BAGS PLEASE

GAME DESIGN DOCUMENT

#### Versión 1.1

FERNANDO MORENO DÍAZ

DENIS GUDIÑA NÚÑEZ

LUIS MIGUEL MORENO LÓPEZ

COLABORACIÓN DE JULIA CAJIGAL MIMBRERA

Índice

[1 Introducción: 3](#_Toc23189068)

[1.1 Objetivo del documento: 3](#_Toc23189069)

[2 *Game Overview*: 3](#_Toc23189070)

[2.1 Concepto de juego: 3](#_Toc23189071)

[2.2 Genre: 3](#_Toc23189072)

[2.3 Público objetivo: 3](#_Toc23189073)

[2.4 Resumen del *Game Flow*: 3](#_Toc23189074)

[2.5 *Look and Fee*l: 3](#_Toc23189075)

[3 *Gameplay* y mecánicas: 4](#_Toc23189076)

[3.1 *Gameplay*: 4](#_Toc23189077)

[3.1.1 Progresión de juego: 4](#_Toc23189078)

[3.1.2 Estructura de misión: 4](#_Toc23189079)

[3.1.3 Objetivos: 4](#_Toc23189080)

[3.1.4 *Game Flow*: 4](#_Toc23189081)

[3.2 Mecánicas: 5](#_Toc23189082)

[3.2.1 Físicas: 5](#_Toc23189083)

[3.2.2 Economía: 5](#_Toc23189084)

[3.2.3 Objetos: 5](#_Toc23189085)

[3.2.4 Agentes: 6](#_Toc23189086)

[3.2.5 Player 7](#_Toc23189087)

[3.2.6 *Screen Flow* 8](#_Toc23189088)

[3.2.7 Guardado y reinicio 8](#_Toc23189089)

[4 Niveles 9](#_Toc23189090)

[5 Interfaz 9](#_Toc23189091)

[5.1 Sistema visual: 9](#_Toc23189092)

[5.2 Control: 9](#_Toc23189093)

[5.3 Audio, música, efectos de sonido: 9](#_Toc23189094)

# 1 Introducción:

En este documento especificamos el diseño de gameplay del juego titulado *Bags, please*. En este documento se reflejan las ideas habladas en varias reuniones por las siguientes personas: Denis Gudiña Nuñez, Fernando Moreno Diaz, Luis Miguel Moreno López. También contamos con el apoyo de Julia Cajigal Mimbrera, que será miembro del equipo para la asignatura de *Gestión y Dirección de Proyectos*.

## Objetivo del documento:

El propósito de este documento es servir a programadores, diseñadores y artistas durante el desarrollo del juego, así como a profesores y cualquier interesado.

# *Game Overview*:

## Concepto de juego:

*Bags please* es un juego de gestión de recursos en el que el usuario debe dirigir un supermercado.

## Genre:

Se trata de un juego de gestión y administración de recursos.

## Público objetivo:

## Resumen del *Game Flow*:

El juego constara de varias pantallas incluyendo menú de inicio, selección de niveles. Por el momento, sólo se considera hacer un único nivel.

También contará con pantalla de final de juego (puntuación), pausa, créditos y ajustes.

## *Look and Fee*l:

El juego tendrá una estética minimalista priorizando el *low-poly* y tendrá similitud visual con respecto a los colores sin brillo y planos del juego *Untitled Goose Game.*



Ejemplo de cómo queremos que los colores se vean en nuestro juego

# 3 *Gameplay* y mecánicas:

## 3.1 *Gameplay*:

### 3.1.1 Progresión de juego:

El jugador elegirá el nivel que desea jugar, una vez dentro del nivel dispondrá de un tiempo en el que debe cumplir y maximizar objetivos. En función de lo bien que haya completado los objetivos se le recompensará desbloqueando nuevos niveles y otros elementos.

