
SUSPECT USER SIMULATOR

Comportamiento de Personajes

9 DE DICIEMBRE DE 2021

GRUPO 3:

PABLO DE LA HOZ MENÉNDEZ
JUAN JOSÉ PÉREZ ABAD
SERGIO SÁNCHEZ MARTÍNEZ

PROFESORES:

CARLOS GARRE DEL OLMO
AARON SÚJAR GARRIDO

Grado en Diseño y Desarrollo de Videojuegos

Contenido

Descripción general	2
Interacción	3
Especificaciones de los personajes	4
Agentes Trabajadores	4
Tabla de percepciones:	4
Tabla de acciones:	5
Diagrama descriptivo: máquina de estados jerárquica	6
Estructura de datos:	6
Agentes traidores	7
Tabla de percepciones:	9
Tabla de acciones:	10
Diagrama descriptivo: máquina de estados jerárquica	10
Estructura de datos:	12
Interacción con el entorno/mundo	13
Descripción de algoritmos	15
Resultados	16
Reparto de tareas	19
Pablo	19
Juan José	19
Sergio	19
Referencias	20

Descripción general

Se presenta una simulación que pretende representar una variante de partida del videojuego [*Among Us*](#), en el cual existen dos tipos de agentes: **Trabajadores** e **Impostores**. Se trata de un juego de suma nula en el que ambos tipos de agente pueden ganar la partida a través de dos formas, dando lugar a cuatro posibles finales de simulación:

- Los Trabajadores podrán ganar realizando un número objetivo de tareas (impuesto antes de comenzar la simulación) o identificando y expulsando a todos los Impostores a través de votaciones.
- Los Impostores podrán ganar si superan en número a los Trabajadores (pudiendo matarles para conseguir este objetivo) o si logran causar un sabotaje durante el tiempo suficiente.

El escenario consta de distintas salas provocando la separación de los agentes, de forma que puedan ocurrir estados ideales de asesinato para los agentes Impostores. Los agentes Honestos han de realizar tareas en zonas específicas del mapa, evitando ser asesinados; los agentes Impostores han de fingir la realización de estas tareas a la vez que van asesinando a los Trabajadores.

Cuando un Trabajador encuentra un cadáver puede desplazarse hacia el botón designado para iniciar una votación, en la que el juego entrará en estado discusión; así, los distintos agentes tendrán que decidir quién fue el asesino y expulsar a un agente.

Interacción

La interacción se presenta de dos formas: antes de la simulación y durante la simulación.

Para empezar, es **muy importante** iniciar el proyecto desde la escena **MainMenu**, para que de esa forma se pasen los datos de forma correcta.

Antes de la simulación se podrá interactuar con una escena de configuración que permita definir ciertos parámetros de la simulación, a saber: nº de Trabajadores, nº de Impostores y nº de tareas necesarias para completar la simulación a favor de los Trabajadores.

Durante la simulación se podrá navegar a través de la escena a través de un cámara controlada por el usuario. Aunque no es posible alterar a los agentes, sí lo es mover la cámara libremente para observarles, pudiendo trasladarla, rotarla, alejarla y acercarla. Los controles para la cámara son:

- Traslación: teclas WASD (adelante, izquierda, atrás, derecha).
- Rotación: teclas QE para girar en ambos sentidos o mantener botón central del ratón.
- Acercar/Alejar: girar la rueda del ratón.

Especificaciones de los personajes

Agentes Trabajadores

Un honesto estará deambulando hasta que aparezca una tarea disponible, en ese momento deberá ir directamente hacia el punto donde debe realizar la tarea. Si un honrado entra en una sala en la que hay un cadáver, se deberá desplazar hasta el botón de votación.

Un honrado, recordará los personajes de las dos últimas salas en las que hayan estado.

Este personaje, tendrá un sistema de sospecha para cada uno de los distintos agentes, que aumentará o disminuirá según avance la partida y según las siguientes acciones:

1. Presenciar asesinato de agente supone un aumento al 100% de sospecha.
2. Que se encuentre en la sala del cadáver al encontrarlo, 80% de sospecha.
3. Cercanía (en salas) al cadáver. 30% de sospecha en primera, 15% en segunda.

Si se produce un “sabotaje”, el personaje Honrado deberá dejar la tarea que esté realizando e ir al punto donde se ha producido el sabotaje para arreglarlo, para este evento tendrá una probabilidad de actuación de 60%, ya que puede “dar por hecho” que otros agentes arreglarán el sabotaje.

Una vez explicada la descripción general de un agente Honrado, se pasan a especificar las subcategorías que salen de este. Estas subcategorías tendrán en común todas las percepciones, acciones, diagramas y estructuras, siendo su única diferencia los lugares a los que se acercan para realizar sus tareas. Con esto se pretende una mayor riqueza en el movimiento de los personajes y establecer rutas específicas para cada uno, simulando a trabajadores de verdad en un recinto.

