Proyecto
"Iki"

Grupo
"IKIGAI"

"DOCUMENTO DE ESPECIFICACIÓN"

Hito: 0
Fecha entrega: 5-10-2016
Versión: 2

Componentes:
- Juan Sánchez Almendro
- Pablo Campos Albert
- Álvaro Mas Menéndez
- Mateo Bernal Montoya
- Adrián González Herrera

Contenido

1. Introducción

1.1. Propósito

En este documento analizaremos y comentaremos las especificaciones para el proyecto de Creación y Entretenimiento digital: Iki, un videojuego para un jugador de sigilo/acción.

El documento servirá fundamentalmente para el equipo de trabajo, para definir y aclarar el funcionamiento y las mecánicas del juego. A su vez, permitirá a los clientes conocer en mayor profundidad nuestro juego.

2. Descripción general

2.1. Ámbito del sistema (Contexto e historia).

Iki es un videojuego para un jugador enmarcado en el género de sigilo/acción. Se trata de un tercera persona, con cámara aérea. Nuestras referencias más claras son:

- Metal Gear de la MSX.
- Monaco what's yours is mine.

Especialmente el primero; nuestras mecánicas serán muy parecidas al igual que el desarrollo general del juego. No obstante, en nuestro caso estaría más enfocado a exteriores y a escapar en vez de infiltrarse.

Uno de nuestros puntos distintivos es quizás la historia:

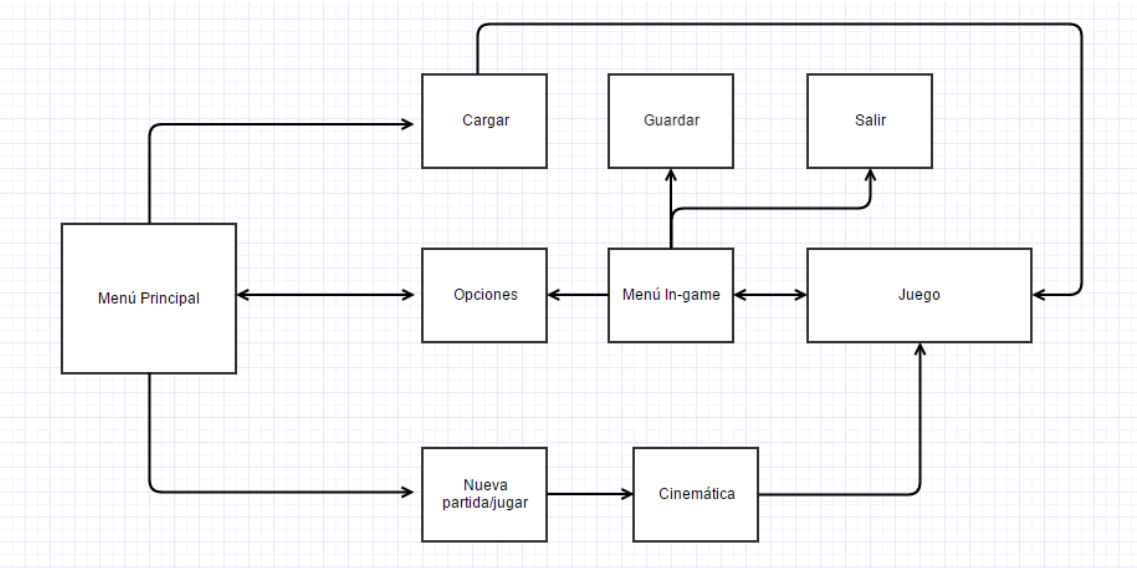
En una época futurista, los robots forman parte de la vida cotidiana en la tierra y son la principal fuerza laboral de la especie humana. Los constantes avances tecnológicos provocan que ciertos modelos queden obsoletos a medida que se mejoran cada vez más las características de los robots. Actualmente, los robots “SG” (de segunda generación) están desbancando cada vez más a los “PG” (primera generación), hasta el punto que se desprecia y persigue a todos aquellos robots obsoletos, abandonados por el hombre.

Comenzamos el videojuego con nuestro protagonista despertándose en un desguace rodeado de chatarra y de robots de primera generación, se da cuenta rápidamente de que corre peligro: Es un robot y lo quieren reciclar. Debe salir de allí cuanto antes aunque en realidad no recuerda nada; no sabe “qué” o “quién” es exactamente, ni cómo ha llegado allí, ni por qué están destruyendo a los robots antiguos.

A lo largo del juego, irá descubriendo pistas sobre su identidad y sobre la situación en la que se encuentra el mundo mientras escapa del desguace. De esta manera le damos un toque misterioso, casi de thriller. Nuestro personaje pese a ser un robot parece tener pensamiento humano, lo que le permite superar infinidad de obstáculos que de otra manera no podría. Pero: ¿Porque tiene un pensamiento humano? Esperamos que esta pregunta mantenga atento a nuestro jugador.

Además de hacer nuestras las mecánicas de Metal Gear manteniendo una jugabilidad sólida y sencilla pero modificándose ligeramente, aparte del cambio obvio en la ambientación, temática e historia.

2.2. Funcionalidades generales.



Este sería el diagrama de flujo básico de nuestro juego. Habría que detallar más ciertas partes como la de opciones (ya que se sub dividirá probablemente en "controles", "gráficos", "sonido" etc) entre otras pero por el momento, consideramos que es suficiente.

2.3. Características de los personajes.

Protagonista. Es un robot humanoide con una apariencia endeble, está oxidado y no posee ningún tipo de arma (al inicio). Su rol es atravesar el desguace sorteando a los guardias en la medida de lo posible. Tiene dos posibilidades de movimiento: andar rápido (velocidad normal) y andar lento (sigiloso). Por supuesto cada una será útil según qué situaciones. A parte de esto el protagonista puede recoger y usar objetos que encontrará conforme avance en el juego (se definen en el punto 3.1.1.2), podrá almacenarlos en un "slot" en el que solo cabrá uno al mismo tiempo; Tendrá también un ataque a corta distancia (también definido en el punto 3.1.1.1) y por supuesto, una cantidad de vida o HP (hit points).

Este sería un prototipo de la interfaz.

Vida a la izquierda

Rayo paralizador y slot de objeto a la derecha

Guarda de asalto. Robot con apariencia humanoide, menos endeble que el anterior; es un modelo nuevo y se nota en el brillo de su coraza. No es demasiado tosco pero harán falta varios golpes para noquearlo. Su rol, vigilar que no pasa nada extraño en el desguace, y para poder cumplirlo tiene un abanico de posibilidades: puede patrullar por zonas amplias o zonas más concretas, o bien quedarse quieto en una posición clave esperando para asaltar a cualquier sospechoso. Si detecta que pasa algo raro irá a revisarlo y si alcanza al protagonista lo atacará desde una distancia corta (todo definido en detalle en el apartado de mecánicas). Tanto este como el resto de enemigos poseen una barra de vida, si llega a cero porque el jugador ataca, el robot será destruido.

Además, los guardias de asalto tienen una característica adicional: la batería.

La batería afecta a las mecánicas de atacar y correr del guardia de asalto. Es una barra que se vacía y se recupera según las acciones del guardia. Determina la distancia que puede correr seguido y/o la cantidad de ataques y la potencia de los mismos.

Robot médico. Modelo femenino. Su rol es ayudar al resto de robots, usando sus distintas habilidades. Es un robot tímido y si te ve huirá de ti y avisará a los guardias. Recuperará a los enemigos que han sido dañados por el jugador.

Dron de reconocimiento. Tiene un aspecto bastante simple a diferencia de los otros dos, y este va flotando por el escenario a un metro de altura, más o menos. Su única función es moverse por el escenario y dar la alarma al detectar a un sospechoso para llamar a una patrulla de guardias que acaben con la amenaza. Este enemigo no puede detectar el ruido.

Torreta defensiva. Entidad sin IA, aspecto de torreta. Su función es disparar (láser) a todo lo que se cruce en su ángulo de visión. Antes de comenzar a disparar escaneará a la entidad rápidamente para asegurarse de que no ataca a un aliado.

Todos los enemigos tienen una característica importante: **el grado de sospecha**.

El grado de sospecha utiliza lógica difusa y tiene consecuencias en el comportamiento y las mecánicas de los enemigos. Evoluciona individualmente en cada enemigo en función de la mecánica escanear/escuchar definida más adelante. Puede cambiar de manera semi-permanente o no dependiendo de la gravedad del acto:

- Generalmente, si el acto es leve o no alcanza digamos el 40% del total posible de sospecha, el enemigo olvidará rápidamente lo ocurrido, regresando a un 0%.
- En cambio, si su grado de sospecha sobrepasó el 40%, tardará más en reducirse, aumentando exponencialmente cuanto más nos acerquemos al 100%. Ver cómo eliminamos a un compañero, combatir contra nosotros o descubrir una modificación importante en el mapa o un “cadáver” de un compañero son eventos que tendrán un impacto del 100% por ejemplo.

Lo desarrollaremos más en el apartado de mecánicas.

