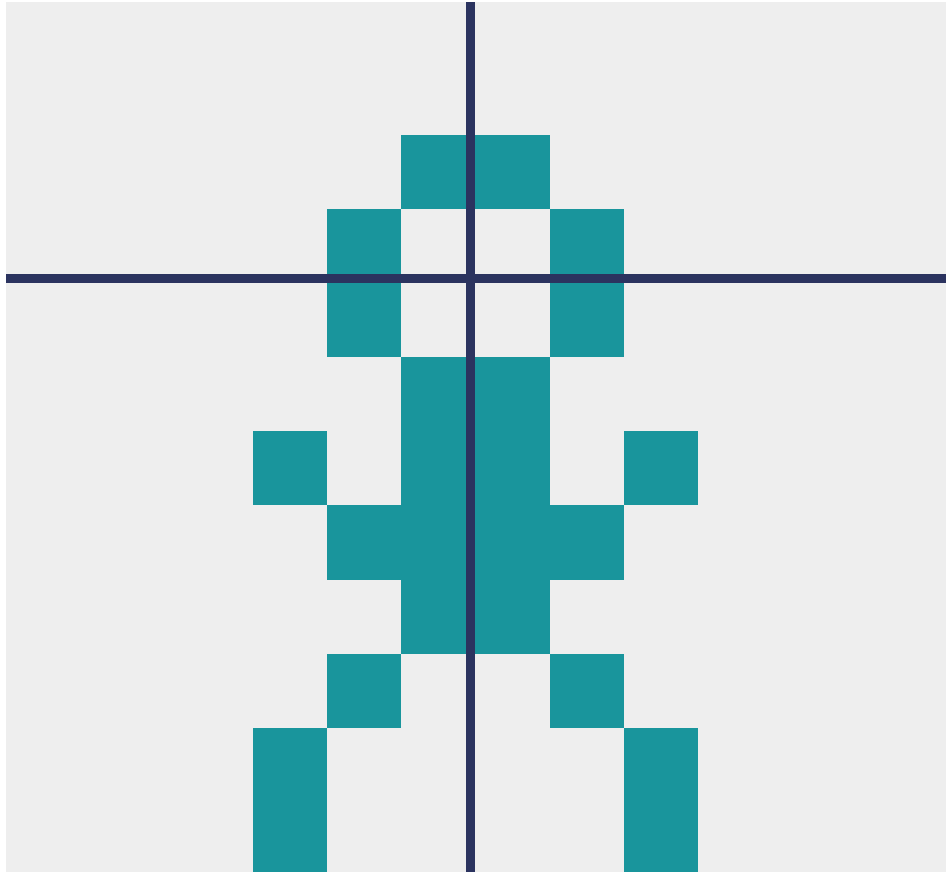


MEMORIA LEMMINGS



JAVIER LÓPEZ MORALES
CARLOS HERRANZ FONTÁN

ÍNDICE

1. Introducción	3
2. Instrucciones	3
3. Clases utilizadas y sus métodos	3
I. Herramientas	3
II. Marcador	4
III. Board	5
IV. Lemming	6
V. App	8
4. Aportaciones extras al juego	9
5. Conclusión	9

INTRODUCCIÓN

Lemmings es un juego en el que el jugador tiene que conseguir que unos individuos llamados lemmings consigan llegar a la meta, partiendo desde la salida. Para ello el jugador tiene la opción de colocar diversos objetos para que los lemmings no mueran y para facilitarles el camino hacia la victoria. Los lemmings saldrán siguiendo un intervalo de tiempo y el usuario tendrá un número limitado de objetos, no serán ilimitados. Al estar todos los lemmings muertos o salvados la partida se dará por concluida. El juego hace uso de la entrada por ratón por lo que la entrada de herramientas es más sencilla y rápida.

INSTRUCCIONES

Para que el juego se ejecute a la perfección se deberán cumplir las siguientes condiciones:

- El fichero con las imágenes y el resto de archivos en los que esté cada clase deberán situarse en el mismo archivo
- El archivo zip impórtalo a un proyecto de Pycharm o cualquier otro IDE.

Una vez hecho esto, disfruta del juego.

CLASES UTILIZADAS Y SUS MÉTODOS

- **HERRAMIENTAS**

Comenzando desde la parte más independiente hemos creado cinco clases que representen a las cinco herramientas que hemos usado: Las escaleras que suben hacia la derecha, las que suben hacia la izquierda, los paraguas, los bloqueadores y las palas. La mayoría de estas clases son prácticamente iguales, ya que ellas solo los atributos de todas ellas son muy similares: la posición en el eje x, la posición en el eje y y una lista en la que se van a guardar las coordenadas en x y en y. Sin embargo, la clase pala tiene un atributo que no tienen el resto. Este atributo indica si la pala ya se ha utilizado anteriormente. Recurrimos a este recurso ya que teníamos problemas para que ningún lemming usara una pala ya utilizada. Para esto lo usamos, para que no vuelvan a utilizarse y para que una vez utilizada se haga transparente. Todos estos atributos los hemos incluido a través del método init.

