**Comportamiento de Personajes**

**Grupo 1**

Juan Manuel Carretero Ávila

Alejandra Casado Ceballos

Pedro Casas Martínez

**08/01/2021**

**Profesores**

Dan Casas Guix

Carlos Garre del Olmo

**Grado en diseño y desarrollo de videojuegos**



**GAME OF DRONES**

Especificación de proyecto

**Índice**

[1. Descripción general 1](#_Toc60744076)

[2. Especificación de los personajes 2](#_Toc60744077)

[Drones 2](#_Toc60744078)

[Drones de ataque cuerpo a cuerpo y a distancia 2](#_Toc60744079)

[Reclutador 9](#_Toc60744080)

[Torreta 12](#_Toc60744081)

[3. Interacción con el entorno/mundo 14](#_Toc60744082)

[4. Resultados 16](#_Toc60744083)

[5. Reparto de tareas 17](#_Toc60744084)

[6. Webgrafía 19](#_Toc60744085)

**Índice de figuras**

[Figura 1. Sistema de utilidad de dron 5](#_Toc60744036)

[Figura 2. Gráficas funciones de utilidad 6](#_Toc60744037)

[Figura 3. Máquina de estados de dron ir al punto de espera 6](#_Toc60744038)

[Figura 4. Máquina de estados de dron conquistar o defender una base 6](#_Toc60744039)

[Figura 5. Máquina de estados ataque de dron a distancia 7](#_Toc60744040)

[Figura 6. Máquina de estados ataque de dron cuerpo a cuerpo 7](#_Toc60744041)

[Figura 7. Máquina de estados recluta 10](#_Toc60744042)

[Figura 8. Máquina de estados torreta 13](#_Toc60744043)

[Figura 9. Máquina de estados generación agentes 14](#_Toc60744044)

[Figura 10. Árbol de comportamientos conquista de base 15](#_Toc60744045)

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. Esta simulación es escalable, por lo que se puede configurar el numero de equipos, cambiar el escenario, la cantidad de agentes y saldrán resultados diferentes. En nuestro caso, se ha decidido crear un total de 3 simulaciones distintas para ver cómo se adapta perfectamente.

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 agentes 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**

Tanto los drones de ataque a distancia, como los de ataque cuerpo a cuerpo como los reclutadores, tienen una clase padre de la que heredan.

Esta clase tiene unos atributos modificables que afectan a los drones que heredan de esta:

* Vida máxima: vida que tiene el dron correspondiente.
* Velocidad de movimiento: velocidad del dron correspondiente.

## **Drones de ataque cuerpo a cuerpo y a distancia**

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 y seguir las órdenes del reclutador, 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.

El estado inicial por defecto es el de aproximamiento de la FSM de ataque. Este estado se ha programado de forma que, en un principio, los drones se mueven de forma aleatoria, haciendo que todas las simulaciones sean diferentes, aunque los parámetros no cambien.

* 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 cierta vida | 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:

Diagrama

Descripción generada automáticamente

Figura 1. Sistema de utilidad de dron

La primera gráfica refleja la función de utilidad de la rama de Atacar al enemigo más cercano, la segunda las ramas de Ir al punto de espera del reclutador y Conquistar base enemiga, y la tercera la de Defender base aliada.

En el caso de Atacar al enemigo más cercano, se desea que funcione de forma que cuando la percepción tiene un valor menor, es decir, el enemigo está muy cerca, su valor sea muy alto y que este valor decrezca poco a poco al alejarse de un enemigo.

Respecto al de Ir al punto de espera del reclutador y Conquistar base enemiga, las percepciones pueden tener valor de 0 en caso de que no haya recibido orden de ir al punto de espera o no deba conquistar una base enemiga, o 1 en caso de que estas órdenes existan. Siendo así, sus funciones de utilidad multiplican estos valores de 0 o 1 por la influencia de estas acciones, de forma que en caso de que la orden sea positiva, el valor de utilidad irá determinado por la influencia.

Por último, en el de Defender base aliada, se pretende que, cuando una base muy cercana al dron está siendo atacada este acuda, teniendo un valor de utilidad muy cercano a 1, pero que sin embargo, al estar muy lejos el valor sea muy cercano a 0.

