

# Proyecto Programado 2

1<sup>st</sup> Roberto José Garita Mata  
*ITCR. Ingeniería en Computación*  
*IC 1803 Taller de Programación*  
Cartago, Costa Rica  
robertojose99@hotmail.com

2<sup>nd</sup> Danny Jossiel Jiménez Sevilla  
*ITCR. Ingeniería en Computación*  
*IC 1803 Taller de Programación*  
Cartago, Costa Rica  
danny.jimenez@estudiantec.cr

## I. INTRODUCCIÓN

El siguiente problema que vamos a solucionar es un juego llamado Exploding Kittens. Este juego consiste repartir de una baraja de 56 cartas entre 4 jugadores y cada jugador tendrá una carta llamada defuse y 7 cartas aleatorias sin incluir bombas, luego cada jugador en su turno puede jugar alguna de sus cartas (skip, future, attack, nope y combinaciones de comodines). Después de jugar sus cartas tendrá que tomar una carta de la baraja y esperar que no le salga una carta bomba, si sale una carta bomba tiene la posibilidad de utilizar una carta defuse para así no perder, finalmente se continua hasta que solo quede un jugador. Nota: la cantidad de bombas en un juego depende de la cantidad de personas que esten jugando, por ejemplo si hay 4 jugadores en el juego solo deberán haber 3 bombas.

Para este trabajo programado, utilizamos el uso de listas para la manipulación de las cartas del juego, además de esto también se usaron para la cantidad de jugadores, para los mazos de cada jugador, para la baraja central y para las cartas desechadas (cartas usadas).

Para la comunicación entre cliente y servidor, nosotros usamos los sockets e hilos para mantener comunicación entre ellos 2, para así mantener una comunicación estable entre ambos y que el programa no se caiga y tenga una comunicación constante, independientemente del tiempo que se mantenga abierto el servidor.

Cuando planificamos la distribución de todas las partes del proyecto, ya sabíamos desde un inicio que cuando iríamos a realizar la parte de los sockets e hilos, sabríamos que iba a ser difícil ya que es un tema nuevo de programación y tendríamos que empezar a investigar desde cero sobre el tema. Tampoco sabíamos desde un inicio que iba a ser necesario el uso de una librería llamada Json.

Durante la realización del proyecto nos encontramos con diferentes problemas, los cuales fueron la manipulación de datos con Json y con los sockets e hilos.

## II. DESCRIPCIÓN DE LA SOLUCIÓN

Para llegar a la solución de este proyecto, primero se dividió el proyecto en diferentes partes: el juego principal, la conexión entre aplicaciones (sockets, hilos) y Json, por último se procedió a hacer cada parte correspondiente para luego juntar

todas las partes necesarias y así que el proyecto funcionara de manera correcta.

Para la parte del juego principal se utilizaron los métodos de las clases (POO), el cual la función principal se llama inicializar, la cual corre la ventana principal, es quién carga las imágenes, esta pendiente de las acciones de los usuarios y esta pendiente de los datos que envia el servidor al cliente y viceversa. Esta función inicializar no recibe parámetros ya que es la función que inicia el programa.

En la parte de conexión entre aplicaciones (sockets, hilos), la función principal lo que hace es crear un servidor, genera 2 hilos secundarios para aceptar una conexión y procesar los datos, también esta a la escucha de la palabra salir, para así cerrar la conexión del servidor. Al igual que la función inicializar, esta función que crea el servidor igual no recibe parámetros ya que es quién crea el servidor.

Por otra parte, en la creación del juego también utilizamos una librería llamada Json, el cual hace la manipulación de datos, hace la lectura de datos, y hace la escritura de los datos necesarios que utiliza el juego.

Además de las funciones principales implementadas, dentro de cada una de estas, se encuentran muchas otras mas funciones, con trabajos en específico, las cuales tienen una explicación dentro de la documentación interna del código del juego, además de su lógica y sus parámetros correspondientes a cada una.

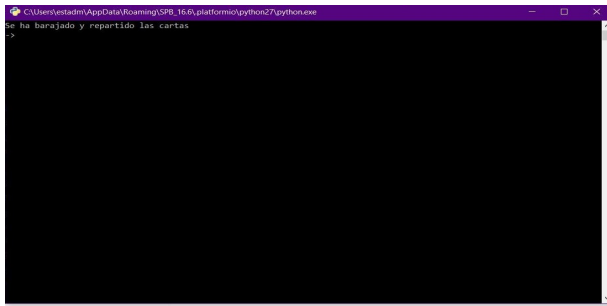
En cuanto a la comunicación de funciones del juego, dentro se encuentran muchas mas funciones las cuales hacen llamadas a otras funciones, para así que el juego corra de manera correcta.

En los problemas específicos encontrados, se resolvió el problema con los sockets e hilos, el cual se solucionó con un desbloqueo de las líneas de escucha de datos para poder escuchar a más de un cliente a la vez. Con respecto a la librería Json, logramos resolver también los problemas que teníamos con un poco más de investigación y algunos videos explicativos sobre el mismo y así ya logramos el resultado a que queríamos llegar.