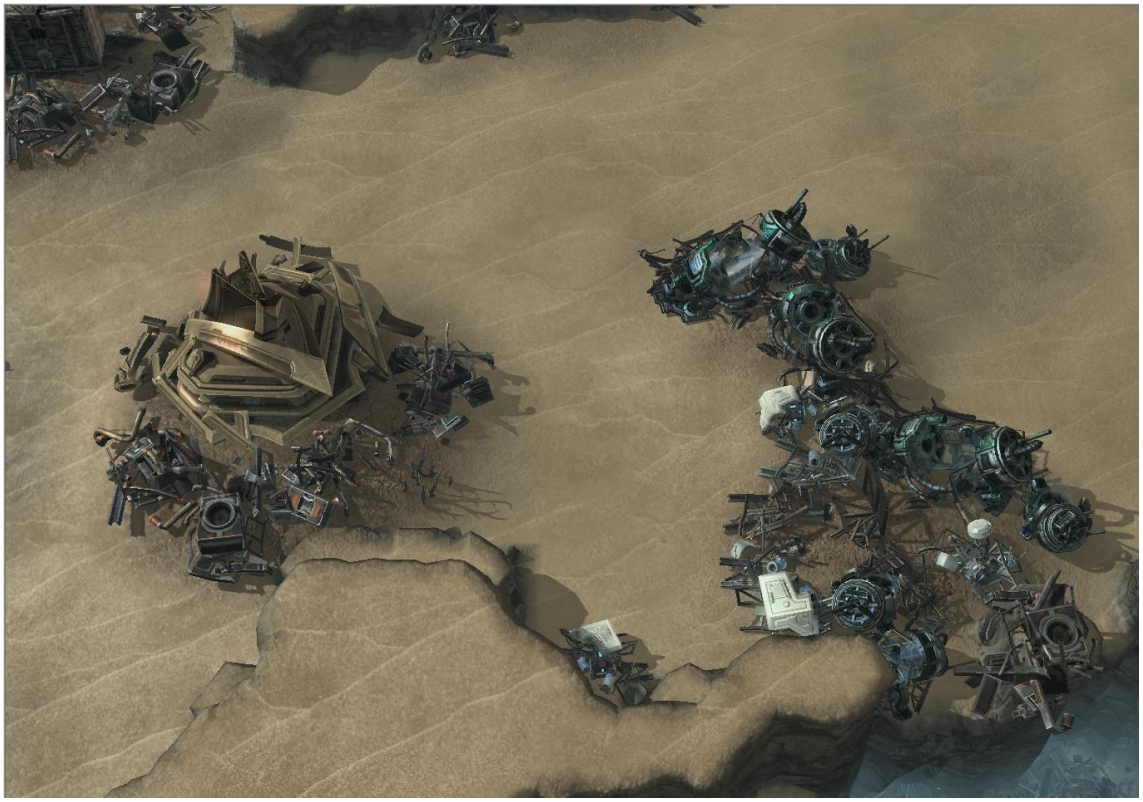
2.4. Escenarios.

El mundo donde se desarrollará la historia de Iki será una especie de desguace de robots. Comenzaremos en una pequeña sala cerrada, en la cual tan solo habrá deshechos, poca iluminación y una puerta hacia el exterior. En un principio los escenarios serán principalmente en el exterior, en él desguace. No obstante, habrá estructuras de diferentes tamaños en las que podremos entrar y que serán fundamentales para el juego. Estas estructuras no llevarán a otro tipo de escenario, el de interior, con diferentes texturas y ambientes.

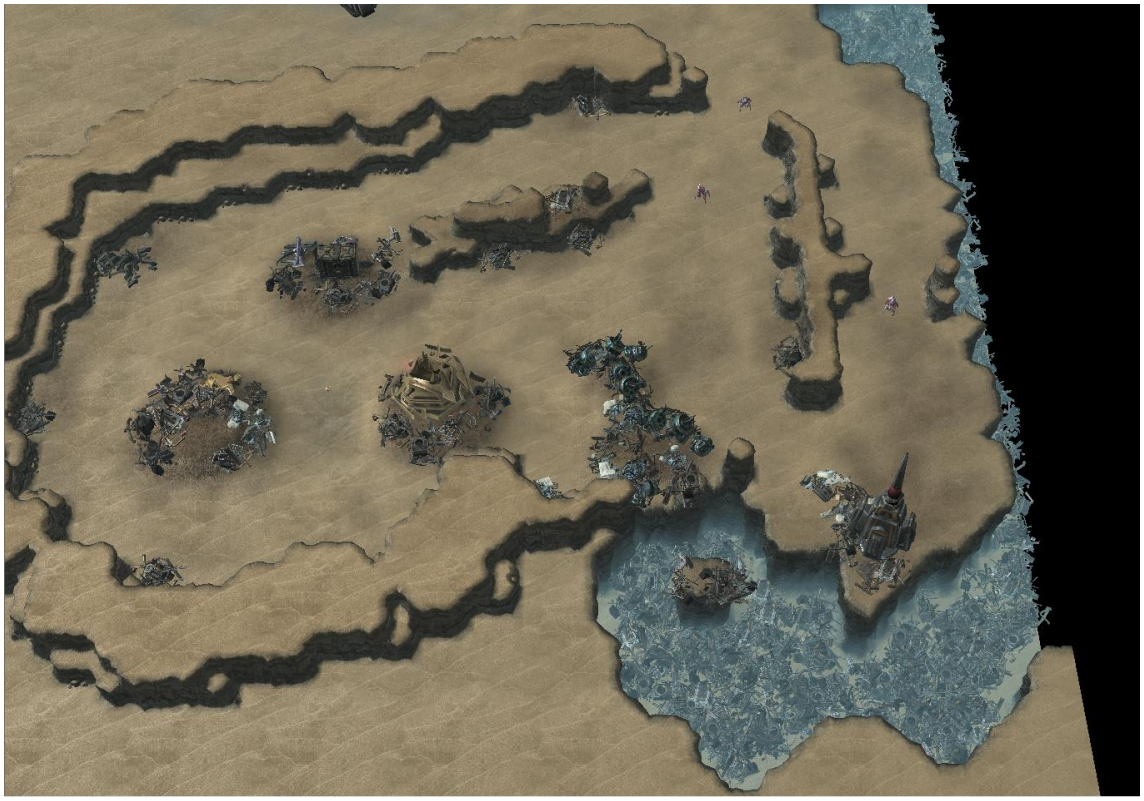
La cámara será de seguimiento, en tercera persona, no habrá “niebla de guerra”, podremos ver todo los que nos rodea con claridad.

Nos hemos tomado la libertad de prototipar algún que otro escenario de ejemplo en el editor de mapas de “Starcraft 2”, aunque no hemos querido abusar de ello, en especial para el apartado de mecánicas ya que el aspecto visual distraería de lo importante.

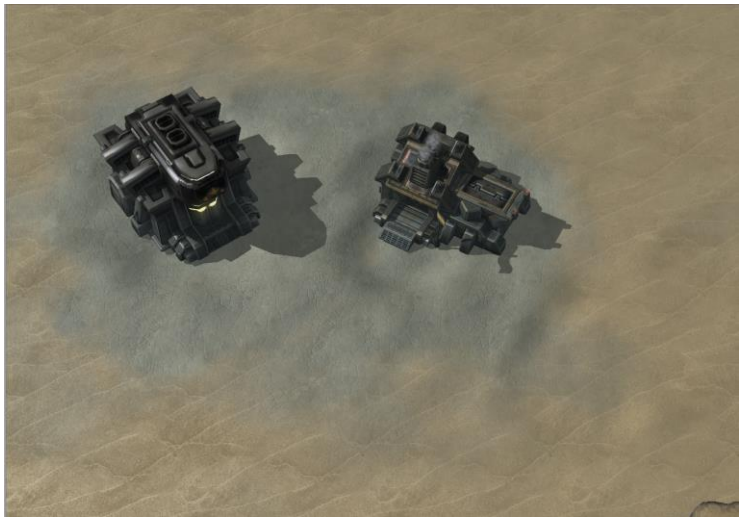
Este sería un ejemplo de parcela del desguace:



Los restos, la chatarra, las elevaciones y los acantilados nos marcarán los caminos a la vez que nos permitirán ocultarnos de los enemigos.



