A kliens az üzenetet elküldi a szervernek, de mást nem csinál vele, csak a szerver által kapott üzenetre reagál, így biztos, hogy mind a két kliensen ugyanaz történik (pl.: nem rajzolja ki a saját pöttyét amíg a szerver vissza nem üzente a lépést mindkét játékosnak).

Minden üzenet egy karakterrel kezdődik, ami megmondja, mit kell vele csinálni.

Ha több részből áll az üzenet, akkor ’;’ -vel válasszuk el, pl.: „J;Gipsz Jakab”, a szóköz azért nem jó, mert az a chat üzenetben is lehet. A chatben gondoskodni kell róla, hogy az üzenetben ne legyen pontosvessző, vagy más karaktert is lehet választani.

A nevek csak azért kellenek, hogy a chat-ben ki lehessen írni, a szerver úgyis a client sockettel fogja azonosítani a klienseket, illetve szerveroldalon a játékosokat névvel érdemes azonosítani (vagyis nem lehet két egyforma).

A játékhoz 20x20-as pálya (400x400-as canvas) elég - szerintem -, a lépéseket a szerveren is nyilván kellene tartani, már csak azért is, mert a győztest ott érdemes ellenőrizni, hogy azt ne kelljen kétszer.

kliens szerver

küldő kliens másik kliens

**J**(oin) *név*: a kliens saját neve **L**(ist) *nevek* -

pl.: J;Walaky L;Grosics;Buzánszky;Lóránt

A saját nevét ne írja ki a kliens.

**G**(et): lista kérése **L**(ist) *nevek -*

Ezt update-hez lehet használni.

**P**(lay) *név*: a meghívott neve P(lay) *név*: a meghívott neve P(lay) *név*: a küldő neve

Szende hívja Kukát

Szende: P;Kuka P;Kuka P;Szende

Ha a szerver egy játékost elküldött játszani, akkor azt már ne lehessen meghívni és ő se hívhasson mást amíg tart a játék. Ne programozzunk multitaskingot (bár az sem lenne nehéz...).

**A**(ccept) név: elfogadás A;Kuka A;Szende

A megívás elfogadása.

**D**(ecline) D;Kuka D;Kuka

A meghívott elutasította.

**S**(tep) név *x y: (pl.: S;Kuka;4;6)* S;név;x;y S;név;x;y

A kliens dolga, hogy olyan mezőt ne küldjön, ami már foglalt, a névből lehet tudni, hogy saját lépés vagy az ellenfélé.

**C**(hat) név *üzenet*: C;név;üzenet C;név;üzenet

**W**(in) *név*: W;név W;név

Győzött valaki

**B**(reak): abbahagyás B B

Ezt csak játék közben lehessen elküldeni.

Összeszögeltem egy kliensoldali vázat és feltettem a Git-re (működik is, értelemszerűen kell módosítani).

* GameController: a GUI JAT szál, ez olvassa és küldözgeti az üzeneteket a GameNET-nek, bekommenteztem, hova kell megírni a lényeges kódokat. A go() eljárás indítja el azt a szálat, ami kezeli az üzeneteket (ez most addig fut, amíg a GameNET.isDoit == true).

Ha Pane.Canvas-ra rajzolunk, akkor egy ilyen eseménykezelőt kell írni a mező kiválasztásához:

*Pane.*setOnMouseClicked(e -> {

(int)Math.floor(e.getX() / 20) /\* X koordináta \*/

(int)Math.floor(e.getY() / 20) /\* Y koordináta \*/

...

}

400x400-as canvason 20x20-as lesz egy mező, ezzel a módszerrel 0-19 közöttiek lesznek a koordináták.

* GameNET: ez végzi (majd) a kommunikációt a szerverrel, én most már csak ezzel fogok foglalkozni.
* GameWindow: a GUI
* Main: (ezt nem mondom meg, mi).