Tabla de percepciones:

Nombre	Implementación	Acceso
Está trabajando	Comprobar si está activado el flag trabajando.	Pull
Está en objetivo	Comprobar que está en el punto para realizar su tarea, pulsar el botón que avisa de un cadáver o en la mesa de reunión para la votación.	Pull
Cadáver encontrado	Recibir evento desde el mundo de que se ha encontrado un cadáver y quien está en esa sala.	Push

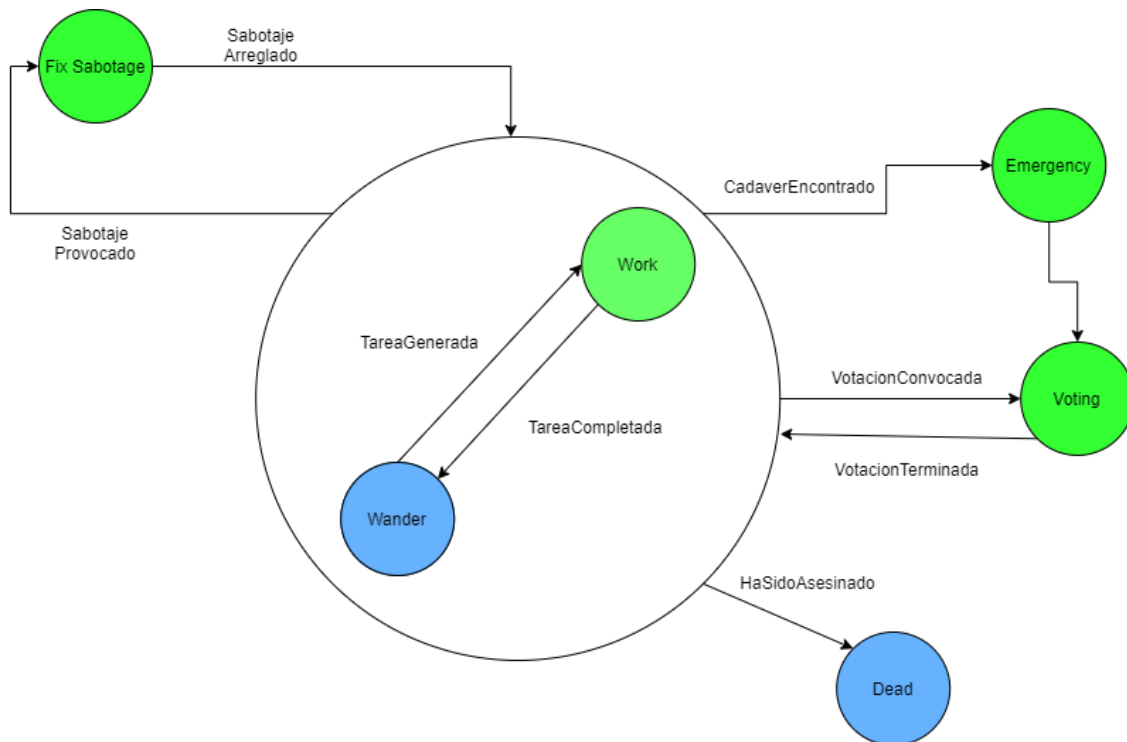
Votación convocada	Recibir evento desde el mundo de que se ha convocado una votación.	Push
Asesinato presenciado	Recibir evento desde el mundo de que en la sala está ocurriendo un asesinato.	Push
Cambio de sala	Recibir evento de que está en una sala nueva (recibiendo la lista de agentes que hay en ella).	Push
Agente nuevo en sala	Recibir evento de que un nuevo agente ha entrado en la sala.	Push
Nueva tarea	Recibir evento del mundo de que se ha generado una nueva tarea.	Push
Estás siendo asesinado	Recibir evento de que el agente está siendo asesinado.	Push
Estás muerto	Recibir evento de que el agente ha sido asesinado.	Push
Sabotaje creado	Recibe evento de que un sabotaje se ha creado para ir a arreglarlo.	Push
Está votando	Comprobar si está activado el flag de votando.	Pull
Agente expulsado	Recibir evento de que un agente ha sido expulsado del juego.	Push

Tabla de acciones:

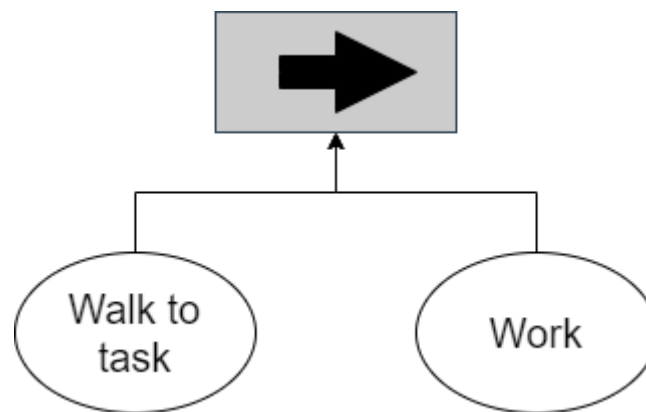
Nombre	Implementación	Efecto
Avanzar	Iniciar el avance del personaje (corriendo).	El personaje avanzará hacia el objetivo.
Parar	La velocidad del movimiento del personaje se coloca a 0.	El personaje detiene su movimiento.
Trabajar	Tras un período de tiempo, se notifica al mundo que se ha realizado una tarea.	El personaje realiza un trabajo durante un período de tiempo determinado.
Estar muerto	El personaje se queda quieto y se cambia su <i>sprite</i> al cadáver.	El personaje detiene su movimiento y cambia su <i>sprite</i> .
Acusar	Añadirá un voto a otro personaje durante el momento de votación.	Vota a otro personaje con los datos obtenidos.

Diagrama descriptivo: máquina de estados jerárquica

Máquina general:



Estado *work*:



Estructura de datos:

El honesto tiene 2 atributos, siendo el segundo modificable:

Tipo de agente	Expresado con una de las clases definidas.
Velocidad de avance.	Expresada en metros por <i>frame</i> .

Además, tiene 5 estructuras de datos modificables:

Diccionario de agentes en la sala que está.	La clave será el hashcode del agente.
Diccionario de agentes que tiene en su memoria (dura dos salas).	La clave será el hashcode del agente. Sirve para trackear temporalmente dónde han estado los agentes.
Diccionario de agentes de los que sospecha.	La clave será el hashcode del agente, el valor será un porcentaje de sospecha (de 0 nada de sospecha a 100 totalmente seguro).
Objetivo actual	Localización a la que debe ir para trabajar en una tarea.
Cadáver encontrado	Identidad del cadáver que ha encontrado.
Sala de cadáver	Sala en la que se encuentra el cadáver encontrado (si lo hubiera).

Agentes traidores

Un agente Traidor tiene como objetivo acabar con la vida de todos los agentes no Traidores. A la vez, debe esconder su identidad (haciéndose pasar por agente Honrado) para que los Honrados no le descubran y expulsen.

Un Traidor estará deambulando por el recinto hasta que su habilidad de matar deje de estar en *cooldown*, es decir, recargándose. Si ve que puede matar a un Honrado, lo hará y cambiará de sala lo antes posible. Después, esperará a que su acción de matar se recargue y volverá a buscar su próximo objetivo.

La acción de matar o no a un agente Honrado se basa en una toma de decisión en la que influyen diferentes factores, a saber:

- La cantidad de agentes en la sala guarda una relación directa con la necesidad de fingir, e inversa con la posibilidad de matar (salvo cuando haya 0 agentes). Si solo hay 1 único agente Honrado en la sala con el Traidor, sus posibilidades de matar valdrán 1. En cualquier otro caso, valdrán 0. Si hay más de 1 agente Honrado en la sala con el Traidor, su necesidad de fingir valdrá 1. En cualquier otro caso, será 0. La necesidad de fingir determinará si el Traidor decide fingir.
- Si un segundo agente Honrado acaba de salir de la sala (es decir, ha dejado solos al Traidor y víctima honrada), el miedo del Traidor a ser descubierto por este valdrá 1.

- Si en la sala anterior se ha encontrado a agentes Honrados (es decir, hay alguien cerca), el miedo del Traidor a ser descubierto por ellos aumentará de forma lineal, siguiendo una relación directa.
- Ambos miedos afectarán en partes iguales (50% cada uno) al miedo del traidor a ser descubierto.
- El número total de tareas completadas por los Honrados afecta al riesgo de perder del Traidor de forma directa, siguiendo una curva sigmoide. Este riesgo determinará la acción de sabotear o no.
- El riesgo de perder por tareas completadas afecta en un 20% de forma directa a las ganas de matar. El miedo a ser descubierto afecta de forma inversa en un 30% a las ganas de matar. La posibilidad de asesinar afecta en un 50% de forma directa a las ganas de matar. Las ganas de matar determinan la acción de asesinar.

Todo esto se encuentra explicado de forma esquemática en el apartado “Diagrama descriptivo”, mediante un Sistema de Utilidad.