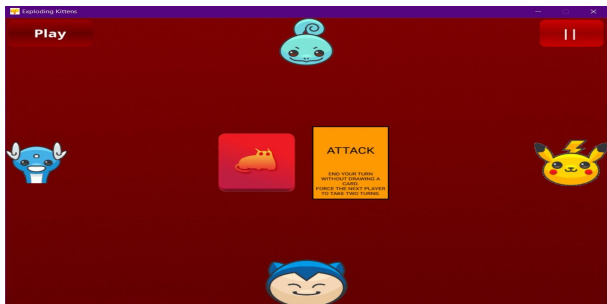
## III. MANUAL DE USUARIO

Este manual va dirigido hacia los usuarios que deseen jugar este juego llamado Exploding Kittens en una computadora. Este juego lo pueden jugar hasta 4 jugadores simultáneamente en una misma computadora. Este manual de usuario tal vez

no es para cualquier persona ya que el juego no esta tan elaborado, pero si puede ser para alguna persona que tenga conocimientos básicos de programación. Primero para iniciar el juego primero se debe de correr en algún lenguaje de programación que soporte el lenguaje de Python, luego se deben de tener 2 documentos abiertos, el cual uno es el servidor y el otro es el cliente (juego). Para iniciar el juego, se debe correr el documento del servidor, cuando ya este corriendo se verá una ventana como esta y por este lado estará listo.



Seguidamente se debe correr el documento del cliente dependiendo de la cantidad de jugadores que van a jugar, si por ejemplo van a jugar 4 se corre 4 veces el documento del cliente. Finalmente se desplegará una ventana como esta y ya los jugadores podrán disfrutar del juego. Si ya terminaron de jugar, simplemente presionan el boton de X para salir.



#### IV. ANÁLISIS DE RESULTADOS

##### A. ¿Qué funciona?

En este proyecto funciona la conexión entre el servidor y cliente, la importación de librerías, el manejo de datos a través de Json, la actualización de ventanas a través de la interfaz pygame y el juego en sí según sus reglas, además con los puntos extra que eran con las cartas adicionales (attack, skip, nope, comodín de 3 cartas).

##### B. ¿Qué no funciona?

En este proyecto lo que podríamos decir que no funciona, es un chat de juego para la comunicación entre jugadores como lo es el juego real para móvil.

##### C. ¿Qué se pudo haber hecho de mejor manera?

Consideramos que a la hora de repartir cartas se pudo haber hecho mucho mejor, con menos líneas de código y se pudo haber hecho con una mayor eficiencia. También pensamos que las animaciones se pudieron mejorar para una mayor calidad de gráficos y transiciones. Con respecto a la interfaz se pudo haber de una manera más amigable para el uso del usuario y con mas menús para que fuera más interactivo con el usuario. Por último también se pudo haber realizado una interfaz para el usuario del servidor, en caso de que no supiera nada de programación.

##### D. Lecciones Aprendidas

Para este proyecto ya sabíamos bastantes cosas sobre programación que implementamos a lo largo de la realización del mismo, pero podríamos decir que con este proyecto aprendimos a la utilización de sockets e hilos, a la manipulación de librerías y de datos con Json.

#### V. CONCLUSIONES

En conclusión lo que presentamos en este trabajo fue un juego programado en en lenguaje de programación Python, llamado Exploding Kittens, el cual es un juego creado para un máximo de 4 jugadores en una misma computadora. El programa consta de 3 principales partes que son el juego principal, la comunicación entre aplicaciones (sockets e hilos) y la librería Json. Los resultados de este programa fueron bastantes prometedores ya que el juego funciona con 4 jugadores jugando al mismo tiempo, el manejo de datos a través de la librería Json funciona, la actualización de las ventanas a través de la interfaz pygame funciona y lo más importante el juego funciona por sí solo. Pensamos que se pudo mejorar en varios aspectos, pero el más importante puede ser que a la hora de repartir las cartas a los jugadores, no es la forma más eficiente de hacerlo. Finalmente si hubieramos tenido más tiempo hubiéramos hecho una intefaz para el usuario del servidor porque pensamos que sin interfaz es un poco complicado para quién lo vaya a usar.

#### VI. LINK PARA INGRESAR AL PROGRAMA

<https://github.com/robertogarita/Proyecto-2-Taller-de-Programacion.git>

#### REFERENCES

- [1] Conocimientos propios de programación
- [2] Exploding Kittens, "Exploding Kittens Reglas del juego". 2018. (PDF brindado por el profesor)
- [3] H. Castilla, "PYTHON SOCKETS - CHAT TUTORIAL". Julio 1, 2016.
- [4] M. Bussonnier, "JSON encoder and decoder". Abril 9, 2019.
- [5] M. Islington, "Socket — Low-level networking interface". Enero 1, 2020.