**Comportamiento de Personajes**

**Grupo 1**

Alejandra Casado Ceballos

Pedro Casas Martínez

Juan Manuel Carretero Ávila

**22/12/2020**

**Profesores**

Dan Casas Guix

Carlos Garre del Olmo

**Grado en diseño y desarrollo de videojuegos**



**NOMBRE DEL JUEGO??**

Especificación de proyecto

**Índice**

[1. Descripción general 1](#_Toc59725205)

[2. Especificación de los personajes 2](#_Toc59725206)

[Drones 2](#_Toc59725207)

[Reclutador 7](#_Toc59725208)

[Torreta 9](#_Toc59725209)

[3. Interacción con el entorno/mundo 11](#_Toc59725210)

[4. Descripción de algoritmos 13](#_Toc59725211)

[5. Resultados 14](#_Toc59725212)

[6. Reparto de tareas 15](#_Toc59725213)

[7. Webgrafía 16](#_Toc59725214)

1. **Descripción general**

Se trata de una simulación en la que se pueden ajustar los parámetros de forma que se pueden probar cómo varían los distintos comportamientos de los cinco agentes existentes al cambiarlos. Los agentes se tratan de un dron de ataque a distancia, un dron de ataque cuerpo a cuerpo, un reclutador, una torreta y finalmente un entorno que genera estos drones de forma inteligente.

Se ha decidido que el diseño sea 3D con modelos propios desarrollados para esta práctica.

La simulación consiste en una batalla de distintos equipos por conquistar el mayor número de bases en el tiempo de duración de una partida. El mundo está dominado por distintos equipos diferenciados por su color. Las bases consisten en una serie de nodos repartidos por el escenario que sirven como punto de reaparición para las tropas del equipo al que pertenezcan. En el inicio de una partida todos los equipos están equilibrados con el mismo número de bases y tropas.

Los equipos deberán defender sus bases al mismo tiempo que se coordinan para conquistar bases enemigas. Hay distintos tipos de enemigos que interactúan entre ellos para formar estrategias de ataque. A nivel visual, la simulación tendrá una apariencia minimalista ambientada en una sala de experimentación de un laboratorio gigante y los agentes son robots.

Respecto a la implementación del proyecto, no se ha utilizado la librería facilitada debido a que la inteligencia artificial es un campo que interesa a los miembros del equipo y que se pretende estudiar más en profundidad durante el máster. Por este motivo, se ha desarrollado una máquina de estados, un árbol de decisión y un sistema de utilidad propios y genéricos que han servido para el correcto desarrollo de la práctica.

1. **Especificación de los personajes**

## **Drones**

1. Nombre y descripción textual detallada

Son los soldados encargados de combatir contra los enemigos, defender y conquistar bases. Su comportamiento está basado en un sistema de utilidad de forma que dependiendo de la situación consideren más importante una acción u otra. Estos drones pueden ser reclutados por los reclutadores y seguir sus órdenes, aunque podrían dar prioridad a otra acción si su influencia es muy fuerte como atacar a enemigos cercanos o defender una base aliada que esté siendo conquistada muy cerca. Su sistema de utilidad es jerárquico, dando lugar cada una de las acciones a una máquina de estados que define su comportamiento.

* Tipos de dron: hay dos tipos de drones diferenciados por su velocidad, vida y forma de atacar:
  1. Cuerpo a cuerpo: son más lentos que los drones a distancia, pero tienen más vida. Su forma de atacar consiste en acercarse al enemigo más cercano y golpearle hasta debilitarlo.
  2. A distancia: son más rápidos, pero poseen menos puntos de vida. Atacan a distancia disparando, para lo que previamente tendrán que apuntar a su enemigo.