Además, hemos utilizado un método draw en cada clase para dibujar las imágenes de cada objeto en las posiciones marcadas por los atributos.

Los objetivos de cada herramienta son los siguientes:

1. Escaleras: Ayudar al lemming a subir a una plataforma que esté por encima. Además, también se pueden usar para bajar y al chocar con una pared el lemming cambia de dirección y comienza a bajarlas.
2. Palas: Cada vez que un lemming pasa sobre una pala el bloque que estaba ahí desaparece.
3. Bloqueadores: Cuando un lemming se encuentra con un bloqueador mientras camina, este cambia de dirección.
4. Paraguas: Son fundamentales para que el lemming no se muera al caer, ya que si cae sin paraguas, este muere.

- **MARCADOR**

Tras las herramientas hemos creado la clase marcador. Esta clase va a ir creando objetos que sirvan como marcadores de los resultados de cada partida. Los objetos de esta clase, como las herramientas, se guardan en un atributo de la clase Board. Para poder incluir información de la partida en la clase Marcador hemos recurrido a dar entrada de dos valores en esta clase: la lista que contiene a todos los lemmings y la lista con todos los objetos creados para poder comprobar cuantas herramientas disponibles le quedan al usuario o para ver cuantos lemmings han muerto, cuantos se han salvado o cuantos quedan por salir.

Dentro de esta clase hemos utilizado el método init para crear los atributos de esta: las palas iniciales y las que quedan por usarse, lo mismo con las escaleras, con los bloqueadores y los paraguas, los lemmings iniciales, los que se van salvando, los que mueren y los que quedan por salir y dos atributos que indican si la partida ha finalizado y si el jugador ha conseguido ganar la partida o no.

Además dentro de esta clase hemos definido un método update que va a ir cambiando los valores de los atributos, ya que las herramientas se utilizan y los lemmings salen, mueren y se salvan. A su vez hemos incluido en este método un condicional que se ejecutará cuando se haya finalizado la partida(si han muerto o se han salvado todos los lemmings) y decidirá si el jugador ha ganado o ha perdido. La victoria se consigue cuando se consiguen salvar al menos la mitad de los lemmings. Este método update será llamado por el método update de la clase Board.

Por último, hemos utilizado un método draw para dibujar los resultados sobre la pantalla. Este método también será llamado por el método draw de la clase Board.

- **BOARD**

Una de las clases más fundamentales que hemos creado ha sido la clase Board, que crea objetos que nos servirán como tablero de juego. En esta clase lo primero que hacemos es crear una matriz en la que hemos introducido unos y ceras siguiendo una pauta para que aparezcan 6 plataformas de una longitud determinada, con un espacio vacío entre ellas, y hemos respetado el espacio que ocupará el marcador. Después, hemos hecho que la última fila tenga unos ya que va a ser la correspondiente al suelo, que las columnas laterales tengan otro número para poder dibujar las paredes y hemos otorgado una posición a la puerta y otra a la salida siguiendo el siguiente algoritmo:

- Establecemos aleatoriamente las filas de la puerta y de la salida.
- Establecemos aleatoriamente la que sería la columna de la entrada (la primera o la última). Sin embargo, en su fila va a comenzar a buscar un bloque sobre el que posicionarse, empezando siempre sobre la primera o la última. Cuando haya encontrado ese bloque, se buscará la posición de la puerta del mismo modo pero empezando a buscar desde la derecha. (Al principio este modo nos dio algún error, pero lo que ocurría era que recorría uno de los bucles while y nunca encontraba solución por lo que el juego nunca llegaba a abrirse. Lo solucionamos aumentando el rango de búsqueda del bloque sobre el que posicionarse).
- Guardamos las coordenadas de la posición de la entrada en un atributo para poder pasarla después a los objetos de clase lemming para establecer su comienzo.
- Por último, cambiamos los números de la matriz que corresponden a las posiciones de la puerta y de la salida para que estas se dibujen.

Además, hemos incluido más atributos en esta clase: una lista con los objetos, una lista con los lemmings, el marcador, dos contadores que se explicaran más tarde y un atributo que indica si se ha finalizado la partida.

Después del init hemos usado un método que comprueba si hay una herramienta colocada bajo la posición sobre la que se encuentra el cursor. Si resulta que si hay una herramienta este método devolverá un True y si no la hay, un False.

