

## Grupo N°6: Porfiao-man19

Pablo Riquelme  
Montserrat Saavedra  
Bárbara Reyes  
Mario Rozas  
Profesora: Anelis Pereira  
Ayudante: Angelo Romano  
Programación, sección 2.

### Descripción y diseño:

El proyecto se basa en generar conciencia sobre la pandemia que se vive actualmente en el mundo, para esto se recrea el juego pac-man con la temática del covid-19. La idea de utilizar un juego para concientizar es porque de esta forma la información llega a las personas de una manera más amigable y efectiva.

Las modificaciones gráficas utilizadas por un lado fueron reemplazar a los fantasmas por el virus, el pacman por una persona, los coins fueron cambiados por vitaminas, el fondo utilizado en el juego busca representar una ciudad y las zonas que se encuentran en peligro de infección por el virus. Por otro lado en la parte derecha de la pantalla se creó una ventana la cual entrega la información que se debe tener en cuenta acerca del virus y de esta forma lograr el objetivo principal del juego. En la parte superior de esta se encuentran la cantidad de vitaminas obtenidas y al lado de este la vida que tiene, en este caso una con la idea enseñarle a las personas que este virus y esta pandemia que se vive y puede llegar a ser mortal. En la parte inferior de la pantalla se encuentra Dr. simi para generar una cercanía con las personas.

### Código:

Para la creación del código se utilizó principalmente la biblioteca PYGAME, la cual es un conjunto de módulos de Python diseñados para escribir videojuegos. Las otras bibliotecas importantes utilizadas fueron TIME, que es un módulo para monitorear y usar el tiempo, y la biblioteca RANDOM que contiene una serie de funciones relacionadas con los valores aleatorios.

El código consta de dos partes principales, una es donde se definen las variables y las funciones, y la otra es dentro de un ciclo infinito en donde suceden las mayorías de funciones y se ejecuta el juego.

Para dar inicio al juego se utiliza "pygame.init()", que es para inicializar todos los módulos de pygame importados. Para mostrar en la pantalla lo que se desee, se utiliza la variable definida "pantalla.blit(variable, (x,y))", la cual será usada dentro del ciclo infinito.

Luego de empezar el ciclo a través de la función `pygame.time.get_ticks()` se empezará a contar el tiempo y gracias a ciertas condiciones haremos un contador limpio (desde 0) de tiempo el cual se utilizará dentro del código para distintas finalidades.

El ciclo `for event in pygame.event.get()` para recorrer los distintos eventos que tiene pygame, y luego con `event.type` lo usamos como condición, si este es igual a funciones tales como `pygame.KEYDOWN`, lo cual significa que se apreta una tecla, con lo cual sucederá algo. En todo el código se utiliza esto para empezar el juego, para mover al jugador y para salirse del juego con `pygame.QUIT`.

Sus detalles se definen con funciones como `pygame.display.set_caption`, que define el título de la ventana actual, mientras para el ícono de la ventana de visualización utilizamos `pygame.display.set_icon`, y para el tamaño e inicializar la ventana con `pygame.display.set_mode`. Con `pygame.image.load` cargamos las imágenes a utilizar que queremos dentro del código. Además utilizamos la función `pygame.mixer.music.play(10)` que se utiliza para repetir la música de fondo, y se repetirá 10 veces desde fuera del ciclo infinito.

Para la pantalla de inicio, se utilizará una variable llamada `condicion tiempo` con valor `False`, con ella se verá una condición que hará mientras su valor sea tal que aparezca momentáneamente todo el comienzo hasta que se cambie a `True`, que será cuando se presione cualquier tecla, y allí recién comenzará el juego.

Luego se construye el mapa dibujando rectángulos con `pygame.draw.rect`, en donde cada rectángulo está guardado en una variable con coordenadas con `pygame.Rect`, esto para luego crear una función `def rectdib(q):` que los ponga en pantalla, para así comprobar su posición e ir armando el mapa.

Luego definimos los personajes en forma de sprite con `pygame.sprite.Sprite` junto a su respectiva imagen, para que sean objetos de juego visible y tengan la posibilidad de chocar con otros sprites, además de otras tantas funciones. Para definir el movimiento para el personaje principal, se utilizan dos variables `vx` y `vy`, las cuales inician con un valor 0, y se van actualizando cada vez que el jugador aprete una tecla, estas variables van tomando valores de 5 o -5, las cuales corresponden un avance o retroceso en píxeles. Cuando el jugador suelta la tecla, el personaje actualiza su valor a 0, por lo que se queda quieto. El movimiento se produce a través de la función `variable.rect.move_ip`. Para los enemigos se realiza lo mismo con sus respectivas variables y funciones.

