

# POSTMORTEM: Kill to Heal del equipo XTeam

La historia real de una programadora que se sumergió en el proceso artístico:

Castigo y supervivencia en el desarrollo de un videojuego en Unity3D.

Mi nombre es Ángeles Fuentes, viví siempre en Tenerife hasta que, en 2012, dejé a mi familia y amigos (la mayoría personas tóxicas que no entienden que hay que luchar por los sueños) para empezar mi vida universitaria como Ingeniera de Software en la isla vecina, Gran Canaria.

Desde 2014 participo en desarrollos de videojuegos en **LudumDare**, en la ciudad ahora llamados **IslandJam**. Hasta ahora he realizado 6 juegos muy pequeños, algunos desastrosos por supuesto y *Kill to Heal*. Recuerdo que la primera vez sólo me dediqué a programar la UI del juego, pero esto no hizo que dejara de profundizar por mi cuenta en la parte de programación.

En 2018 me matriculé en el Experto Universitario en Diseño y programación de Videojuegos, ¿por qué hice esto? Volví de la RWTH de Alemania y no quería verme ociosa, y como aún no he elegido un máster orientado a Neurociencia, sólo tenía como opción este experto. Escribir este *postmortem* puede que me ayude a deconstruir esta respuesta, que puede parecer vaga, pero os prometo que hay mucha metralla detrás.

El equipo *XTeam* surgió en noviembre de 2018. Al comenzar el curso nos pusieron en grupos aleatorios, o no tan aleatorio, para desarrollar con nuestros conocimientos base un videojuego muy tonto. De esta prueba surgió *GetOutPennis*, y el equipo *XTeam*, formado por Aitor García, Sara Arribas, Jesús Ojeda y yo. 😊 Para ponerte en situación he de decir que durante este proceso participé como programadora junto a Aitor y Sara, pero “junto” sobra aquí. Me vi tan sola en el proceso de programación que estuve a punto de abandonar el grupo después de la entrega y presentación, cosa que no hice básicamente porque vi una joya en Jesús Ojeda (diseñador artístico). Hacía mucho tiempo que no me encontraba con una persona con tanta implicación.

*Kill To Heal* se desarrolló partiendo de una idea propuesta en clase por mi compañero Aitor, compañero del gremio. Yo seguía en el mismo equipo, que finalmente no abandoné, pero me negué a trabajar “junto” a los programadores :P y decidí unirme al proceso de desarrollo artístico junto a Jesús.

*Kill to heal* sería un juego 3D, *shooter* en primera persona, cinco niveles, variedad de enemigos y cinco jefes finales más uno extra. Ambicioso.

Muy importante comentar, que en mi *postmortem* hablaré principalmente de Arte, pues este era mi “departamento”, pero siendo programadora me permitiré opinar sobre el trabajo de programación, visto desde lo que sería producción. Si alguno se siente ofendido sabrá cómo contactarme.

## Empezando el juego

La idea concebida en enero, la entrega en junio. Bien. Tenemos seis meses.

### Personajes y enemigos

Jesús y yo, a partir de ahora equipo artístico, dividimos los bocetos: a mí me hacía ilusión el personaje, pues cuando era pequeña dibujaba nada más que gente, y hacía casi diez años que no hacía un garabato. Jesús se cogió tres enemigos. Y los jefes los íbamos cogiendo de tarjetas de nuestro tablón de *trello* cuando acabamos los bocetos.

#### Lo que podía haber salido mal y ni tan mal:

Pues los tutores de arte nos pidieron tres bocetos por cada bichejo. Entregado. Pero los tutores de producción nos pedían bocetar al menos veinte. No les hicimos caso. No quiero decir que recomiende a otros hacer esto, para nada, claro está que cuantos más bocetos tengas, más opciones para elegir.

#### Lo que sentimos con nuestro *feedback*:

Nuestros bocetos eran justo lo que queríamos, los enseñábamos a los programadores y también estaban contentos con el resultado.

Si quieres leer cómo fue el desarrollo del personaje puedes hacerlo en

<https://killtohealgame.wordpress.com/2019/04/07/personaje-post-mortem/>

### Diseño de niveles

#### Lo que salió mal:

Pues básicamente todo al principio.