### 3.1.2 Estructura de misión:

Dentro de un nivel, se desarrollarán los siguientes acontecimientos. En un primer momento el usuario contará con unos recursos base para empezar a funcionar, a la vez que empieza a generar economía y superar dificultades y objetivos. Habrá un par de puntos en el desarrollo del nivel en el que el jugador debe sentir el agobio o una mayor presión para que, al superarlos, resulte satisfactorio. Por último, también es importante que en los momentos finales del nivel se le inste al usuario a realizar un último esfuerzo en la gestión para poder maximizar su puntuación a la par que se le relaja con la satisfacción de llegar al final.

Siendo más explicito los puntos de mayor tensión serán reflejados en una mayor afluencia de clientes/ladrones en el supermercado. La duración de cada nivel varía entre los 5 y 10 minutos para no agobiar demasiado al jugador.

### 3.1.3 Objetivos:

Los objetivos del juego son dos principalmente:

El **primero** de ellos y el más básico para el jugador es **evitar la quiebra** del supermercado hasta el final del nivel (dado por un tiempo). El jugador debe ante todo resistir las dificultades (robos y clientes insatisfechos) que le puedan ocasionar la caída del local. Es un objetivo básico de supervivencia.

El **segundo** objetivo es **maximizar** la puntación obtenida al final del nivel. Esta puntación se calcula en base a varios parámetros:

* Ladrones pillados
* Clientes Satisfechos
* Dinero final
* Reponedores adquiridos
* Seguratas contratados

Se premia por tanto que el jugador haga una gestión idónea y responsable de los recursos (no premia la acumulación innecesaria de alimentos en el almacén, el jugador debe ser responsable a la hora de comprar recursos de tipo alimenticio. Si se premia la adquisición de empleados ya que además de devolver en puntos su valor económico, habría una plusvalía de puntos por los ladrones pillados y clientes satisfechos que estarán relacionados con los empleados contratados).

### 3.1.4 *Game Flow*:

Aunque hemos hablado de el algo en puntos como el 3.1.2. Más profundamente sabemos que la dificultad de los niveles de uno a otro deberá aumentar para mantener al jugador en el Flow.

Además, dentro de un propio nivel se reflejarán momentos de mayor o menor tensión (dados por un mayor número de clientes/ladrones) que tendrán como objetivo mantener al jugador concentrado. También se disponen momentos para el clímax que reflejaremos como una especie de “hora del cierre”.

## 3.2 Mecánicas:

### 3.2.1 Físicas:

Se usarán las físicas proporcionadas por el motor. Hay que tener en cuenta colisiones entre paredes y agentes y agentes y agentes. Lo resuelve el movimiento de la IA

### 3.2.2 Economía:

Dado que el juego es un gestor de recursos este es el punto más importante respecto a mecánicas.

El jugador contará con los siguientes recursos:

-Dinero: Se trata de la moneda del juego, será utilizada para adquirir tanto alimentos como para contratar empleados (reponedores y seguratas)

-Número de reponedores: Se trata del número de agentes de tipo reponedor con los que cuenta el jugador en activo.

-Número de seguratas: Se trata del número de agentes de tipo segurata con los que cuenta el jugador en activo.

-Almacén:

* Verduras: Número de verduras que hay en el almacén para que los reponedores puedan reponer los estantes.
* Huevos: Número de huevos que hay en el almacén para que los reponedores puedan reponer los estantes.
* Carne: Número de carne que hay en el almacén para que los reponedores puedan reponer los estantes.
* Leche: Cantidad de leche que hay en el almacén para que los reponedores la coloque en estantes.

Fuera de lo que es el universo del juego el jugador ganara estrellas en función de la puntuación obtenida en el nivel (explicada anteriormente en otro punto cómo se obtiene). Siendo posible un mínimo de 0 estrellas y un máximo de 3 por nivel. Estas estrellas servirán para desbloquear ciertos elementos a modo de recompensa (cosméticos, étc).

### 3.2.3 Objetos:

Existen una serie de objetos importantes que tendrán importancia en el desarrollo del juego y sus dinámicas, estos son:

* Estanterías: Estos objetos podrán ser interaccionados tanto por reponedores como por clientes y ladrones. (Queda definir si también seguratas). En el caso de los reponedores estos se encargarán de rellenar los estantes cuando estos se encuentren vacíos (aunque quizá también medio vacíos). En el caso de los clientes, estos irán cogiendo los productos de ellos y los irán vaciando. Las estanterías contaran con un estado de vacío y con tres estados por producto (repleta, medio llena, medio vacía).