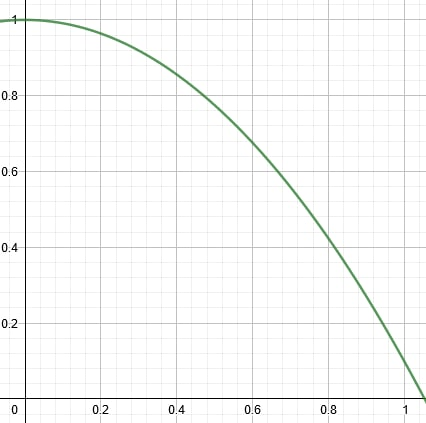
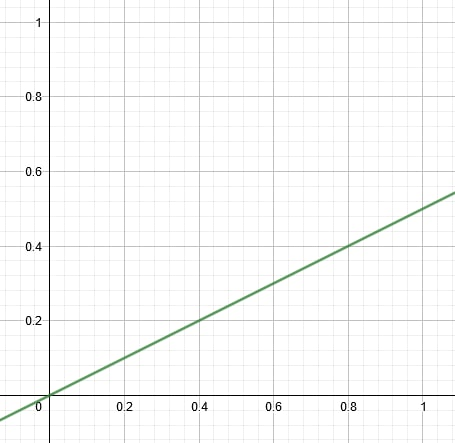
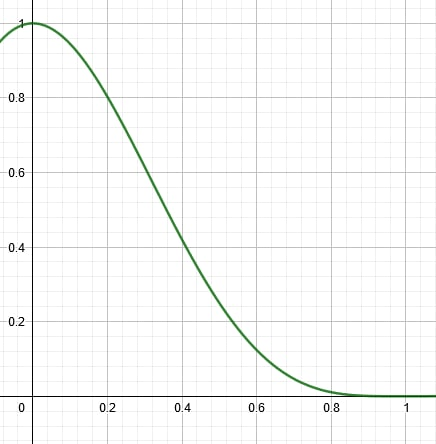
  

Figura 2. Gráficas funciones de utilidad

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

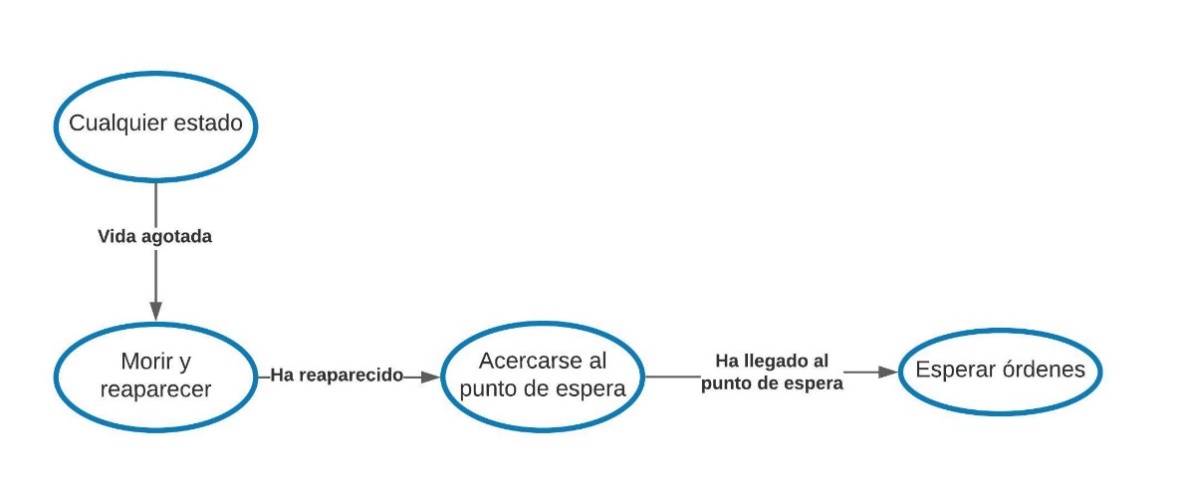


Figura 3. Máquina de estados de dron ir al punto de espera

FSM Nivel 2 (Conquistar o defender base):

Diagrama

Descripción generada automáticamente

Figura 4. Máquina de estados de dron conquistar o defender una base

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

Diagrama

Descripción generada automáticamente

Figura 5. Máquina de estados ataque de dron a distancia

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



Figura 6. Máquina de estados ataque de dron cuerpo a cuerpo

1. **Estructuras de datos**

El dron tiene cuatro atributos modificables generales:

* Rango de detección de enemigos
* Rango de protección
* Daño 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.