Como ambos deben colisionar con las paredes, se crean diferentes funciones llamada `choque` donde el número que acompaña a la función define los choques con las respectivas paredes. Dentro del loop infinito, se crearán variable llamadas `olx` y `oldy` para cada sprite, y al momento de chocar con la función `variable.sprite.rect.colliderect(x)` dentro de `choque`, las coordenadas de estos serán las variables antes mencionadas, por lo que no podrán pasar más allá.

La función definida “choque” es diferente para los virus, ya que además al momento de chocar, estos sprites cambiarán su dirección, haciendo que se muevan aleatoriamente al momento del choque.

Los choques entre el jugador y el virus mediante “colliderect” se realizan activando una condición, de esta forma la vida del jugador cambia a 0 y el jugador muere, por lo que ya no se puede mover. Además cuando esto ocurre se inicia el sonido de muerte del jugador (canción). También al jugador se le impedirá el movimiento. Creamos una variable “impacto” que se vuelve verdadera y la pantalla cambiará mostrando las imágenes de “game over” o “you died” y mostrando textos de lo mismo.

A continuación se crean variables para las vitaminas las cuales serán verdaderas y que posteriormente serán utilizadas dentro del gran ciclo. Luego se define la función “Vitamina”, la cual creará otro sprite, y hará que aparezcan en pantalla estos puntos y que desaparezcan al momento del impacto con el jugador, además de sumar 1 a la variable “puntos”. En el ciclo se llamará a “vitamina” dentro de una condición, que solo dejará de suceder luego del impacto, estas condiciones son usadas con 1ras variables. Cuando las variable “puntos” lleguen a cierto número y se acaben en la pantalla, una condición volverá las 1ras variables a su valor original, haciendo que todos las vitaminas se restablezcan.

Luego crearemos cuadros de texto en la parte superior derecha de la pantalla. Estos mensajes se logran a partir de las funciones “escribir” y “escribir2”, en la cuales los parámetros serán el tamaño de letra del mensaje, la información, los colores y las coordenadas. Para esto utilizamos los módulos “pygame.font.Font” que es para crear un objeto Font a partir de las fuentes del sistema, y la variable.render que es para ayudar en la creación de los textos. Llamaremos a la función con intervalos de tiempo para ir actualizando los mensajes, que son de suma importancia para entregar la información necesaria que se requiere sobre le covid-19.

Para poder ejecutar este juego desde el computador (solo apto para computador), primero se descarga el archivo comprimido, luego el archivo se debe descomprimir y abrir el archivo llamado “porfiao-man19.exe”, para volver a jugar se debe cerrar el juego e iniciar de nuevo.

En resumen todas las funciones utilizadas en el juego son las sgtes y algunas fueron utilizadas más de una vez:

- `pygame.init()`
- `pygame.display.set_caption("Porfiao-man 19")`
- `pygame.image.load('virus.png')`
- `pygame.display.set_icon(icon)`
- `variablesprite.rect.colliderect(x)`
- `variablesprite1.rect.left`
- `variablesprite1.rect.top`
- `random.randint(0, 1)`
- `time.time()`
- `pygame.sprite.Sprite()`
- `pantalla.blit(nada, (x, y))`

- `variable.get_rect()`
- `pygame.draw.rect(pantalla, (0, 0, 0), q)`
- `pygame.display.set_mode((1200, 600))`
- `pygame.mixer.music.load('pactrap.mp3')`
- `pygame.time.Clock()`
- `pygame.Rect(0, 21, 800, 1)`
- `pygame.font.Font("zig.ttf.ttf", 23)`
- `variable.render(texto, 0, (r, g, b))`
- `pygame.time.get_ticks()`
- `pygame.event.get()`
- `pygame.QUIT`
- `pygame.KEYDOWN`
- `pygame.K_LEFT`
- `pygame.K_RIGHT`
- `pygame.K_UP`
- `pygame.K_DOWN`
- `pygame.KEYUP`
- `efectosonido.play()`
- `spritepersonaje.rect.move_ip(vx, vy)`
- `pygame.display.update()`
- `pantalla.fill((255, 255, 255))`
- `pygame.quit()`

## Bibliografía:

“12 Videojuegos con Python y Pygame: Colisiones”. Youtube. [Consulta: 25 de Junio de 2020]

“CÓMO USAR SPRITES para HACER ANIMACIONES ► PYTHON [ pygame ] ✓”. Youtube. [Consulta: 15 de Junio de 2020]

“Curso de pygame 2020: Creando una ventana y estructura de nuestro juego”. Youtube. [Consulta: 15 de Junio de 2020].

“Curso de pygame 2020: Dibujando figuras en la pantalla”. Youtube. [Consulta: 19 de Junio de 2020]

“Curso de pygame 2020: Imagen de Fondo (Video 9)”. Youtube. [Consulta: 15 de Junio de 2020].