Yo apostaba por un estilo de niveles orgánico, mientras mi compañero de arte hacía un nivel bastante *flat*. En los primeros bocetos parecía que iban a cuajar, cosa que no pasó cuando modelamos y pasamos los escenarios al editor.

Busqué muchísimos tutoriales para modelar cuevas, horas invertidas, que se han convertido en conocimiento que vale oro, pero que no sirvieron para avanzar el proyecto.



Bocetos finales de nivel Tutorial y nivel Intestinos.

#### Lo que salió bien:

Después de casi dos meses, empezamos a tener soltura en modelado con Blender, y los tutores nos dieron directrices porque nos veían muy perdidos al no encontrar solución a los

modelos finales, ni los estilos encajaban y yo me estaba derrumbando, perdida en el abismo de tutoriales de youtube intentando plasmar mi idea.

Así que nos dijeron: ¡FLAT!, usad estilo *flat*, organizaos para que no choquen los niveles. Y lo conseguimos. ClapClap.

### **Modelos en general**

La mayor parte de los modelos son de mi compi Jesús, es una máquina de modelar, y además me ha comentado que es lo que más le gusta. Así que le di todo mi apoyo con ideas, programando cosas muy chorras para darle sentido a sus modelos, animando o corrigiendo detalles en sus animaciones para pulir al máximo con nuestros conocimientos de principiantes.

### **Integrando modelos y animaciones en el proyecto**

#### Lo que salió mal:

Pues cuando el equipo de arte añade modelos dentro del proyecto, no es sólo soltarlo ahí y bum, magia. Hay que poner *colliders*, crear los materiales con colores, trabajar el árbol de animaciones, generar el *prefab* y colocarlos en toda la escena, muchas de las veces nos olvidamos de algún *collider* y cuando otro del equipo lo veía no se paraba a corregirlo.

Por mi parte, cuando generé siete animaciones para los enemigos clones (clones del personaje en el juego) y las coloqué en el código de los programadores (máquinas de estado finito muy bien organizadas, pues cada una de mis animaciones se basaba en este código), los programadores decidieron (sin consultar a arte), eliminar esas máquinas de estado, supongo que porque no consiguieron dar solución a algo, la verdad es que no lo sé porque a mi nunca me lo comentaron, y cortaron por lo rápido y cutre. Así todas las animaciones que tenía este enemigo se quedaron ahí, sin activarse nunca. Adiós a varios días de trabajo animando un humanoide. JAJA.

Esto hace que ahora, el enemigo al disparar siga caminando con una animación que no estaba preparada para ese código, y claro, la culpa es mía porque la animación no es “buena”. Que se puede arreglar usando *layers* en las animaciones, pero esto a mi ya me trae de cabeza, hice pruebas y el resultado fue peor, así que lo dejo estar.

### **VFX y efectos de partículas**

A las partículas les tenía miedo, mucho miedo. Y decidí encargarme de ellas. Mark Sureda consiguió quitarme este terror con un curso implacable sobre Iluminación técnica en Unity3D, tanto que llegué a casa y dediqué esa semana entera a estudiar diferentes efectos de partículas.

Y son una pasada. Tenemos muchas partículas en el juego.

Unas no son más, las hizo el equipo de programación porque sí.

En concreto:

Partículas de fuego para nivel intestinos, están muy logradas y dan vida al nivel intestino, además son las partículas que rodean al jefe que representa la *Fiebre*.

Las partículas de humo tóxico, en concreto, consumen muchísimos recursos. Usuarios que han probado el juego nos comentan que les baja el rendimiento, los *frames per seconds*. Cuando abrí el sistema de partículas primero vi que no existe limitaciones en la generación, como millones de partículas en pantalla. Corregí esto, pero aún hoy, este sistema de partículas puede hacer que sea imposible jugar en un pc de gama media-baja. Lo siento si te pasa esto. Fue algo que no comprobamos hasta hace poco en otros ordenadores que no fueran los nuestros. 😞

### FPS y Rendimiento del juego en el ordenador

Cuando los FPS caían, arte fue la diana. Recortar de arte a tres semanas de entrega no es una solución. Que mi tutor me diga, eso es culpa del modelo, pues tampoco es una solución oye.

