Diseño y desarrollo de Videojuegos

Comportamiento de personajes - Memoria de la práctica -

Grupo 4:
Alberto Blanco Barrios
Eva M. Pérez Fernández
Sergio Sánchez-Urán López
José M. Segade de Tena
Jesús Téllez Serrano

3 de diciembre de 2018

Índice

1. Descripción general

El juego consiste en sobrevivir en un espacio estrecho, dónde hay una serie de *zombies* enemigos que intentarán acabar con el jugador. Es un *Survival Horror* clásico, en el cual el jugador tendrá que enfrentarse a los espacios cerrados y la escasez de munición.

El jugador maneja una cámara en primera persona. En la pantalla puede observar cuantas balas tiene en el cargador, y la munición máxima. Este control de *First Person Shooter* se ha implementado por comodidad, para ver la escena del juego de manera que se parezca a otros similares ampliamente conocidos. Es un añadido para poder interactuar por la escena cuyo propósito es la navegación, aunque el sistema de control no está perfectamente pulido, pues el contenido se aleja de la asignatura. Los controles de la cámara son los clásicos de cualquier *Shooter*:

- **W** avanzar.
- **S** retroceder.
- **A** moverse a la izquierda.
- **D** moverse a la derecha.
- **R** recargar el arma.
- **Botón Izdo del Ratón** disparar el arma.

Se ha optado por una estética 3D porque resultaba más realista y, al no ser los *assets* propios, no supone una excesiva dificultad a la hora de la implementación.

Los agentes que intervienen en el juego son:

- **Jugador** Interactúa con los *zombies* disparándoles, huyendo de ellos o escondiéndose. También puede modificar las decisiones de los supervivientes, que intentarán robarle o ayudarle.
- **3 Zombies normales** Persiguen al jugador siempre que le vean. También huyen de otros *zombies*.
- **2 Zombies rápidos** Persiguen al jugador siempre que le vean, aunque atacarán a otros *zombies* si el jugador no se encuentra en su visión.
- **Zombie fuerte** Ataca a los jugadores mediante objetos que hay en el escenario. Similar a los *zombies* normales, pero atacan desde la distancia en caso de que hayan cogido un objeto.
- **3 supervivientes** Pueden atacar a otros *zombies* y según su salud y el número de balas, ayudará al jugador o le perjudicará, robándole, dándole munición o atacándole.
- **Entorno** Permite al jugador esconderse y define las zonas de navegación tanto del jugador, como de los supervivientes y los *zombies*. Proporciona objetos que tirar.

2. Descripción detallada de los personajes

2.1. Zombie normal

Es un *zombie* lento, con una vida de 100 puntos, que ataca solamente al jugador y a los supervivientes que encuentra en su camino. Cuando tiene hambre necesita comer y buscará un *zombie* o un superviviente abatidos para alimentarse. El estado de hambre prevalece sobre el de patrullar, alerta o atacar. Tiene varios puntos de patrulla a los que va accediendo a través del *NavMesh* por métodos de búsqueda ya incorporados.

Tabla de percepciones:

NOMBRE	IMPLEMENTACIÓN	ACCESO
Ver al personaje	Comprobar si el agente está suficientemente cerca y se encuentra contenido en el ángulo de visión del jugador y no existe ningún elemento que lo tape.	A través de los *Collider* y los *RayCast* que notifican al agente.
Seleccionar objetivo	Buscar el objetivo más cercano al *zombie*. Distancia $<$ 3m.	Recorrer una lista con los elementos que estén cerca del *zombie* y se hayan detectado previamente.
Atacar	Busca si el objetivo está en una distancia muy cercana.	*Collider* que notifica al agente.
Hambre	Mediante una variable que disminuya en el *update* del objeto.	Se accede en cada *update* y notifica al agente que debe pasar a un estado de hambriento si la variable llega a 0.
Recibir daño	Saber si el personaje ha disparado hacia el agente y si éste se encuentra en el camino de la bala.	El personaje lanza un *RayCast* que notifica al agente mediante un método de una interfaz abstracta común para todos los agentes

Tabla de acciones:

NOMBRE	IMPLEMENTACIÓN	EFECTOS
Moverse	Navigation Mesh	Update Position
Recibir daño	Lanzar animación	Disminuir salud
Atacar	Lanzar animación	Notificar a jugador
Comer	Lanzar animación y sistema de partículas	Disminuir hambre
Morir	Lanzar animación	Desactivar comportamientos

Máquina de estados:

IDENTIFICADOR	NOMBRE DE ESTADO	PUEDE IR A
1	Patrullar	2, 3, 4, 5
2	Alerta	1, 4, 5
3	Atacando	1, 2, 5
4	Comiendo	1, 2, 3, 5
5	Sufriendo	1, 2, 3, 4, 6
6	Muerto	∅

2.2. Zombie rápido

Es un *zombie* que avanza rápido, con una vida de 100 puntos, que ataca al jugador y a los supervivientes que encuentra en su camino y además ataca a otros *zombies* rápidos y normales. Cuando tiene hambre necesita comer. Si tiene hambre, pero ve a otro *zombie*, prioriza atacarle antes que comer de un *zombie* o superviviente que ya esté muerto. Tiene varios puntos de patrulla a los que va accediendo a través del *NavMesh* por métodos de búsqueda ya incorporados. Este *zombie* corre en la dirección de su objetivo cuando está en modo de alerta y va a una velocidad de andar cuando está en el estado de patrulla.

Tabla de percepciones:

NOMBRE	IMPLEMENTACIÓN	ACCESO
Ver al personaje	Comprobar si el agente está suficientemente cerca y se encuentra contenido en el ángulo de visión del jugador y no existe ningún elemento que lo tape.	A través de los *Collider* y los *RayCast* que notifican al agente.
Seleccionar elemento para comer	Buscar el objetivo más cercano al agente	Recorrer una lista con los elementos que estén cerca del agente y se hayan detectado previamente.
Seleccionar objetivo	Buscar el objetivo más cercano al *zombie*. Distancia < 3m.	Recorrer una lista con los elementos que estén cerca del *zombie* y se hayan detectado previamente.
Atacar	Busca si el objetivo está en una distancia muy cercana.	*Collider* que notifica al agente.
Hambre	Mediante una variable que disminuya en el *update* del objeto.	Se accede en cada *update* y notifica al agente que debe pasar a un estado de hambriento si la variable llega a 0.
Recibir daño	Saber si el personaje ha disparado hacia el agente y si éste se encuentra en el camino de la bala.	El personaje lanza un *RayCast* que notifica al agente mediante un método de una interfaz abstracta común para todos los agentes

Tabla de acciones:

NOMBRE	IMPLEMENTACIÓN	EFECTOS
Moverse	Navigation Mesh	Update Position
Correr hacia el objetivo	Navigation Mesh y lanzar animación	Update Position
Recibir daño	Lanzar animación	Disminuir salud
Atacar	Lanzar animación	Notificar a jugador
Comer	Lanzar animación y sistema de partículas	Disminuir hambre
Morir	Lanzar animación	Desactivar comportamientos

Máquina de estados:

IDENTIFICADOR	NOMBRE DE ESTADO	PUEDE IR A
1	Patrullar	2, 3, 4, 5
2	Alerta	1, 4, 5
3	Atacando	1, 2, 5
4	Comiendo	1, 2, 3, 5
5	Sufriendo	1, 2, 3, 4, 6
6	Muerto	∅

2.3. Zombie fuerte

Es un *zombie* más lento y grande que el normal, con una vida de 100 puntos, que ataca a *zombies*, al jugador y a los supervivientes que encuentra en su camino. Cuando tiene hambre necesita comer y buscará un *zombie* o un superviviente abatidos para alimentarse. El estado de hambre prevalece sobre el de patrullar, alerta o atacar. Tiene varios puntos de patrulla a los que va accediendo a través del *NavMesh* por métodos de búsqueda ya incorporados. Este *zombie*, sin embargo, utiliza los objetos repartidos por el mapa para atacar a los demás. Ataca con más distancia y tiene la capacidad de recoger y lanzar elementos del mapa.

