

Scribble.io clone

<https://github.com/mprostakk/scribble.io-clone>

Członkowie projektu:

- Maciej Prostak
- Malika Wawer

Ogólne

Program zbudowany jest na wielu wątkach, w którym komunikacja odbywa się za pomocą "Thread safe" kolejki w Pythonie. Obsługiwanych jest wielu użytkowników jednocześnie. Architektura to klient-serwer. Mamy wiele różnych rodzajów wiadomości, gdzie każda z nich ma odpowiednie nagłówki oraz walidatory.

Serwer jest typu zdarzeniowego. Wiadomości są szyfrowane. Praca odbywała się najczęściej w trybie pracy pair-programming.

Wymagania

Na zielono zrobione :)

1. zaimplementowania logiki, która wykorzystuje więcej niż 3 rodzaje przesyłanych wiadomości oraz
2. wykorzystania co najmniej 1 mechanizmu z poniższych:
 - a. synchronizacja wątków,
 - b. obsługa większej liczby graczy niż dwóch jednocześnie,
- c. obsługa wielu procesów równoległe,
 - ii. uzyskanie większej liczby punktów niż 65 wymaga:
 1. spełnienia warunków z punktu powyższego oraz
 2. wykorzystania co najmniej 2 z poniższych mechanizmów:
 - a. inna architektura niż klient-serwer,
 - b. asynchroniczność (p2p),
 - c. serwer zdarzeniowy,
 - d. połączenia szyfrowane.

Obowiązki

Malika

- zbudowanie struktury serwera
- zaimplementowanie receive i send
- zaimplementowanie logiki rund
- poprawienie wyglądu GUI
- zaimplementowanie szyfrowania
- dokumentacja requestów

Maciej

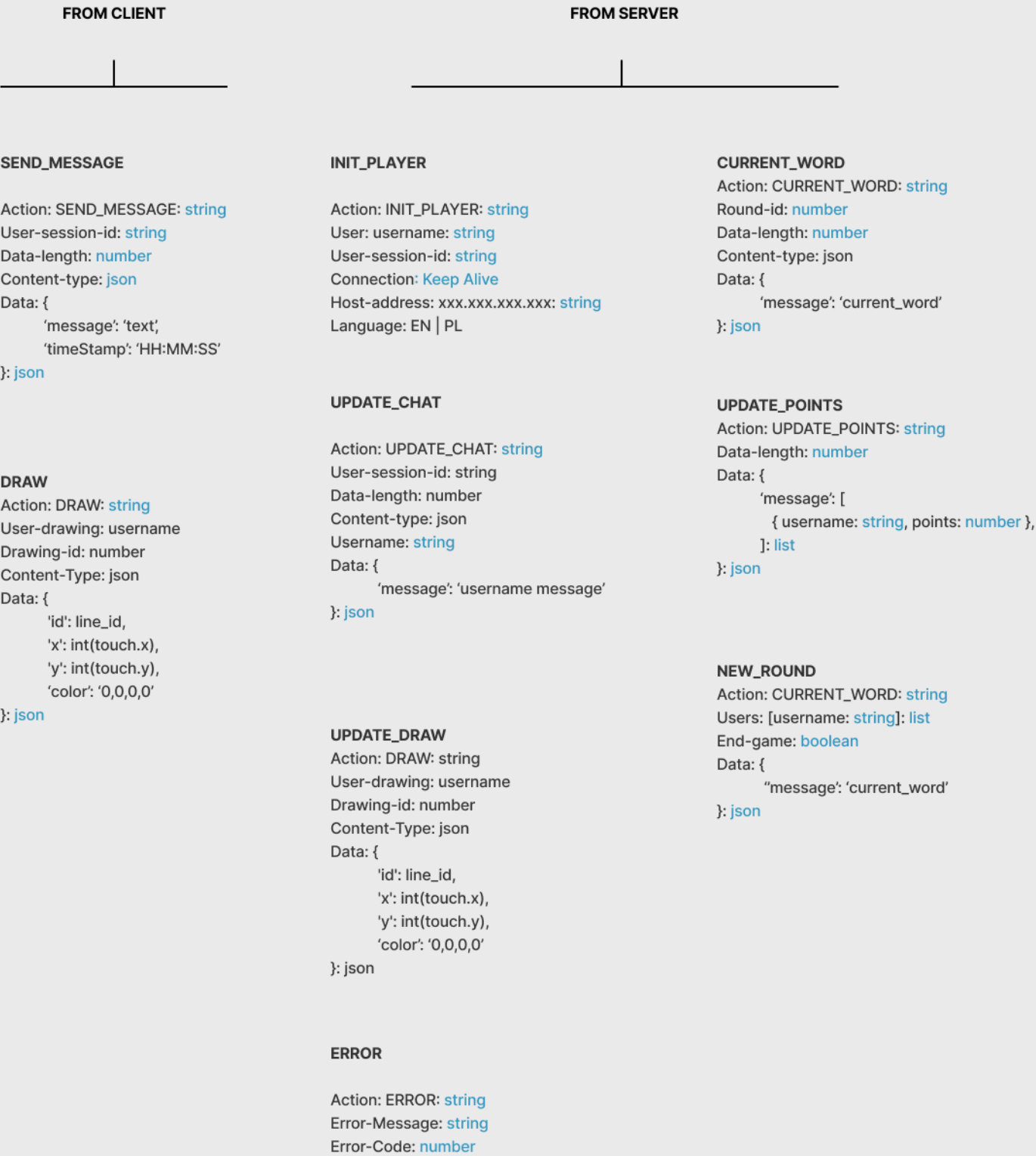
- zaimplementowanie wielowątkowości
- zaimplementowanie wielu użytkowników
- zbudowanie GUI w Kivy z możliwością rysowania
- zaimplementowanie wysyłania rysunków
- zaimplementowanie prostego chatu

