

Politechnika Wrocławska  
Wydział Elektroniki  
Kierunek Elektronika

---

# DOKUMENTACJA KOŃCOWA LUGO\_GAME APLIKACJA W JĘZYKU PYTHON

---

*autorzy:*

DOMINIKA KANTY  
MIKOŁAJ CZERNIAK  
MICHAŁ STANISŁAWSKI  
TOMASZ KOPCIŃSKI

**Prowadzący:**  
**mgr inż. Konrad Kluwak**  
Zajęcia: Środa 9<sup>15</sup> – 11<sup>00</sup>  
Kod grupy: E06-42f

Rok akademicki 2019/2020

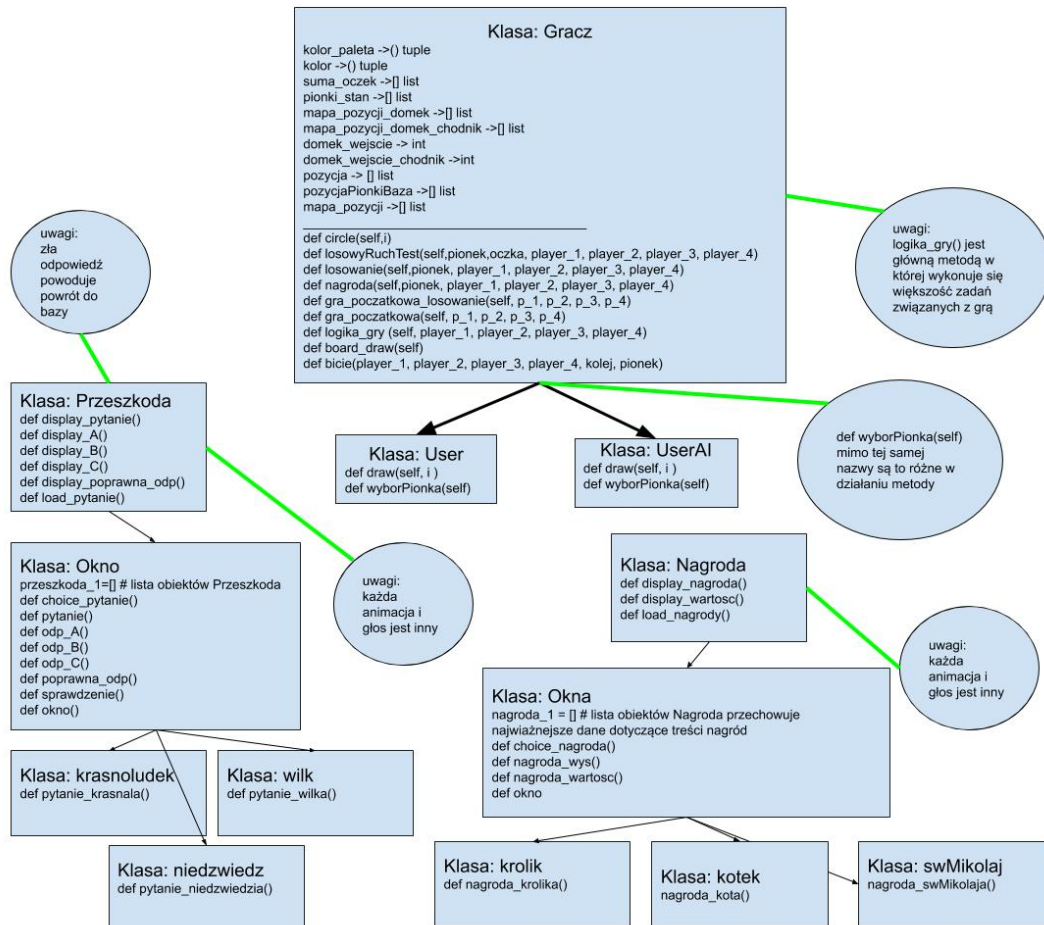
## Spis treści

<b>Wprowadzenie</b>	<b>2</b>
<b>Opis klas:</b>	<b>3</b>
Gracz . . . . .	3
User(Gracz) . . . . .	3
UserAI(Gracz) . . . . .	3
LUDO_nagrody . . . . .	4
LUDO_przeszkoda.py . . . . .	6
<b>Karty CRC:</b>	<b>8</b>

# Wprowadzenie

Projekt którego podjęliśmy się jest to tradycyjny ”Chińczyk”, jednak zostały w nim umieszczone specjalne przeszkody i utrudnienia tak, aby uatrakcyjnić grę. Grafika planszy została napisana dzięki pygame.draw. Wszystkie klasy zostały rozpisane na poniższych stronach wraz z wytłumaczeniem.