Hemos creado un método “final” que detectará si una partida ha acabado y le restará valores a uno de los contadores del init para mostrar los resultados de la partida y que no los muestre instantáneamente. Dependiendo de la conclusión del juego mostrará diferentes textos. Además,

se indica al usuario lo que tiene que hacer si quiere seguir jugando. Este método es llamado por el método draw que se explicará posteriormente.

En el método update damos entrada a todas las herramientas que el usuario quiera o pueda colocar. Para colocar cada objeto, se deberá pulsar una tecla determinada, y se utilizarán un método y dos ifs para comprobar que no se coloca una herramienta sobre otra y que no se colocan sobre el suelo, salvo la pala que solo puede colocarse sobre las plataformas. Al colocar una herramienta, el objeto que la representa se guarda en el atributo que guarda la lista con los objetos.

Hemos incluido la posibilidad de que el usuario pueda quitar un objeto, si se arrepiente de su uso. Para eso tendrá que dejar pulsada la tecla D. En el momento en el que comienza a pulsarla. Se va restando valores a un contador hasta que se hace cero y borra la herramienta sobre la que está el cursor. Para saber que herramienta borrar, hemos usado un bucle que recorre la lista con los objetos, lista por lista y al encontrar el objeto mediante su atributo coordenadas, lo eliminamos.

Teclas para cada herramienta:

1. A -> Palas
2. E -> Escaleras que suben hacia la derecha
3. T -> Escaleras que suben hacia la izquierda
4. U -> Paraguas
5. X -> Bloqueadores

Para finalizar el update de la clase tablero, llamo al update de la clase Marcador para que se realice.

Por último el último método que hemos utilizado en esta clase ha sido el método draw que recorrerá la lista con los objetos para llamar al método draw de cada herramienta, llamo al método draw del marcador, dibujo un cuadrado para saber la casilla que marca el cursor, dibujo el método que “final” explicado anteriormente, y dibujo las plataformas, las paredes, la puerta y la salida.

- **LEMMING**

Esta clase la hemos creado para crear objetos que representarán a cada uno de los lemmings que aparezcan. Esta clase tendrá un valor de entrada que será un objeto de la clase tablero ya

que el lemming necesita conocer la posición de los bloques, de las herramientas y de la entrada, entre otras cosas.

En el método `init` de esta clase hemos guardado varios atributos como: la matriz del objeto de la clase `Board`, la posición inicial dependiendo de donde esté la puerta, la velocidad inicial del lemming, la gravedad, la velocidad de caída con el paraguas y distintos atributos que indicarán si se está usando una herramienta para ejecutar los métodos convenientes.

A continuación hemos creado distintos métodos que devolverán las coordenadas de todos los objetos correspondientes a una misma herramienta. Estos los usaremos para comprobar si el lemming se encuentra o no con una herramienta. Hemos usado estos métodos para poder reutilizarlos más de una vez, ya que la mayoría se usan más de una vez. Con esto, logramos un código bastante más fácil de entender.

A partir de ahora viene una de las partes más laboriosas del juego, definir el movimiento del lemming.

Hemos utilizado un método para cada tipo de movimiento:

Método caminar: Se usa para que un lemming camine sobre los bloques. En este método se buscan continuamente herramientas que usar, como las escaleras, las palas y los bloqueadores. Si no los encuentra sigue caminando hasta que se encuentra con una pared que va a hacer que cambie de dirección o hasta que finalice en una posición sobre la que no hay superficie y empiece a caer. Lo complicado de este movimiento ha sido tener en cuenta la posición real de lemming ya que esta la marca la esquina superior izquierda de la imagen que lo dibuja. Por este motivo hemos tenido que ajustar posiciones para que choque con las paredes justamente cuando el “cuerpo” del lemming choque con ellas, que descienda correctamente cuando los dos pies del lemming se encuentren sobre el vacío o, para nosotros lo más complicado, que comience a subir una escalera cuando el pie adecuado llegue hasta ella.

Métodos subir escaleras hacia la derecha o hacia la izquierda: Hemos creado dos métodos uno para cada tipo de escalera. En estos métodos se comprueba si hay otra escalera sobre la que seguir subiendo y si no la hay, lo primero que comprueba es si hay un bloque sobre el que caminar, y si esto no ocurre empieza a descender. Además, hemos añadido la posibilidad de que rebote si se encuentra con una pared o con un bloque mientras sube, por que empezaría a bajar esa escalera.

Métodos bajar escaleras hacia la derecha o hacia la izquierda: Este método es muy similar al de subir escaleras, pero sin embargo, las posiciones que detecta son las que van hacia abajo, ya que

es lo obvio ya que el lemming está bajando. Al igual que al subir, los lemmings rebotan cuando se encuentran con una pared y comienzan a bajar.

