

Proyecto de Metodología de la Investigación

Instructivo del juego “La Tripulación”



Alumnos: Bulich, Yamila; Martin, Mariano; Pasini, José.

Fecha: 23/6/2021

Link de Github:

<https://github.com/UTN-FRM-TUP/ejercicio-1-tm-grupo-p.git>

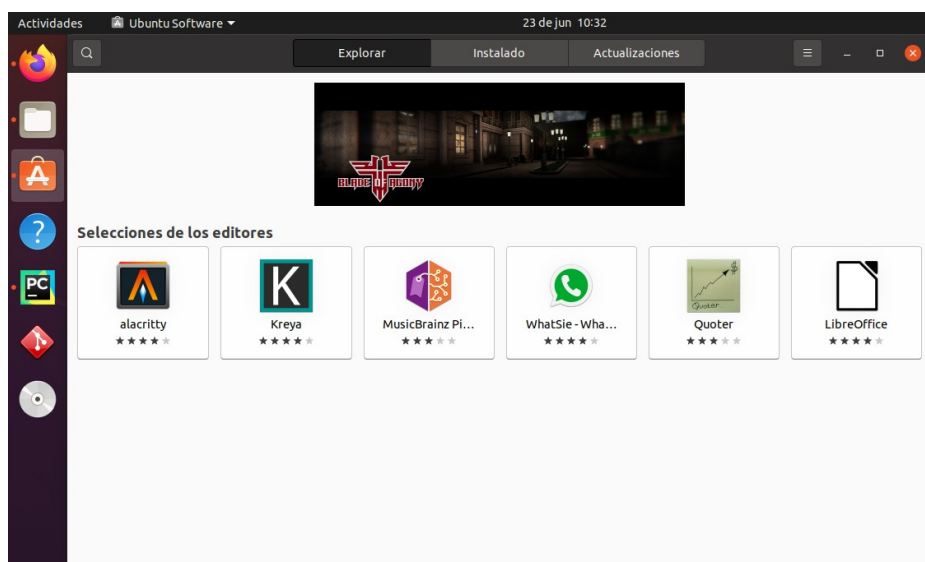
En qué consiste:

Nuestro proyecto consiste en crear una versión del juego de mesa “La

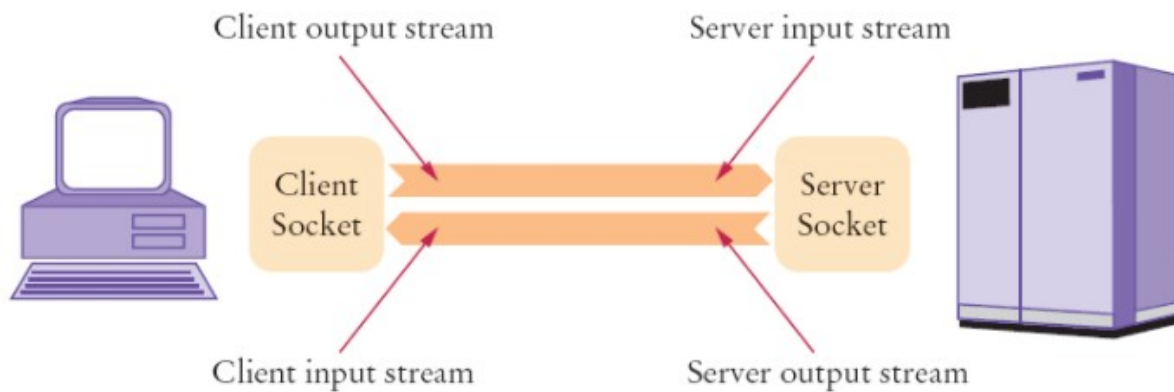
tripulación” para computadores, usando el lenguaje Python (ya incluido en el sistema operativo Ubuntu de Linux) en el IDE PyCharm.



Explicación de la instalación de pycharm usando Ubuntu Software:
Primero que nada lo abrimos, en el buscador de arriba a la izquierda tecleamos Pycharm y debemos instalar la versión de la comunidad PyCharm CE.



Para llevar a cabo esto, tuvimos la necesidad de crear un servidor para que el juego se pueda jugar de varios jugadores ya que, llevando este juego a la práctica no puede ser jugado por dos o más personas en una computadora por lo que, los demás jugadores verían las cartas del primero.



Una vez ya creado el servidor usando sockets teníamos que empezar con la creación del juego y su interfaz gráfica ya que este tipo de juego resultaría muy confuso de jugar en consola. Para hacer esto decidimos usar PyGame una librería para crear interfaces gráficas de juegos.