Napotkane trudności

- nauka Kivy
- stylowanie interfejsu Kivy
- refaktorowanie kodu bez testów
- testowanie kodu wątkowego
- wydajne wysyłanie rysunków (na początku bardzo się cięło)

Rodzaje wiadomości:

Diagram dostępnych wiadomości w protokole.



1. SEND_MESSAGE (from Client) - Przesyła na serwer wiadomość wpisaną przez użytkownika na czacie. Forma zależy od tego czy wiadomość była obecnie odgadywanym hasłem - jeśli tak, pozostali gracze otrzymują tylko informację pozytywnym odgadnięciu hasła przez gracza wysyłającego wiadomość. (Broadcast wiadomości do graczy UPDATE_CHAT)
2. DRAW (from Client) - Przesyła na serwer informacje o narysowanej kresce przez użytkownika podanego w polu User-Drawing. Żądanie jest wysyłane za każdym razem kiedy użytkownik zmienia pozycję kursora - dzięki czemu pozostali użytkownicy mają widok live. Wraz z punktami x oraz y wysyłany jest odpowiedni ID linii, dzięki czemu wiemy jak łączyć punkty w jedną linię.
3. UPDATE_DRAW (from Server) *Broadcast* - Przekazuje dalej wiadomość od klienta do pozostałych graczy o nowym rysunku.
4. INIT_PLAYER (from Server) - Żądanie jest wywoływane gdy nowy socket klienta jest akceptowany przez serwer. Ustawiane jest nowe połączenie, obiekt klienta jest zapisywany do kolekcji, użytkownik otrzymuje session_id.
5. UPDATE_CHAT (from Server) *Broadcast* - Przekazuje dalej wiadomość od klienta do pozostałych graczy o nowej wiadomości. Serwer również wysyła informacje o tym, że runda się zakończyła lub ktoś odgadł hasło.
6. CURRENT_WORD (from Server) *Broadcast* - Do wszystkich graczy wysyłana jest informacja o nowo wylosowanym słowie. Osoba rysująca dostaje słowo w czytelnej postaci, pozostali w ukrytej znakami “_”.
7. UPDATE_POINTS (from Server) *Broadcast* - Do wszystkich graczy wysyłana jest informacja o aktualnym stanie punktów. Stare wartości nadpisywane są nowymi.
8. NEW_ROUND (from Server) *Broadcast* - Inicjalizacja nowej rundy. Po stronie serwera - zmiana aktualnie rysującego gracza, zmiana hasła do odgadnięcia, wysłanie hasła do graczy w odpowiedniej formie (jawnej lub ukrytej). Po stronie klienta - wyczyszczenie obszaru do rysowania, ustawienie nowego hasła w górnym widżecie.
9. ERROR (from Server) - Każda wiadomość posiada mechanizm validate. Jeśli funkcja validate rzuci wyjątek w momencie sprawdzania nagłówka, to zostanie zwrócony do żądającego informacja o błędzie (kod błędu oraz jej wytłumaczenie)