Poniżej zostanie zaprezentowany schemat naszego projektu.



Rysunek 1: Schemat

## Opis klas:

### Gracz

- **\_\_init\_\_** konstruktor odpowiedzialny za przyporządkowanie odpowiedniego koloru pionka, pozycji początkowej oraz innych początkowych parametrów
- **circle** metoda odpowiedzialna za rysowanie wybranego pionka w zadanej pozycji
- **losowanie** metoda zwracająca losową liczbę oczek
- **nagroda** metoda odpowiedzialna za wywołanie klas odpowiedzialnych za wydawanie nagród i egzekwowanie wartości przez nich zwracanych
- **przeszkoda** metoda odpowiedzialna za wywołanie klas odpowiedzialnych za wyświetlanie przeszkód (tj. pytań, egzekwowanie wartości przez nich zwracanych tj. odsyłanie pionków na pola bazowe dla złych odpowiedzi)
- **gra\_początkowa\_losowanie** metoda odpowiedzialna za wybranie gracza rozpoczynającego rozgrywkę
- **logika\_gry** główna metoda, w której wykonywana jest cała logika gry, jej zasady i mechaniki
- **board\_draw** metoda odpowiedzialna za rysowanie planszy korzystająca z funkcji rysowania wektorowego pygame
- **bicie** metoda odpowiedzialna za sprawdzanie kolizji z pionkami, wykonywanie bicia tj. odstawianie pionków do baz

### User(Gracz)

- **draw** metoda wywołująca metodę Gracz.circle
- **wyborPionka** metoda odpowiedzialna za wybór pionka za pomocą klawiatury, umożliwia też ona wyjście z gry

### UserAI(Gracz)

- **draw** metoda wywołująca metodę Gracz.circle
- **wyborPionka** metoda odpowiedzialna za wybór pionka w sposób automatyczny

## LUDO\_nagrody

Projekt LUDO\_nagrody.py składa się z 5 klas. Klasa Nagroda odpowiada za przechowywanie danych zawierających treści nagród. Klasa Okno jest odpowiedzialna za losowanie oraz wyświetlenie nagrody. Natomiast klasy krolik, swMikolaj i kotek dziedziczą z klasy Okno i wyświetlają odpowiednie dla siebie obrazy oraz odtwarzają odpowiednią muzykę. Do wyświetlenia okna użyta została biblioteka *tkinter*

Do wyświetlenia okna użyta została biblioteka *tkinter*, natomiast aby otworzyć dźwięk użyto biblioteki *winsound*.

Klasa *Nagroda*, odpowiedzialna jest za przechowanie danych na temat nagród.

- **def \_\_init\_\_** konstruktor, odpowiedzialny za tworzenie elementów klasy Nagroda. Dane wczytywane są w odpowiedniej kolejności, oddzielone znakiem "\_\_\_",
- **def display\_nagroda** metoda odpowiedzialna za zwrócenie wartości *a*, odpowiadającej treści nagrody,
- **def display\_wartosc** metoda odpowiedzialna za zwrócenie wartości *a*, odpowiadającej treści nagrody,
- **def load\_nagrody** metoda odpowiadająca za ładowanie treści nagród z pliku. Plik zawierający bazę pytań nazwany został *nagrody.txt*.

Klasa *Okno*, dziedziczy z klasy *Nagroda* jest odpowiedzialna za wylosowanie pytania oraz wyświetlenie okna. Okno wyświetlane jest z pomocą biblioteki *tkinter*.

- **def choice\_nagroda** metoda odpowiedzialna za losowanie wartości z przedziału od 1 do 8
- **def nagroda\_wys** metoda zwracająca wartość *a* odpowiadającą treści nagrody z wylosowanej nagrody,
- **def nagroda\_wartosc** metoda zwracająca wartość *a* odpowiadającą wartości z wylosowanej nagrody
- **def okno** metoda odpowiedzialna za wyświetlenie wyświetlenie okna z treścią nagrody, obrazkiem przypisanym dla danej klasy nagrody. Metoda ta odpowiada także za odtworzenie muzyki

Klasa *krolik* dziedziczy z klasy *Okno*. Odpowiada ona za wyświetlenie treści nagrody oraz Obrazka królika.

- **def nagroda\_krolika** metoda wyświetlająca nagrodę z odpowiednim obrazkiem oraz muzyką, która zostaje przzerwana najpóźniej w chwili zamknięcia okna. Metoda ta zwraca wartość odpowiadającą wartości nagrody w pliku *nagrody.txt*

Klasa *swMikolaj* dziedziczy z klasy *Okno*. Odpowiada ona za wyświetlenie treści nagrody oraz Obrazka Świętego Mikołaja.