Instalación

```
sudo apt install python3.9
```

Ingresar a la terminal de linux y escribir lo siguiente:

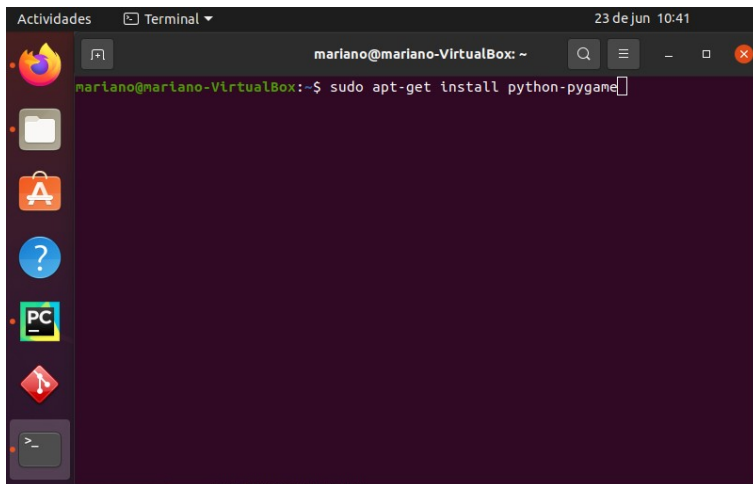
```
sudo apt update
```

Actualiza la lista de paquetes disponibles y sus versiones, pero no instala o actualiza ningún paquete. Esta lista la coge de los servidores con repositorios que tenemos definidos en el sources.list.

```
sudo apt upgrade
```

Una vez el comando anterior ha descargado la lista de software disponible y la versión en la que se encuentra, podemos actualizar dichos paquetes usando este comando. Instalará las nuevas versiones respetando la configuración del software cuando sea posible.

Instalación de PyGame en el sistema hay que teclear en la terminal el siguiente comando : `sudo apt-get install python-pygame` y dar enter.



Luego teclear en terminal esto que sera lo que instale el pip:

```
sudo apt install python3-pip
```

Luego teclear en terminal esto que acaba la instalación de pygame:

```
pip3 install pygame
```

Luego de instalar pygame reiniciar la máquina virtual.

Guía de arranque del juego para probarlo individualmente:

Para empezar necesitamos saber la ip de nuestro internet más específicamente el inet que en la imagen es: 10.0.2.15. Para hacer esto hay que abrir la consola de linux y teclear `ifconfig` (en el caso de no funcionar se debe teclear `sudo apt install net-tools` en terminal, instalarlo y intentar de nuevo con `ifconfig`), ahí estará nuestra dirección ip la cual hay que pegar en clase configuración del proyecto donde marca la flecha roja entre las comillas.

```

mariano@mariano-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::327d:6346:3027:dec7 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:0c:8b:05 txqueuelen 1000 (Ethernet)
    RX packets 105785 bytes 156855358 (156.8 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 47504 bytes 3120809 (3.1 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Bucle local)
    RX packets 530 bytes 53038 (53.0 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 530 bytes 53038 (53.0 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

```

LaTripulacion5 > configuracion.py
1 # client settings
2 ANCHO = 1350
3 ALTO = 680
4
5 # server settings
6 DIRECCION_SERVIDOR = '***.***.*.*' # Nos conectamos a la red local, podemos obtener el número de ip con la consola de linux
7 #tipiando "ipconfig" nos dará el número de ip.
8 PUERTO = 5555 # Este es un puerto que generalmente se encuentra desocupado.
9
10 # game settings
11 PUNTOS_TOTALES = 100

```

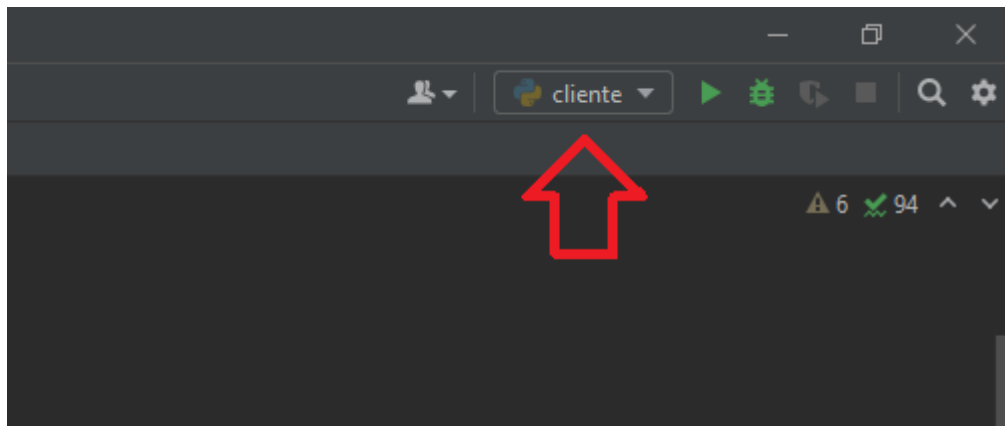
Una vez cambiada la ip tiene que dirigirse a la clase servidor y iniciarlo

