

THE ROOKIE THIEF

CPZenon Games - Yolanda Cruz Girona & Antonio Ríos Vila

CÓMO SE HIZO



Nosotros sabíamos que ser desarrolladores de videojuegos es una profesión dura, complicada. Sin embargo, también teníamos en cuenta que los resultados al final siempre, de una manera u otra, son satisfactorios. La experiencia que hemos vivido durante este mes desarrollando un juego para la *CPCRetroDev 2018* ha sido, cuanto menos, complicada y enriquecedora a la vez. Desde luego que ha sido algo interesante de relatar. Nosotros somos Yolanda y Antonio, conocidos en este ámbito como CpZenon Games. Somos estudiantes de cuarto de Ingeniería Multimedia en la Universidad de Alicante, y esta es nuestra pequeña historia.

1 – UN RETO Y UNA IDEA

Todo comenzaba el 11 de septiembre en clase de Videojuegos I, una asignatura de nuestra carrera. En ella, el profesor, Francisco Gallego (no creemos que haga falta presentarle en este ámbito), no reparó en rodeos a la hora de presentarnos la actividad que nos ocuparía todo este mes: desarrollar un juego para un Amstrad CPC 464 y presentarlo el 31 de octubre en la *CPC RetroDev 2018*. Desde luego, todo un reto ¡Desarrollar un juego durante poco más de un mes! Para ello se nos proporcionó un posible calendario que era, cuanto menos, intenso.

Además, el juego debía cumplir dos condiciones: estar programado en lenguaje Ensamblador y contener Inteligencia Artificial básica. Es evidente que, aparte de trabajar en el desarrollo de nuestro juego, deberíamos dedicar mucho tiempo al aprendizaje y comprensión del lenguaje Ensamblador del Z80.

Sin embargo, no conviene ir tan deprisa. Un videojuego no puede ser concebido sin una idea previa. Nuestro objetivo durante aquella tarde fue decidir qué íbamos a desarrollar. Tras conversar distintas ideas y propuestas, pensamos en que sería divertido crear un juego similar a *Kung-Fu Master*. A pesar de ello, nos vimos obligados a replantear el concepto de juego, ya que esta referencia carecía de una Inteligencia Artificial interesante.

Al final, tras conversar con el profesor, decidimos crear un videojuego con las mecánicas de *Kung-Fu Master* junto a la exploración e Inteligencia Artificial de la segunda parte de *Navy Moves*.



La idea ya estaba clara y el reto establecido. Desde luego que para ser nuestro primer juego de Amstrad, junto a la poca experiencia de programación en Ensamblador y las limitaciones de la máquina, el reto era considerable. Pero nosotros no teníamos miedo, ¡era hora de desarrollar!

2 – LAS MECÁNICAS DE JUGADOR Y LA NAVEGACIÓN POR EL MAPA

Durante el periodo de aprendizaje decidimos que nuestro juego se trataría de un ladrón que entra a robar en una casa para llevarse todo el dinero y las joyas que pudiese. Los guardias patrullarían por la casa. Si un guardia encontraba al ladrón, éste tendría la oportunidad de huir o combatir. Además, queríamos hacer un combate exigente, es decir, que el jugador solo tuviese una vida y cayera de un solo golpe, como suele ser en la realidad.

Junto a esta propuesta, definimos la siguiente lista de mecánicas que nos interesaba incluir:

- El jugador se movería a la derecha y a la izquierda.
- El jugador puede saltar para evitar a los enemigos.
- El jugador puede golpear a los enemigos.
- El jugador tiene la opción de bloquear los golpes de los enemigos de forma limitada.
- Existirán objetos que son recogidos por el jugador, y le sumarán puntos.

- El jugador podrá desplazarse a través de las múltiples salas que tiene el mapa.
- El jugador podrá utilizar puertas, algunas necesitarán desbloquearse con llaves, para explorar el mapa y huir de sus enemigos.
- El jugador tendrá a su disposición unos armarios con los que podrá esconderse para que sus enemigos dejen de perseguirle.

Lo primero que desarrollamos, tras nuestro periodo de aprendizaje, fueron las mecánicas del jugador y cómo establecer la navegación por el mapa de una forma coherente.

A la primera semana de desarrollo ya disponíamos de un primer prototipo con el jugador y los mapas cargados en memoria. Éste podía desplazarse y navegar por el entorno tal y como habíamos planteado en un principio.