Además de poder matar otros agentes, un Traidor podrá también realizar “sabotajes” en el recinto. Para realizarlos deberá acercarse a un punto específico. Estos sabotajes se marcarán con una alarma en el recinto para alertar a todos los Honestos. Si los trabajadores Honestos no desactivan este sabotaje en un tiempo determinado, los Traidores ganan.

Solo los Traidores tienen la habilidad de viajar entre “conductos de ventilación” situados de forma estratégica en diferentes salas. Este viaje les permite acceder a una sala sin pasar por las que se encuentren en medio (de A a C sin pasar por B).

Los agentes traidores siguen el mismo esquema de reproducción explicado en los agentes Honrados.

A la hora de votar para expulsar un agente, un traidor escogerá a uno de los que lo votaron en la última votación. En caso de no haber ninguno, elegirá un agente al azar.

Tabla de percepciones:

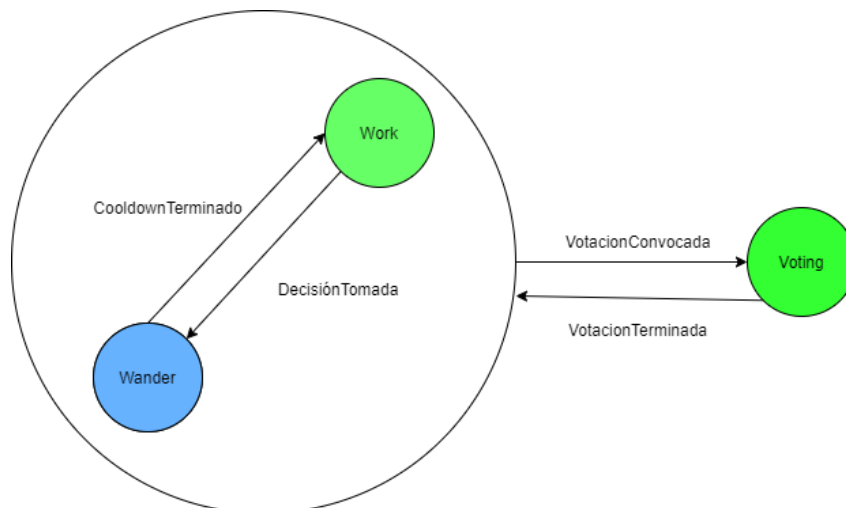
Nombre	Implementación	Acceso
Está en objetivo	Comprobar si la posición del personaje y la del objetivo coinciden	Pull
Está matando	Comprobar si finaliza la animación de matar	Pull
Está fingiendo	Comprobar si está activado el flag de fingir	Pull
Está sabotando	Comprobar si finaliza la animación de sabotear	Pull
Cambio de sala	Recibir evento de que está en una sala nueva (recibiendo la lista de agentes que hay en ella).	Push
Agente nuevo en la sala	Recibir el evento de que un nuevo agente ha entrado en la sala.	Push
Votación convocada	Recibir evento de que un agente ha convocado una votación (interrumpiendo lo que se estuviera haciendo).	Push
Está votando	Comprobar si está activado el flag de votando.	Pull
Se te ha votado	Recibe el evento de que otro agente lo ha votado.	Push
Agente expulsado	Recibir evento de que un agente ha sido expulsado del juego.	Push

Tabla de acciones:

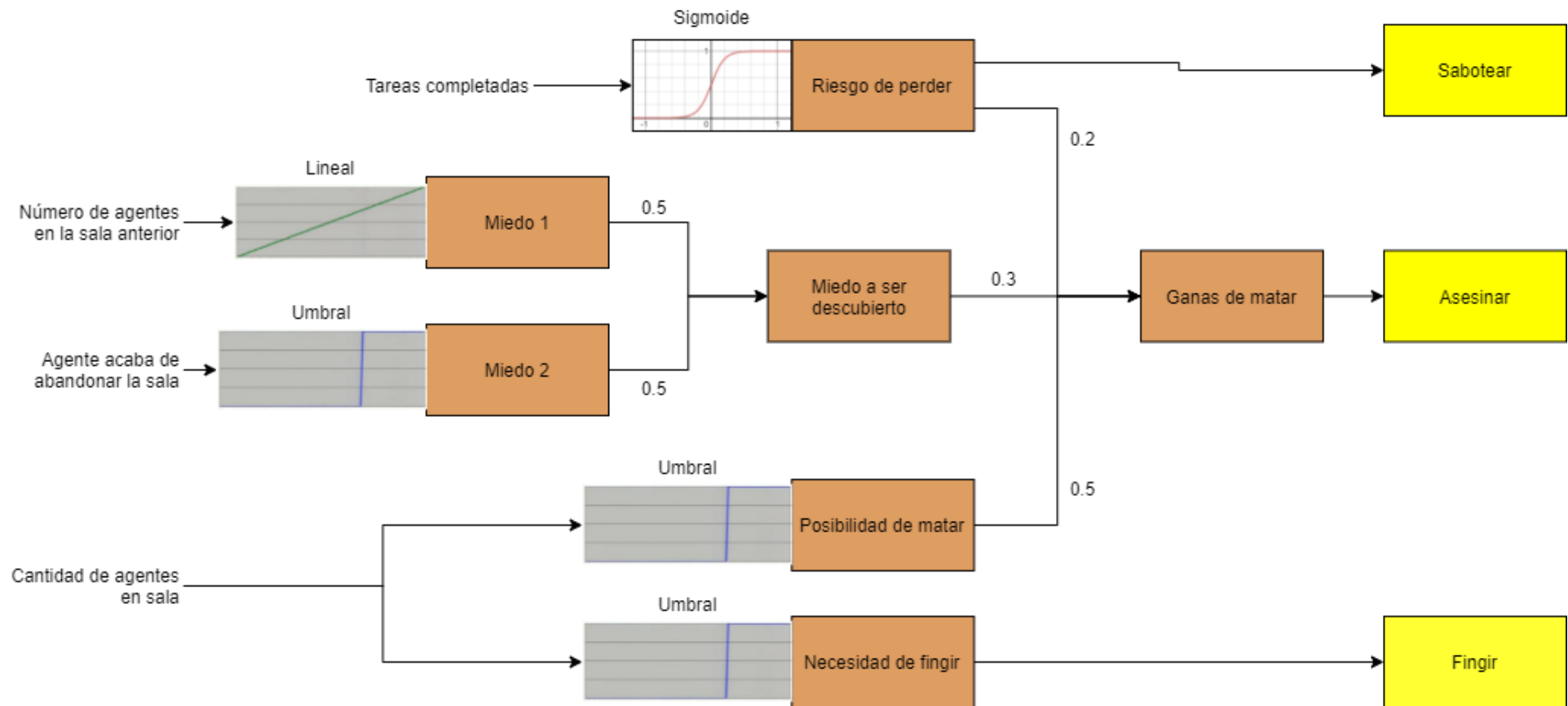
Nombre	Implementación	Efecto
Avanzar	Iniciar el avance del personaje (corriendo)	El personaje avanzará hacia el objetivo.
Parar	La velocidad del movimiento del personaje se coloca a 0.	El personaje detiene su movimiento.
Matar	El personaje interrumpe a un agente Honrado para matarlo.	Un agente Honrado muere.
Fingir trabajo	El personaje va a un puesto de trabajo y activa la animación de trabajar, pero sin contribuir a las tareas realizadas.	El personaje realiza un trabajo durante un período de tiempo determinado.
Sabotear	El personaje utiliza su habilidad para sabotear el recinto.	Los agentes Honrados son redirigidos a solucionar el sabotaje si no quieren perder la partida.
Acusar	El personaje añadirá un voto a otro personaje durante el momento de votación.	El personaje vota a otro personaje con los datos obtenidos.

Diagrama descriptivo: máquina de estados jerárquica

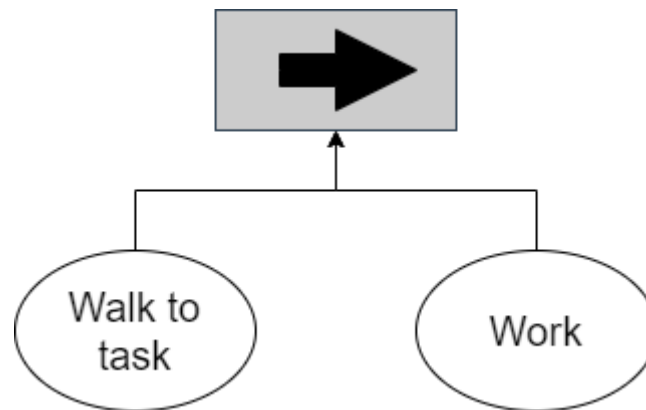
La generación de tareas del traidor se corresponde con la finalización del tiempo de *cooldown* o espera entre asesinatos:



El estado *work* del traidor contendrá el siguiente sistema de utilidad:



Fingir contendrá el árbol de comportamientos de los agentes honrados:



Estructura de datos:

El traidor tiene 4 atributos, los 3 primeros modificables:

<i>Cooldown</i> para decisión (asesinato, sabotaje o fingir).	Expresado en segundos.
Velocidad de avance.	Expresada en metros por <i>frame</i> .
Tipo de agente.	Expresado con una de las claves definidas (siempre será traidor).

Además, tiene 5 estructuras de datos modificables:

Diccionario de agentes en la sala que está.	La clave será el hashcode del agente.
Diccionario de agentes que tiene en su memoria (dura dos salas).	La clave será el hashcode del agente. Sirve para seguir temporalmente dónde han estado los agentes.

