisku "Oszust!" przy poprawnych zasadach gry.

test_8_checkReset [link](#)

Wpisanie 10 odpowiedzi z błędnymi cyframi. Wciśnięcie "Reset". Wpisanie 5 odpowiedzi z błędnymi cyframi. Sprawdzanie czy licznik tur resetuje po wciśnięciu "Reset".