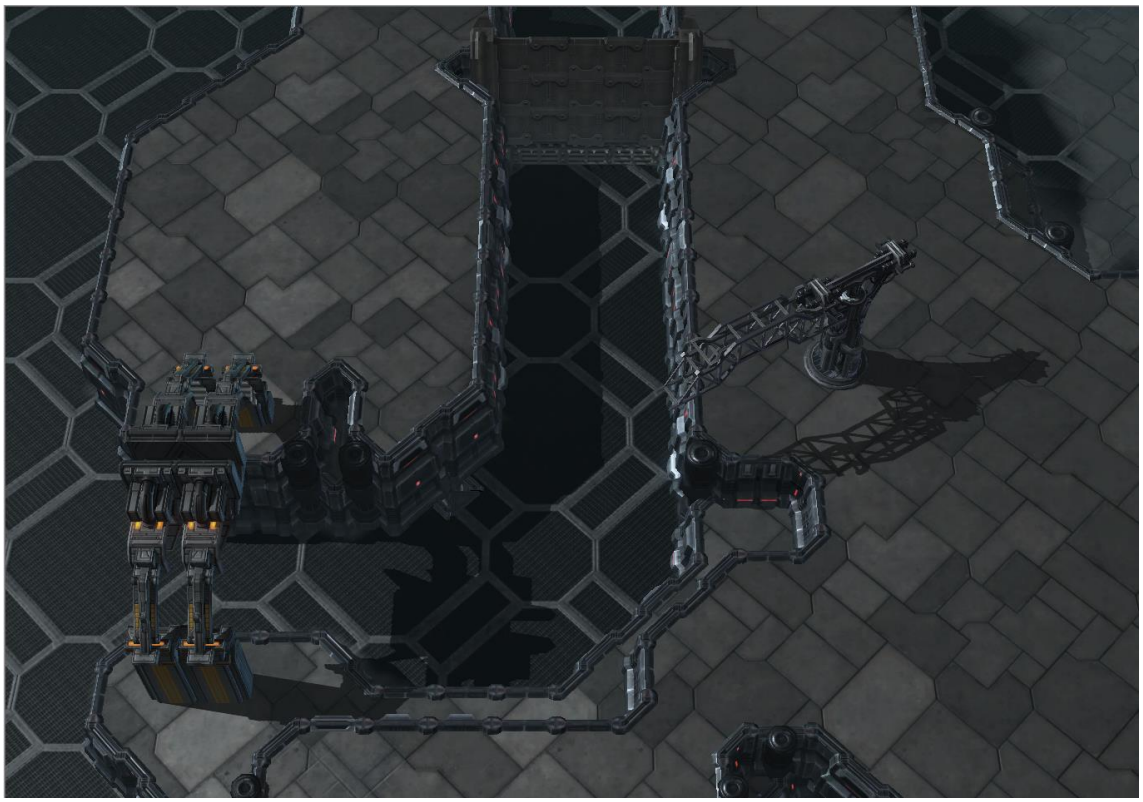
Esto sería una vista de ejemplo de parcela completa vista desde arriba. En principio el juego se dividirá con este tipo de parcelas. Serán como habitaciones pero en exterior. Daremos la sensación de que es un escenario continuo en todo momento pero cada parcela será una unidad semi-individual en la acción, tendrá sus propios puzzles y los enemigos en principio no cambiarán de parcela aunque si se da el caso, pueden hacerlo.



Estas, serían algunas estructuras pequeñas que nos encontraremos durante el juego. Una se trata de una "fábrica de robots", definida en el 3.1.1.2. La otra podría ser decorativa o podría contener una mini sala en la que habría algún ítem, palanca o evento de importancia.

Las estructuras más grandes sí que contendrán más complejidad en su interior (una o varias parcelas). Recordamos que el juego se desarrollará principalmente en exterior pero podrá haber tramos del juego más o menos largos que se desarrollen en interior, sobre todo de cara al final del juego.

(Entramos)



Ejemplo de interior: Espacios más cerrados, más oscuros, mecánicas ligeramente diferentes.

En principio esos 2 tipos de escenario serían los únicos realmente distinguibles en nuestro juego. Pretendemos que en el resto no haya una separación brusca, quizás el decorado cambie un poco a medida que avanzamos en el juego: zonas del desguace con diferentes tipos o cantidades de chatarra, con vegetación cambiante, acantilados cambiantes pero en lo fundamental similares a lo largo del juego.

Encontraremos elementos con los que podremos interactuar, como puertas, palancas...Lo definiremos en el apartado 3.1.1.2.

La estimación de tiempo/niveles que hemos pensado es de unos 5 minutos por nivel/parcela/sector y un total de 20 niveles, aproximadamente, para hacer un total de unos 100 minutos de juego aproximados.

2.5. Requisitos (suposiciones y dependencias).

Algunos factores deberán cambiar si decidimos añadir nuevas funciones al juego o funciones definidas como opcionales en el apartado 3. El juego podría cambiar su jugabilidad drásticamente dependiendo de lo que cambiemos, por ejemplo una mecánica de disparos supondría tener que pensar una serie de colisiones y físicas que de otra manera no existirían. A su vez, dependiendo del tiempo, es posible que algunas mecánicas que no aparecían como opcionales se queden atrás también. Intentaremos que el "core" del juego sea correcto pero muchos factores estéticos o secundarios tendrán una prioridad menor.

Por no mencionar que las restricciones del punto 2.6 también pueden afectar al resultado final.

Lo indicamos en el punto 2.7, pero adaptar el juego a diferentes S.O en un futuro sería fundamental.

2.6. Restricciones.

Como restricciones tenemos que desarrollaremos en Windows para Windows, y utilizaremos C++ como lenguaje de programación. Se utilizará un control de versiones git, haciendo uso de GitHub. Todavía no se ha decidido el entorno de trabajo, aunque probablemente usemos Code::Blocks o NETBeans.

Los documentos de texto y las hojas de cálculo serán editados en línea a través de Google Drive, donde se puede trabajar de manera simultánea.

En cuanto al juego, se jugará con mando o teclado y tendrá una resolución mínima será de 1280x720.

Además, debemos tener en cuenta que el juego deberá poder funcionar correctamente en los ordenadores de la universidad, que tienen las siguientes características básicas:

- Windows 7 (64 bits)
- RAM: 8 GB.
- CPU: 2.90 Ghz.
- Nvidia GTX 480.

Aunque lo más seguro es que no represente un problema.

2.7. Requisitos futuros.

Como hemos explicado en el punto 2.5 y en algunas ocasiones en el documento, algunas partes opcionales y/o adicionales podrían cambiar notablemente la experiencia de juego. Además, una vez el producto acabado, sería posible añadir "expansiones" al juego, ya sea más niveles, objetos, enemigos… Aparte de la posibilidad de plantearse adaptar el juego a otras plataformas (S.O), principalmente móviles.

3. Requerimientos específicos

3.1. Requerimientos funcionales.

3.1.1. Mecánicas.

3.1.1.1. De los jugadores

- **Movimiento.** Nos podemos mover en cualquier dirección haciendo click sobre el terreno (digital), colisionaremos con todas las entidades físicas correspondientes.
- **Movimiento sigiloso.** Manteniendo la tecla SHIFT, nos moveremos igual pero haciendo menos ruido y pasando más inadvertido.
- **Atacar**. Podremos atacar en cualquier dirección a lo que haya en frente nuestra. Si estamos a la distancia apropiada, dañaremos al enemigo. También podemos usar el ataque para llamar la atención (ya que hará ruido). Los ataques por la espalda y por los lados dañan más.
- **Acción.** Cuando estemos cerca de algún elemento del escenario con el podamos interactuar, podremos activarlo pulsando una tecla. (puertas, palancas y objetos).
- **Activar objeto.** Activaremos el efecto del objeto guardado en ese momento en el slot
- **Rayo paralizador.** Lo tendremos disponible sobre la mitad del juego. Lanzará un rayo de corta-media distancia delante nuestra que paralizará momentáneamente a un enemigo, se le podrá dañar normalmente. Ese enemigo también perderá "la memoria", es decir, si salimos de su campo de visión, no se acordará de haber sido agredido, se comportará normalmente y su grado de sospecha bajará a cero. El rayo se recargará con el tiempo.

3.1.1.2. De objetos y NPCs

--Objetos--

Los objetos que nos vamos encontrando se almacenarán en nuestro slot y nos proporcionarán mejoras permanentes o temporales al usarlos

Lista de objetos:

- Aceite: Proporciona una mejora temporal de velocidad además de que hacemos menos ruido.
- Engranaje: Restauran una cantidad de vida fija.
- Tarjeta: Permitirá abrir un tipo de puertas. Solo se puede conseguir si destruyes un guardia que la lleve.

- ❖ Rayo paralizador: Sobre la mitad del juego recogeremos una mejora permanente que nos aportará una mecánica nueva (descrita en el 3.1.1.1.)

--NPCs –

Recordemos que los enemigos tienen unas características importantes (grado de sospecha y batería) que fueron definidos en el 2.3.

Mecánicas comunes a todos los enemigos:

Vigilar:

- ❖ Ciertos enemigos permanecerán en este modo por defecto, para potenciar alguna característica del escenario.
- ❖ Cuando están en modo estático la mecánica escanear/escuchar es más precisa.
- ❖ Podrán rotar su visión en función del escenario y de sus compañeros (no mirarán puntos inútiles).
- ❖ Los enemigos que patrullan pueden cambiar a este modo si:

    - Van a compartir ruta con otro compañero pero se encuentran demasiado juntos/solo cabe uno por el pasillo, esperarán vigilando hasta que la distancia sea adecuada.
    - Pueden también cambiar de forma arbitraria en zonas en las que tenga sentido; generalmente zonas abiertas en las que pueda aprovechar su mejora de percepción y siempre y cuando lleven ya un mínimo de distancia patrullada (para que no puedan pararse cada 2 por 3).
    - Si ven el “cadáver” de un compañero o alguna modificación sustancial del escenario se colocarán al lado del evento en cuestión y entrarán en modo vigilar rotando 360 grados rápidamente, entrarán después en modo inspeccionar (profundo), usando ese punto como centro.