Błędy z kodami

SYNTAX_ERROR - 400

Authorization Failed - session_id unknown

AUTH_FAILED - 401

Syntax Error - could not parse incoming request

UNKNOWN ACTION_TYPE - 402

Unknown Action Type

FAILED_UPDATE_POINTS - 412

Points update failed - unknown username

INTERNAL_SERVER_ERROR - 500

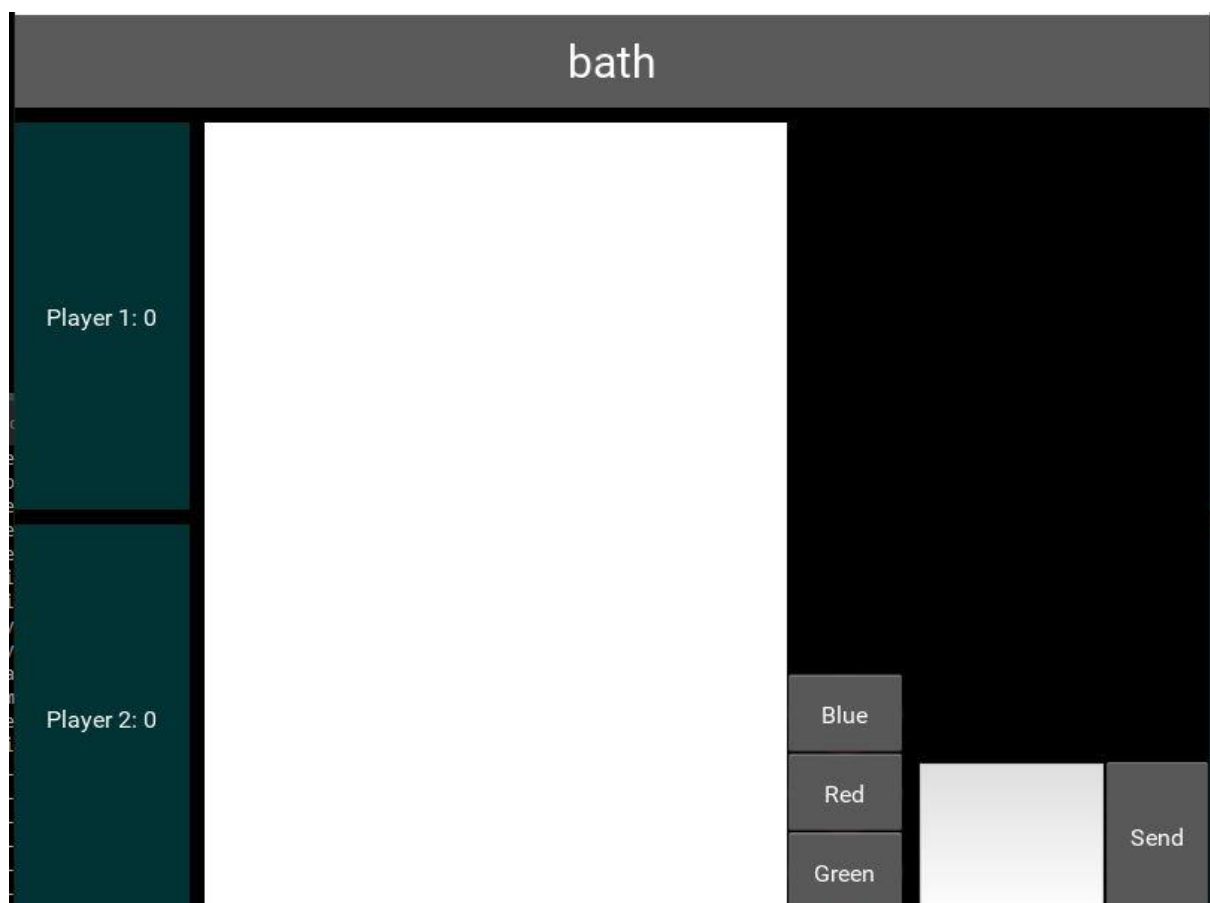
Internal server error

Aplikacja

Ekran rysującego

Pierwsza osoba ma możliwość rysowania. Losuje się słowo, które następnie będzie rysować odpowiednio na białym wyznaczonym canvas. Wiadomości możemy wysłać, ale bez możliwości wysłania poprawnego hasła.