Lo que hace un total de 13 estados posibles.

* Cajas registradoras: en este caso cada nivel contara con un número máximo y mínimo de este objeto ya establecido. Se irán abriendo y cerrando en función de los clientes que se encuentren en el interior del supermercado. Sirven para que los clientes que han realizado su compra paguen y el jugador reciba el dinero.
* Almacén: aunque podemos considerarlo como un lugar de interés más bien, también se puede definir en esta categoría. La función del almacén es retener el stock de los distintos alimentos que compre el usuario. Además, los reponedores que se queden sin mercancía acudirán a el para seguir rellenando. También puede llegar a acceder un segurata para devolver objetos robados (Esto último quizá se caiga)

- Puerta de acceso: Los clientes interaccionaran con estos elementos para entrar/salir del local

### 3.2.4 Agentes:

Dado que en nuestro juego la IA es realmente imprescindible, en este apartado pasamos a definir cada uno de los agentes que encontramos en nuestro juego y sus acciones.

-Reponedores: Los reponedores son los encargados de rellenar las estanterías con más producto cuando estas se vacían. Irán patrullando por el supermercado hasta que encuentren una estantería vacía (Deben saber si se esta rellenando o no) , es entonces cuando , si cuentan con el alimento que puede ir en ese estante (es decir deben saber si pueden colocar algo en esa estantería, por ejemplo no estaría bien tener cuatro estanterías seguidas del mismo producto) , lo rellenan con el alimento que tienen en su inventario. Una vez se quedan sin stock propio acuden al almacén a por más producto (tampoco están allí eternamente hasta que el usuario compra, sino que están un rato y si no encuentran nada que coger se marchan y vuelven al rato).

Además, también pueden recibir mercancía robada por parte de los seguratas para que la coloquen.

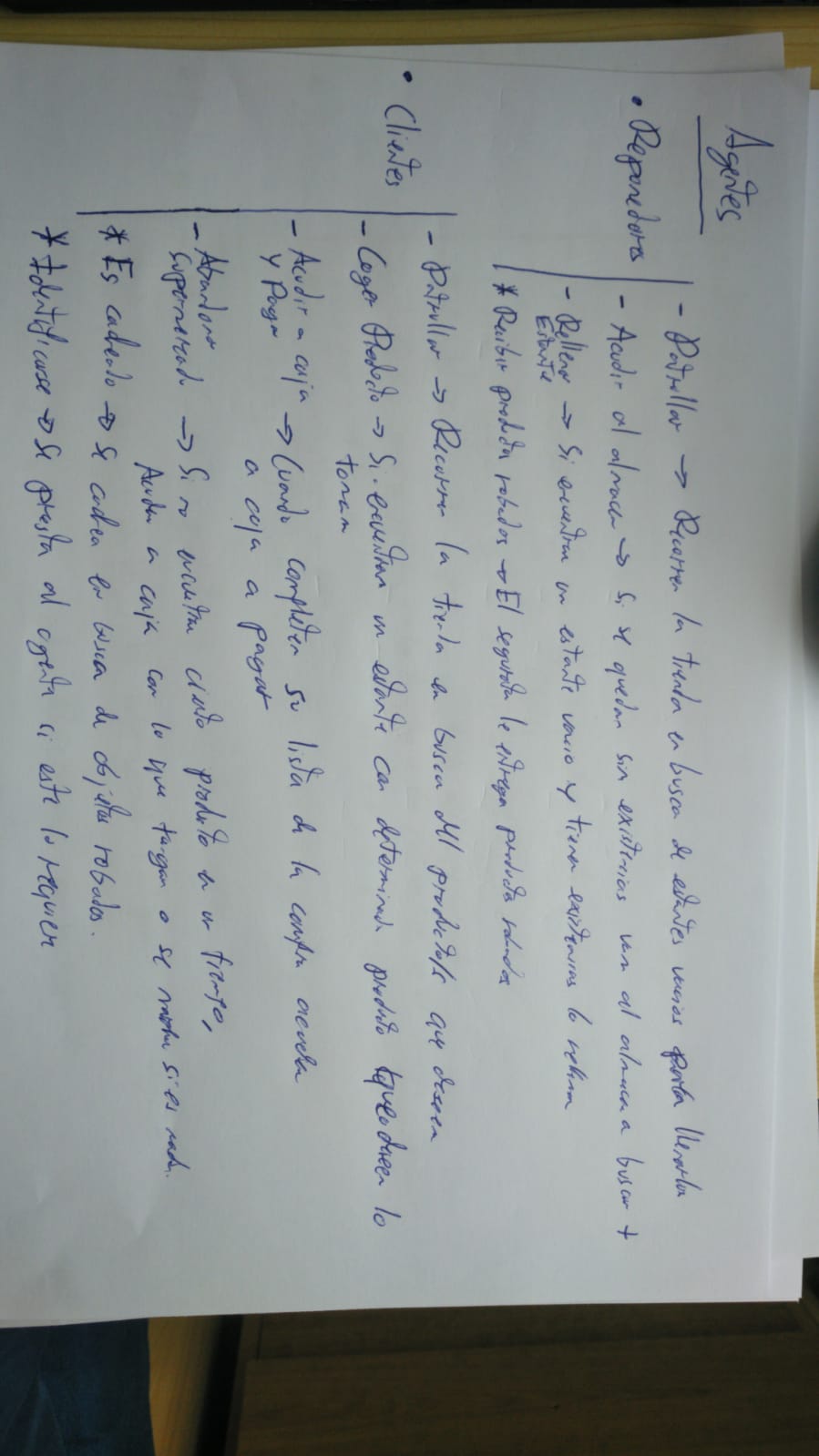
Se debe estipular también cuanta cantidad de producto cogen del almacén (seguramente un numero aleatorio entre ciertos valores por cada alimento).