- ❖ Tardarán un tiempo de entre 5 y 11 segundos en salir del estado de vigilancia (si no era su estado natural permanente), que decidirán teniendo en cuenta que deben aproximarse a la media (8 segundos) en el total de sus paradas. Además tendrán en cuenta la posición de sus compañeros.
- ❖ El grado de sospecha afecta de la siguiente manera a esta mecánica:

    - Si es leve (es decir si estamos a mucha distancia, hacemos muy poco ruido o nos ve durante muy poco tiempo), el enemigo simplemente rotará hacia esa dirección y se mantendrá durante unos segundos, si no ocurre nada más, volverá a su rotación habitual.
    - Cualquier otro grado de sospecha hará que el enemigo abandone el modo vigilar (ya sea permanente o no) y cambie al modo inspeccionar.

Escanear/Escuchar:

- ❖ Esta mecánica está activada de manera permanente en los enemigos.
- ❖ Esta mecánica es la que modificará constantemente el grado de sospecha en función de lo siguiente:

    - La distancia entre el jugador y el enemigo.
    - La gravedad de la acción.
    - El tiempo durante el que nos ven y/u oyen, la persistencia.
    - Modificaciones importantes en el mapa (puertas abiertas etc...) o cadáveres.
    - Comportamiento de los compañeros: si ve a un compañero suyo persiguiéndonos o buscandonos, aunque no nos vea directamente, también aumentará su grado de sospecha. Se aplican las mismas reglas de distancia y persistencia con respecto al compañero.

Patrullar:

- ❖ Los enemigos que no tengan el modo vigilar fijo, patrullarán siguiendo diferentes rutas:

    - Habrá rutas predefinidas básicas en cada parcela de terreno (entre una y tres por enemigo dependiendo del escenario y de la cantidad de enemigos en la parcela), el enemigo empezará aleatoriamente en una de ellas y escogerá al final de cada una dependiendo de la cercanía de las mismas y de la posición de sus compañeros. Las rutas estarán diseñadas de manera que haya puntos de conexión entre ellas.
    - Si alteramos el curso natural de su patrulla, haciéndole por ejemplo tener que inspeccionar un área alejándose de su ruta, en caso de no encontrar nada, escogería la ruta más cercana alrededor de la zona que inspeccionó y empezaría desde allí.
    - 2 robots pueden estar en la misma ruta mientras no estén muy cerca el uno del otro. 3 no.
    - En caso de no poder escoger la ruta más cercana por el motivo anterior, escogerían la segunda mejor.
    - En caso extremo de bloqueo, ya sea por una modificación del escenario por parte del protagonista o por colapsamiento de rutas por parte de compañeros, el enemigo entrará en modo vigilancia hasta poder encontrar una ruta, si no pudiese encontrar una, quedaría en modo vigilancia permanentemente.

- ❖ Cualquier modificación en el grado de sospecha hará que el enemigo entre en modo vigilar y a partir de ahí tomará su decisión siguiendo esa mecánica.

Inspeccionar:

- ❖ En cualquier momento, si el grado de sospecha es superior a 5% y tiene una ubicación posible a la que desplazarse, el enemigo inspeccionará la zona en la que haya visto u oído algo.

- La inspección consistirá en desplazarse hasta la zona en cuestión por la ruta más corta y ponerse en modo vigilancia en el epicentro del evento.
- Cuanto más alto sea el grado de sospecha, más rápido se desplazará a la zona (en el caso de los guardias de asalto, utilizarán la mecánica correr). Además, la inspección durará más tiempo una vez haya llegado al punto en cuestión, siendo posible que el robot se desplace por la zona, buscando más en profundidad: buscará hasta haber comprobado visualmente todas las zonas en un radio alrededor del epicentro o hasta que se agote el tiempo y su grado de sospecha baje.
- Si varios enemigos inspeccionan la misma zona simultáneamente, se coordinarán siempre y cuando sea posible: Si hay varias rutas para entrar/salir a la zona de inspección cada uno irá por una distinta. En caso de haber solo una posible (o si las alternativas son muy complicadas o demasiado largas) actuarán de la siguiente manera:
    - Si son 2, uno inspeccionará la sala (modo vigilancia o búsqueda en profundidad) y el otro se quedará en modo vigilancia en la única entrada/salida que haya.
    - Si son 3, harán exactamente lo mismo, y el tercero inspeccionará la sala complementando a su compañero: Si su compañero está en modo vigilancia, utilizará la búsqueda en profundidad y viceversa.

- Más de 3 enemigos no pueden inspeccionar la misma área si solo tiene una entrada, no obstante los robots que hubiesen entrado en modo inspeccionar cancelarán su patrulla y se pondrán en modo vigilar temporalmente, además su grado de sospecha aumentará en consecuencia.
- Se suma un robot posible por cada entrada adicional al área (dependiendo de quien llegue antes, hará parte de los 2 que inspeccionan o cubrirá las entradas en modo vigilar).
- No obstante, no siempre van a corresponder el número de robots máximo con los que realmente vayan. Que haya por lo menos uno inspeccionando es prioritario, a partir de ahí, cubrir entradas es más importante (aunque faltará siempre alguna).
- Cuando varios tipos de enemigo coinciden inspeccionando un área se establece el siguiente orden de prioridades:

    - Los robots médico tienen la máxima prioridad para inspeccionar.
    - Los drones inspeccionarán si no hay médicos suficientes.
    - Los guardias de asalto no inspeccionarán si hay otras opciones.

- Mientras están inspeccionando, el resto de mecánicas siguen presentes, siguen escuchando/viendo, pueden rotar en el modo vigilancia incluso pueden volver a entrar en el modo inspeccionar: Si mientras están inspeccionando (ya sea cubriendo entradas o dentro), vuelven a escuchar o a ver algo brevemente, re inspeccionarán tomando como centro ese nuevo punto. Tened en cuenta que el grado de sospecha ya será alto en esas situaciones y aumentará más si ocurre esto, los enemigos se moverán con rapidez y sus sentidos estarán más alerta, será complicado escapar. Los que están cubriendo las entradas permanecerán ahí, aunque su grado de sospecha aumentará.
- Cuando el grado de sospecha es ya demasiado alto (nos han visto claramente, han oído claramente muy cerca suyo un sonido de combate o de destrucción, oyen la alarma de un dron, les avisa un médico...), el enemigo abandonará todas sus tareas y pasará a uno de estos modos dependiendo del tipo:
    - Guardia de Asalto: Persecución/Ataque.

- Dron de reconocimiento: Alarma.
- Robot médico: Huir/Avisar.

Mecánicas específicas:

--Guardia de asalto--

Perseguir:

- ❖ Una vez nos han detectado, el guardia de asalto nos perseguirá haciendo uso de la mecánica "correr "cuando crea conveniente (lo explicamos después). Nos marcará como un punto que podrá seguir calculando el camino más corto.
- ❖ Si varios guardias nos persiguen simultáneamente, se coordinarán de una manera similar a la de inspeccionar:
    - Cogerán rutas alternativas siempre que sean relativamente óptimas (no estén demasiado lejos o representen un desvío absurdo). Intentarán cortarnos el paso colocándose delante nuestra si es posible.
    - Como prioridad siempre habrá un enemigo detrás de nosotros siguiendo nuestra ruta.
    - En caso de que no haya rutas óptimas disponibles, nos perseguirán en bloque o en fila dependiendo del escenario
    - En todo momento evitarán obstaculizarse entre ellos

- ❖ La persecución de uno o varios robots es algo que hará bastante ruido.
- ❖ Una vez esté a una distancia a la que nos puede alcanzar, utilizará su ataque.

Atacar:

- ❖ Cuando esté a la distancia adecuada, el guardia de asalto cargará su ataque, un cono de rayos. Cuanto más lo cargue más alcance y daño tendrá pero más batería consumirá y más tardará en hacerlo. Recordemos que cuando nos está persiguiendo también hace uso de la batería así que tendrá que gestionar su uso siguiendo los siguientes patrones:

    - El cono de rayos se puede esquivar, el enemigo tendrá en cuenta las posibilidades que tiene el objetivo de esquivarlo según el terreno: Si está en un pasillo estrecho no tendrá miedo a cargarlo ya que no hay muchas opciones para esquivarlo. En cambio durante una persecución complicada en campo abierto o con múltiples opciones, tendrá más sentido cargarlo poco para mermar poco a poco nuestra vida por ejemplo.
    - Si no está solo en la persecución, tendrá en cuenta las acciones de sus compañeros. Si hay uno que está cargándolos poco y que se mantiene cerca del objetivo corriendo, otro quizás se anime más a intentar cargarlos. O quizás en alguna situación, interesaría que ambos lo cargarán desde diferentes ángulos o cuando nos tienen acorralados.
    - Se adaptará a los movimientos del objetivo, si percibe que estás esquivando fácilmente los ataques más poderosos, probará con ataques más cortos. Al contrario, si ve que no te mueves demasiado y que eres presa fácil, insistirá con los largos.