1. Tabla de percepciones

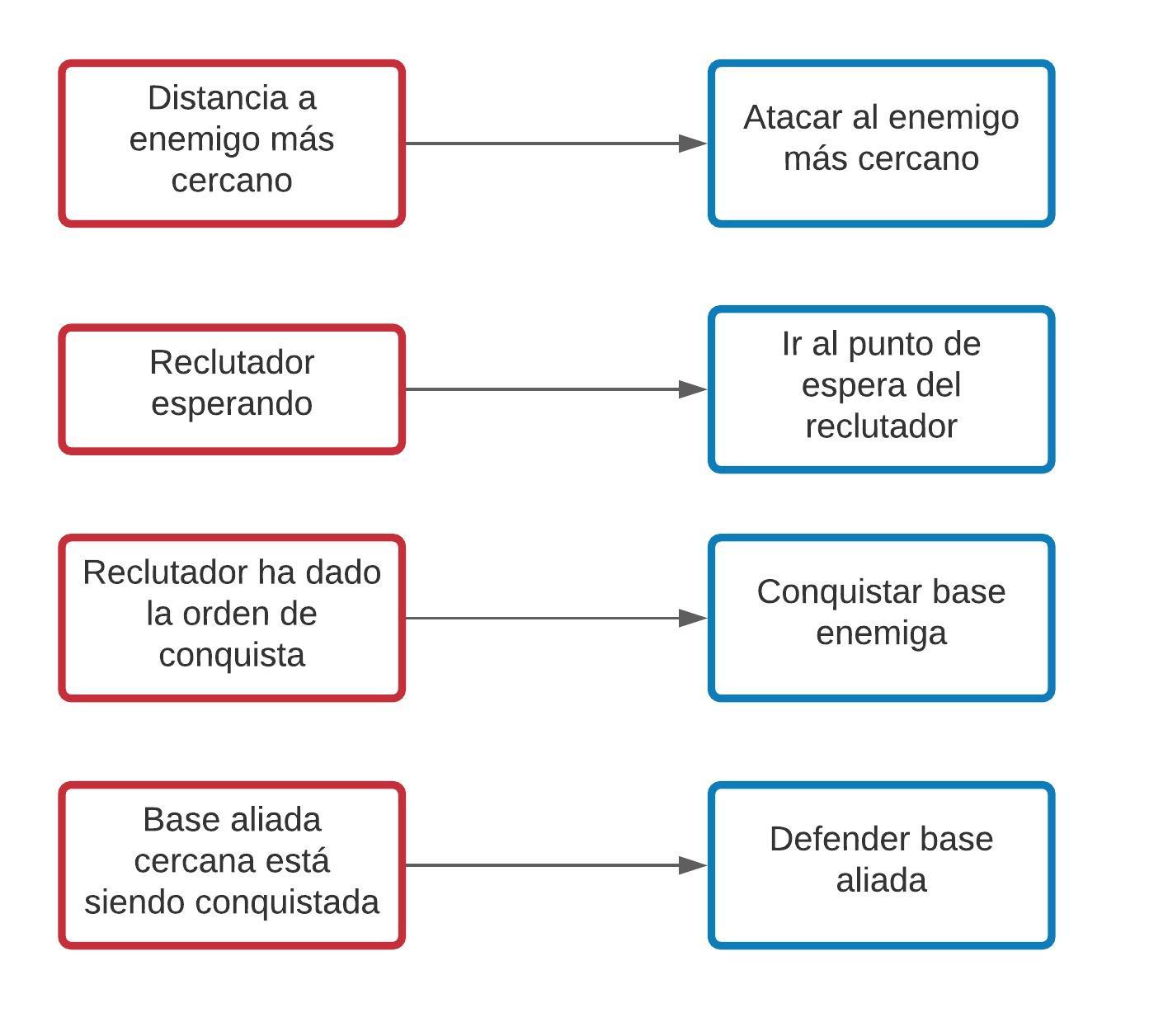
|  |  |  |
| --- | --- | --- |
| **Nombre** | **Implementación** | **Acceso** |
| Distancia a enemigo más cercano | Calcular la distancia del enemigo más cercano | *Pull* |
| Reclutador esperando | Si ha sido reclutado, escuchar evento del reclutador de movilizarse al punto de espera | *Push* |
| Reclutador ha dado la orden de conquista | Escuchar evento del reclutador de orden de ataque | *Push* |
| Base aliada cercana está siendo conquistada | Comprobar la distancia a la base neutral más cercana que esté siendo conquistada por los enemigos | *Pull* |
| Vida agotada | Comprobar si el valor de vida ha llegado a cero | *Pull* |
| Ha reaparecido | Escuchar evento del mundo de que ha reaparecido en alguna base | *Push* |
| Ha llegado al punto de espera | Comprobar si está en el rango de espera especificado por su reclutador | *Pull* |
| Ha llegado al interior de la base objetivo | Comprobar si está dentro de la base que tiene como objetivo | *Pull* |
| Enemigo en rango de ataque | Comprobar si un enemigo está en el rango de ataque | *Pull* |
| Ha disparado | Comprobar si ha generado un proyectil en dirección al objetivo | *Pull* |
| Enemigo en punto de mira | Comprobar si está apuntando a un enemigo | *Pull* |
| Ha atacado | Comprobar si ha terminado de atacar cuerpo a cuerpo a un enemigo | *Pull* |

