

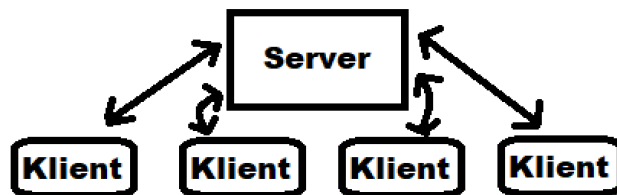
## Sprawozdanie sk2

Temat : Jungle Speed

### 1. Opis projektu

- krótki opis  
Jungle speed jest grą zręcznościową na 3-8 graczy, my zaimplementowaliśmy go na 4 by GUI wyglądało schludnie, lecz zmieniając zmienną server może obsłużyć większą ilość graczy. Gracz w swojej turze, która będzie wskazywana strzałką może wyciągnąć kartę ze stosu kart zakrytych na odkryty stos przed sobą (klawiszem N) . Gdy 2 gracze mają taki sam kształt na odkrytej karcie , ten kto pierwszy podniesie totem (spacją) daje swoje karty odkryte i karty odkryte przeciwnika i wkłada je na zakryty stos przegranego. W razie pomyłki , gracz który podniósł totem zbiera wszystkie karty wszystkich graczy. Wygrywa gracz który pierwszy pozbędzie się wszystkich kart. Gra się wtedy kończy. Server może obsłużyć równoległe dowolną ilość gier.
- użyta technologia
  - Do GUI i programu gracza użyto Pythona i biblioteki Pygame
  - Do Servera użyliśmy C

### 2. Opis komunikacji pomiędzy serwerem i klientem



Komunikacja odbywa się za pomocą BSD Sockets API. Server nasłuchuje i akceptuje wszystkich graczy którzy próbują się podłączyć pod adres który musi być ustalony wcześniej w programie. (Adres ipv4 servera). Następnie tworzy wątek (pthread) który ma za zadanie komunikację z tym jednym graczem. Gracz może asynchronicznie przesłać serverowi za pomocą tego socketu jeden z 3 komunikatów wyrażanych liczbowo :

1. : Chęć podniesienia karty, co server wykona jeśli jest kolej gracza wysyłającego komunikat.
2. : Chęć podniesienia totemu, co server wykona niezależnie od kolejki i odpowiednio zmieni stan gry.
3. : Chęć rozłączenia, co spowoduje zamknięcie socketu i usunięcie gry z pamięci servera. Reszta graczy nie zostanie rozłączona zaraz po tym.

Server graczowi na tym samym sockecie wysyła 18 znakowy ciąg znaków w którym zakodowany jest filtr i obecny stan gry. Zapewnia to synchronizację w uaktualnianiu gry.

### 3. Podsumowanie

- Najważniejsze informacje o implementacji

Gracz nie zna stanu gry. Na podstawie dostarczanej od servera wiadomości wyświetla odpowiednie liczby i karty na ekranie. Server informacje o każdej grze trzyma na dynamicznej liście. W chwili gdy do servera dołączy 4 graczy, gra jest zerowana (rozdawane są karty) i ta wrzucana jest na listę, a następnie server tworzy nową poczekalnię do której dołączyć będą mogli nowi gracze. Po wyjściu z gry nie da się do niej wrócić, a server usuwa informacje o niej, jednocześnie rozłączając resztę graczy, jako że ta jest już niegrywalna.

- Co sprawiło trudność

1. Trudność sprawiła asynchroniczność, liczba wątków musiała słuchać klientów i później operować na tej samej strukturze opisującej grę. Warunek postępu jest spełniony, lecz technicznie bezpieczeństwa nie. Nie powinno być to problemem, przy tak małej ilości graczy, lecz w razie dużego pecha, 2 gracze mogą podnieść totem w tej samej chwili, i jeśli jeden wątek nie zdąży zablokować drugiego, obaj będą starali się dać karty przeciwnikowi i jeśli ten wyjątek nie będzie obsługiwany przez server, ten nie tylko skończy krytycznym błędem obecną rozgrywkę, ale wszystkie prowadzone w danym momencie.

2. Również trudne było filtrowanie szumu na sockecie. Ten z jakiegoś powodu powstawał na początku pracy nad projektem, a także w czasie bardzo szybkiego wysyłania komunikatów przez jednego lub więcej graczy. Filtr będący trzema liczbami 110, dodawanymi przez server na początek komunikatu skutecznie odfiltrowywał takie szumy, a program klienta i servera po otrzymaniu wiadomości z innym filtrem rozpoznaje ją jako szum i ignoruje. Filtrowanie jest również potrzebne by server i gracz potrafił zamknąć socket i skończyć program w chwili gdy gdy ten drugi skończył się wskutek przerwania lub krytycznego błędu.

3. Również wiele błędów spowodowało nie używanie wskaźników w miejscu gdzie były potrzebne lecz to jest domena programowania funkcyjnego w C, bardziej niż Sieci Komputerowych.