-Clientes: En el caso de los clientes estos acceden al supermercado con una “lista de la compra” (serie de recursos que desean comprar y que se estipula de manera entre aleatoria y dificultad del nivel). Una vez en el interior se dedican a patrullar el supermercado en busca de una estantería con el producto que desean y si lo encuentran lo cogerán. En el caso de que no lo encuentren podrán marcharse del supermercado o pasar a otro punto de la lista (aunque realmente estarán pendientes de todos a la vez). Si han encontrado algún producto pasaran por caja, pero si no han cogido nada se marcharán sin más.

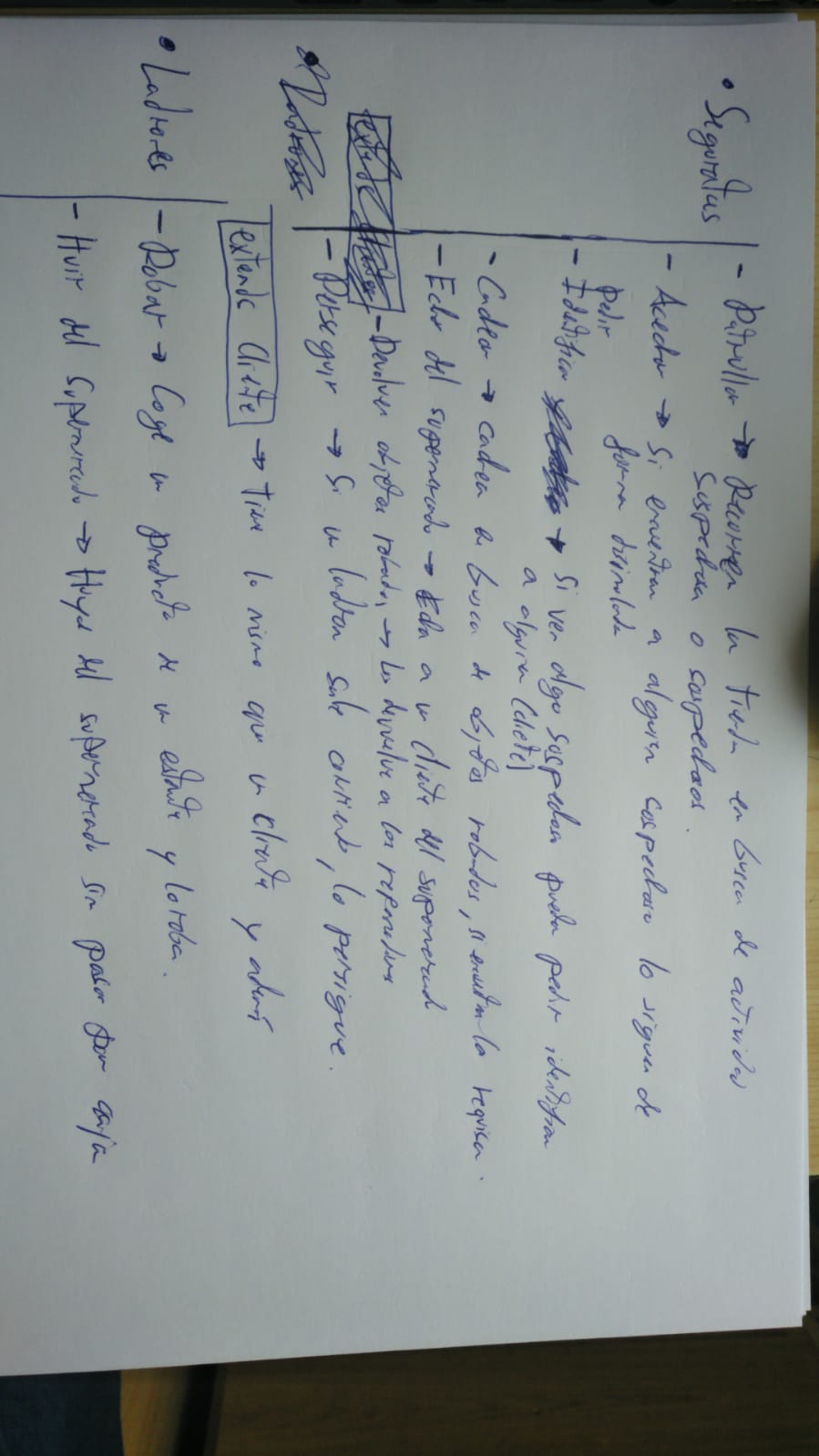
Pueden ser requeridos por un segurata para que se identifiquen o sean cacheados.

-Seguratas: Los miembros de seguridad recorren la tienda en busca de infractores. Cuando encuentran un posible sospechoso se dedican a acecharlo. Pueden pedir que se identifique cierto cliente si es sospechoso de algún delito de robo, también pueden cachear. Si encuentran a un ladrón lo echan del supermercado y le requisan lo robado y al terminar vuelven a patrullar. Además, si un sospechoso se da a la fuga este lo persigue y lo atrapa para posteriormente requisarle lo robado y echarlo del establecimiento. (Vuelve a patrullar al acabar)

Por último, devuelve los objetos robados a un reponedor.

-Ladrones: los ladrones actúan exactamente igual que un cliente exceptuando que además en cierto momento pueden decidir robar un artículo (creen que no les están viendo, no roban porque sí). También, si se ven en una situación de peligro (vayan a ser identificados o cacheados) pueden decidir huir. Si les pillan (es decir se les puede atrapar) son expulsados del lugar y les quitan los objetos robados.

A continuación, adjuntamos imágenes con esquemas más sencillos de lo explicado anteriormente:



### 3.2.5 Player

Partiendo que el jugador tendrá una vista privilegiada (cámara cenital inclinada), el jugador podrá mover la cámara y hacer zoom.

Además, podrá comprar alimentos para el almacén y cuando compra un empleado podrá decidir donde lo coloca arrastrándolo sobre la escena.