Interacción con el entorno/mundo

El mundo se encarga de informar a los agentes de los siguientes eventos:

- Aparición de una nueva tarea (con el tipo de tarea).
- Se ha convocado una votación. Todos los agentes reciben la información de qué agente ha muerto y dónde ha muerto.
- Se te ha votado en la votación (individual para cada agente).
- Se ha expulsado a un agente.
- Se ha provocado un sabotaje. Todos los agentes reciben dónde ir a solucionar el sabotaje.
- Se ha solucionado el sabotaje.
- Un agente te ha asesinado (privado para el agente asesinado).
- Cadáver encontrado. Cuando un agente encuentra un cadáver, se le informa solo a él. Deberá convocar una votación.
- Cambio de sala e información de la nueva sala (agentes en la sala en la que acaba de entrar).
- Asesinato presenciado. Tiene que estar en la sala donde se está cometiendo. Individual de cada agente.
- Agente nuevo en la sala. Individual de cada agente, se da la información de quién ha entrado en la sala.
- Fin de partida (por cualquier motivo).

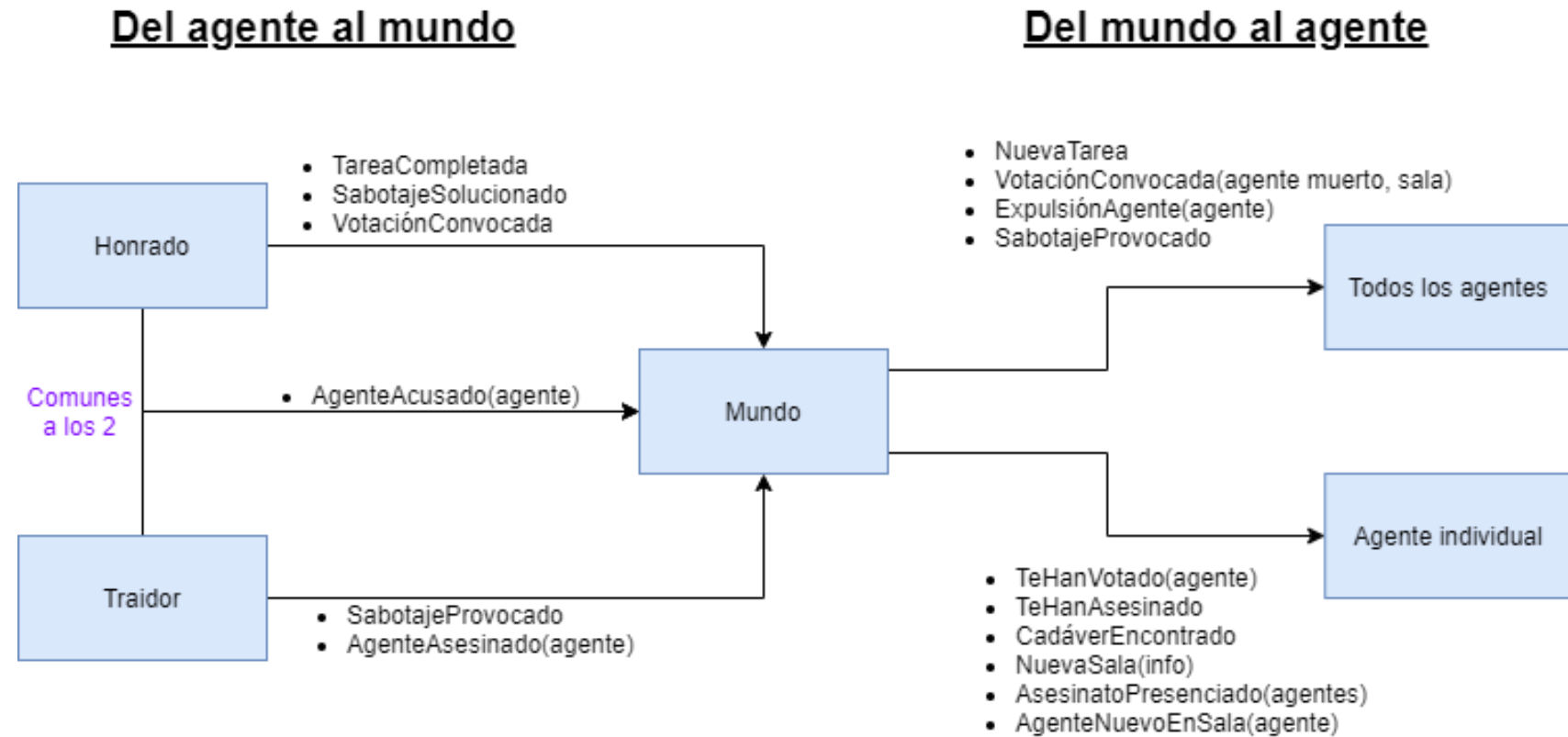
Los agentes se encargan de informar al mundo de los siguientes eventos:

- Ha completado la tarea que estaba haciendo.
- Ha solucionado un sabotaje.
- Ha convocado una votación.
- Ha votado a otro agente durante el proceso de votación.