Esto ha sido sin duda la parte más complicada de realizar, ya que en algunos testeos que hemos ido realizando, el lemming al finalizar la escalera siempre caía aunque se encontrase un bloque para caminar y en otros intentos no seguía subiendo si se encontraba otra escalera. Esto ha sido un problema derivado de la ubicación real de los lemmings comentada anteriormente ya que esta se encuentra sobre la esquina superior izquierda de la imagen. Para poder arreglarlo hemos tenido que ir manejando la siguiente posición del lemming sumándole o restándole valores para colocar la ubicación en el pie del lemming correspondiente.

Método descenso: Con este método definimos el movimiento del lemming cuando desciende. Esto también lo usamos para detectar si se encuentra con un paraguas y para definir la velocidad con la cae, dependiendo esta de si lleva o no un paraguas. Además dentro de este método comprobamos si se choca con el suelo y comprobar si muere o no dependiendo de si lleva o no un paraguas. Para comprobar si hay paraguas usamos el método creado anteriormente, que devuelve una lista con las coordenadas de los paraguas.

Método excavar: En este método lo único que hacemos es cambiar el valor de los atributos que indican si el lemming está cavando o si está cayendo, ya que cuando desaparece el bloque, el lemming va a dejar de cavar y va a empezar a descender.

Una vez definidos cada movimiento del lemming usamos el método update. En este método dependiendo los valores de los atributos establecidos en el init y siguiendo un bucle va a realizar el método correspondiente al tipo de movimiento que debe llevar el lemming. Es decir, en este método se llaman a los métodos que definen el movimiento del lemming. Si está muerto el lemming no se moverá(obviamente).

Para finalizar esta clase, hemos definido el método draw en el que definirá la imagen que corresponde a cada lemming, ya que no es la misma si camina, si sube o baja, si desciende, si muere o si llega a la meta.

- **APP**

Por último, la última clase del juego, la principal, la que hace que todo funcione, la que guarda en ella toda la información, la clase app.

En el comienzo de esta clase, hemos usado el método init para establecer sus atributos principales. Entre ellos, se encuentran las dimensiones del juego, un atributo que indicará si el

juego está en pausa, una lista con los lemmings que van a ir apareciendo, un atributo en el que se va a guardar un objeto de la clase Board, el máximo de lemmings y un contador para regular la entrada de lemmings. Además en el init incluimos todas las propiedades de pyxel que necesitamos: que se muestre el cursor por pantalla, que se cree el juego(con pyxel init), que suene la música del juego, que se carguen los ficheros con las imágenes y que se ejecuten el update y el run.

Dentro del update incluimos la opción de pausar el juego pulsando la tecla de espacio. Esta opción es muy útil si quieres colocar herramientas antes de que un lemming llegue. Además, hemos incluido una opción para reformatear el nivel si al usuario no le gusta, bastará con pulsar la tecla R y aparecerá un nivel nuevo que comenzará desde cero. A su vez, hemos incluido la opción de salir del juego mediante la tecla Q. A continuación, vamos dando la entrada a los lemmings restándole de 1 en 1 al marcador del init. Por último, realizamos un update de cada lemming añadido a la lista de lemmings y llamamos al método update del tablero.

Por último, el método draw. En este hacemos que se establezca un fondo negro, y mediante este método llamamos a los métodos de draw del tablero y los de cada lemming de la lista de lemmings.

APORTACIONES EXTRAS AL JUEGO

Como ya se ha mencionado, hemos incluido la pala como herramienta que puede usar el lemming, su uso es muy sencillo, el usuario la coloca sobre un bloque y cuando el lemming pasa por encima la usa y el bloque desaparece.

Además, hemos incluido la opción de que el lemming se pueda mover hasta de 5 en 5 pyxels sin que los objetos fallen.

Por último, hemos incluido la opción de poder pausar el juego pulsando la tecla del espacio.

CONCLUSIÓN

Hemos finalizado el juego cumpliendo todas las exigencia pedidas en la práctica. Además hemos conseguido introducir una herramienta extra como es la pala o la posibilidad de poder poner el juego en pausa.

Sin embargo, debido a la falta de tiempo, no hemos podido implementar otras cosas en las que habíamos pensado como una pantalla de inicio, la posibilidad de mostrar una pantalla con el objetivo del juego y la utilidad de cada tecla del teclado o la opción de introducir un trampolín que desplazase al lemming hacia arriba siguiendo también una dirección en x.

Pese a todo, concluir afirmando que la realización de esta práctica ha sido bastante difícil debido al escaso tiempo que teníamos y a la falta de experiencia y conocimientos inicial. Sin embargo, tras la realización se han adquirido muchos conocimientos que no se podrían haber obtenido de otro modo.

Javier López Morales

Carlos Herranz Fontán