Po lewej stronie GUI widzimy punkty, które są aktualizowane w momencie poprawnego wpisania hasła.



Osoba zgadująca

Zgadujący użytkownik widzi tylko ile hasło ma liter. Może wpisywać wiadomości w chat'cie. Zablokowana jest możliwość rysowania. Ta osoba, która odgadnie hasło dostaje 10 punktów i zaczyna się kolejna runda. Kolejnym rysującym jest osoba następna w kolejności.

— — — —

Player 1: 0

Player 2: 0

Blue

Red

Green

Send

Chat

Każda kolejna wchodząca osoba do gry otrzymuje od serwera odpowiedni nick. Jeśli wiadomość jest od użytkownika, widzimy przy wiadomości "Player x:"

Serwer również wysyła wiadomości mówiąca o aktualnym stanie gry.

Jeżeli wiadomość zawiera aktualnie zgadywane hasło, wysyłana jest tylko informacja "Answer correct!".

Jeżeli aktualnie rysujący próbuje wysłać na czacie hasło, wiadomość nie jest wysyłana dalej.

angry

Player 1: 0

Player 2: 10

Player 2: hello
Player 2: test
Player 2: Answer correct!
New round, Player 2 is drawing

Blue

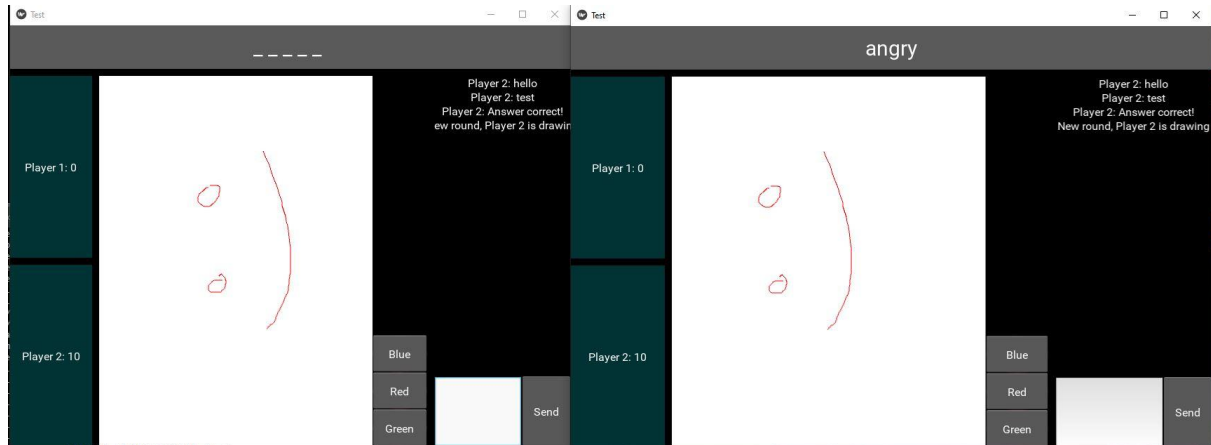
Red

Green

Send

Rysowanie

W czasie rzeczywistym widzimy co aktualnie jest na canvas rysującego. W momencie odgadnięcia hasła, canvas jest u wszystkich czyszczony.



Baza Słów:

Aktualne hasło, jest losowane z pliku words_list.py i jest powiązane z podobnymi formami tego słowa. Dzięki temu jeżeli użytkownik zgadnie słowo akceptowanych jest kilka, przypisanych do tego słowa form.

Do zaimplementowania kiedyś:

- pełna aktualizacja rysunku w momencie wejścia do gry przez nowego gracza
- pokoje z kodami
- dokończenie rysowania kolorami
- feature'y ułatwiające malowanie (wielkość pędzla, gąbka do usuwania, paint filler)
- dodanie możliwości wyboru własnego username

DIAGRAM UML