Tabla de percepciones:

NOMBRE	IMPLEMENTACIÓN	ACCESO
Ver al personaje	Comprobar si el agente está suficientemente cerca y se encuentra contenido en el ángulo de visión del jugador y no existe ningún elemento que lo tape.	A través de los *Collider* y los *RayCast* que notifican al agente.
Atacar	Busca si el objetivo está en una distancia muy cercana y si tiene un objeto.	*Collider* que notifica al agente.
Recoger objeto	Detecta el objeto si está en una distancia cercana e incorpora el objeto al agente para posteriormente lanzarlo.	*Collider* que notifica al agente y accede a las variables de transformación del objeto.
Hambre	Mediante una variable que disminuya en el *update* del objeto.	Se accede en cada *update* y notifica al agente que debe pasar a un estado de hambriento si la variable llega a 0.
Recibir daño	Saber si el personaje ha disparado hacia el agente y si éste se encuentra en el camino de la bala.	El personaje lanza un *RayCast* que notifica al agente mediante un método de una interfaz abstracta común para todos los agentes

Tabla de acciones:

NOMBRE	IMPLEMENTACIÓN	EFECTOS
Moverse	Navigation Mesh	Update Position
Recibir daño	Lanzar animación	Disminuir salud
Coger objeto	Lanzar animación	Transferir objeto para usarlo
Atacar	Lanzar animación	Notificar a jugador y soltar objeto
Comer	Lanzar animación y sistema de partículas	Disminuir hambre
Morir	Lanzar animación	Desactivar comportamientos

Máquina de estados:

IDENTIFICADOR	NOMBRE DE ESTADO	PUEDE IR A
1	Patrullar	2, 3, 4, 5
2	Alerta	1, 4, 5
3	Atacando	1, 2, 5
4	Comiendo	1, 2, 3, 5
5	Sufriendo	1, 2, 3, 4, 6
6	Muerto	∅

2.4. Superviviente

Es un agente que detecta a cualquier otro tipo de agente. Según los valores que tenga almacenados puede realizar un tipo de acción u otra. Lo supervivientes atacarán a los *zombies* dependiendo de si tienen munición, en caso contrario, procurará huir. Siempre que acabe una acción relacionada con el jugador, su comportamiento será de huir. Tiene una variable si dice si el superviviente es bueno o es malo. La variable de bondad será un número del 0 al 100, siendo de 0-30 baja, de 30-50 media y superior a 50 alta. Cuando se encuentre con otro superviviente y deba realizar una acción relacionada con el jugador, se le notificará al otro superviviente para que no haya conflicto y no se repita la acción. En el vídeo que se adjunta con los comportamientos de cada agente, se ha decidido no incluir las acciones de curar ni robar porque no se diferencia de la implementación de dar munición y no se iba a apreciar.

Tabla de percepciones:

NOMBRE	IMPLEMENTACIÓN	ACCESO
Ver al personaje	Comprobar si el agente está suficientemente cerca y se encuentra contenido en el ángulo de visión del jugador y no existe ningún elemento que lo tape.	A través de los *Collider* y los *RayCast* que notifican al agente.
Seleccionar objetivo	Buscar el objetivo más cercano al *zombie*. Distancia $<$ 3m.	Recorrer una lista con los elementos que estén cerca del *agente* y se hayan detectado previamente.
Recibir daño	Saber si el personaje ha disparado hacia el agente y si éste se encuentra en el camino de la bala.	El personaje lanza un *RayCast* que notifica al agente mediante un método de una interfaz abstracta común para todos los agentes
Ir a curar	Detectar al jugador y la bondad de este agente es alta	Variable de bondad y acceso a la salud del jugador.
Ir a dar munición	Detectar al jugador y la bondad de este agente es alta	Variable de bondad, acceso a la salud del jugador y notificación de la munición.
Ir a robar	Detectar al jugador o a otro superviviente y la bondad de este agente es media	Variable de bondad y notificación de la munición del jugador.
Disparar a jugador o superviviente	Detectar al otro agente si la bondad de este agente es baja	Variable de bondad, mediante *RayCast* encontrar al jugador y acceder a su salud.
Disparar a *zombie*	Detectar al *zombie* y la bondad de este agente es baja	Variable de bondad, mediante *RayCast* encontrar al jugador y acceder a su salud.