Por otro lado, también conseguimos desarrollar los objetos que el jugador podía recoger para ganar puntuación. Finalmente, conseguimos realizar también durante esa misma semana unas puertas por las que el jugador se podía desplazar y un prototipo de enemigo que perseguía y mataba al ladrón. ¡Desde luego que fue una semana productiva!

Sin embargo, la velocidad con la que desarrollamos todas estas características nos trajo muchos problemas a la hora de implementar nuevas funcionalidades. En concreto, el doble *buffer* para evitar el parpadeo de las imágenes que se desplazaban por el entorno causó un problema de colisión con la pila de programa que nos retrasó cuatro días de nuestros planes. Nos vimos obligados a tirar mucho código durante el proceso. Todo este retraso afectó al desarrollo del videojuego.

3 – ¡LOS GUARDIAS SE ESTÁN ORGANIZANDO!

Como mencionamos antes, nuestro juego debía contar con una Inteligencia Artificial básica que le plantease un reto al jugador. La clave de nuestra Inteligencia Artificial estaría basada en dos puntos principales:

Comportamientos de enemigos

Existen dos tipos de enemigos a los que nos enfrentaremos durante el juego:

- Guardia *Stalker* : Este enemigo no atacará directamente al jugador, sino que se dedicará a perseguirle y mantenerse a una distancia prudencial de éste.
Pasados unos segundos, cambiará de comportamiento a uno más agresivo, en el que atacará al jugador y lo perseguirá por las distintas salas.
- Guardia perseguidor: Este enemigo es el más agresivo, ya que en cuanto detecte al jugador por primera vez, le perseguirá a través de las habitaciones para matarlo.

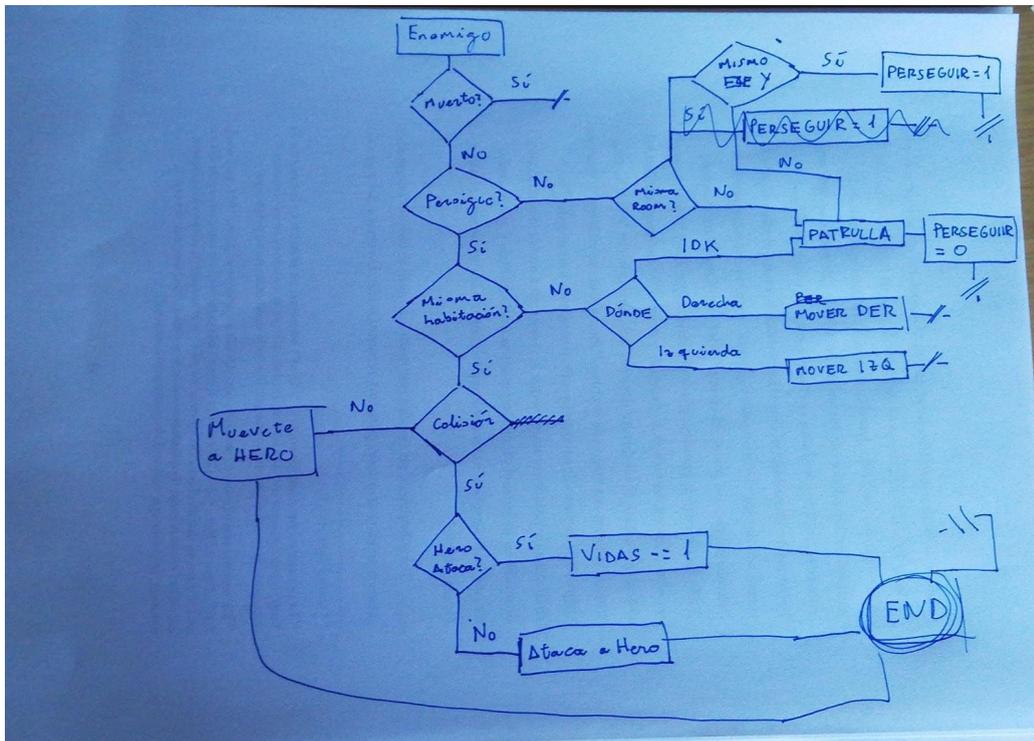
Si no han detectado al jugador, ambos pasarán a un estado de patrulla, en el que se moverán de una sala a otra.