Además de los anteriores, los agentes traidores se encargan de informar al mundo de lo siguiente:

- Ha asesinado a un agente.
- Ha provocado un sabotaje.

Siguiendo el siguiente diagrama:



Descripción de algoritmos

Debido a una serie de fallos obtenidos en la toma de decisiones mediante Sistemas de Utilidad al utilizar la API proporcionada, se decidió implementar desde cero uno que cumpliera las necesidades buscadas. Para ello, se han creado dos funciones:

- Una para obtener el valor de un factor de decisión normalizado entre 0 y 1 a partir de una curva definida por una serie de puntos dados por el programador y otro factor de entrada (también normalizado).
- Otra para obtener el valor de otro factor de decisión normalizado a partir de la suma con pesos de varios factores proporcionados por el programador.

Una vez obtenidos los factores decisivos (normalizando los datos de entrada y usando las funciones anteriormente descritas en los casos necesarios), se ejecuta el comportamiento asociado al factor de mayor valor.

Se han desarrollado sistemas dedicados al correcto seguimiento del estado de todos los agentes en el mundo, incluyendo la sala en la que se encuentran ellos mismos y los demás, sistemas de sospecha individual y priorización de tareas.

Se han desarrollado diferentes estados finales de juego a los que se pueden llegar dependiendo en gran factor de las condiciones iniciales de la simulación, aunque pueden variar debido al azar que se encuentra naturalmente en este tipo de simulaciones.

Además, se ha incluido un algoritmo de generación aleatoria dentro del *NavMesh* al que los agentes puedan caminar cuando no están ociosos, evitando de esta forma que se queden parados y dándole dinamismo a la simulación.

Para terminar, los agentes Trabajadores tienen un “cono de visión” para detectar cadáveres en las habitaciones. El funcionamiento es simple, al entrar en una sala el agente recorre una lista que contiene los cadáveres de la misma. Se calcula la distancia del agente al cadáver y, si esta es mayor que la distancia de visión del agente, se ignora. Si la distancia entre ambos es correcta, se comprueba el ángulo formado entre el vector *forward* y un vector que los une. En caso de que el ángulo entre dentro de la amplitud de visión, se guarda qué agente ha muerto y dónde, y el trabajador vivo va a dar la alarma.

Resultados

Se presenta un simulador capaz de crear situaciones muy complejas de interacción entre agentes. Observar el recorrido y toma de decisiones hace que el estado del juego se vuelva interesante y, aunque posible, difícilmente predecible.



Interesante es destacar la importancia que implica la división entre habitaciones a la hora de detectar asesinatos. Un agente Impostor podría salvarse de sospecha si mata a un agente e inmediatamente cambia de habitación, pudiendo así encadenar asesinatos.

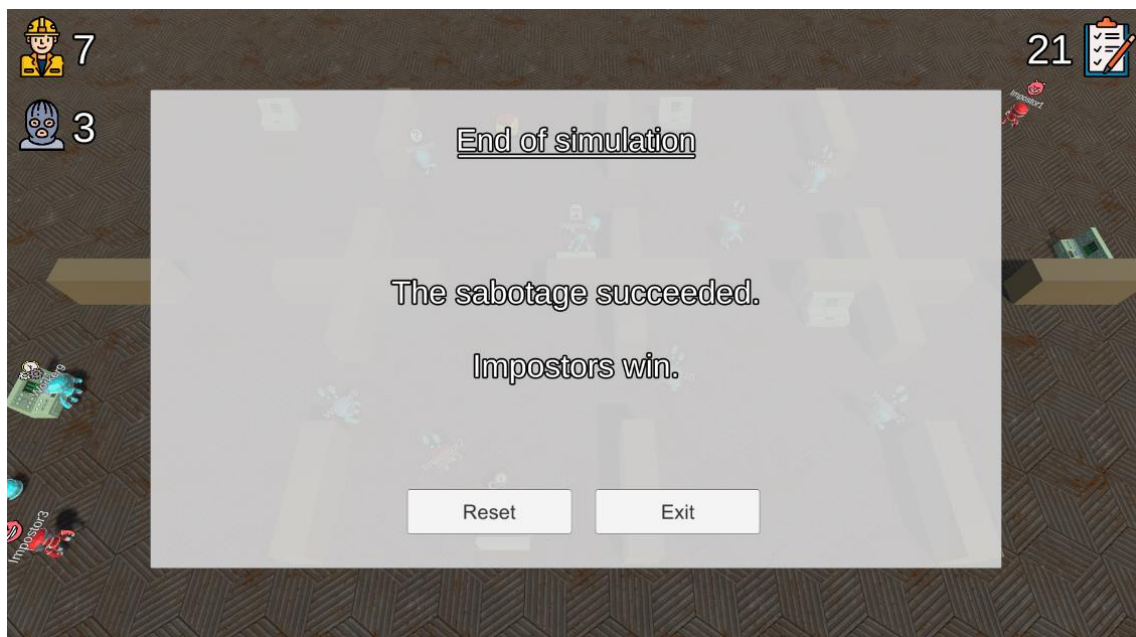
En situaciones en las que exista igualdad numérica, los Impostores estarán en desventaja, ya que elegirán mantener su papel de trabajador antes de asesinar y ser descubiertos.