```

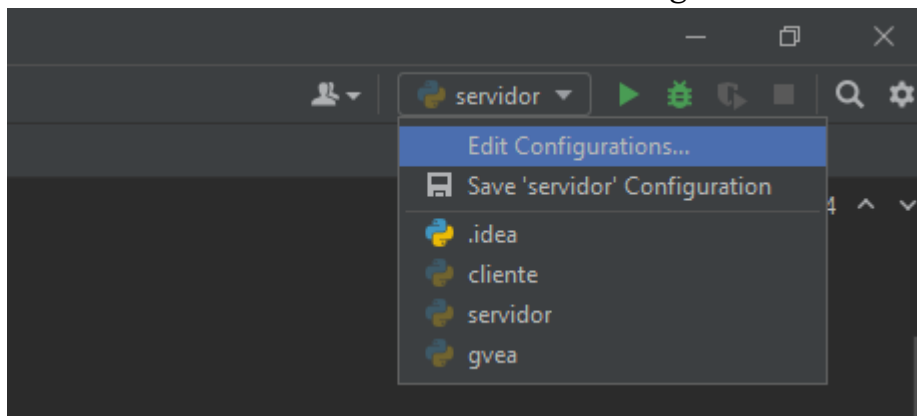
PC File Edit View Navigate Code Refactor Run Tools VCS Window Help LaTripulacion5 - servidor.py
LaTripulacion5 > servidor.py
1 import socket
2 from _thread import *
3 import pickle
4
5 from juego import Juego
6 from configuracion import *
7
8 #servidor= '192.168.1.103'
9 #puerto=5555
10
11 # Este es el script del ser

```

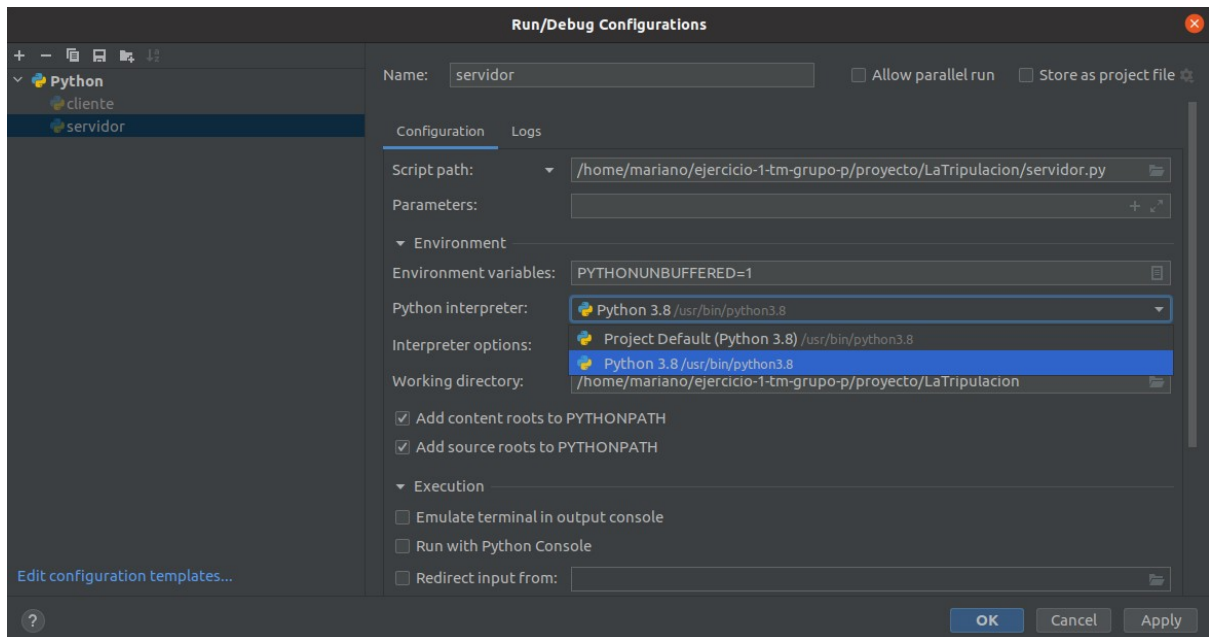
En caso de no poder ser iniciado el servidor por un error hay que cambiar el intérprete de esta forma; Hay que ir a esta pestaña y seleccionarla.



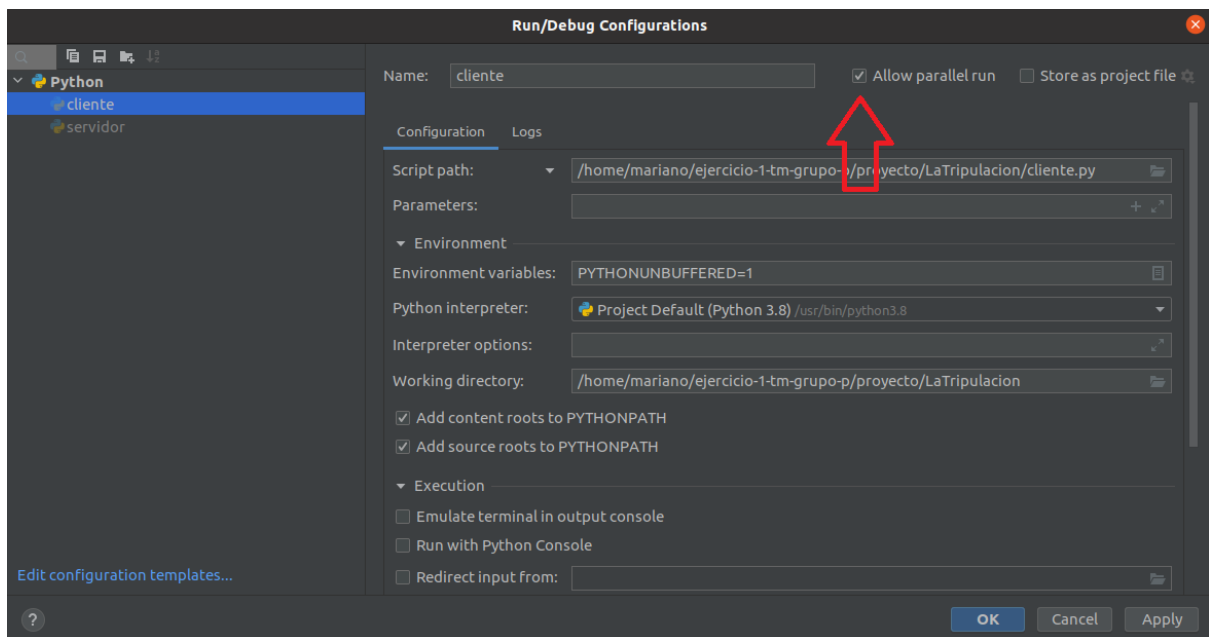
Una vez abierta seleccionar Edit Configurations



Se abrirá la siguiente pestaña y tendrá que clicar en Python Interpreter
Y seleccionar esta la segunda opción Python 3.8:

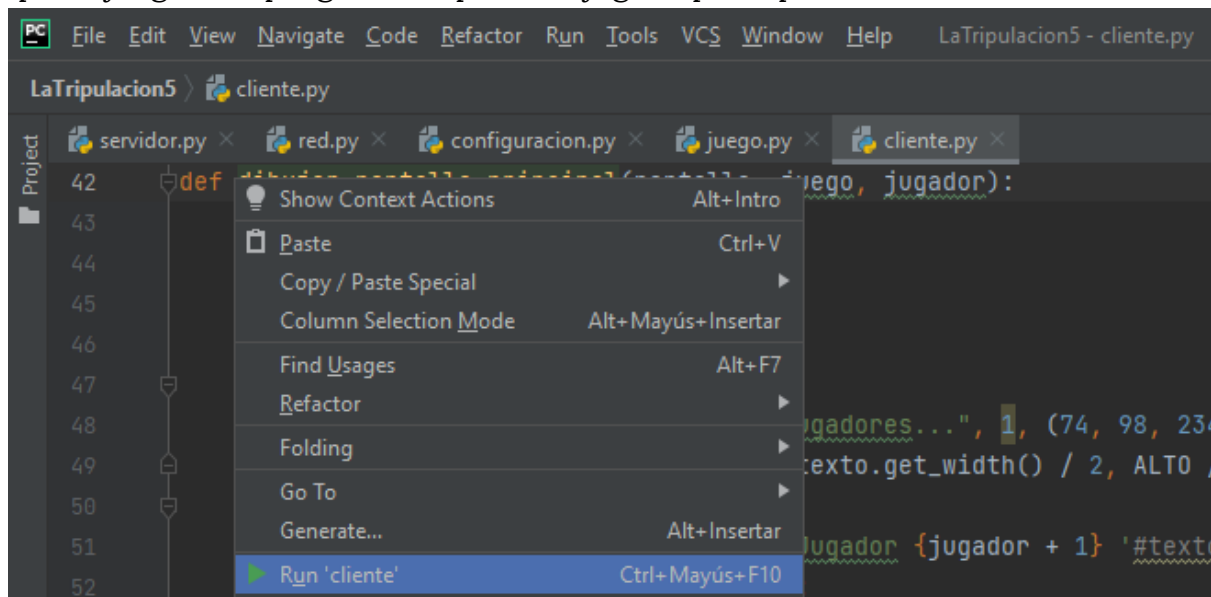


Además de suma importancia habilitar el: ALLOW PARALLEL RUN en la clase cliente

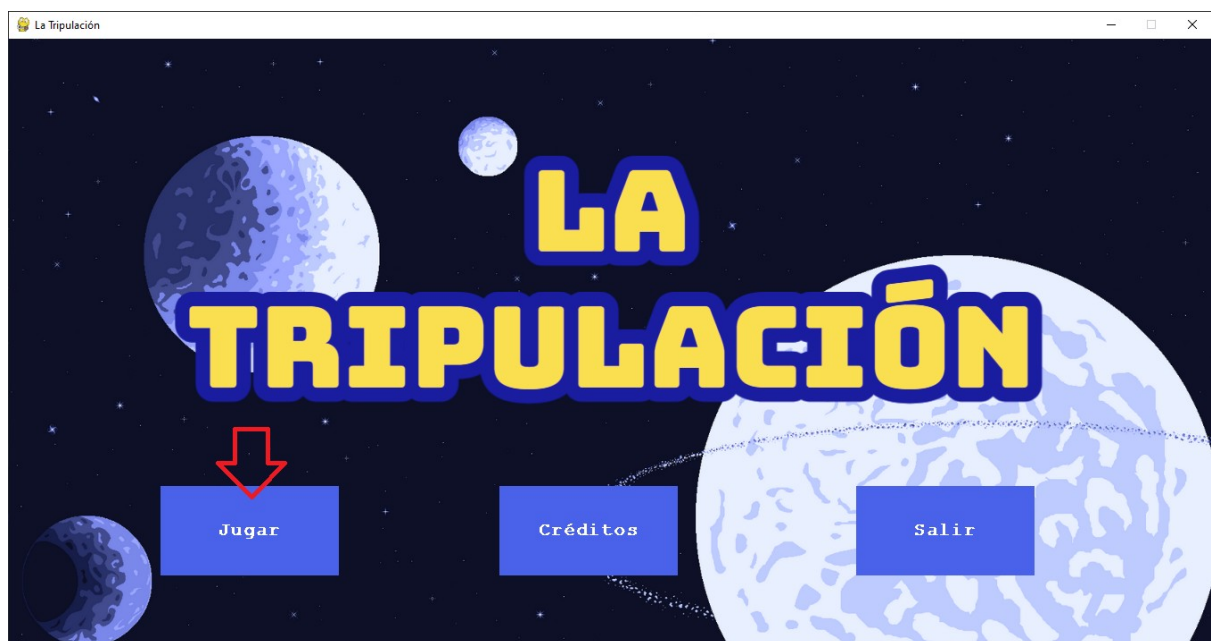


Una vez iniciado el servidor diríjase a la clase cliente e iniciarlo 4 veces ya

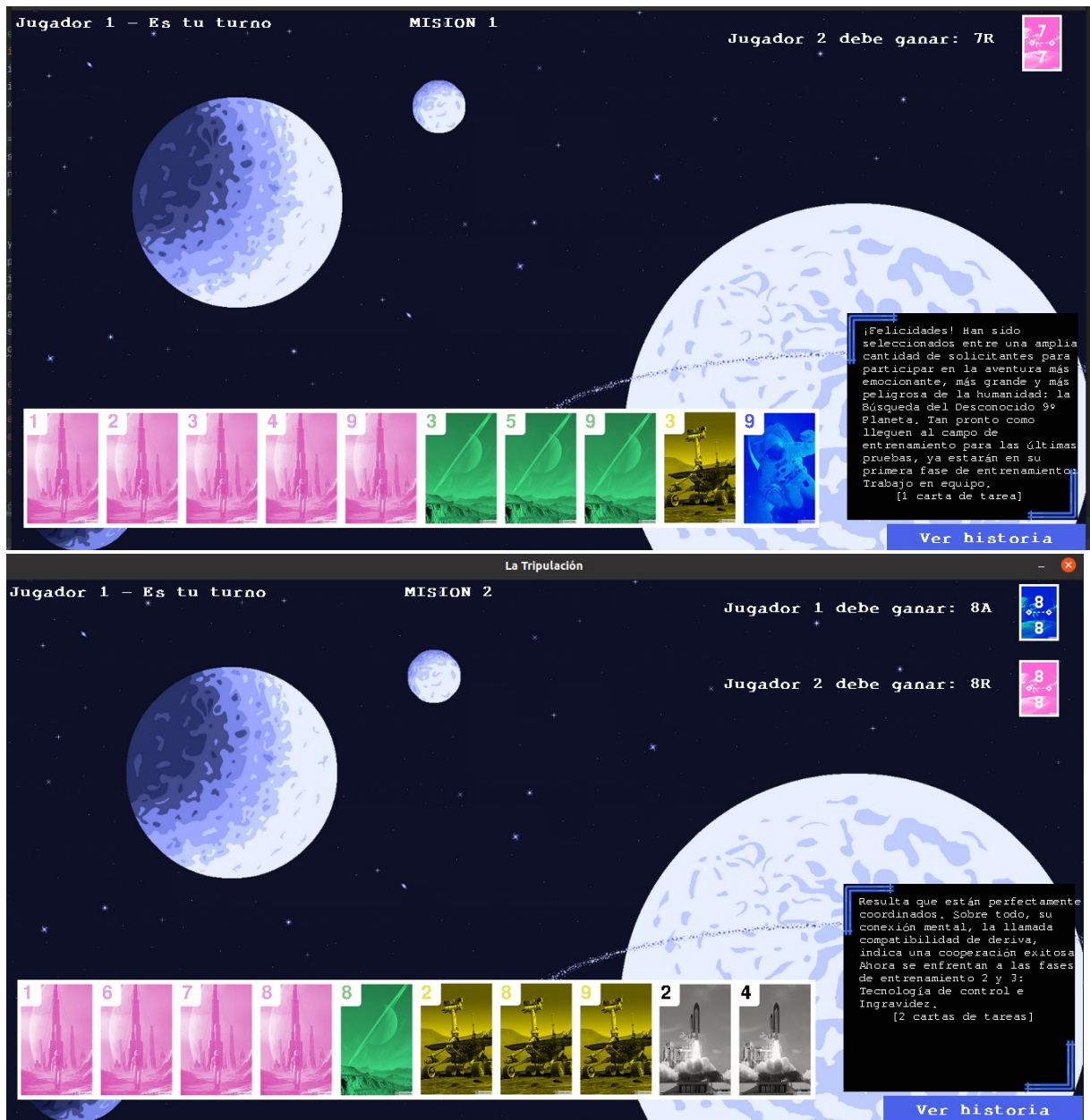
que el juego está programado para ser jugado por 4 personas.