Tabla de acciones:

NOMBRE	IMPLEMENTACIÓN	EFECTOS
Moverse	Navigation Mesh	Update Position
Recibir daño	Lanzar animación	Disminuir salud
Atacar jugador	Lanzar animación	Notificar a jugador
Atacar *zombie*	Lanzar animación	Notificar a agente
Curar	Lanzar animación	Notificar al jugador
Robar	Lanzar animación	Notificar al jugador y aumentar munición
Morir	Lanzar animación	Desactivar comportamientos

Máquina de estados:

IDENTIFICADOR	NOMBRE DE ESTADO	PUEDE IR A
1	Patrullar	2, 4, 5, 6, 8
2	Atacar	1, 3, 4, 5, 6, 8
3	Huir	1, 2, 7, 8
4	Curar	1, 2, 4, 6, 8
5	Dar Munición	1, 2, 4, 6, 8
6	Robar	1, 2, 4, 6, 8
7	Descansar	1, 2, 4, 5, 6, 8
8	Muerto	∅

3. Flujo y estructuras de datos

Flujo de información principal entre los agentes:

AGENTE	MODIFICA INFORMACIÓN DE	A TRAVÉS DE
Cualquiera	Movimiento en el entorno	NavMesh
Agente inteligente	Salud del resto de agentes	Notificación por método en el momento de ataque
Agente inteligente	Posición del resto de agentes	Colliders
Jugador	Salud a *zombies* y supervivientes	RayCast
Superviviente	Intención de interactuar con el jugador	Notificación por método
Superviviente	Munición o salud del jugador	Notificación por método comprobando etiqueta
Zombie	Hambre de manera interna	Interacción por distancia mediante colisión con otro agente en estado muerto
Zombie fuerte	Posición de objetos	Suscripción del objeto

Las estructuras de datos empleadas para el funcionamiento correcto han sido principalmente para almacenar los estados de cada uno de los agentes de la escena. Existe una superclase que se llama `zombie` de la que heredan los *zombies* de la escena, que tiene que ver principalmente con el hecho de recibir el daño. Todos los agentes que se encuentran dentro de la escena, además, llevan un componente *NavMesh* para poder moverse por la escena de manera que siempre estén en el camino correcto, y dependiendo del tipo de agente, se le hace seguir una ruta de patrulla o no. El comportamiento viene definido en cada uno de los agentes por un *script* propio. De manera general todos contienen de internamente una máquina de estados, detectan a otros agentes por medio de *RayCast* y detectan el entorno y las distancias entre los agentes mediante *Colliders*. Las variables a las que tienen acceso unos agentes sobre otros es la salud, que pueden quitar mediante los ataques, notificando siempre al agente correspondiente. También se tiene en cuenta de manera interna a cada *zombie* el hambre que tiene, que modificará su conducta. En cuanto a la escena, existen objetos repartidos que se acceden a través de las colisiones, lo cuales pueden ser modificados según su posición. En cuanto al jugador, su forma principal de interacción es notificar a los agentes de su posición, cuando le detectan, o de proporcionar daño, cuando éste dispara.

4. Descripción de algoritmos

La visión de los agentes se ha programado de manera que existe un *Collider* que se encarga de la percepción. Este *Collider* marca una distancia circular amplia, que capta todos los objetos que se encuentran en el radio fijado (en el caso del juego 8m. de radio). Cuando se atraviesa, el agente es notificado para lanzar un rayo de visión a través de un *RayCast* que recoge todos los elementos entre el agente y el personaje, siempre y cuando el agente se encuentre en un ángulo de visión de 60º con respecto al eje del personaje. Una vez se han recogido todos los elementos que cumplen las condiciones tanto de ángulo como de cercanía, se recorre en orden la lista con los objetos; los cuáles se han obtenido por haber colisionado con dicho rayo. Si primero se encuentra un objeto que no sea una pared, quiere decir que dicho objeto está dentro de la visión del agente. En caso contrario, hay una pared ocluyendo al personaje.

5. Reparto de tareas

Alberto Blanco Programación del superviviente.

Eva M. Pérez Programación de *zombie* fuerte.

Sergio Sánchez-Urán Programación de la cámara FPS, *NavMesh*, y *zombie* normal.

José M. Segade Lógica de los agentes y redacción de la memoria.

Jesús Téllez Programación de *zombie* rápido.