1. Tabla de acciones

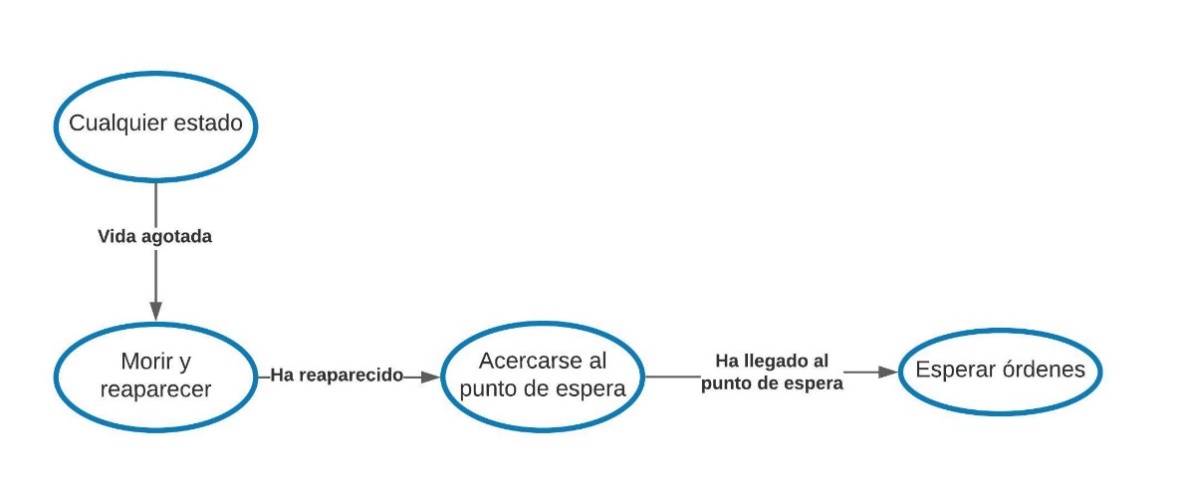
|  |  |  |
| --- | --- | --- |
| **Nombre** | **Implementación** | **Efecto** |
| Atacar al enemigo más cercano | El dron calcula el enemigo situado a menor distancia y a través de una FSM le ataca | El dron ataca a un enemigo |
| Morir y reaparecer | Al agotar sus vidas la posición del dron cambiará a la de una base aliada, en caso de no existir base aliada, muere para siempre | El dron desaparece y vuelve a generarse en una base aliada |
| Acercarse al enemigo más cercano | Si el dron es de tipo cuerpo a cuerpo, al detectar un enemigo en su rango de ataque y mediante un algoritmo de *pathfinding*, se aproxima a este | El dron se mueve hacia el enemigo |
| Atacar | Si el dron es de tipo cuerpo quita al enemigo el daño correspondiente | El dron genera la animación de atacar cuerpo a cuerpo |
| Ir al punto de espera del reclutador | El dron a través de una FSM se acerca a la zona establecida y posteriormente espera a las órdenes del reclutador | El dron se dirige hacia el punto de espera y se queda quieto |
| Acercarse al punto de espera | El dron mediante un algoritmo de *pathfinding* se dirige hacia el punto de espera establecido por el reclutador | El dron se mueve hacia el reclutador |
| Esperar órdenes | El dron se queda esperando a que le den alguna orden | El dron se queda quieto |
| Conquistar base enemiga | El dron a través de un FSM se dirige hacia una base enemiga y la conquista | Nueva base conquistada para el equipo del dron |
| Entrar en el interior de la base objetivo | Mediante un algoritmo de *pathfinding* el enemigo se dirige al centro de la base a conquistar | El enemigo entra en la base a conquistar |
| Defender base aliada | Mediante un FSM el dron entrará en la base que está siendo atacada y atacará a los enemigos que estén intentando conquistarla | El dron defiende su base |

