

REUNIÓN – 16 ABRIL 2024

DOCUMENTO

Entrega 5-A **prácticamente terminada**, sujeta a **revisión** (ligeros cambios [CU10 precondition, CU 31 postcondicion y escenario principal iguales], formateo y revisión total de VP). **A terminar antes del**

Fecha de entrega jueves, 25 de abril de 2024, 23:59

Entrega 6-A **no empezada**. Trata de los modelos de dominio, de hacer un **diagrama de clases** esencialmente. Para ello primero debemos tener las clases, y para tener las clases necesitamos **avanzar más en el juego**. Más info en el Trello. **A terminar antes del**

Fecha de entrega jueves, 2 de mayo de 2024, 23:59

JUEGO

Muchos avances, mucho trabajo por hacer. Toneladas de sprites que diseñar, managers que crear y requisitos que implementar.

REGLAS

1. Pon en la **primera línea del script**: **/* Author:nombre1, nombre2... */**
Para que, si no entendemos algo, no tengamos que ponernos a buscar quien hizo el commit.
2. **Comenta el código**, por favor. Piensa que lo tiene que leer otra persona y entender qué estabas pensando/haciendo en ese momento.
3. Pon los **nombres de las cosas en inglés** y a poder ser usando la **clásica forma** sin espacios y con el uso de mayúsculas (ejemplo: LogicManager, PlayButton...), por favor. Por claridad y por mantener un diseño consistente.
4. Cuando metáis objetos visibles (UI, etc), usad **ANCLAJES** para que no se rompa el juego en otras resoluciones.
5. **En los archivos .csv, cuidado con escribir cosas fuera de las columnas preasignadas**. Al parsearse a un txt separado por ;, hay que ser riguroso y escribir solo en las columnas preasignadas (la primera línea del csv siempre se ignora, está así puesto en el código).
6. Para cargar los archivos del csv, **ejecutad primero el juego en el editor**. El **ObjectLoader no funciona en Build**.

***LAS SUGERENCIAS DE ASIGNACIÓN SON, VALGA LA REDUNDANCIA, SUGERENCIAS. SE PUEDEN CAMBIAR SIMPLEMENTE HABLANDO SOBRE QUIÉN HACE QUÉ. PROBABLEMENTE MÁS GENTE SE INVOLUCRE EN TODAS LAS SECCIONES DEL JUEGO DE UNA FORMA U OTRA.**

TAREAS

Hay que ponerse con las **estadísticas**. Inspiración fuerte del Cookie Clicker, pero con un toque personal. Quizá algo más de estadísticas generales y con el **apartado de logros aparte (POR DETERMINAR)**.

Ideas para las estadísticas: **script StatsManager**, que viva en ClickingScreen, mantenga la cuenta de cada parámetro y guarde a memoria con el sistema de guardado de Eduardo. Deben ir codificadas al guardarse. Todo lo que no sea playerPrefs no debe de saberse como se modifica, porque si no podrían “hackear” el juego. No cuesta mucho porque es usar un método de Edu. También hay que **diseñar la escena o panel (por determinar)**, aquello que sea más fácil) donde se desplegarán las estadísticas.

PERSONAS QUE PODRÍAN DESEMPEÑAR ESTA TAREA* => **Artur y Jorge**. Este último me mencionó que lo estaba mirando para hacer (no hizo nada al final) y Artur porque mola cantidubi.

También hay que seguir con la **tienda**. Juanma ha hecho ya mucho trabajo sucio de cómo será la tienda, pero queda por hacer. Rubén creó un **ObjectLoader** que carga (dándole al editor, no va al Build, es imposible) y guarda ScriptableObjects (Juanma escribió Scriptable porque tiene down) a una carpeta en concreto. Se puede usar para **crear y actualizar en masa objetos de tienda (generadores y mejoras) y logros**. Ahora mismo solo carga generadores, de un archivo **generators.csv** (archivo tipo Excel que separa las columnas por ;, fácilmente parseable en código) situado en **Assets/Resources**. Hay otros archivos creados, pero vacíos y sin un método que los use, en la carpeta Resources: *power-ups.csv* y *achievements.csv*. Además (**UI**), hay que buscar una forma de que las **mejoras no aparezcan si no las tiene desbloqueadas el jugador**, y de que los **generadores aparezcan ocultos si no los tiene desbloqueado el jugador**. Similar al Cookie Clicker (en los generadores, no hace falta que aparezcan ocultos todos, solo los siguientes):



Tampoco estaría de más una forma de cargar sprites usando su nombre a los ScriptableObjects con el **ObjectLoader** (aunque esto es más secundario, creo). También hay que **pensar una forma de hacer que cada mejora y cada generador nuevo cueste algo más** (sin que sea demasiado barato o costoso). Los eventos aleatorios para el futuro, que se avecina oscuro.