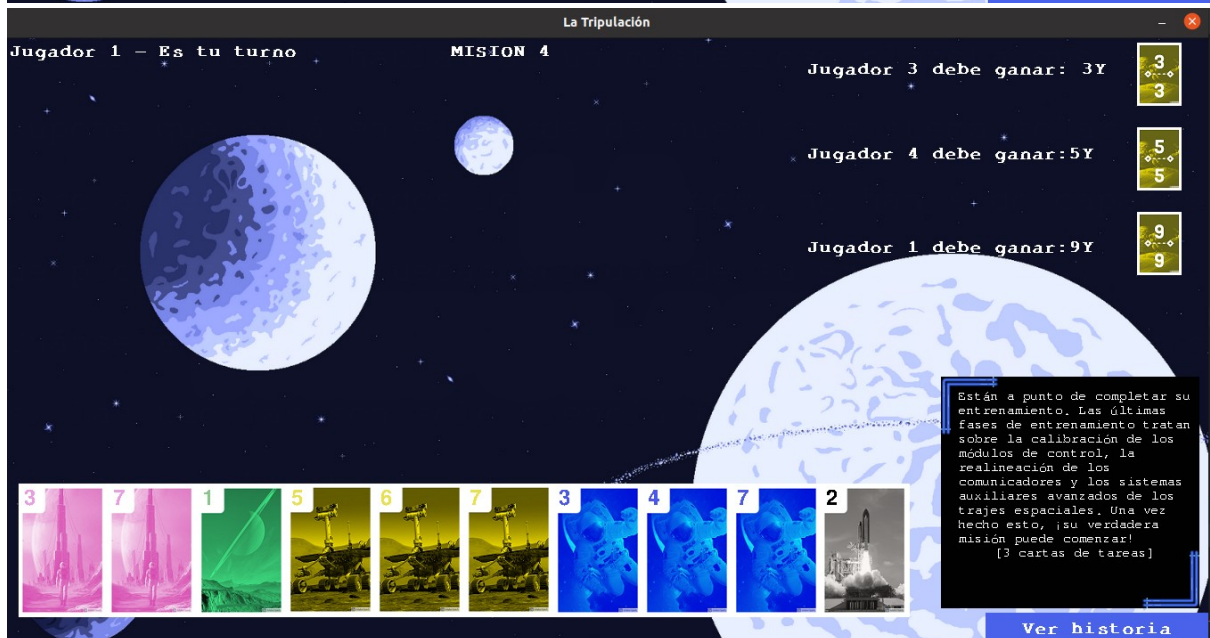
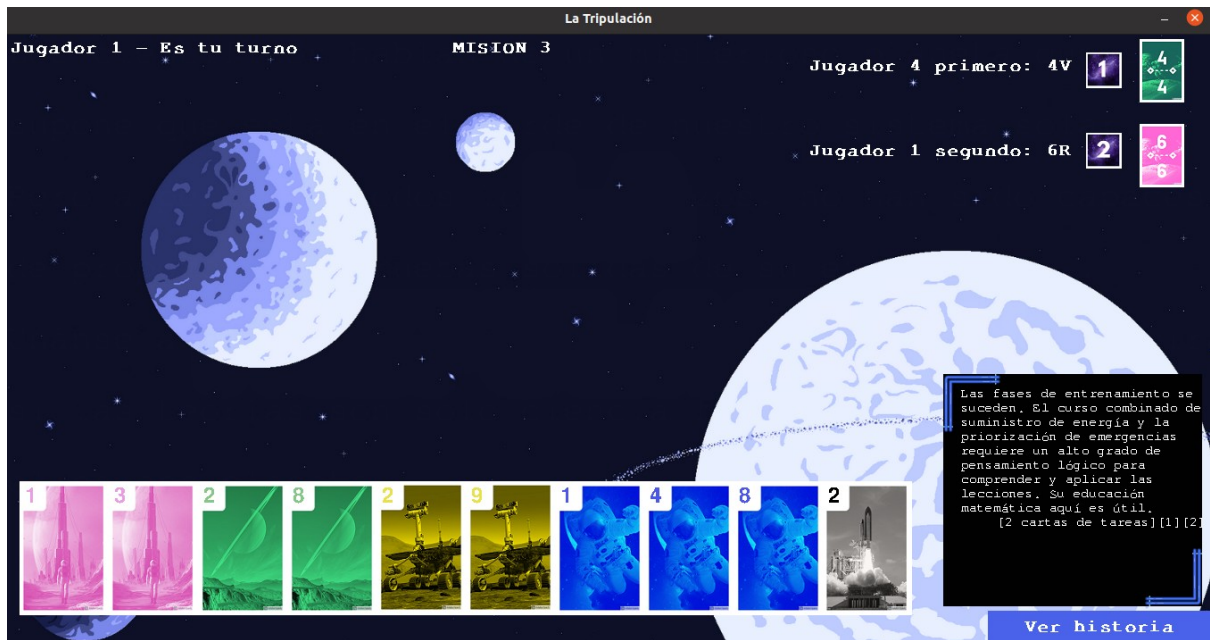


Una vez estén todos los clientes abiertos presione jugar en cada uno de ellos.



Imágenes in game:







Hecho eso se puede comenzar a jugar por lo que daremos una breve introducción a la mecánica del juego.

Mecánica del juego:

El juego consiste en cumplir misiones ganando bazas, las bazas se ganan al tirar una carta que tenga mayor valor comparada a las que lanzaron otros jugadores.

En total dentro del juego hay 40 cartas jugables, 36 “normales” y 4 denominadas “cohetes”.

Las “normales” van del 1 al 9 por colores (amarillo, verde, rosa, azul) y los “cohetes” que van del 1 al 4 además de tener más valor que las normales.

Si el primer jugador en tirar lanza una carta por ejemplo: “1 verde” todos los jugadores están obligados a tirar una carta verde a menos de no tener carta de ese color, en ese caso hay dos caminos:

- Puede tirar una carta normal de otro color pero ésta perderá así su valor sea más alto que el “1 verde” como por ejemplo: un “9 azul”.