“Curso de pygame 2020: Keyboard/Teclado”. Youtube. [Consulta: 19 de Junio de 2020]

“Curso de pygame 2020: Keyboard/Teclado”. Youtube. [Consulta: 19 de Junio de 2020]”. Youtube. [Consulta: 20 de Junio de 2020]

“How to Convert any Python File to .EXE”. Youtube. [Consulta: 4 de Julio de 2020]

“Juego en Pygame - Shooter parte 1”. Youtube. [Consulta: 19 de Junio de 2020].

“Making PacMan AI in python pygame”. Youtube. [Consulta: 15 de Junio de 2020].

“Pacman in Python - Part 18 - Smart Enemies!”. Youtube. [Consulta: 28 de Junio de 2020].

“PAC-MAN Theme (Trap Remix)”. Youtube. [Consulta: 2 de Julio de 2020]

“Player Death Scream Sound Effect Fourth Variation”. Youtube. [Consulta: 1 de Julio de 2020]

“Programacion Juegos : Pygame 20 - Sprites”. Youtube. [Consulta: 17 de Junio de 2020]

“Programacion Juegos :Pygame 19 - Velocidad de Movimiento”. Youtube. [Consulta: 27 de Junio de 2020]

“Programming Pac-man in Python”. Youtube. [Consulta: 20 de Junio de 2020]

“Pygame (Python Game Development) Tutorial - 24 - Collision Detection”. Youtube. [Consulta: 25 de Junio de 2020]

“Pygame Functions: Making Sprites”. Youtube. [Consulta: 18 de Junio de 2020].

“pygame tutorial 4 ~~ Uso de Funciones y dibujos Geométricos # 1”. Youtube. [Consulta: 17 de Junio de 2020].

“Python - Nivel 29 - Reto 23 - Mapas con pygame”. Youtube. [Consulta: 24 de Junio de 2020].

“Python - Nivel 29 - Reto 5 - Coordenadas en pygame”. Youtube. [Consulta: 15 de Junio de 2020].

“Python - Nivel 29 - Reto 7 - Movimiento en pygame”. Youtube. [Consulta: 16 de Junio de 2020].

“Python and Pygame - Text Box Tutorial”. Youtube. [Consulta: 22 de Junio de 2020]

“Tutorial pygame 8~~Colisiones con objetos en movimiento choque frontal”. Youtube. [Consulta: 25 de Junio de 2020]

*Archivo:SARS-CoV-2 without background.png*. Imagen Digital. Wikipedia. Imagen Digital. Sitio web. 28 de Junio de 2020.

Bannet Noam. Pac-Man rules. 2020. Imagen digital. Twitter. Imagen digital. Sitio web. 29 de 2020.

*Fondo Celeste*. Imagen Digital. Freepik. Sitio web. 30 de Junio de 2020.

*Free Sprites*. Imagen digital. Blogoscoped. Sitio web. 17 de Junio de 2020.

Gobierno de Chile, <<https://www.gob.cl/coronavirus/>>. Página web. 2 de Julio de 2020.

Gomez, Cesar. Shooter-pygame. <<https://github.com/mundo-python/shooter-pygame>>. Página web. 14 de Junio de 2020.

Leamy, Devin. Pacman. <<https://github.com/DevinLeamy/Pacman/commit/00cbc1f6984d7309dda5dae9bc4a3f0afc6342ac#diff-683b787dc9ce5ddee67870a627ddb26eR319>>.Página web. 14 de Junio de 2020.

*Marco Negro*. Imagen Digital. Pngocean. Sitio web. 2 de Julio de 2020.

*Marco rojo, marco, marco borde rojo PNG Clipart*. Imagen Digital. Pngocean. Sitio web. 2 de Julio de 2020

Minsal, <<https://www.minsal.cl/nuevo-coronavirus-2019-ncov/casos-confirmados-en-chile-covid-19/>>. Página web. 17 de Junio de 2020.

*Organización mundial de la Salud Cuidado de la Salud salud Pública*. Imagen digital. Freepng. Sitio web. 28 de Junio de 2020.

Pygame, <[www.pygame.org](http://www.pygame.org)>. Página web. 12 de Junio de 2020.

Recursos Python, <<https://recursospython.com/guias-y-manuales/caja-de-texto-entry-tkinter/>>. Página web. 17 de Junio de 2020.

S. Brandon. *Zig Fond*. 2002. Imagen digital. 1001fonts. S. Brandon. 2002. Sitio web. 2 de Julio de 2020.

Shin [Herbert.](#) Moriste Lámina artística. 2007. Redbubble. Imagen Digital. Sitio web. 28 de Junio de 2020.

Smith Kriz. *Pacman's Power Pellet*. 2018. Imagen Digital. Pngocean. Sitio web. 2 de Julio 2020.

Stackoverflow, <<https://stackoverflow.com/questions/37317840/pygame-colliderect-precision-issue>>. Pagina web. 16 de Junio de 2020.

[Yudha](#). Game Over. 2005. Imagen digital. Pngtree. Sitio web. 3 de Julio de 2020.