### 3.2.6 *Screen Flow*

El manejo de pantallas se realiza de la siguiente forma. En primer lugar, contaremos con una pantalla de inicio/titulo que de la bienvenida al juego. Después de esa pantalla deberíamos poder acceder a la selección de niveles (en principio deberíamos realizar al menos 3 de dificultad creciente). Una vez seleccionado el nivel que se desea jugar cargara la escena del nivel en si donde el usuario jugara la partida. Una vez el tiempo de partida estipulado finalice o el jugador quiebre. Si pasara a la pantalla de puntuación (puede estar superpuesta encima de la de juego) Una vez se complete esta pantalla y se guarde nuestra puntuación final (estrellas y eso) volveremos a la pantalla de selección de nivel.

Aunque el objetivo de esta asignatura es el desarrollo de la IA no cabe menospreciar pantallas que deberían estar presentes como son la de pausa, la cual se puede ejecutar en cualquier momento del nivel y sirve para salir y volver al menú principal.

En el caso de querer añadir una pantalla de créditos y/o ajustes seria necesario una pantalla entre el selector de niveles y la de titulo la cual actuaría como menú principal con opciones para acceder a los dos puntos anteriores y obviamente al selector de niveles.

De forma más visual:

La pantalla de inicio será una especie de video de fondo con el título a lo “terraria” y con un texto que nos indique pulsa para continuar.

La siguiente pantalla si es el menú principal tendrá una lista de botones en vertical cada uno accederá a lo que ponga en ellos.

En el caso de la selección de niveles que es el más relevante en este punto y dado el objetivo de la práctica.

Constara de una pantalla visualmente similar a la selección de niveles como “angry birds” en las que cada botón nos permite acceder a un nivel (numeración de 1 a …) además de mostrarse la puntación máxima alcanzada y el número de estrellas.

Una vez dentro del nivel, y dado que estamos explicando el paso de escenas de una a otra, nos interesa decir que habrá un botón que nos permitirá abrir el menú de pausa para después salir al menú principal desde este.

Al acabar la partida se abrirá una escena que muestre puntaciones y permita volver al menú principal.

### 3.2.7 Guardado y reinicio

Se deberá guardar un perfil de usuario que recoja las puntaciones máximas obtenidas en cada nivel y también aquellas recompensas desbloqueadas con las estrellas (a falta de definir cuales mas allá de nombrar elementos estéticos).

# 4 Niveles

Realizaremos tres niveles con distinta disposición de estanterías, cajas registradoras, almacen, puertas de acceso. También el número de máximo de clientes/ladrones, tiempo y otros factores serán determinantes.

También hay que determinar los recursos iniciales del jugador en cada nivel.

# 5 Interfaz

## 5.1 Sistema visual:

En anteriores puntos hemos explicado visualmente los menús o al menos la gran mayoría.

En el caso de estar jugando un nivel, el HUD debe mostrar el tiempo que queda a la partida, el dinero que tiene el jugador y el número de alimentos que hay en el almacen.

Además, en el caso de los agentes, se deberá mostrar encima de ellos a modo de iconos sobre un recuadro (podría funcionar como un bocadillo de comic con forma de rectángulo) que alimentos tiene cada uno encima, esto se aplicaría solo a reponedores y a seguratas ya que si todos los agentes lo tuviesen seria un caos visual para el usuario. Además, que realmente como el jugador solo tiene poder de decisión sobre estos (más o menos) pues son los que necesita información constante. Por el contrario, si un cliente no encuentra cierto alimento podría indicarse con un bocadillo de pensamiento con la imagen del producto.

También se valora la posibilidad de mostrar un mensaje de bocadillo cuando un cliente es atendido en la caja registradora dando así retroalimentación al jugador de que todo va de forma correcta.

Otro elemento visual que se debe mostrar es la detección de un ladrón por un miembro de la seguridad (quizá una exclamación).

## Control:

La mayoría son elementos de pulsar con el ratón, a excepción de colocar los agentes que se colocaran moviendo el ratón y soltando o volviendo a pulsar.

## Audio, música, efectos de sonido:

Se añadirán efectos que ayudaran y complementaran a los elementos visuales.

Además, la música será acorde al tipo de juego ya que debe facilitar la concentración.