- **def nagroda\_swMikolaja** metoda wyświetlająca nagrodę z odpowiednim obrazkiem oraz muzyką, która zostaje przzerwana najpóźniej w chwili zamknięcia okna. Metoda ta zwraca wartość odpowiadającą wartości nagrody w pliku *nagrody.txt*

Klasa *kotek* dziedziczy z klasy *Okno*. Odpowiada ona za wyświetlenie treści nagrody oraz Obrazka Świętego Mikołaja.

- **def nagroda\_kota** metoda wyświetlająca nagrodę z odpowiednim obrazkiem oraz muzyką, która zostaje przzerwana najpóźniej w chwili zamknięcia okna. Metoda ta zwraca wartość odpowiadającą wartości nagrody w pliku *nagrody.txt*

## LUDO\_przeszkoda.py

Projekt LUDO\_pytania.py składa się z 5 klas. Klasa Przeszkoda odpowiada za przechowywanie danych nt pytań oraz odpowiedzi. Klasa Okno jest odpowiedzialna za wylosowanie, wyświetlenie oraz sprawdzenie czy odpowiedź była poprawna, natomiast klasy krasnoludek, wilk i niedzwiedz dziedziczą z klasy Okno i wyświetlają odpowiednie obrazy oraz odtwarzają odpowiednią muzykę.

Klasa Przeszkoda - odpowiada za przechowywanie wszystkich danych na temat pytań oraz pozwala wyświetlić odpowiednie elementy.

- **\_\_init\_\_** - konstruktor odpowiadający za tworzenie elementów klasy Przeszkoda. Wczytuje dane ustawione we właściwej kolejności i oddzielone znakiem — ze stringa.
- **display\_pytanie** - metoda odpowiedzialna za zwrócenie wartości pytania w stringu.
- **display\_A** - metoda odpowiedzialna za zwrócenie wartości odpowiedzi A w stringu.
- **display\_B** - metoda odpowiedzialna za zwrócenie wartości odpowiedzi B w stringu.
- **display\_C** - metoda odpowiedzialna za zwrócenie wartości odpowiedzi C w stringu.
- **display\_poprawna\_odp** - metoda odpowiedzialna za zwrócenie wartości poprawnej odpowiedzi od 1 do 3 w intcie.
- **load\_pytania** - funkcja ładująca pytania z pliku pytania.txt do tablicy.

Klasa *Okno*, dziedziczy z klasy *Przeszkoda* jest odpowiedzialna za wylosowanie pytania oraz wyświetlenie okna za pomocą biblioteki tkinter oraz sprawdzenie czy wybrana odpowiedź jest poprawna.

- **def choice\_pytanie** metoda zwracająca losową wartość z przedziału od 1 do 81,
- **def odp\_A** metoda zwracająca wartość *a* odpowiadającą treści odpowiedzi A z wylosowanego pytania,
- **def odp\_B** metoda zawierająca wartość *a* odpowiadającą treści odpowiedzi B z wylosowanego pytania,
- **def odp\_C** metoda zwracająca wartość *a* odpowiadającą treści odpowiedzi C z wylosowanego pytania,
- **def poprawna\_odp** metoda zwracająca wartość *a* odpowiadającą wartości poprawnej odpowiedzi z przedziału od 1 do 3,
- **def sprawdzenie** metoda sprawdzająca poprawność udzielonej odpowiedzi i zwracająca 1 w przypadku dobrej odpowiedzi oraz 0 w przypadku błędnej odpowiedzi,

- **def okno** metoda wyświetlająca okno z pytaniem, obrazkiem oraz 3 możliwymi do wyboru odpowiedziami oraz rozpoczyna odtwarzanie muzyki.

Klasa *krasnoludek*, klasa dziedzicząca z klasy *Okno* odpowiada za wyświetlenie pytania wraz z właściwą muzyką oraz obrazem.

- **def pytanie\_krasnala** metoda wyświetlająca pytanie z odpowiednim obrazem oraz muzyką zwracająca wartość 0 jeżeli udzielona odpowiedź była błędna lub 1 jeśli była poprawna,

Klasa *wilk*, klasa dziedzicząca z klasy *Okno* odpowiada za wyświetlenie pytania wraz z właściwą muzyką oraz obrazem.

- **def pytanie\_wilka** metoda wyświetlająca pytanie z odpowiednim obrazem oraz muzyką zwracająca wartość 0 jeżeli udzielona odpowiedź była błędna lub 1 jeśli była poprawna,

Klasa *niedzwiedz*, klasa dziedzicząca z klasy *Okno* odpowiada za wyświetlenie pytania wraz z właściwą muzyką oraz obrazem.