Correr:

- Los guardias de asalto pueden correr, consumiendo batería pero moviéndose más rápido que el jugador.
- Gestionarán su batería de manera similar que con el ataque:

    - Dependiendo del ataque que quieran realizar, correrán más o menos guardando la batería suficiente para realizar el ataque. El ataque tiene prioridad en el orden.
    - Si no están solos en la persecución, tendrán en cuenta las acciones de sus compañeros: Si ya hay un compañero corriendo delante suya con ventaja, tenderá a conservar batería
    - Tendrán en cuenta la distancia con el objetivo.

--Dron de reconocimiento--

***Particularidad*: A diferencia de los demás enemigos, el dron de reconocimiento no puede oír, tan solo escanear (ver).

Alarma:

- Cuando nos haya detectado, el dron de reconocimiento dará la alarma: se transformará ligeramente y emitirá una luz y un sonido particularmente perceptibles para sus compañeros.
- Durante el proceso será invulnerable y se autodestruirá después de terminar la alarma.
- La alarma tiene efectos diferentes según la situación y el nivel en el juego:

    - El total de enemigos alertados por la alarma debe ser de 2 o más en las primeras etapas del juego, de 3 o más en las etapas medias y de 4 o más hacia el final.
    - Si no llega a ese número, caerá una cápsula del cielo conteniendo un guardia de asalto por cada enemigo que falte para llegar a la cifra. Caerá alrededor del dron.
    - Si una alarma alerta a otro dron de reconocimiento, se autodestruirá sin dar una alarma adicional pero caerán 2 cápsulas extra sin importar cuantos enemigos alertó la primera alarma. (recordemos que los drones no pueden oír, así que no será una situación excesivamente común).

- Los enemigos alertados por la alarma obtienen un grado de sospecha muy elevado enseguida. Dependiendo de la cercanía, lo normal será que entren en modo perseguir inmediatamente.
- Los que caen de las cápsulas entran automáticamente en modo perseguir pase lo que pase. Si de alguna manera conseguimos escapar de ellos gracias al escenario, seguirán

las mismas reglas que sus compañeros: elegirán una ruta disponible en la parcela y si no entrarán en modo vigilancia.

--Robot médico--

Huir/avisar:

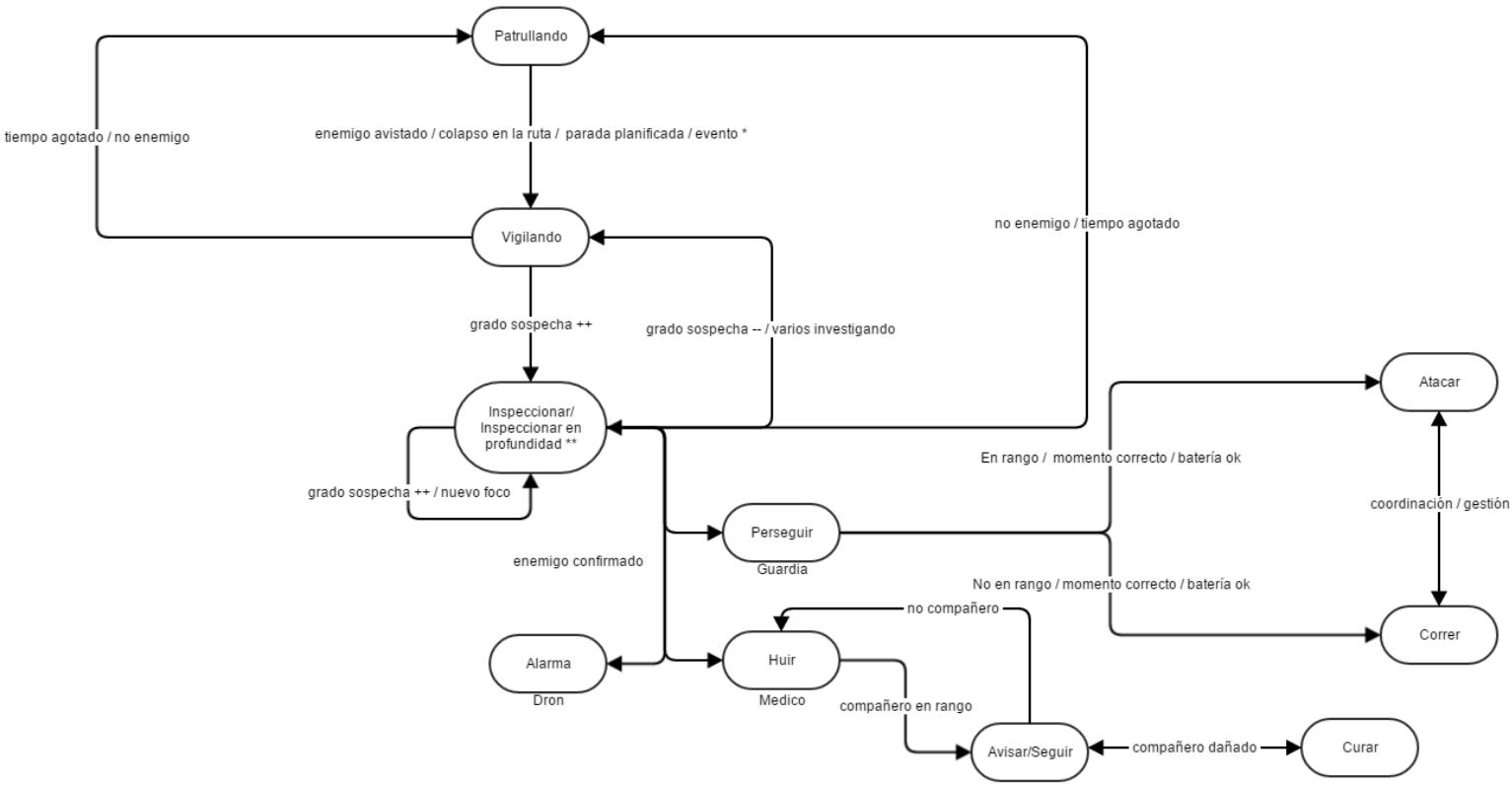
- Los robot médico no pueden luchar, cuando nos hayan detectado tendrán 2 posibilidades:
    - Si están solos, huirán buscando a un guardia de asalto o dron de reconocimiento. Cuando lo encuentren, vendrán a toda velocidad ("correr" para los guardias de asalto) hasta el punto donde el robot nos vio/oyó. El grado de sospecha de los enemigos aumentará mucho cuando esto ocurra, si pese a eso no nos encuentran después, seguirán las normas anteriores (escogerán rutas en esa parcela o quedarán en modo vigilancia)
    - Si no están solos, permanecerán siempre cerca de un guardia de asalto y utilizarán la mecánica "curar" con él.
        - Si estaba con un dron, simplemente asistirá a los guardias de asalto que acudan a la alarma.
        - Si estaba con otro robot médico, huirán intentando tomar rutas diferentes siempre que sea posible.
    - Si no pueden huir debido al escenario, o a un bloqueo en particular quedarán inmóviles y nos suplicarán clemencia. Si el bloqueo se anulase, volverían a intentar huir.

Curar:

- Los robot médico pueden curar solo a los guardias de asalto, esto hará que necesitemos el doble de tiempo para matar al guardia de asalto (si decidimos seguir ese orden y no matar primero al médico). Es decir, cura el 50% del daño que hacemos a nuestro mismo ritmo.
- 2 (o más) robot médico pueden curar al mismo guardia de asalto, esto hará que sea imposible matar al guardia de asalto si no eliminamos primero al menos a un médico.
- Si el guardia no necesita curarse, el robot médico simplemente dará vueltas a su alrededor (si está estático luchando contra nosotros) o lo seguirá (si nos está persiguiendo).
- Mientras cura, debe permanecer inmóvil y a una distancia corta del guardia de asalto.
- Si el guardia de asalto al que estaba curando muere y no quedan guardias de asalto, entrará en modo huir de nuevo.

Potenciar (pasiva):

- Los robot médico tienen un área activa a su alrededor permanentemente:
    - Los robots de todo tipo dentro del área ven sus sentidos aumentados (vista y oído).
    - La batería de los guardias de asalto se reduce mucho más lentamente.