Así que dediqué varias tardes a investigar el rendimiento del juego usando el *Profiler* de Unity3D (gracias al tutor Aitor Lozano que me enseñó a entender esta herramienta):

- En el nivel cerebro hay 360 neuronas flotando estáticas, que tenían *collider* y una esfera de unity3D, con miles de caras, ¡que pedían renderizado y *fuah!* El *shader* que tenían esas esferas no ayudaba nada. ¡Arreglarlo fue fácil!
- Los enemigos colisionan contra muros toooooooooo el tiempo, esto no pasaba antes, cuando el enemigo tenía aquella famosa máquina de estados. Gracias a que puse esto en común con los profesores el equipo de programación entendió que había un problema en el código y medio lo solucionaron. Aún las arañas siguen chocando, pero bueno, fue un avance.
- Las partículas de humo tóxico: comentadas antes en VFX no hay solución con esto. Se trata de tema de renderizado, están en toda la pantalla, se mueve, aparecen y desaparecen. **Solución posible:** El humo podía haber sido estático, las partículas están en la pantalla, pero no se renderizan cada vez y en cada movimiento. Como digo, soy una novata en partículas, si que conocía métodos de optimización como *OcclusionCulling* y quemar las luces de una escena usando *BakeLights*, pero el rendimiento con partículas es otro mundo, que estoy empezando a entender.
- *Canvas* de vida en cada enemigo: Cada enemigo renderizaba en cada *frame* su barra de vida. Liada pardísima. Cuando ocultamos las barras de vidas vemos que el rendimiento aumenta considerablemente. El tutor me dio la idea de esconderlas todas y activar sólo la del enemigo al que estás atacando. No es ideal, porque no ves la vida hasta que le has dado el primer golpe, pero como pensamiento rápido me vale. Podía haberse hecho con el *raycast* del *player* al enemigo, así verías su vida desde el momento en el que le apuntas. Después de este cambio los jefes del juego se volvieron inmortales, así que hubo que tocar más código. Pero bueno, algo para tener en cuenta a partir de ahora.

### Contar la historia

También fue tarea de arte contar la historia del juego, ya que la tarjeta estuvo en *trello* sin ser tocada por nadie durante un par de semanas.

Llegados a dos semanas antes de la entrega del proyecto, entendemos que no tenemos historia contada. Jesús y yo montamos unas escenas y como programación estaba saturado de bugs, decidí programar esos cambios de escena.

### Lo que salió mal:

Dos de las escenas ocurren mientras estás jugando, al principio funcionaban, pero luego empezó a llenarse de fallos, a uno no le funciona, al otro sí y luego de repente a nadie le funciona.

Dejar estas cosas para el final, si tienen que estar dentro del nivel jugable, no lo recomiendo. Se rompe el *GameLoop*, el juego tiene que parar, esperar por la escena, etc. El código no estaba preparado para añadir *escenas in situ*, así que puede ser una liada parda.

### **Experiencia general**

Bueno, cada uno trabaja en su parte y ya está ¿no? Pues yo no opino así. Puede que mi forma de plantear el trabajo en equipo no sea la mejor. Porque al final no éramos un equipo de cuatro, sino de dos y dos.

Ya había trabajado con los programadores en el proyecto anterior, y algo que hizo que me estallara la cabeza fue encontrarme una con 10 *if* anidados, sin parámetros ni nada. Desde mi punto de vista un desastre de alguien que pretende hacer algo rápido y no “sabe” que está escribiendo 40 líneas de código en vez de 4. Por esto me negué desde el principio del proyecto de Kill o Heal a mirar el código, ni por morbo. Pero tuve que hacerlo para implementar animaciones, y acabé de alguna forma programando funcionalidades extras, cosa que creó un ambiente hostil entre el grupo de programación y yo, ya que parecía que no aceptaban ningún tipo de ayuda, y que cuando el equipo de arte decidió añadir una nueva zona en el nivel Cerebro, en concreto el ascensor, empezaron reproches estilo “ahora voy a probar por dónde puede fallar” cosa que no se ocuparon de hacer en el juego en general, y a esto yo lo llamo intento de sabotaje, que no funcionó porque apliqué TDD (Desarrollo dirigido por pruebas) así que poco de ese intento de “a ver que te lo rompo” ya estaba más que probado, si lo hicieron, tiempo que perdieron tontamente.