Además, tiene algunos propios de cada máquina de estados de segundo nivel, que son:

* Respecto a la FSM de ataque a distancia:
  + Cadencia de tiro
  + Velocidad de apuntado
  + Ángulo máximo de tiro
  + Cantidad de movimiento aleatorio al inicio del ataque
* Respecto a la FSM de ataque cuerpo a cuerpo:
  + Rango de ataque
  + Velocidad de ataque
  + Desplazamiento en idle
  + Velocidad de giro

## **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

Figura 7. Máquina de estados recluta

1. **Estructuras de datos**

El reclutador tendrá cinco atributos modificables:

* Número máximo de aliados
* Rango del punto de espera: distancia máxima respecto a su posición para detectar otros drones están esperando con él.
* Rango de reclutamiento: distancia máxima a la que puede reclutar a un dron
* Porcentaje mínimo de aliados en punto de espera para iniciar un ataque
* Tiempo límite de espera en el punto de espera a los aliados
* Distancia de seguridad: Distancia respecto a la distancia establecida de espera a las bases, la distancia predeterminada viene dada por la distancia de la base a la torreta mas el rango de ataque de la torreta, pretendiendo que el reclutador no pueda ser disparado mientras espera.

**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

**Figura 8. Máquina de estados torreta**

1. **Estructuras de datos**

La torreta tendrá cuatro atributos modificables:

* Velocidad de apuntado
* Cadencia de tiro
* Rango de ataque
* Ángulo de ataque
* Ángulo de disparo
* 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:

Diagrama

Descripción generada automáticamente

Figura 9. Máquina de estados generación agentes

El generador de agentes tendrá tres atributos modificables:

* Tiempo de reaparición de agentes

Además, existe un Team Manager que tiene los siguientes atributos modificables:

* Número de drones cuerpo a cuerpo por equipo
* Número de drones a distancia por equipo
* Número de drones reclutadores por equipo
* Número de equipos
* Colores de los equipos

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:

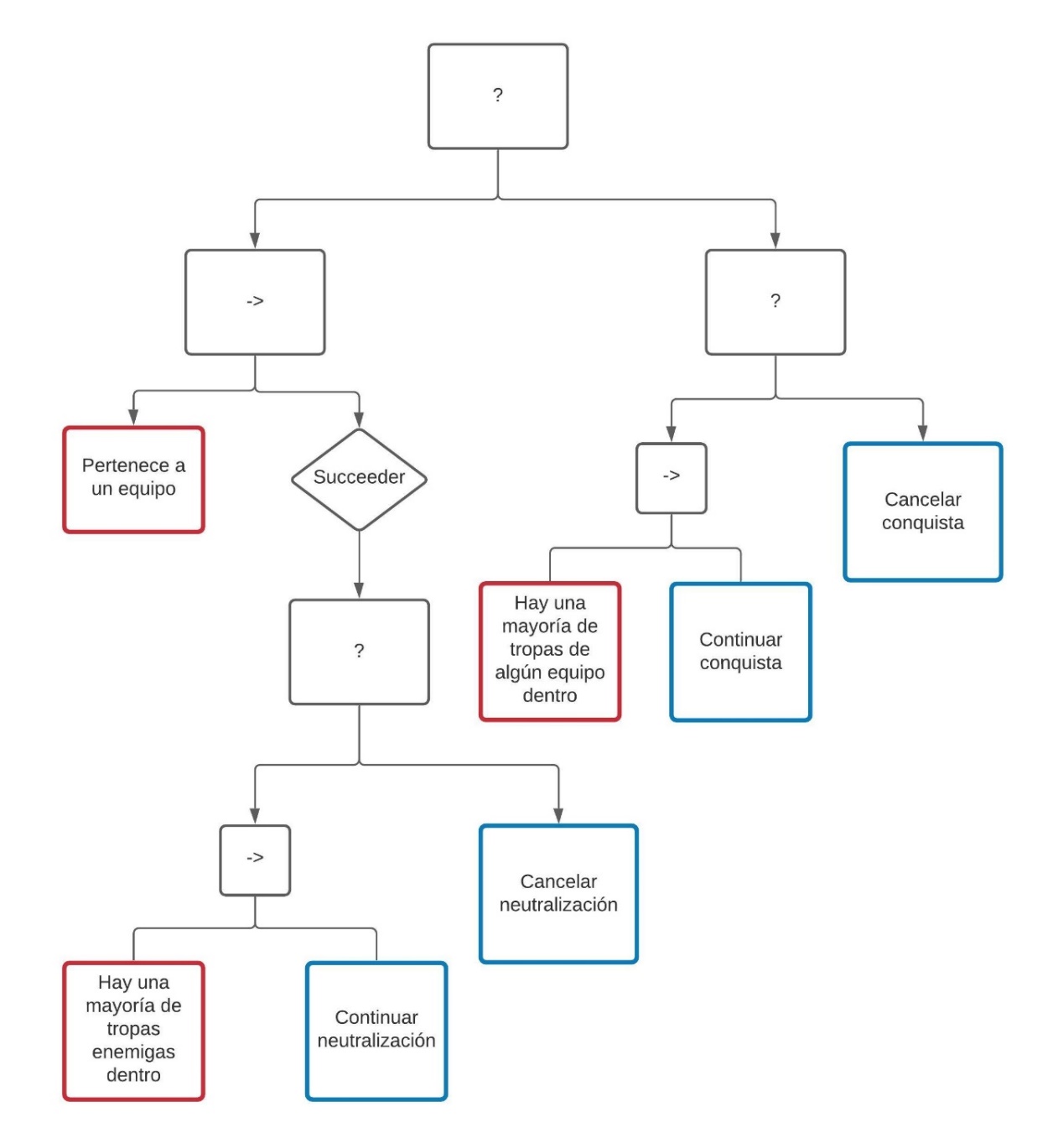


Figura 10. Árbol de comportamientos conquista de base

Las bases tendrán tres atributos modificables:

* Tiempo de neutralización
* Tiempo de conquista
* Equipo al que pertenece

# **Resultados**

Aquí podéis mostrar de manera libre e informal el resultado obtenido, mediante capturas de pantalla, explicando algunas situaciones especialmente interesantes, etc. Es importante que nos “vendáis” lo bueno que es vuestro proyecto resaltando los puntos fuertes, aunque también se valorará la sinceridad y la autocrítica mostrando los posibles fallos o puntos más flojos que hayan podido quedar pendientes.

# **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 agentes.
  + Programación de los sonidos en el juego.
* 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.
* 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. (09/01/2014). SOUND OF AN ANTI-GRAVITY HOVER DEVICE STARTING UP. Recuperado el 31 de diciembre de 2020 de <https://freesound.org/people/peridactyloptrix/sounds/213384/>
* DayCraftMC. (18/02/2016). LARGE LASER BEAM. Recuperado el 1 de enero de 2020 de <https://freesound.org/people/DayCraftMC/sounds/337112/>
* JohnBuhr. (30/10/2015). SWORD CLASHES. Recuperado el 1 de enero del siguiente enlace <https://freesound.org/people/JohnBuhr/sounds/326800/>
* beskhu. (28/03/2013). SIMPLESONG (MAC) CONSONANTS. Recuperado el 31 de diciembre del siguiente enlace <https://freesound.org/people/beskhu/sounds/182153/>
* HenryRichard. (18/11/2018). NES SOUND EFFECTS ROUND 1. Recuperado el 31 de diciembre del siguiente enlace <https://freesound.org/people/HenryRichard/sounds/448266/>
* josepharaoh99. (15/11/2016). ENGINE DYING. Recuperado el 31 de diciembre del siguiente enlace <https://freesound.org/people/josepharaoh99/sounds/368512/>
* Sergenious. (23/06/2008). LASER2. Recuperado el 1 de enero del siguiente enlace <https://freesound.org/people/Sergenious/sounds/55836/>