Este diagrama (que intenta simular una máquina de estados) resumiría toscamente el comportamiento general de los enemigos a falta de diversos matices y aclaraciones que deberán de leerse en los apartados anteriores.
Concretemos al menos los más importantes:

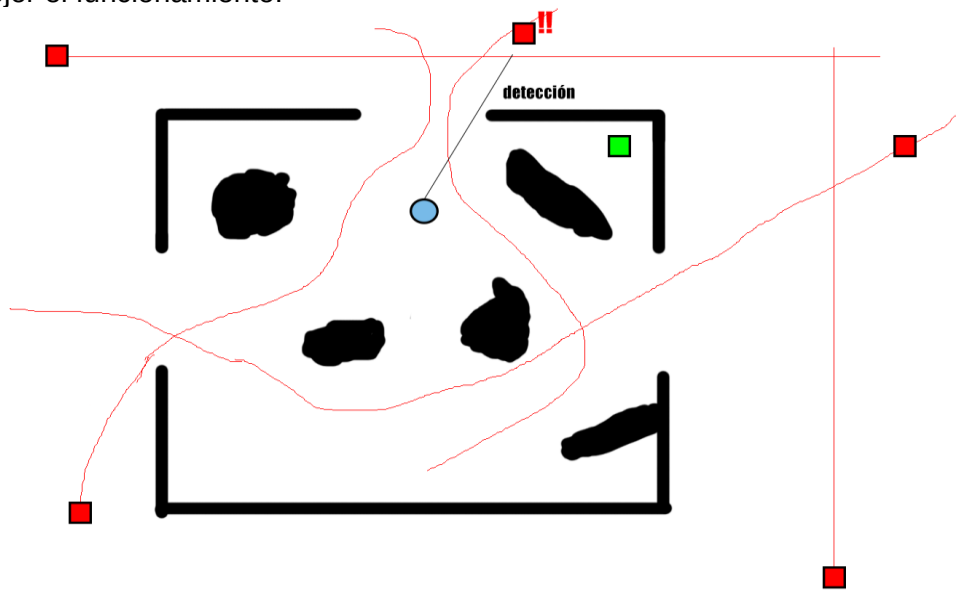
- * evento se refiere a una multitud de posibilidades que no cabían o eran demasiado concretas para ponerlas en el diagrama, tales como avistamiento de cadáveres, modificaciones en el mapa, bloqueos concretos en pasillos…

- ** Inspeccionar en profundidad quizás debería ser una mecánica a parte de la de inspeccionar, pero lo hemos decidido representar así. Inspeccionar a secas es ir al lugar y vigilar rotando. En profundidad sería comprobar escondites en un radio alrededor del lugar. Se decide en función del grado de sospecha.

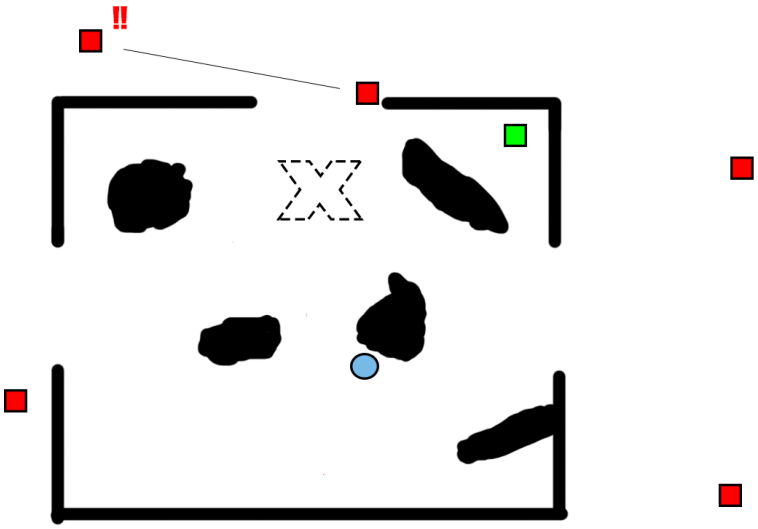
- El guardia puede usar “correr” en la inspección si el grado de sospecha lo requiere. En la persecución “momento correcto” hace referencia a si el escenario y las acciones de los compañeros.

BOCETOS

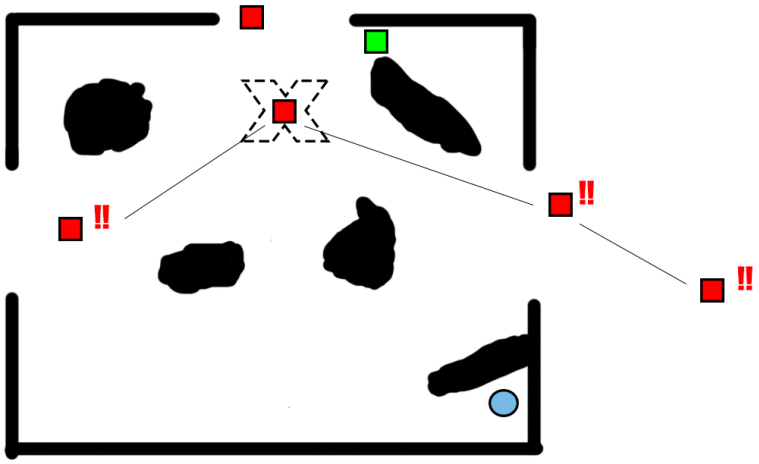
A continuación intentaremos ejemplificar alguna situación completa para entender mejor el funcionamiento:



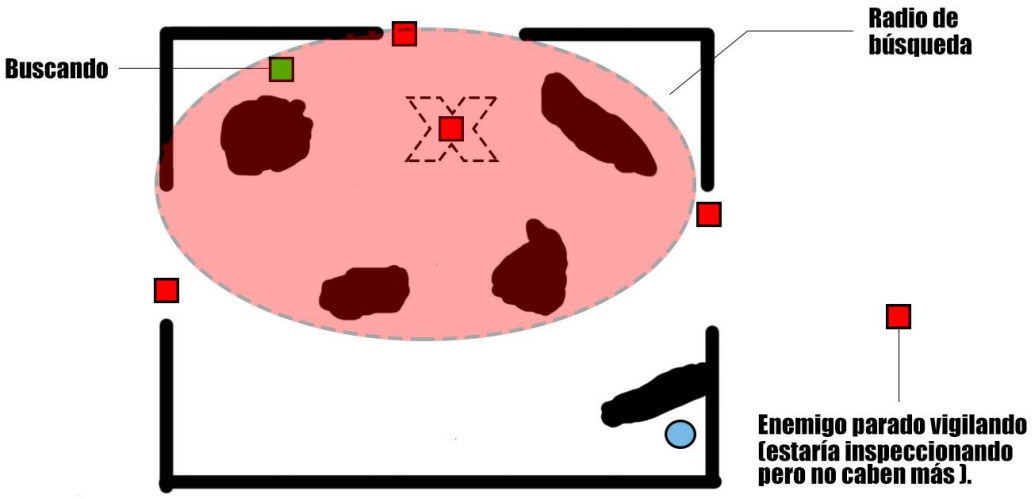
Los cuadrados rojos representan guardias de asalto, el cuadrado verde es un robot médico, la elipse azul es el protagonista, las partes negras son muros/chatarra, las líneas rojas son las rutas de patrulla de los enemigos.
Aquí han visto al protagonista un corto periodo de tiempo desde una distancia media.



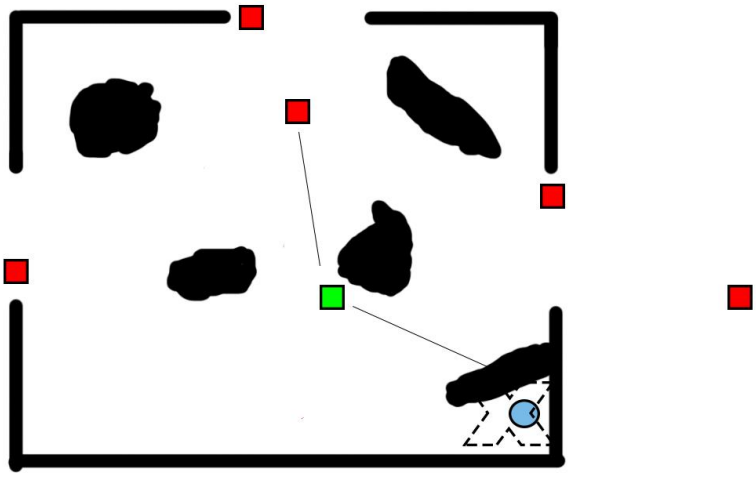
El protagonista se esconde rápidamente, el enemigo va a investigar al punto y un compañero lo ve e inspeccionará con él. El resto siguen sus patrullas.



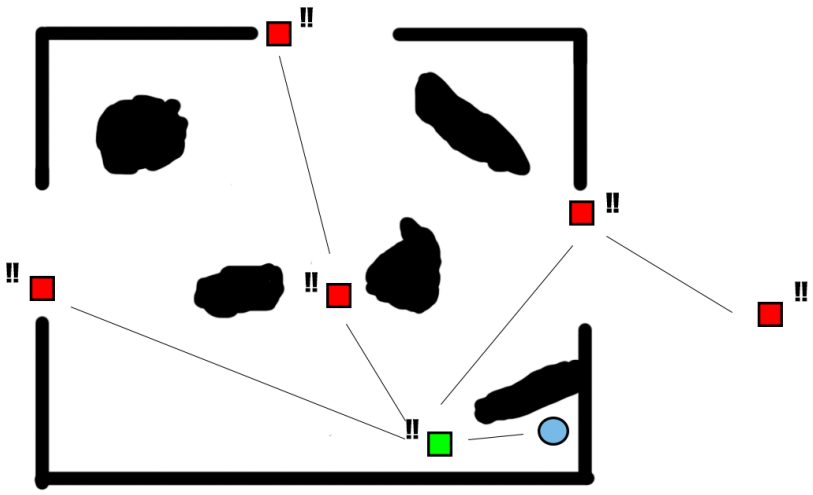
El protagonista se esconde sigilosamente, el robot médico y el resto de guardias ven a sus compañeros, inspeccionarán todos por orden de llegada y por prioridad (al médico). El que llegó primero vigila en el centro.