Con respecto a qué debería de tener cada **generador**, quizá debería ser algo como (**ID, TITLE, DESC, PRICE, UNLOCKED (booleano), ¿POINTS PER SECOND A AÑADIR?, ¿NÚMERO DE ITEMS?, SPRITE**) y cada **mejora**, quizá debería ser algo como (**ID, TITLE, DESC, PRICE, UNLOCKED (booleano), ¿MEJORA QUE SE PONE?, SPRITE**).

Otras cosas a añadir: el Sprite de la **mejora correspondiente se pone en tu cursor** y al clicar sale una **pequeña animación** como en el CookieClicker

PERSONAS QUE PODRÍAN DESEMPEÑAR ESTA TAREA* => **Juanma y Javi**. El primero porque ya tiene experiencia en la tienda y el segundo para ver si espabila y hace algo. Del tema del ObjectLoader se encargará **Rubén**.

Hay que empezar también con los **logros**. Esto va a depender del ObjectLoader y probablemente sea más fácil que la tienda. Una **SortedList** que ordene primero porque **SÍ** esté desbloqueado, y después por cualquier orden (alfabético, por ejemplo). Visualmente (UI), es una especie de grid en la que hay X logros por fila y una cantidad Y de columnas. Estaría bien usar marcos hechos con **4 palos en pixel art** (ejem, ejem, **Eloy**). De nuevo, nos inspiramos en Cookie Clicker y algo (poco) en los logros de Steam, que “se ordenan” (no exactamente, porque están como divididos) primero porque **SÍ** esté desbloqueado y después por algún criterio que desconozco.



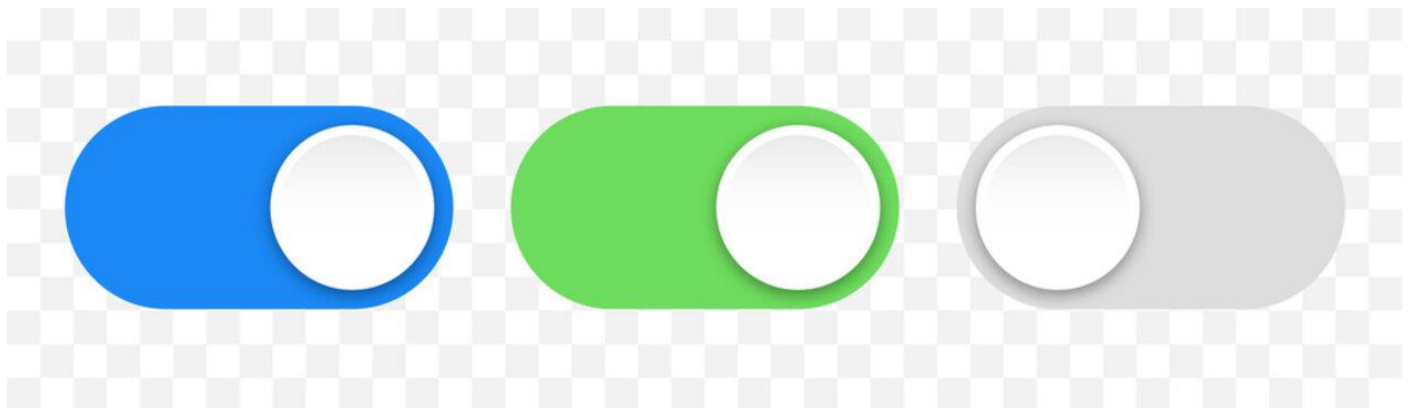
Con respecto a qué debería de tener cada Object o ScriptableObject (probablemente mejor lo segundo), quizá debería ser algo como (**ID, TITLE, DESC, CONDITION TO UNLOCK, UNLOCKED (booleano), SPRITE, ¿MEJORA POR LOGRO?**).

PERSONAS QUE PODRÍAN DESEMPEÑAR ESTA TAREA* => **Eduardo y quizá Eloy**. Dijo que lo quería hacer y probablemente hacer el sistema no sea algo titánicamente difícil. Además, creo que hay tutoriales en internet. Del tema del ObjectLoader se puede encargar **Rubén**, aunque tampoco es 100% necesario ya que Eduardo sabe cómo va el escribir y leer de fichero (es su actividad favorita de todos los lenguajes).

Menús. El menú principal está prácticamente finalizado, salvando que el título que hay es un castañón y que no hay fondo de pantalla. Estaría bien robar (perdóname Dios) algún vídeo o animación decente y poner el vídeo de fondo de pantalla (Unity no admite GIFs. Si es un .gif, hay que pasarlo a .mp4 o similares). Rubén y Jorge en su día lo intentaron sin éxito. No sé si tenía que ver con el modo de renderización del canvas, que se cambió para que Eloy pudiese meter la animación del destello. Puede que ya se pueda poner fácilmente (no tengo ni idea).

