

Proyecto: Explodding Kittens

1st Ian Murillo Campos

2nd Josue david Quesada Garita

I. INTRODUCCIÓN

El problema principal a solucionar es la creación en código Python de una versión del juego "Explodding Kittens" utilizando Sockets para la jugabilidad con 2-3 o 4 personas, aparte, se requiere de una interfaz gráfica, la cual se realizara con la herramienta de PyGame para su creación, el juego consiste en cartas con diferentes poderes, donde la manera de ganar es siendo la última persona sin eliminar, teniendo cartas como explosion, que son la derrota instantánea, la defuse para evitar el destino anterior o los comodines los cuales funcionan como otra manera de ganar. Como se mencionó se utilizará Python para la creación del código de este juego, junto con las funciones que ofrecen Sockets y PyGame para la creación de interfaz gráfica y compatibilidad para de entre a 2 y 4 jugadores, aparte se utilizara PyGame para crear las cartas que aparecerán en la pantalla de cada jugador y manejar el juego.

Se realizo un código que incluye una interfaz hecha con la herramienta PyGame y para el multijugador se utilizó Sockets, el código de la interfaz se basa en poner un fondo, una "caption" y un icono, relacionados con el juego, aparte se utiliza para definir las imágenes de las cartas y el mazo, el código del juego consiste en definir las funciones de las cartas utilizando funciones de listas, las cuales se llaman a la hora de dar click viéndolas como una especie de botón, al dar click en la carta se realizará su función correspondiente así para el que juegue en servidor como en cliente, y el código del servidor y los clientes, no obstante la solución considerada optima no fue alcanzada.

II. MANUAL DE USUARIO

Luego de correr el juego simplemente se debe empezar a jugar

A. *paso 1*

Se toma automáticamente una carta de la pila y se decide si jugar alguna carta del mazo del jugador o pasar al turno del siguiente jugador.

B. *Paso 2*

Existen diversas cartas que ayudan a los jugadores a conseguir la victoria, siendo estas las cartas: 1. Defuse: si aparece una carta explosion la invalida y se pasa al turno del siguiente jugador.

2. Favor: obliga a cualquier jugador a darte una carta de su mano

Identify applicable funding agency here. If none, delete this.

3. Shuffle: Si se juega se vuelve a barajar la baraja de cartas a tomar.

4. See the future: El jugador que juega esta carta tiene la posibilidad de ver las primeras 3 cartas de la baraja.

C. *Paso 3*

Se pueden jugar las cartas que se quieran en cada turno, este termina al tomar una carta de la baraja y se pasa al siguiente jugador.

III. ANALISIS DE RESULTADOS

El mayor problema fue la implementación de Sockets y la búsqueda de información sobre los diferentes módulos de PyGame debido a las no muy claras explicaciones encontradas en línea. Por otro lado, el orden del código y las diferentes maneras de aplicar una interfaz gráfica complicaron el proceso ya que no se sabía cuál resultaría ser la más sencilla o práctica. Por otro lado, disponer de una fuente que aportará recomendaciones acerca de cómo realizar juegos de calidad habría supuesto una diferencia considerable en los resultados obtenidos, pues habría definido un modelo de desarrollo más eficaz resultando en una reducción de la cantidad de tiempo dedicado a la investigación que podría haberse utilizado para la elaboración del juego.

IV. CONCLUSIONES

El proyecto presentaba un nivel de dificultad considerable, pero los módulos de PyGame ayudaron a su desarrollo, del mismo modo, el sistema de Git fue también de gran ayuda para la transferencia de información con nuestro debido compañero y durante la realización del proyecto, se logró aprender sobre los diferentes módulos que ofrece la aplicación de PyGame. Se lograron adquirir algunos conceptos elementales de los requerimientos para la elaboración de juegos y el uso de PyGame, no obstante no se lograron encontrar fuentes suficientemente claras acerca de los sockets. Fue posible a través de este proyecto la importancia de organización y orden al programar dada la cantidad de código manipulada y la necesidad de comprender conceptos de programación orientada a objetos. Un ejemplo importante es la comprensión de las clases en Python y la forma en que se pueden crear estancias dentro de una clase.