- Puede tirar un cohete que siempre ganara ante una carta normal a menos que otro jugador lance un cohete de mayor valor.

Aclarado esto pasamos a explicar en qué consisten las misiones:

Las misiones son objetivos que nos da el juego como por ejemplo: “El jugador 1” debe ganar la baza que contenga el 4 rosa, por lo que los demás jugadores al ser un juego cooperativo deberán ayudarlo a cumplir la misión así poder pasar a la siguiente.

Vamos a hacer un pequeño ejemplo práctico de la anterior situación; “El jugador 1 debe ganar la baza contenedora del 4 rosa”.

verde; rosa; amarillo; azul; cohete

Cartas del jugador 1: 1, 7, 8, 3, 1, 9, 4, 1, 5, 8.

Cartas del jugador 2: 2, 4, 5, 7, 8, 9, 3, 5, 1, 4.

Cartas del jugador 3: 4, 1, 4, 6, 7, 3, 3, 9, 6, 2.

Cartas del jugador 4: 2, 3, 5, 2, 7, 8, 6, 9, 6, 2.

Empieza jugando el jugador 1:

Turno del jugador 1: Tira 7.

Turno del jugador 2: Tira 4.

Turno del jugador 3: Tira 6.

Turno del jugador 4: Tira 2.

Ganó la baza el jugador 1 y cumplió la misión ya que se llevó el 4 rosa – Este ejemplo corresponde a la primera misión del juego.