Aproximación a un *AI Director*

Queríamos ofrecer una experiencia de juego más dinámica y diferente al jugador, por lo que decidimos no establecer el número de enemigos presentes en cada habitación desde el principio. Para ello, diseñamos un sistema en el cual el número de enemigos que aparecen en las habitaciones adyacentes al jugador depende de un medidor de “intensidad” de la partida. Con esto, si el jugador se encuentra en una situación comprometida, rodeado de enemigos, el juego no permitirá que aparezcan más antes de que haya acabado con los que ya se encuentran en el juego.

Por otro lado, si el jugador se encuentra en habitaciones vacías durante demasiado tiempo, este medidor bajará, por lo que el sistema hará que aparezcan enemigos en las habitaciones tanto a su derecha como a su izquierda.

Diagrama de clases del comportamiento de la IA:



4 – HORA DE DARLE VIDA A NUESTRO JUEGO: ARTE Y BANDA SONORA

Pese a no haber implementado la mecánica de sigilo con los armarios planteada anteriormente, dimos por finalizado el desarrollo de funcionalidades de nuestro videojuego a una semana y media de la entrega final del producto.

Al ser dos personas de distintas disciplinas, Yolanda tiene experiencia en arte tanto 2D como 3D y Antonio contaba con algunos conocimientos de composición musical y experiencia con programas de diseño sonoro, se dividieron para realizar cada una de las facetas comentadas previamente. Por lo tanto, ahora dividiremos las voces de este documento para que cada uno comente su experiencia:

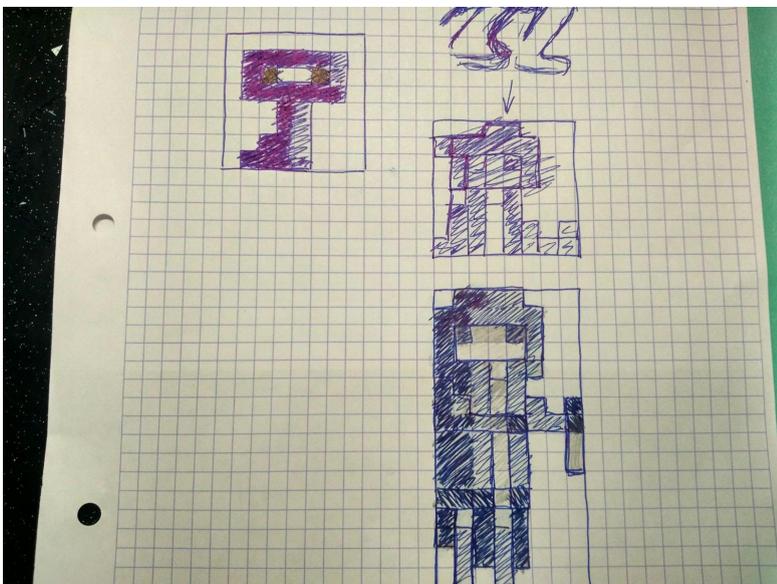
Arte 2D – Yolanda Cruz

Al principio, quisimos darle un estilo retro al juego utilizando la paleta de colores de la Game Boy classic. Sin embargo, ésto nos supuso muchos problemas ya que, en un estado muy avanzado del desarrollo, nos dimos cuenta de que estábamos trabajando en Modo 0. Del mismo modo, las pantallas de los equipos en los que trabajamos tienen configuraciones de color diferentes, por lo que era difícil saber si el juego se estaba viendo bien o no. Una vez lo probamos en un Amstrad CPC, supimos que debíamos cambiar la paleta de colores.

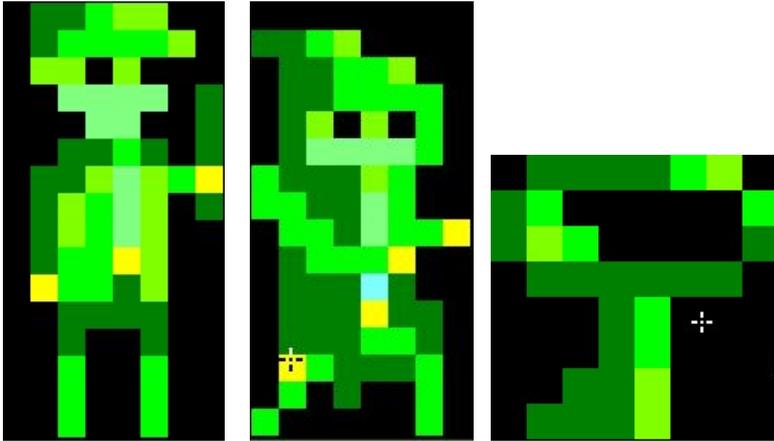
Así, decidimos finalmente decantarnos por una paleta de azules y violetas.

