**HangmanPy**

Haladó programozás

**Neumann János Egyetem  
GAMF  
Informatika Tanszék**

2020.10.19.

Feladat leírás

A játékosok szobákat hozhatnak létre, illetve csatlakozhatnak azokba. A játékmester egy szót választ, a többi játékosnak pedig azt kell kitalálnia úgy, hogy körönként egy betűt tippelhetnek.

Feladat megvalósítás:

A backend oldalt python-flask keretrendszerrel valósítottam meg. A szerver csak mint egy api működik a megjelenítést a frontend-en dolgoztam ki js-React könyvtár segítségével. A csatornák létrehozására menedzselésére Pusher api-t használtam. A program a jelenlegi verziójából hiányzik még a játékmester, körök, randomizált szó kiválasztás megvalósítása, nem lehet nyerni ,veszteni, nem működik az újraindítás.

**Backend**

**app.py:flask,pusher init, útvonalak**

Globális változók

app: Flask inicializációja

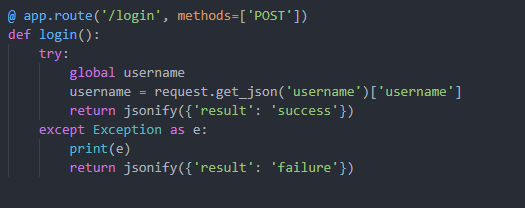
gameInstances: dictionary, a szobák nevei (kulcs) es Hangman példányok (érték) alkotják

username: a felhasználó nevét tárolja (ebben a verzióban nincs felhasználva)

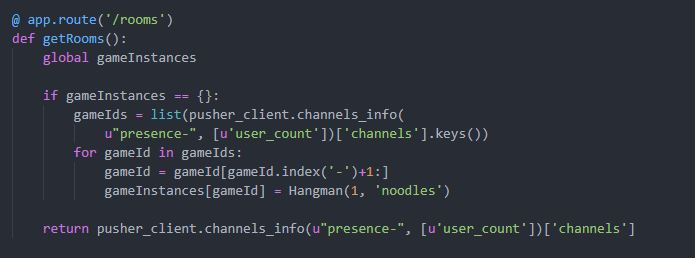
pusher\_client: a Pusher inicializációja

Útvonalak:

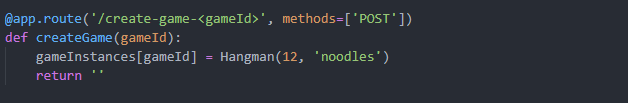
/login:username válótózóba POST metódussal elmenti a beérkaző felhasználó nevét



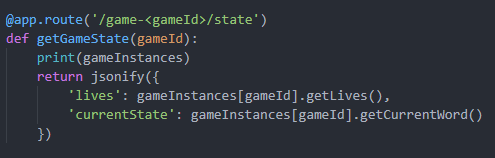
/rooms: feltölti a pusher api segítségével a szobákat a gameInstance dictionary-be, ha az nem üres, visszatérési értéke egy json objektum a csatorna nevével es a felhasználók számával



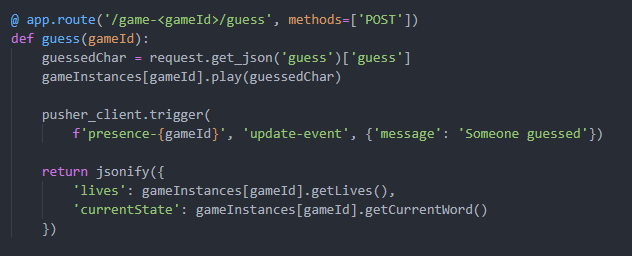
/create-game-<gameId>: létre hoz egy új kulcs érték párt a gameInstances dict.-be



/game-<gameId>/state: visszatér a pillanatnyi élettel és eddig kitalált szóval



/game-<gameId>/guess: POST metódussal megkapja tippelt betűt, meghívja a Hangman play metódusát és triggereli az update-event eseményt, visszatér a pillanatnyi élettel és eddig kitalált szóval

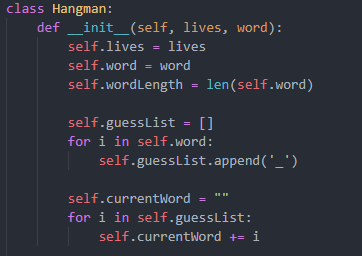


/pusher/auth: a Pusher presence szoba autentikációja

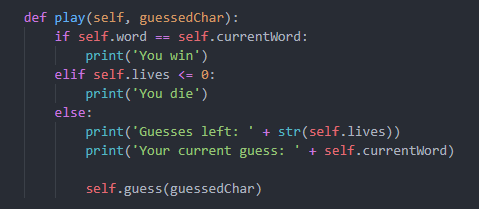


**hangman.py:osztály, a játék logikáját valósítja meg**

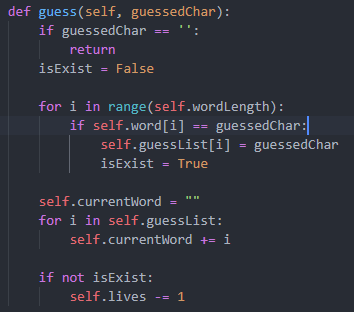
\_\_init\_\_(self, lives,word): beallítja a paraméterként kapot értékeket az osztály szintű változókra, kiszámolja a szó hosszát, létrehozza az eddig tippelt betűk és az eddig kitalált szó listáját



play(self, guessedChar): ellenőrzi, hogy nyert-e vagy veszített-e a játékos, ellenkező esetben meghívja a guess metódust



guess(self, guessedChar): ellenőrzi, hogy a karaktert adtunk-e meg, illetve, hogy benne van-e a szóban3



getCurrentWord(self): getter. eddig kitalált szó

getLives(self): getter, pillanatnyi életek