1. Diagrama descriptivo: Sistema de utilidad jerárquico

Sistema de utilidad Nivel 1:



FSM Nivel 2 (Ir al punto de espera del reclutador):



FSM Nivel 2 (Conquistar o defender base):

Diagrama

Descripción generada automáticamente

FSM Nivel 2 y 3 (Atacar enemigo más cercano dron a distancia):

Diagrama

Descripción generada automáticamente

FSM Nivel 2 y 3 (Atacar enemigo más cercano dron a cuerpo a cuerpo):



1. Estructuras de datos

El enemigo tendrá seis atributos modificables:

* Velocidad de movimiento
* Rango de ataque
* Vida
* Daño de ataque
* Velocidad de ataque
* Influencia del reclutador: se implementará como un cambio en los pesos del sistema de utilidad. Cuanto más alta sea esta influencia menos probable será que el dron de prioridad a acciones que no sean las órdenes de su reclutador.

## **Reclutador**

1. Nombre y descripción textual detallada

El reclutador es un tipo de dron especial que no se encarga de defender bases o atacar enemigos sino de reclutar aliados. Irá acercándose a sus aliados para reclutarlos en su equipo y cuando haya reclutado suficientes hará un llamamiento a sus reclutas para reunirse en un punto de espera. Este punto de espera es decidido por el reclutador como un lugar cercano a la base no aliada más cercana desde donde no puedan ser atacados por sus torretas. Cuando los reclutas lleguen al punto de espera, el reclutador dará la orden de ataque para intentar conquistar la base. Si no consigue que la cantidad esperada de reclutas lleguen al punto de espera, volverá a reclutar más drones hasta lograr su objetivo. De este agente depende en gran medida la victoria del equipo, pues los drones por sí mismos solo se encargan de defender sus bases por lo que, en el mejor caso, terminarían la partida como empezaron. Sin embargo, este agente permite al equipo ganar territorio a sus enemigos durante la partida.

1. Tabla de percepciones

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Implementación** | **Acceso** |
| Aliado en rango de reclutamiento | Comprobar si el aliado se encuentra dentro del rango de reclutamiento | *Pull* |
| Ha reclutado suficientes tropas | Comprobar si el número de tropas reclutadas es suficiente | *Pull* |
| Hay aliados suficientes para atacar la base | Comprobar si hay aliados suficientes para realizar una conquista | *Pull* |
| Base conquistada | Escuchar evento de la base cuando ha sido conquistada | *Push* |
| Vida agotada | Comprueba si el valor de vida es igual a 0 | *Pull* |
| Ha reaparecido | Escuchar evento de que ha sido regenerado en el mundo | *Push* |