En cuanto a la resolución, siempre he trabajado con resoluciones de 32x32 píxeles o más, por lo que ahorrar memoria y realizar sprites de menos resolución que tuviesen forma y fuesen distinguibles del resto de objetos supuso un reto para mí.

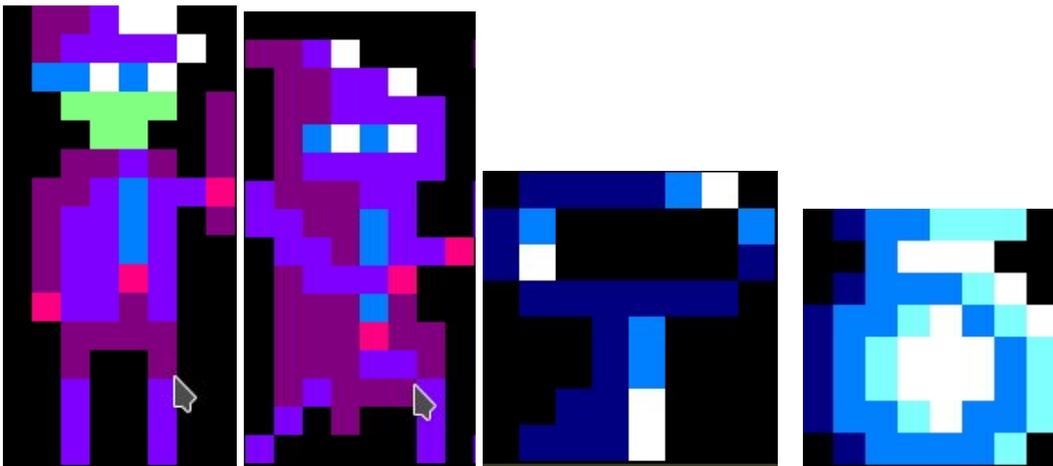
Primeros bocetos de los sprites:



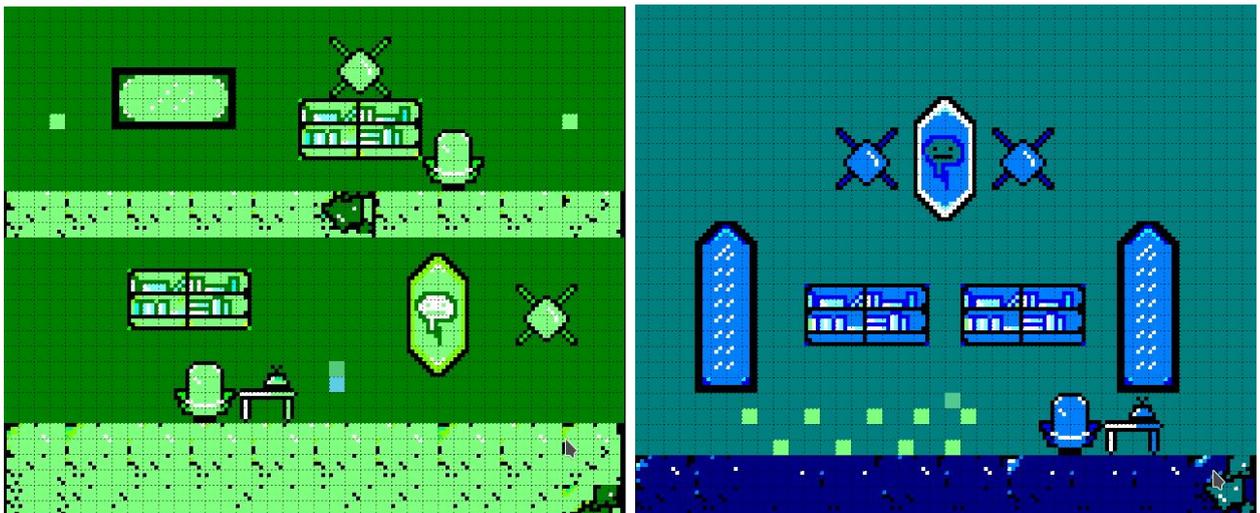
Primera versión de los sprites en verde:



Versión final de los sprites en violeta + azul:



Comparación del acabado inicial y final de los entornos:



Acabado final:



Banda Sonora – Antonio Ríos

Mi experiencia en temas relacionados con la composición musical es nula. Ciertamente tengo algunos conocimientos básicos de solfeo y producción musical. Sin embargo, realizar la banda sonora de nuestro videojuego fue todo un reto.