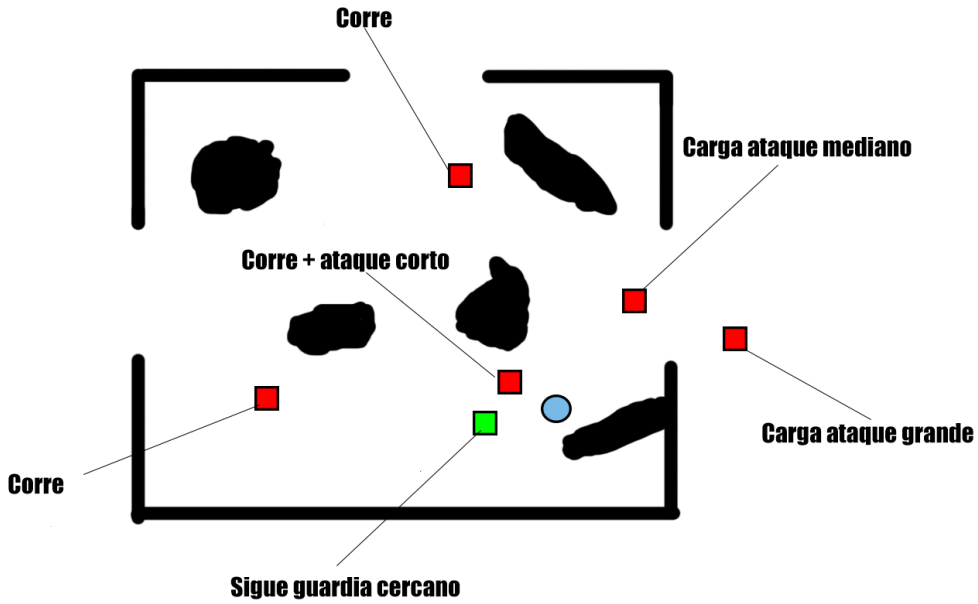
Los guardias toman sus posiciones en cada una de las entradas por proximidad, uno se queda fuera por colapso. Se busca visualmente en el radio hasta que se agote el tiempo o se encuentre.



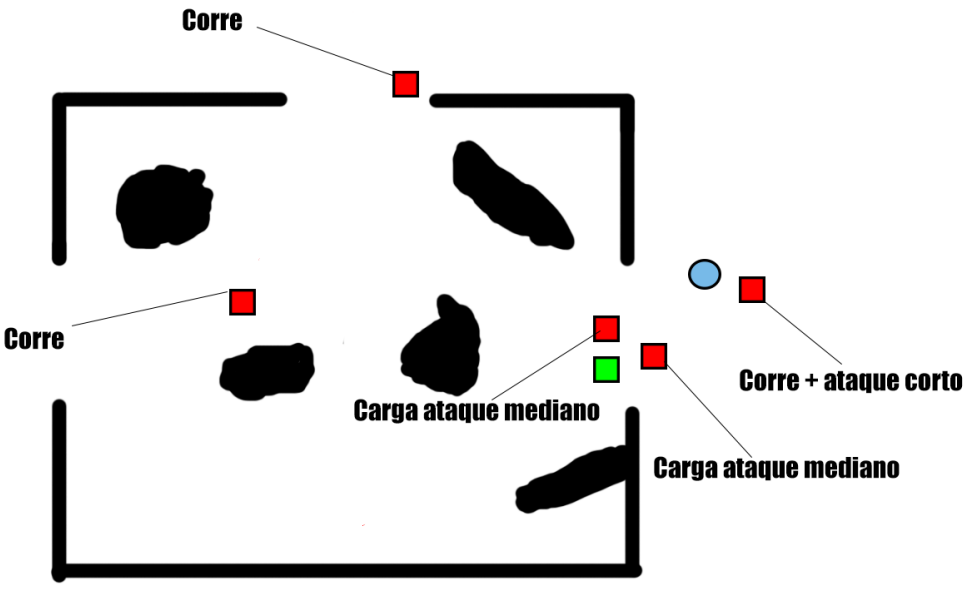
El protagonista hace un ruido sin querer y el robot medico lo escucha, se toma como nuevo punto de inspección. El que vigilaba y el buscador reajustan, el resto se mantiene en su posición.



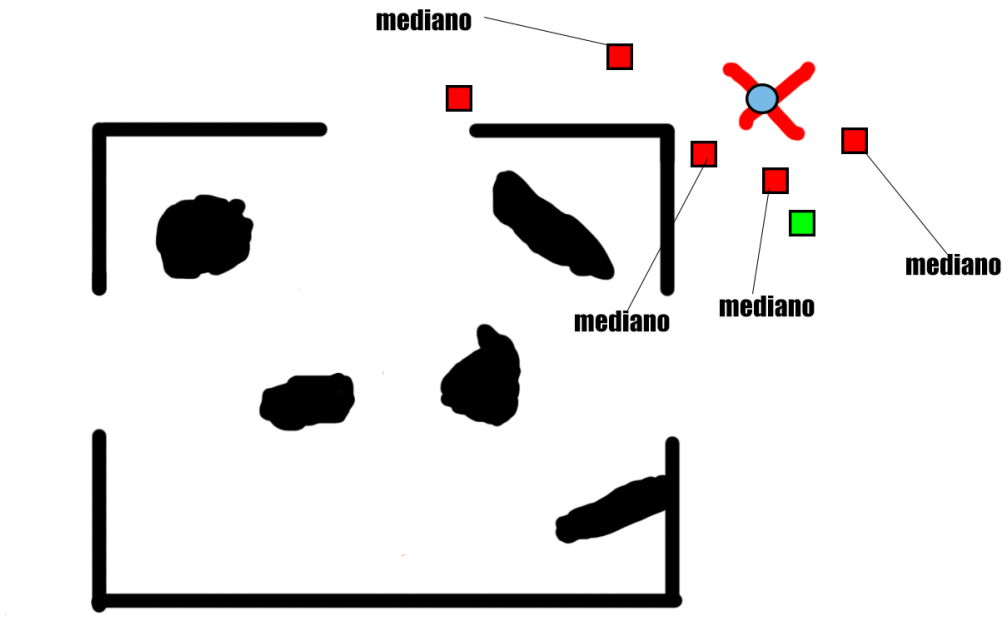
Confirman al prota, en cadena todos entran en modo perseguir y el medico en seguir.



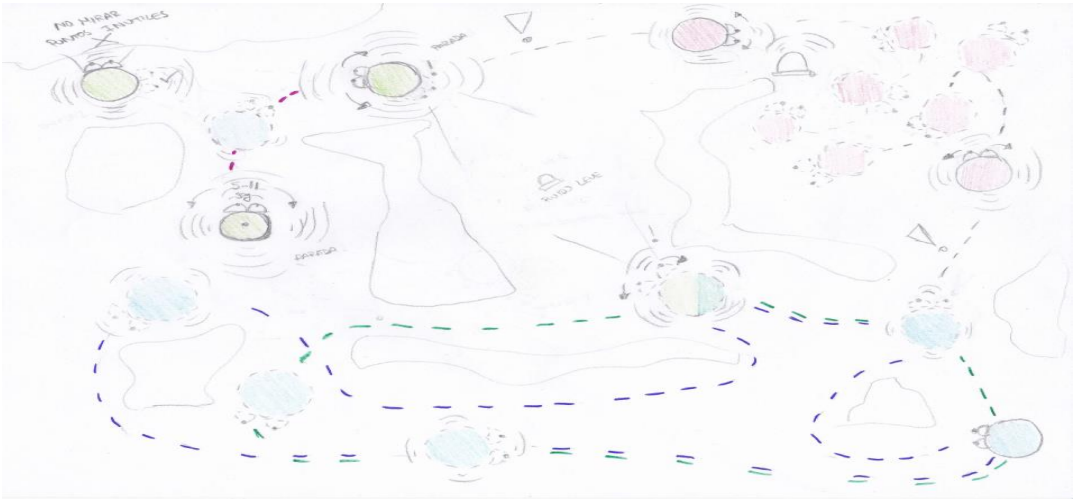
El protagonista intenta huir, los enemigos ajustan su comportamiento en función del escenario, la distancia y el movimiento del protagonista.



Los enemigos de la izquierda ajustan su ruta para atrapar por arriba al protagonista, el resto siguen gestionando du batería y mezclando ataques en función del escenario y respuesta del protagonista.



Los de la izquierda dejan de correr, deciden acabar en campo abierto con múltiples cargas medianas.