REFERENCIAS

- [1]PyGame with Python 3 Game Development. 2014.
- [2]Sockets with Python 3. 2019.
- [3]M. Agarwal, "Implement a Multithreaded Python Server Using Threads", Learn Programming and Software Testing, 2020.

- [Online]. Available: <https://www.techbeamers.com/python-tutorial-write-multithreaded-python-server/>.
- [4] "Socket Programming with Multi-threading in Python [19] GeeksforGeeks", GeeksforGeeks, 2020. [Online]. Available: <https://www.geeksforgeeks.org/socket-programming-multi-threading-python/>.
- [5] R. Python, "Socket Programming in Python (Guide) – Real Python", Realpython.com, 2020. [Online]. Available: <https://realpython.com/python-sockets/handling-multiple-connections>.
- [6] "Socket Server with Multiple Clients — Multithreading — Python", Codez Up, 2020. [Online]. Available: <https://codezup.com/socket-server-with-multiple-clients-model-multithreading-python/>.
- [7] "A Newbie Guide to pygame — pygame v2.0.0.dev7 documentation", Pygame.org, 2020. [Online]. Available: <https://www.pygame.org/docs/tut/newbieguide.html>.
- [8] "8. Compound statements — Python 3.8.5 documentation", Docs.python.org, 2020. [Online]. Available: https://docs.python.org/3/reference/compound_stmts.html.
- [9] R. Python, "Python Exceptions : An Introduction – Real Python", Realpython.com, 2020. [Online]. Available: <https://realpython.com/python-exceptions/>.
- [10] "8. Errors and Exceptions — Python 3.8.5 documentation", Docs.python.org, 2020. [Online]. Available: <https://docs.python.org/3/tutorial/errors.html>.
- [11] R. Python, "The Most Diabolical Python Antipattern – Real Python", Realpython.com, 2020. [Online]. Available: <https://realpython.com/the-most-diabolical-python-antipattern/>.
- [12] H. user? and M. Lorne, "How can I know the number of clients connected to the server and return the number of the connected clients to the user?", Stack Overflow, 2020. [Online]. Available: <https://stackoverflow.com/questions/20621253/how-can-i-know-the-number-of-clients-connected-to-the-server-and-return-the-numb>.
- [13] [1] "Code Faster with Line-of-Code Completions, Cloudless Processing", Kite.com, 2020. [Online]. Available: <https://www.kite.com/python/answers/how-to-check-if-a-variable-is-a-list-or-a-tuple-in-python>:`text=Use isinstance() to check, type class and False otherwise.`
- [14] [2] "Aaron-Buckles/solitaire", GitHub, 2020. [Online]. Available: <https://github.com/Aaron-Buckles/solitaire>.
- [15] [3] L. Python, J. Rooy and A. Martelli, "Length of an integer in Python", Stack Overflow, 2020. [Online]. Available: <https://stackoverflow.com/questions/2189800/length-of-an-integer-in-python>.
- [16] [4] A. pyGame, "Allowing resizing window pyGame", Stack Overflow, 2020. [Online]. Available: <https://stackoverflow.com/questions/11603222/allowing-resizing-window-pygame>.
- [17] [5] "Build software better, together", GitHub, 2020. [Online]. Available: https://github.com/search?q=pygame.display.toggle_fullscreen_type=Codel=Python.
- [18] [6] "Python3 – Modules – Tutorialspoint", Tutorialspoint.com, 2020. [Online]. Available: https://www.tutorialspoint.com/python3/python_modules.htm.
- [7] Index-of.es, 2020. [Online]. Available: <http://index-of.es/Python/Beginning.Game.Development.with.Python.and.Pygame.From>.
- [20] [8] Buildmedia.readthedocs.org, 2020. [Online]. Available: <https://buildmedia.readthedocs.org/media/pdf/pygame/latest/pygame.pdf>.
- [21] [9] "R0X0RE0/Dragon", GitHub, 2020. [Online]. Available: <https://github.com/R0X0RE0/Dragon/blob/d7b92eee9165b65d56692fe7db5>
<https://stackoverflow.com/questions/39274460/pygame-fullscreen-display-flag-creates-a-game-screen-that-is-too-large-for-the-s/3929810739298107>.