En el **apartado de información** se mostrará **información básica de lore** (lo básico de hacer gordo al líder y que 1 caloría = 1 punto de fe) y **notas del parche** (dudo que hagamos alguna actualización después de tener una versión 1.0, pero mola tener eso). Esto seguramente será un panel. Es simplemente texto en un cuadrado bonito.

El apartado de estadísticas y logros del menú principal se construirán conforme se construyan los apartados de **estadísticas y logros**, ya que son esencialmente lo mismo. El **apartado de opciones** está medio creado, pero está **más feo que pegarle a un padre**. Hay que ponerlo bonito, además de terminar de implementar las cosas que faltan. Y dirás, ¿qué falta? Pues falta el **toggle de inicio rápido** (salta directamente a la ClickingScreen si está activado – se salta el menú de inicio), el **slider del volumen de efectos** (está el de música, para este hay que pelearse con el audio, crear un nuevo mixer etc) y un **slider para FrameCap** (esto es una tontería, pero tiene pinta de no ser difícil de hacer y nunca está de más). No hay toggles en Unity (creo), hay checkboxes, así que habría que crear el toggle (hay tutoriales en YT). Esto es un toggle, para los que no manejan la lengua de Cervantes:



Audio. Con respecto a la **música**, estaría bien crear un **pequeño OST** de unas 10 canciones generadas con IA (suno.ai) de un rollo similar a la que ya hay. Una de ellas sería la canción principal, y cuando se jugase se iría cambiando entre todo el OST en bucle (quizá se podría incorporar un **pseudo-algoritmo de shuffle**, para que no siempre estuviesen en el mismo orden... Ya veremos). Los **efectos ya los dejaremos para más adelante**, aunque no serán complejos y funcionarán con el mixer y slider de volumen de efectos que he mencionado antes.

Otros. No estaría de más, y más siendo la clase de tontería que es, meter abajo a la izquierda una **pieza de texto con opacidad al 30-50% que pusiera v. 1.0** (indicando la versión, como en el CookieClicker) y que nunca se destruyera.




Añadir un campo en la ClickingScreen, encima del botón del líder, que pusiera **“La iglesia de <user>”, y el usuario pudiese rellenar ese campo con el string que quisiese.**

Considerar que **vamos a manejar números (quizá) mayores que los números más grandes representables**. Además, **es totalmente ilegible** un número del tipo **18,446,744,073,709,551,615**, así que debería de haber alguna forma de representarlo como “18 octapintisupermillones” o algo así.

Characteristics of the integral types		
C# type/keyword	Range	Size
int	-2,147,483,648 to 2,147,483,647	Signed 32-bit integer
uint	0 to 4,294,967,295	Unsigned 32-bit integer
long	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807	Signed 64-bit integer
ulong	0 to 18,446,744,073,709,551,615	Unsigned 64-bit integer

6 filas más · 29 sept 2022

 Learn Microsoft
<https://learn.microsoft.com/en-us/csharp/builtin-types>

Integral numeric types - C# reference - Learn Microsoft

PERSONAS QUE PODRÍAN DESEMPEÑAR ESTA TAREA (menús, audio y otros)* => **Javi y Rubén en menús y audio**. Son quienes han hecho los menús. Además, Javier tiene un muy buen sentido de lo que es bonito y de lo que no (generalmente). **Todos en otros**. Son cosas concretas, pero quizá importantes, y no pertenecen a ninguna sección en específico.

ARTE

La **sección más crítica del proyecto**. Nadie sabe usar Unity, pero sabemos programar... ¿Pero alguno sabe dibujar? Eloy es aquí nuestra mejor baza (no ha dibujado en su vida, pero usó GIMP en el instituto), nuestro soldado de referencia. Eloy y el Aseprite. Dios los bendiga.

Pero aunque Eloy sea un gran dibujante, probablemente **acabe necesitando apoyo**. Los sprites de las mejoras, por ejemplo, seguramente serán de 64x64 píxeles y el Aseprite es bastante fácil e intuitivo de usar (palabras del soldado de referencia).

PERSONAS QUE PODRÍAN DESEMPEÑAR ESTA TAREA* => **Eloy, Juanma, Rubén e incluso Javier, aunque en realidad todos**. Fueron los que votaron algo que no fuese un no rotundo en aquella encuesta que hizo Rubén. Realmente todos podemos ayudar, ninguno sabe dibujar así que no vamos a obtener resultados grandiosos. Sed humildes y no le echéis 50 horas a copiar un ítem del Isaac cambiando cuatro píxeles. Gracias.