En aquellas situaciones donde un Impostor sea descubierto, pero a la vez se realice un sabotaje, los agentes Trabajadores decidirán priorizar el arreglo de este sabotaje antes de ir al botón de votación, ya que, si pasara demasiado tiempo, perderían el juego al completo.



Existe la posibilidad de que, aun quedando varios agentes Trabajadores en el juego, un sabotaje tenga éxito. Esto se debe a que los Trabajadores pueden decidir no ir al sabotaje si consideran que sus compañeros lo harán, o incluso podría darse el caso de morir de camino a arreglarlo.



Tras realizar diferentes casos de prueba en la simulación, se ha llegado a las siguientes conclusiones:

- El número de Trabajadores debe mantener cierta proporción con el de Impostores para mantener una simulación equilibrada. Grandes diferencias en los números hacen que la balanza se incline mucho hacia uno de los lados.
- Un número elevado de Trabajadores (aproximadamente a partir de 17) reducirá muchísimo las posibilidades de ganar de los Impostores, ya que estos nunca decidirán matar a un Trabajador debido a que es muy poco probable encontrarse a solas con ellos en una habitación.
- El aparente azar a la hora de asignar tareas juega un factor importante durante el juego. En simulaciones equilibradas, la asignación de tareas afectará a la posición de los agentes y por ello a la eficiencia de los Impostores para matarles.
- Los números óptimos para ofrecer una simulación interesante varían entre 5-2 y 13-6 Trabajadores e Impostores respectivamente. El número de tareas a completar debe ser proporcional al número de Trabajadores; en exceso darán ventaja a los Impostores, en defecto darán ventaja a los Trabajadores.

Reparto de tareas

De forma general, los tres miembros del grupo han contribuido de igual forma al contenido de la práctica. Ha habido tareas realizadas en conjunto y otras de forma individual, facilitando así el trabajo en paralelo.

Algunas como las máquinas de estado, comportamiento de los agentes, el controlador del mundo, el sistema de votación y el sistema de generación de tareas han sido realizadas y revisadas en mayor o menor medida por todos los integrantes, por lo que se omiten en las listas individuales.

En cuanto al trabajo individual, a continuación se exponen los diferentes miembros y listadas sus principales funciones.

Pablo

- Implementación del Sistema de Utilidad para la toma de decisiones malvadas de los agentes Impostores.
- Comunicación de los agentes con el controlador del mundo para notificar asesinatos, descubrimiento de cadáveres, sabotajes y votaciones convocadas.
- Cono de visión de los agentes Trabajadores y detección de cadáveres.

Juan José

- Implementación del UI del Menú Principal.
- Sistema de sospecha de los agentes.
- Detección de cambio de habitación y memoria de los agentes respecto a ello.
- Implementación del sistema de navegación (*NavMesh*).

Sergio

- Búsqueda e implementación de recursos visuales para la simulación tales como modelos, animaciones y HUD.
- Condiciones de fin de simulación.
- Cámara principal y controles.
- Generación aleatoria de agentes en la escena (referido a sus posiciones).

Referencias

- Game Dev Guide. (2019, 24 junio). *Building a Camera Controller for a Strategy Game*. YouTube.
<https://www.youtube.com/watch?v=rnqF6S7PfFA&feature=youtu.be>
- @Quaternius. (s. f.). *Cute Robot 3d model. Free download.* | Creazilla. <https://creazilla.com/nodes/5206-cute-robot-3d-model>
- Flaticon. (s. f.). *Free icons designed by Freepik.*
<https://www.flaticon.com/authors/freepik>
- *How to get a random point on NavMesh? - Unity Answers.* (s. f.). Unity Answers. <https://answers.unity.com/questions/475066/how-to-get-a-random-point-on-navmesh.html>