En primer lugar, me di cuenta que disponía de pocos recursos para realizar las melodías que animarían nuestro juego. Por lo tanto, el objetivo era claro: debía componer una melodía simple y llamativa. Como en todo proceso creativo, uno debe inspirarse y tomar referencias. En mi caso, leí diversas partituras. En concreto, centré mi foco en videojuegos de la época: *Castlevania* y *Mega-Man 2* en particular. También observé reinterpretaciones de juegos retro como *Shovel Knight*. Junto a estas referencias, escribí la primera partitura del juego: *The rookie's challenge*, la cual adjunto a continuación:

The rookie's challenge
The rookie thief main theme Antonio Rios

The image displays a musical score for 'The rookie's challenge' by Antonio Rios. The score is presented in two columns. The left column contains measures 1 through 17, and the right column contains measures 20 through 30. The music is written in 4/4 time with a key signature of one flat (F major or D minor). The notation includes a bass line and a treble line. In the right column, measure 30 features a blue rectangular highlight on the final chord in the treble clef.

En ella, se puede observar que se invierte la lógica y las notas que suenan en clave de Fa componen la melodía. Mientras que las notas en clave de Sol se limitan a hacer un acompañamiento al principio. Todo esto está organizado a propósito para indicar que el juego trata de un personaje oscuro con un reto. Por otro lado, también adjunto las partituras de la melodía que suena en el nivel, la cual intenta representar esa sensación de peligro y esperanza del ladrón para conseguir su objetivo, y la del fin del juego, que es muy simple para darle algún contexto y textura.

Into the house
The Rookie Thief Soundtrack Anto

The image shows a musical score for 'Into the house' from 'The Rookie Thief Soundtrack' by Anto. It consists of five systems of piano accompaniment. The first system starts with a treble clef and a 4/4 time signature. The melody is written in the treble clef, and the bass line is in the bass clef. The score includes measure numbers 6, 11, 15, and 21. The music features a mix of eighth and sixteenth notes, with some triplets and rests.

The Rookie Thief
Game over screen Anto

The image shows a musical score for 'The Rookie Thief' from 'The Rookie Thief Soundtrack' by Anto. It is a single system of piano accompaniment for a game over screen. The score is in 4/4 time and features a melody in the bass clef with a dynamic marking of *mf*. The melody consists of eighth and sixteenth notes, with a key signature of one flat. The score ends with a final chord and a rest.

5 – Un final problemático, pero con orgullo

De momento hemos hablado en gran medida de todas las cosas que, en principio, habían salido bien del videojuego.

Sin embargo, todo tiene una cara oscura.

Evidentemente, con el nivel que tenemos y la poca experiencia en el desarrollo de videojuegos que disponemos, era de esperar que sufriéramos fallos y errores que retrasarían nuestro trabajo. Entre los más destacados están:

- Fallos con la gestión de la **pila**. Perdimos cuatro días de desarrollo debido a este problema y afectó severamente al transcurso del proyecto.
- **Mala gestión de la memoria**, que llevaría al primer problema. Ocupábamos mucho espacio para realizar tareas que, en un principio,

requerían poco. También gastamos mucho tiempo en desarrollar un sistema de compresión y descompresión de recursos para ahorrar el máximo de espacio posible.

- **Fallos de planificación** que, si se trata de tu primer proyecto del estilo, es inevitable que ocurran.

Todos estos problemas condicionaron de manera drástica el desarrollo y entrega final del producto. Sin embargo, creo que en estas tres preguntas podríamos resumir toda nuestra experiencia.

¿Es *The Rookie Thief* todo lo que nosotros queríamos desarrollar? No, ojalá hubiésemos tenido más tiempo para haber pulido detalles que no hemos concretado.

¿El juego está exento de fallos? Para nada, seguramente haya muchos *bugs* que no hayamos conseguido ni siquiera detectar.

¿Estamos orgullosos de lo que hemos hecho? ¡Por supuesto! A pesar de todo lo que hemos sufrido para hacer realidad este pequeño juego, estamos satisfechos con lo que hemos logrado y, desde luego, que el camino solo acaba de comenzar.