1. Tabla de acciones

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Implementación** | **Efecto** |
| Acercarse al aliado más cercano | El reclutador determinara mediante *pathfinding* de Unity la posición del aliado y el camino a seguir | El reclutador se acerca a un aliado |
| Reclutar aliado | El reclutador añade a su lista de tropas influenciadas el aliado reclutado | El aliado seguirá las órdenes del reclutador |
| Ir a la base enemiga más cercana | El reclutador calculará entre las bases enemigas cual atacar y enviará una orden a las tropas reclutadas para que le sigan. | El reclutador avanza con sus tropas a una base por conquistar |
| Esperar a las tropas reclutadas | El reclutador espera a que lleguen las tropas reclutadas antes de atacar | Las tropas se colocan en formación para el ataque |
| Atacar la base enemiga | El reclutador manda la orden de ataque y atacan | Las tropas comienzan su conquista |
| Morir y reaparecer | El reclutador pierde todos sus puntos de vida y desaparece del mundo | El reclutador muere |

1. Diagrama descriptivo: Máquina de estados de un nivel

FSM:

Diagrama

Descripción generada automáticamente

1. Estructuras de datos

El reclutador tendrá cinco atributos modificables:

* Velocidad de movimiento
* Vida
* Rango de reclutamiento: distancia máxima a la que puede reclutar a un dron
* Número de drones a reclutar
* Porcentaje de drones mínimo en punto de espera para iniciar un ataque

**Torreta**

1. Nombre y descripción textual detallada

Las torretas están posicionadas en las bases y se encargan de atacar a todos los agentes no aliados de la base. Por tanto, si la base es neutral, atacarán a todos los agentes. Tienen la limitación de que no pueden atacar al interior de la base, por lo que deben intentar acabar con los enemigos antes de que logren entrar.

1. Tabla de percepciones

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Implementación** | **Acceso** |
| Un enemigo está en rango de ataque | Comprobar si el enemigo se encuentra dentro de su rango de ataque | *Pull* |
| Un enemigo está a tiro | Comprobar si se ha calculado la posición aproximada del enemigo | *Pull* |
| Ha disparado | Comprobar si se ha disparado al enemigo | *Pull* |

1. Tabla de acciones

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Implementación** | **Efecto** |
| Idle | La torreta espera a que se acerque algún enemigo a la base | La torreta está quieta |
| Apuntar | Calcula la posición aproximada donde se va a desplazar el jugador | La torreta apunta |
| Disparar | Dispara una bala que si colisiona con el jugador supondrá una disminución en la energía del jugador | La torreta dispara |

1. Diagrama descriptivo: Máquina de estados de un nivel

FSM:

Diagrama

Descripción generada automáticamente

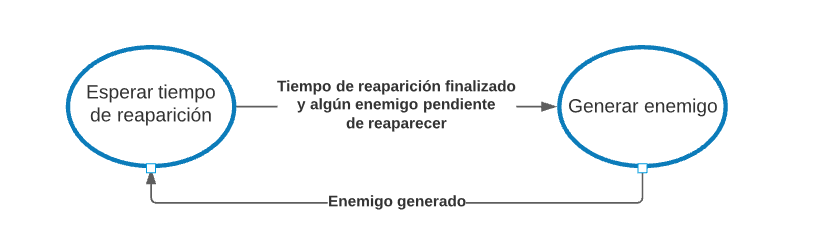
1. Estructuras de datos

La torreta tendrá cuatro atributos modificables:

* Velocidad de apuntado
* Cadencia de tiro
* Rango de ataque
* Daño de las balas

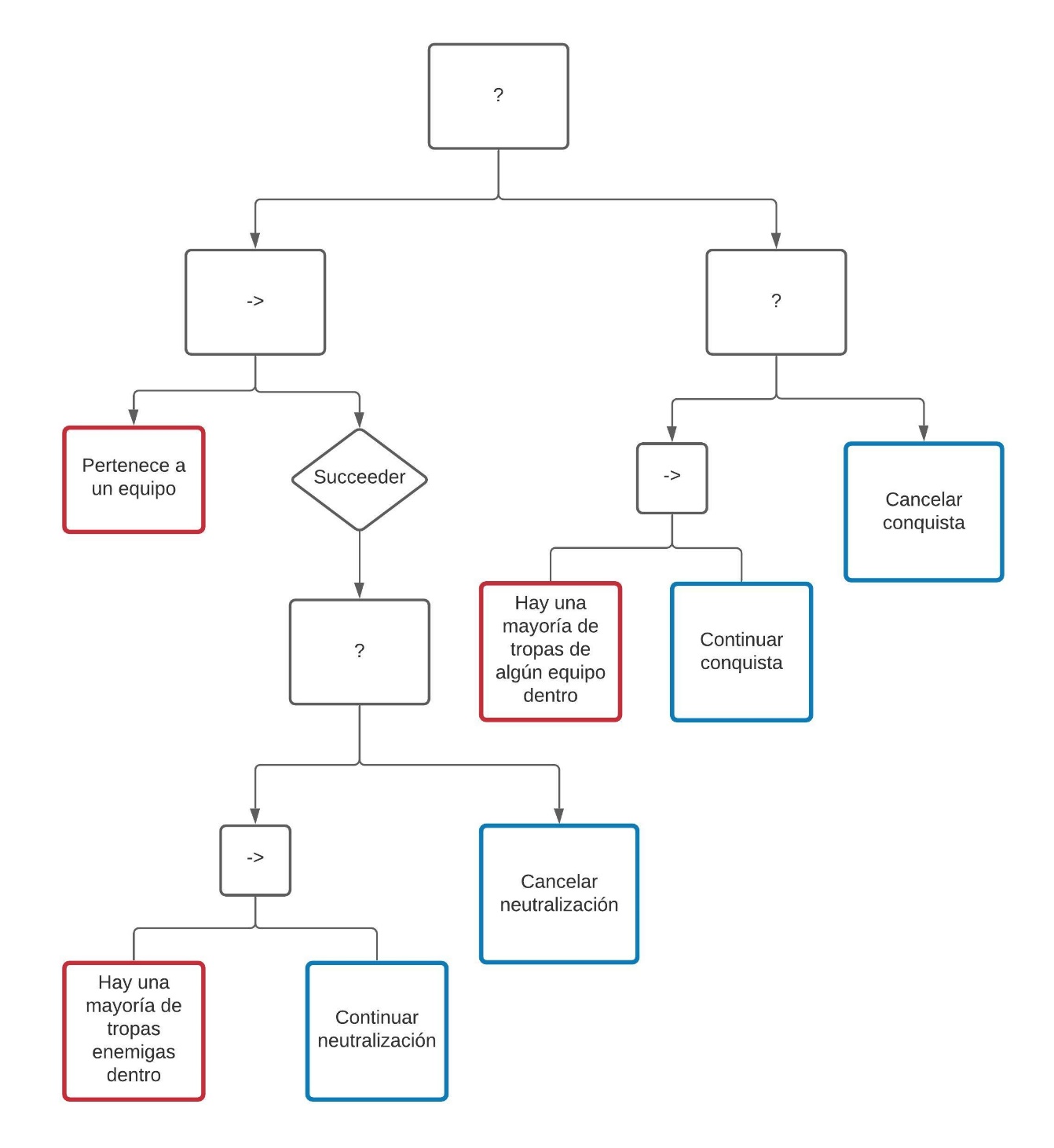
1. **Interacción con el entorno/mundo**

El mundo está dividido en bases de distintos equipos o neutrales. Estas bases consisten en un rango de acción circular protegido por torretas en su perímetro. Las bases sirven a su equipo como punto de reaparición para las tropas, sin embargo, están limitadas a regenerar una tropa cada cierto tiempo. De esta forma, si el equipo pierde muchos drones en poco tiempo, y no tiene las suficientes bases para asumir esa pérdida, no podrá regenerar sus tropas rápidamente y aumentarán sus probabilidades de perder la batalla. El comportamiento de las bases para regenerar enemigos está definido como la siguiente máquina de estados:



Para que un equipo pueda conquistar una base enemiga, primero debe neutralizarla. Para esto, sus tropas deberán pasar un tiempo dentro de la base enemiga siendo mayoría respecto a sus adversarios. Una vez la base es neutralizada, del mismo modo estas tropas deberán resistir en mayoría un tiempo dentro de la base para conquistarla.