Cuando algo falla, buscan un culpable, simple.

Siempre me tomo las cosas con humor, pero en todo el proceso he pasado momentos de querer matar a alguien, sentirme mal, ver caras de culo en el equipo, y acabar teniendo conversaciones del tipo: que te calles ya. Y lo he sentido mucho, porque poca gente puede hacerme perder mi autocontrol, pero cuando lo consiguen, conmigo no cuentas nunca más. Y esto lo digo públicamente: con estos programadores no volveré a trabajar ni aunque me pagues. Y siempre contando con que fue una segunda oportunidad, siempre desde el punto de la profesionalidad, fuera de este ámbito puede que sean unas personas maravillosas, puede ser, pero tampoco lo sabré.

La sensación personal que me quedó fue bastante a disgusto.

Aunque yo decidiera en este proyecto hacerme a un lado en la programación: primero porque no quería verme en las mismas que el proyecto anterior, arte contaba sólo con dos manitas principiantes, ¡y qué mejor que cuatro!

Cuando decidí meterme en este experto, me matriculé de la doble especialización, por lo que tuve las mismas clases de programación que mis compañeros programadores. Cuando tomé la

decisión de no participar en la programación de mecánicas básicas, pero sí programar otras cosas, me sentí excluida por el equipo de programación cuando en las discusiones me comentan “Tú estás en arte” entre otras cosas bastante infantiles como “mira las notas que le han puesto a arte”. Pues lo siento mis niños, pero yo formo parte de un proyecto, y puedo asignarme tareas de arte y/o programación si me apetece.

Me gusta el arte, pero no es lo mío. Lo tuve claro cuando acabé mis estudios medios en bellas artes. Me apasiona la creatividad, pero la que está en código y la programación.

Intenté hacer un póster para el juego, y me vi con pocos recursos, estaban las ganas que no quepa duda. Me dieron una pequeña clase extra sobre diseño de carteles y sensaciones que quieres transmitir, pero no cuento con tiempo, ni ánimos, para arreglarlo, ni esto ni otras cosas del juego. Porque opino que la ilusión se contagia, y que también se agota si no hay gente esponjosa.

Aprendí en estos seis meses muchísimo: modelado, texturizado, animaciones, temas de luces. Una locura todo lo que he aprendido asentando bases y llevándolo a la práctica, dedicándole más de cinco horas diarias, viendo videotutoriales y trasteando Blender.

Sobre mi aprendizaje del módulo de programación, no he podido ponerlo en práctica, pero es lo que me llevo como profesional y desarrolladora en programación. Pues sigo trabajando como programadora de videojuegos.

Aunque ha sido un proceso duro, que me ha afectado a nivel personal en varias ocasiones, llegando a conflictos que me sirven para mejorar como profesional, a contrastar comentarios de compañeros que te dicen “el tutor dice X cosa sobre tu trabajo” sin darte un contexto previo, a entender que un equipo a veces no es el equipo maravilloso que esperas encontrar (cosa que yo ya no esperaba), me quedo con la experiencia de haber visto a mi compañero desarrollar sus destrezas como artista, su motivación con el proyecto y su creatividad. Los profesores, todos y cada uno de ellos, porque pusieron su granito de arena. Los alumnos del experto, que tendrán su historia que contar; y aquellos que además de alumnos se convirtieron en amigos. Con los tutores del proyecto Aitor Lozano y Luis Antón, por escribirnos seguimiento en cada entrega con comentarios más o menos duros, pero disponibles desde las nueve de la mañana. Y, por último, Agustín Trujillo, coordinador del experto, por escucharme en los momentos más críticos, estar en contra de mis varios intentos de “abandono el curso” y entender mi humor, porque no he conocido a nadie con tanta empatía como la suya.

Puede que estos seis meses hayan sido un castigo, no solo para mí, también para los proyectos que he puesto en pausa, amigos que me han escuchado hablar más mal que bien, pero somos unos supervivientes.

A los que he conocido por el camino o me habéis acompañado un trozo de esta aventura, gracias por el calor. A los que no, gracias por vuestra educación. Y al resto, gracias por nada, supongo. *See you in another life, brotha.*