El juego se puede jugar de 4 jugadores usando ZeroTier cada uno desde su pc

conectados a la misma red local.

TABLA DE MISIONES

Misión 1: Se selecciona una carta al azar que debe ser ganada por uno de los cuatros jugadores seleccionados al azar sin orden específico.

Misión 2: Se seleccionan dos cartas al azar que deben ser ganadas por dos jugadores seleccionados al azar sin orden específico.

Misión 3: Se seleccionan dos cartas al azar que deben ser ganadas en un orden específico indicado por el juego seleccionando al azar que jugadores tendrán que obtenerla.

Misión 4: Se seleccionan tres cartas al azar que deben ser ganadas por tres jugadores seleccionados al azar sin orden específico.

Distribución de roles y tareas

Yamila se encargó de crear el servidor, conectarlo a la red y crear la interfaz del cliente, ya que la documentación estaba en inglés y podía entender cómo implementarlo.

José hizo el diseño de las cartas (ya que no disponíamos de las cartas originales digitales). Se encargó de la creación de las bazas y la comparación de números y colores entre las cartas.

Mariano creó los jugadores y métodos que reparten las cartas asignándoles a los jugadores mostrándolas en la interfaz gráfica de Pygame. Creó la clase de tarea y la asignación de las mismas por misión.

Entre todos creamos las diferentes misiones dentro del juego, asimismo la creación de las cartas tarea, y las cartas tareas realizadas.misiones

Planificación versus ejecución

Habíamos planificado crear nuestra versión digital del juego “The Crew”, ya que contamos con el juego de mesa y no estaba disponible en ninguna plataforma.

En cuanto a la ejecución, primero buscamos códigos de otros juegos de cartas para conocer la librería Pygame y cómo se organizaban las clases. Luego

decidimos crear el servidor por un lado y por otro manipular las cartas en la interfaz, para luego unir ambos códigos. Comenzamos trabajando en equipo ya que no contábamos con los conocimientos necesarios para la división de tareas y su ejecución. Una vez que sentamos las bases del juego, pudimos trabajar cada uno por su cuenta, siguiendo la distribución de trabajo planteada anteriormente.

Historial de git usando **history|grep git|tail -n 20**:

```
mariano@mariano-VirtualBox: ~  
mariano@mariano-VirtualBox:~$ history|grep git|tail -n 20  
118  git merge  
119  git status  
120  git add .  
121  git commit -m "subiendo misiones 1 y 2"  
122  git push origin master  
123  git status  
124  git pull  
125  git status  
126  git add .  
127  git commit -m "Mision 3 y 4 de cliente"  
128  git push origin master  
129  git pull  
130  git add .  
131  git commit -m "terminando"  
132  git push origin master  
133  git pull  
134  git add .  
135  git commit -m "cambiando nombre de carpeta principal"  
136  git push origin master  
137  history|grep git|tail -n 20  
mariano@mariano-VirtualBox:~$
```

```
jose@jose-desktop: ~/ejercicio-1-tm-grupo-p/proyecto/LaTripulacion  
jose@jose-desktop:~/ejercicio-1-tm-grupo-p/proyecto/LaTripulacion$ history|grep git|tail -n 20  
572  git add .  
573  git commit -m "Interfaz grafica de las cartas en Cliente"  
574  git push origin master  
575  git fetch  
576  git merge  
577  git commit -am " Solo arreglo la clase juego que se movió"  
578  git push origin master  
579  git status  
580  git commit -am "Con esto terminarian todas las misiones, solo falta cargar las imágenes de las cartas"  
581  git push origin master  
582  git pull  
583  git status  
584  git commit -am "Acá van las cartas tachadas para corroborar las funciones"  
585  git push origin master  
586  git  
588  git fetch  
589  git merge  
590  git pull  
591  history|grep git|tail -n  
592  history|grep git|tail -n 20  
jose@jose-desktop:~/ejercicio-1-tm-grupo-p/proyecto/LaTripulacion$
```

```
yami@yami-VirtualBox:~/PycharmProjects/TRip/ejercicio-1-tn-grupo-p$ history|grep git|tail -n 20
```

```
133 git status
134 git add .
135 git commit -m "main"
136 git push origin master
137 git fetch
138 git pull
139 git push origin master
140 git pull
141 git fetch
143 git remote -v
150 git pull
151 git ifconfig
153 git status
154 git add proyecto/LaTripulacion/juego.py
155 git commit -m "arreglo indentado juego"
156 git push origin master
157 git pull
158 history|grep git
159 history|grep git -n 20
160 history|grep git|tail -n 20
```