El comportamiento de las bases para gestionar el equipo al que pertenecen está definido como un árbol de comportamiento de la siguiente forma:



Las bases tendrán tres atributos modificables:

* Tiempo de neutralización
* Tiempo de conquista
* Tiempo de reaparición de enemigos

# **Resultados**

# **Reparto de tareas**

La tarea de diseño de juego se ha realizado en grupo. El resto de las tareas han sido repartidas entre los tres integrantes del grupo, quedando los roles de la siguiente manera:

* Pedro Casas Martínez: Modelador 3D, programador.
* Juan Manuel Carretero Ávila: Técnico de música y sonido, programador.
* Alejandra Casado Ceballos: Gestora de proyecto, programadora.

Por la parte de modelos 3D, el artista ha realizado los modelos de los agentes y los elementos del escenario para la realización de este proyecto. En cuanto a la música y el sonido, el encargado de ello ha buscado los recursos en internet y los ha editado de forma que se adecuase a los requisitos del proyecto diseñado. Por último, la gestora del proyecto ha redactado la memoria de este y se ha asegurado de que las tareas del resto de miembros fuesen ejecutadas correctamente.

En cuanto a la programación, la división ha sido la siguiente:

* Juan Manuel Carretero Ávila:
  + Creación de la estructura genérica de las máquinas de estados.
  + Programación de la máquina de estados de la generación de enemigos por el mundo.
* Alejandra Casado Ceballos:
  + Creación de la estructura genérica del sistema de utilidad.
  + Programación de la máquina de estados del reclutador.
  + Programación del sistema de utilidad de los drones.
  + Programación de la máquina de estados de “Ir al punto de espera del reclutador” de los drones.
  + Programación de los sonidos en el juego.
* Pedro Casas Martínez:
  + Creación de la estructura genérica del árbol de comportamientos.
  + Programación del árbol de comportamientos del mundo.
  + Programación de la máquina de estados de las torretas.
  + Programación de la máquina de estados de “Conquistar o defender base” de los drones.
  + Programación de la máquina de estados de “Atacar enemigo más cercano dron a distancia” de los drones.
  + Programación de la máquina de estados de “Atacar enemigo más cercano dron a cuerpo a cuerpo” de los drones.

# **Webgrafía**

* Brackeys. (14/03/2018). UNITY NAVMESH TUTORIAL – BASICS. Recuperado el 1 de enero de 2021 en el siguiente enlace <https://www.youtube.com/watch?v=CHV1ymlw-P8>
* Garre, C. (fecha de publicación desconocida). TEMA 2 PARTE 1 ARQUITECTURA DE AGENTES – MÁQUINAS DE ESTADOS FINITAS (FSM). Recuperado el 30 de diciembre de 2020 del aula virtual de la Universidad Rey Juan Carlos.
* Garre, C. (fecha de publicación desconocida). TEMA 2 PARTE 2 ÁRBOLES DE COMPORTAMIENTO (BT). Recuperado el 30 de diciembre de 2020 del aula virtual de la Universidad Rey Juan Carlos.
* Garre, C. (fecha de publicación desconocida). TEMA 2 PARTE 3 SISTEMAS DE UTILIDAD (US). Recuperado el 30 de diciembre de 2020 del aula virtual de la Universidad Rey Juan Carlos.
* Bytes, P. (30/06/2017). STARFIELD SKYBOX. Recuperado el 03 de enero de 2021 del suiente enlace <https://assetstore.unity.com/packages/2d/textures-materials/sky/starfield-skybox-92717>
* peridactyloptrix