

## **Worms:**

### **Programación orientada a eventos**

Se busca realizar un programa que simule el desplazamiento y salto de dos Worms siguiendo las condiciones detalladas abajo:

1. Ambos Worms se posicionan inicialmente en la horizontal que vamos a tomar como piso y  $y = 616px$ . Su posición en  $x$  debe ser tal que  $701px < X < 1212px$ . Dentro de estos parámetros cada grupo puede elegir la posición inicial y orientación de los dos Worms como prefiera.
2. Uno de los Worms responde a las teclas “flecha IZQUIERDA” para moverse hacia la izquierda, “flecha DERECHA” para moverse hacia la derecha y “flecha ARRIBA” para saltar.
3. El otro Worms responde a las teclas “A” para moverse hacia la izquierda, “D” para moverse a la derecha y “W” para saltar.
4. El usuario puede presionar cualquiera de estas teclas en cualquier momento.
5. Si se comanda a dos Worms distintos (por ejemplo se aprieta la “W” y la “flecha IZQUIERDA”) al mismo tiempo, ambos Worms deben responder al mismo tiempo (siguiendo el ejemplo uno de los Worms salta y el otro se mueve hacia la izquierda).
6. Si se presionan dos teclas sobre un mismo Worms se le dará prioridad a una de ellas (siguiendo alguna estrategia, por ejemplo se toma la que primero ve el sistema) y se ignorará la segunda hasta que se termine la condición de dos teclas sobre el mismo Worms apretadas en simultáneo.
7. El desplazamiento de cada Worms está regido de la siguiente manera:
  - a. El Worms se mueve a razón de 27 píxeles por segundo mientras el jugador mantiene alguna de las teclas “flecha IZQUIERDA”, “flecha DERECHA”, “A” o “D” presionada.
  - b. Para que el gusano comience a moverse el jugador debe mantener la tecla presionada por más de 100ms Si la presionara menos tiempo la máquina entiende que el usuario quiere dar vuelta al gusano seleccionado sobre sí mismo si se tratara de la flecha de sentido inverso y es ignorada en otro caso.
  - c. La máquina tomará desplazamientos múltiplos de 27 píxeles. Por lo que luego de entender que el usuario quiere realizar un movimiento, recién a los 900ms volverá a verificar si la flecha está apretada por 100ms para volver a mover el gusano. Si no lo estuviera, se termina el movimiento. Si lo estuviera se continúa.

d. Más información en [http://worms2d.info/Phases\\_of\\_Worm\\_walking](http://worms2d.info/Phases_of_Worm_walking).

8. Tomaremos solamente desplazamientos horizontales en el escenario provisto por la cátedra. Los Worms puede moverse en  $Y=616px$  entre  $701px < X < 1212px$ .
9. El salto se calcula con una velocidad inicial en módulo de 4.5. Se considera que el gusano salta con un ángulo de  $60^\circ$  (resulta por lo tanto una velocidad inicial en el eje vertical ("Y") del salto de  $\sin(60^\circ) * 4,5$ ). Sólo permitiremos "forward jumps" por lo que el gusano salta en la dirección en la que apunta.
10. Mientras el Worms está saltando (el tiempo de duración del salto se determina a partir de las ecuaciones de tiro oblicuo) se ignora el estado de la tecla que comandó dicho salto. Una vez que terminó el salto, se vuelve a considerar.
11. Vale aclarar que tanto el salto como el movimiento tienen un preámbulo conocido como Start-up en donde no se realiza ningún movimiento que se va a discutir en clase. También cuenta dentro del tiempo de salto. Por el contrario, el Start-up está incluido en el tiempo de desplazamiento de 1s.
12. Tomaremos  $g=0.24$ .
13. Los ángulos se miden desde la recta horizontal que representa el piso en donde se apoya el gusano hacia la vertical en la dirección del gusano. Si el ángulo es negativo el mismo se mide por debajo el gusano y si es positivo por arriba de él.
14. La velocidad se mide en píxeles por frame, la aceleración en píxeles por frame por frame.
15. La velocidad de refresco está fija en 50FPS (frames por segundo).
16. Para representar a los Worms usaremos las imágenes provistas por la cátedra.