- **def pytanie\_niedzwiedzia** metoda wyświetlająca pytanie z odpowiednim obrazem oraz muzyką zwracająca wartość 0 jeżeli udzielona odpowiedź była błędna lub 1 jeśli była poprawna,



## Karty CRC:

Gracz	
<ul style="list-style-type: none"> <li>• cała logika gry</li> <li>• mechanika ruchu</li> <li>• bicie pionków</li> <li>• wywoływanie nagród i przeszkód</li> </ul>	<ul style="list-style-type: none"> <li>• pygame</li> <li>• sys</li> <li>• random</li> </ul>

User	
<ul style="list-style-type: none"> <li>• wybór pionka z klawiatury</li> <li>• wywoływanie funkcji Gracz::draw()</li> </ul>	<ul style="list-style-type: none"> <li>• Gracz</li> </ul>

User	
<ul style="list-style-type: none"> <li>• wybór pionka w sposób automatyczny</li> <li>• wywoływanie funkcji Gracz::draw()</li> </ul>	<ul style="list-style-type: none"> <li>• Gracz</li> </ul>

Przeszkoda	
<ul style="list-style-type: none"> <li>• odpowiada za przechowywanie informacji o pytaniach i odpowiedziach</li> <li>• pozwala wyświetlić konkretne pytanie oraz odpowiedzi</li> <li>• odpowiada za wczytywanie danych z pliku "txt".</li> </ul>	<ul style="list-style-type: none"> <li>• tkinter</li> <li>• winsound</li> <li>• random</li> <li>• codecs</li> </ul>

Okno	
<ul style="list-style-type: none"> <li>• odpowiada za wylosowanie liczby od 1 do 81,</li> <li>• pozwala wyświetlić konkretne pytanie oraz odpowiedzi,</li> <li>• odpowiada za sprawdzenie czy odpowiedź jest poprawna,</li> <li>• odpowiada za wyświetlenie okna z pytaniem, 3 możliwymi odpowiedziami, odtworzenie muzyki oraz wyświetlenie obrazu.</li> </ul>	<ul style="list-style-type: none"> <li>• Przeszkoda</li> </ul>

krasnodek	
<ul style="list-style-type: none"> <li>• wyświetla okno z pytaniem, ma przypisany konkretny obraz do wyświetlenia oraz muzykę do odtworzenia,</li> </ul>	<ul style="list-style-type: none"> <li>• Okno</li> </ul>
wilk	
<ul style="list-style-type: none"> <li>• wyświetla okno z pytaniem, ma przypisany konkretny obraz do wyświetlenia oraz muzykę do odtworzenia,</li> </ul>	<ul style="list-style-type: none"> <li>• Okno</li> </ul>
niedzwiedz	
<ul style="list-style-type: none"> <li>• wyświetla okno z pytaniem, ma przypisany konkretny obraz do wyświetlenia oraz muzykę do odtworzenia,</li> </ul>	<ul style="list-style-type: none"> <li>• Okno</li> </ul>
Nagroda	
<ul style="list-style-type: none"> <li>• odpowiada za przechowywanie informacji o nagrodach</li> <li>• pozwala wyświetlić konkretne nagrody</li> <li>• odpowiada za wczytywanie danych z pliku "txt".</li> </ul>	<ul style="list-style-type: none"> <li>• random</li> <li>• codecs</li> <li>• tkinter</li> <li>• winsound</li> </ul>
Okna	
<ul style="list-style-type: none"> <li>• odpowiada za wylosowanie wartości od 1 do 8,</li> <li>• pozwala wyświetlić konkretną nagrodę,</li> <li>• odpowiada za wyświetlenie okna z nagrodą, odtworzenie muzyki, wyświetlenie obrazu, oraz przycisku zamykania okna.</li> </ul>	<ul style="list-style-type: none"> <li>• Nagroda</li> </ul>
krolik	
<ul style="list-style-type: none"> <li>• wyświetla okno z nagrodą, ma przypisany konkretny obraz do wyświetlenia oraz muzykę do odtworzenia,</li> </ul>	<ul style="list-style-type: none"> <li>• Okna</li> </ul>
swMikolaj	
<ul style="list-style-type: none"> <li>• wyświetla okno z nagrodą, ma przypisany konkretny obraz do wyświetlenia oraz muzykę do odtworzenia,</li> </ul>	<ul style="list-style-type: none"> <li>• Okna</li> </ul>
kotek	
<ul style="list-style-type: none"> <li>• wyświetla okno z nagrodą, ma przypisany konkretny obraz do wyświetlenia oraz muzykę do odtworzenia,</li> </ul>	<ul style="list-style-type: none"> <li>• Okna</li> </ul>