En este boceto tenemos tres situaciones de la IA distintas:
- El enemigo de arriba a la izquierda, tiene una ruta preestablecida y se va parando a vigilar/hacer guardia.
- La situación de arriba a la derecha, nos muestra la reacción de un enemigo ante una alarma que ha saltado.
- Por último, la situación de abajo nos muestra a un enemigo patrullando que en un determinado punto de su patrulla escucha un ruido leve y se para a vigilar la zona por donde ha escuchado el ruido.

--Torreta— (caso especial)

- No tiene I.A, ni grado de sospecha, solo usa la mecánica escanear (no oye solo ve).
- Nos atacará mientras estemos en su campo de visión una vez pasado 1 segundo.
- No se puede mover en absoluto, rotará siguiendo un patrón constante o para que el objetivo se mantenga en su campo de visión.

Escenario e interacciones:

--Puertas--

- Existirán 3 tipos de puertas:
  - Puertas automáticas: No requieren nada para abrirlas, sirven para separar espacios (haciendo que sea más difícil ver u oír), aunque hacen ruido al abrirse.
  - Puertas con tarjeta: Requieren una tarjeta que solo podemos obtener de un guardia de asalto o enemigo especial. Fuerzan el desarrollo del juego en una dirección.

- Puertas con palanca: Requieren la activación de una palanca para abrirse o cerrarse. Pueden tener efectos importantes en el desarrollo del juego ya que bloquearán el paso de enemigos (o lo habilitará). Los enemigos no saben usarlas, pero recordarán si estaba abierta o cerrada (para el grado de sospecha).

--Prensa hidráulica--

- Las prensas hidráulicas se pueden activar o desactivar gracias a una palanca, cuando están activadas, destruyen todo lo que hay entre ellas (incluidos nosotros). Nos permitirán tender trampas a los enemigos y destruir muchos de golpe, aunque también nos pueden cortar el paso a nosotros.
- Los enemigos no sospechan de las prensas hidráulicas ni las evitan.

--Palancas--

- Accionan todo tipo de mecanismos, los enemigos pueden detectar que una palanca ha sido activada (y por lo tanto aumenta su grado de sospecha), pero deberán estar más cerca de lo normal.

--Fábricas--

- Las fábricas serán estructuras indestructibles que mantendrán el número de robots constante en la parcela: Si destruimos uno, generará inmediatamente otro del mismo tipo. Esto hará que sea imposible completar el nivel usando fuerza bruta. Las fábricas se podrán desactivar gracias a palancas.

Desarrollo opcional:

--Cintas--

- Cintas transportadoras que se activan y cambian de dirección gracias a palancas, permitirán una mayor cantidad de puzzles y situaciones. (Por concretar…).

--Zonas corrosivas--

- Zonas del mapa que dañaran progresivamente al jugador. Una vez más, añadirán factores a tener en cuenta a la hora de desarrollar el juego e interacciones con enemigos de las que todavía no estamos seguros.

--Boss devastador y otros tipos de enemigo--

- Si podemos, añadiremos “evoluciones” de enemigo o enemigos especiales que fueran ligeramente superiores a los básicos en las etapas medias-altas del juego.
- El devastador se trataría de un boss al que solo podríamos matar usando puzzles, sería una especie de robot gigante que nos persigue y destruye todo a su paso...

--Más objetos, más mecánicas--

- Siempre nos guardamos la posibilidad de añadir algún objeto en última instancia, como un chip que nos vuelva invisibles. Preferimos por el momento ser conservadores y pensar lo básico primero.
- Los disparos tanto de enemigos como del protagonista es una opción que no descartamos por el momento.

--Aliados, robots rescatables--

- Esta sería la más opcional de todas, conseguir diseñar una I.A capaz de ayudarnos a completar el juego y a interactuar con nosotros.

3.1.2. Técnicas y algoritmos a desarrollar.

Vamos a centrarnos principalmente en justificar las técnicas y algoritmos de videojuegos I y II escogidas en el presupuesto:

Máquina de estados y árbol de decisión: Básico para nuestro juego, al tener múltiples enemigos con estados cambiantes según la situación y una toma de decisiones que podría fácilmente tomar la forma de un árbol de decisión, creemos que estas técnicas son fundamentales para un funcionamiento correcto.

Lógica difusa: En nuestro juego hay aspectos que se pueden beneficiar de esta técnica, tanto en la detección (grado de sospecha), como en el combate, nuestros enemigos no toman decisiones binarias. Deben escoger la intensidad con la que atacarán y hay una variabilidad suficientemente alta en esa intensidad. A su vez, a la hora de detectarnos, deberán estimar cuánto realmente nos han detectado en función de una serie de cuestiones como la intensidad del ruido, la distancia…

Following: Creemos que nos podría servir en las persecuciones que realizan los guardias de asalto y el seguimiento que hacen los robots médico a los guardias de asalto.

Steering behavior: Queremos que el movimiento de los personajes sea realista y suave. Acelerarán, frenarán, cambiarían de dirección y tendrán en cuenta la inercia (entre otras cosas) gracias a esta técnica.

Blackboard: Se trata de un coordinador de decisiones. Dado que la colaboración y comunicación entre enemigos es crucial en nuestro juego para que cada uno tome la decisión final, creemos que esta herramienta nos facilitará el trabajo.

Sistema de percepción sensorial: Básico en nuestro juego, queremos que los enemigos vean y oigan de manera realista y óptima.

Dijkstra y pathfinding estratégico: En situaciones en las que los enemigos tengan que cooperar y variar sus rutas será preciso un pathfinding estratégico. Dijkstra podrá ser útil en contadas ocasiones y escenarios en los de alguna manera alertemos a un enemigo que se encuentra entre una complejidad importante de obstáculos o suficientemente lejos como para que el pathfinding sea necesario.

Event manager, LoD, logros y puntuaciones web:

Las 2 primeras son herramientas de optimización que pueden servir casi para cualquier juego.

En cuanto a la parte de red, hemos decidido hacer tan solo logros y puntuaciones ya que a priori nuestro juego no iba a contar con ningún aspecto de red adicional.

Para videojuegos II es un poco más sencillo;

Implementación de menús, HUD, diseño y creación de niveles, cargador de niveles, power-ups, mecánicas de acción, puzzle y mecánicas para entidades sin I.A (torreta) son entregables prácticamente obligatorios y que no tienen demasiado debate. En nuestro juego.

Dadas las características de nuestro juego, también se sobreentiende que la **cámara** será de **seguimiento** y que utilizaremos por lo tanto fundamentalmente "**clipping**" para la optimización.

Decidimos implementar un **editor de niveles** ya que pensamos que construir los niveles de nuestro juego sin él sería demasiado complicado ya que en nuestro caso necesitamos que las mecánicas funcionen muy bien con respecto al escenario. El nivel de level design es lo suficientemente importante como para planteárselo. De paso, incluimos **formato propio para la definición de niveles.**

En el apartado de gráficos tampoco representa una elección sustancial: el juego podría prescindir de **skybox, partículas y vegetación** pero tenemos interés en que sea atractivo y no hará falta forzar nada, para las **partículas** tendremos multitud de posibilidades, desde chispas hasta matices en los lasers o polvo en el ambiente. Para la **vegetación** en un desguace probablemente no sea abundante, verde y colorida, pero habrá.

En cuanto a las **físicas**, debido a que utilizamos control de movimiento por steering behaviors en videojuegos I, suponemos que nuestro **motor será dynamic** pero no estamos del todo seguros si está completamente relacionado. Dado que vamos a utilizar físicas añadimos pues los entregables relacionados que consideramos útiles: **Utilización de propiedades físicas en las mecánicas del juego** (aquí realmente las posibilidades son prácticamente infinitas), **utilización de joints en las mecánicas jugables, trazado de rayos, depuración visual de las físicas.** Realmente no teníamos ninguna preferencia o restricción en ese sentido. No son imprescindibles pero pueden ayudar a mejorar la experiencia de juego.

Finalmente utilizaremos una **arquitectura basada en objetos** por comodidad/sencillez para nuestros programadores principalmente.

3.2. Requerimientos no funcionales.

En cuanto a requerimientos estéticos tendremos que buscar un estilo homogéneo que vaya acorde a nuestra temática (futurista robótica) en todas las disciplinas: modelado, sonido o incluso postproducción. Buscamos algo realista pero desenfadado al mismo tiempo. Al tener 3 miembros matriculados en diseño de sonido, se pretende que ese apartado tenga una presencia notable en nuestro juego.

Por otro lado, si publicáramos nuestro juego, además de añadirle funcionalidades en red, deberíamos preocuparnos por una serie de cuestiones:

- Copyright: Sobre todo si queremos monetizarlo, deberíamos asegurarnos que no infrinjamos ninguna normativa y que nuestro contenido, tanto arte como música sea original y/o se pueda usar sin problemas.
- Las funcionalidades en red deberían ser en tiempo real, fiables y seguras.
- De cara a la promoción del juego, habría que trabajar un mínimo todo el marketing correspondiente.
- Una vez publicado el juego, deberíamos seguir atentos a la comunidad y a posibles parches futuros. Disponibilidad y mantenibilidad básicas.

4. Apéndices

4.1. Referencias