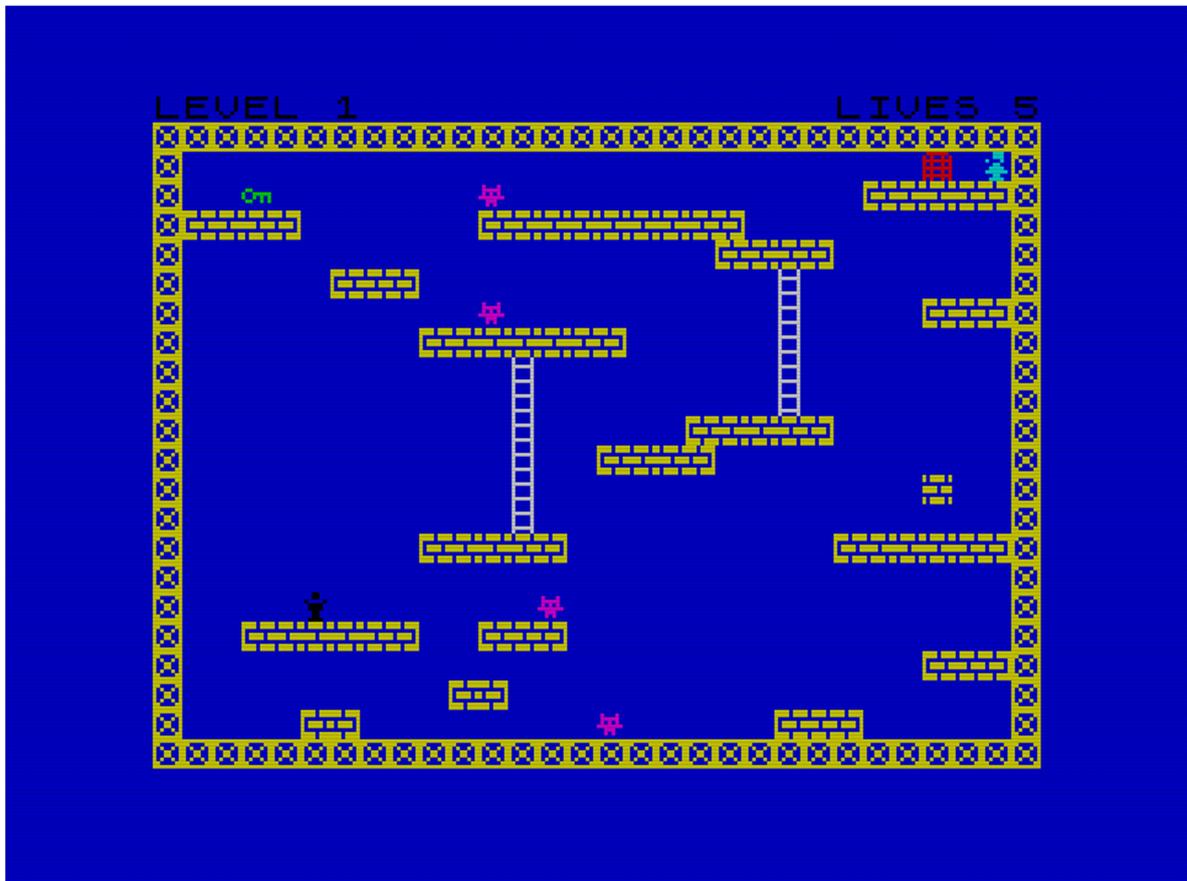


# PRINCESS RESCUE



## Objetivo de juego

El objetivo del juego es rescatar a la princesa a lo largo de nueve niveles que se desarrollan dentro de las estancias del castillo.

Para ello tendremos que recoger distintas llaves que nos permitirán abrir la puerta donde se encuentra presa la princesa.

No todo va a ser tan sencillo, ya que los guardias intentaran por todos los medios que no realicemos nuestra heroica misión. Su contacto con nuestro héroe será mortal de necesidad.

Nuestro héroe tan solo dispondrá de su habilidad de movimientos para esquivar a los guardias.

Podremos saltar en diagonal y movernos a lo largo de cada una de las pantallas para lograr nuestro objetivo.

## Ejecución del juego

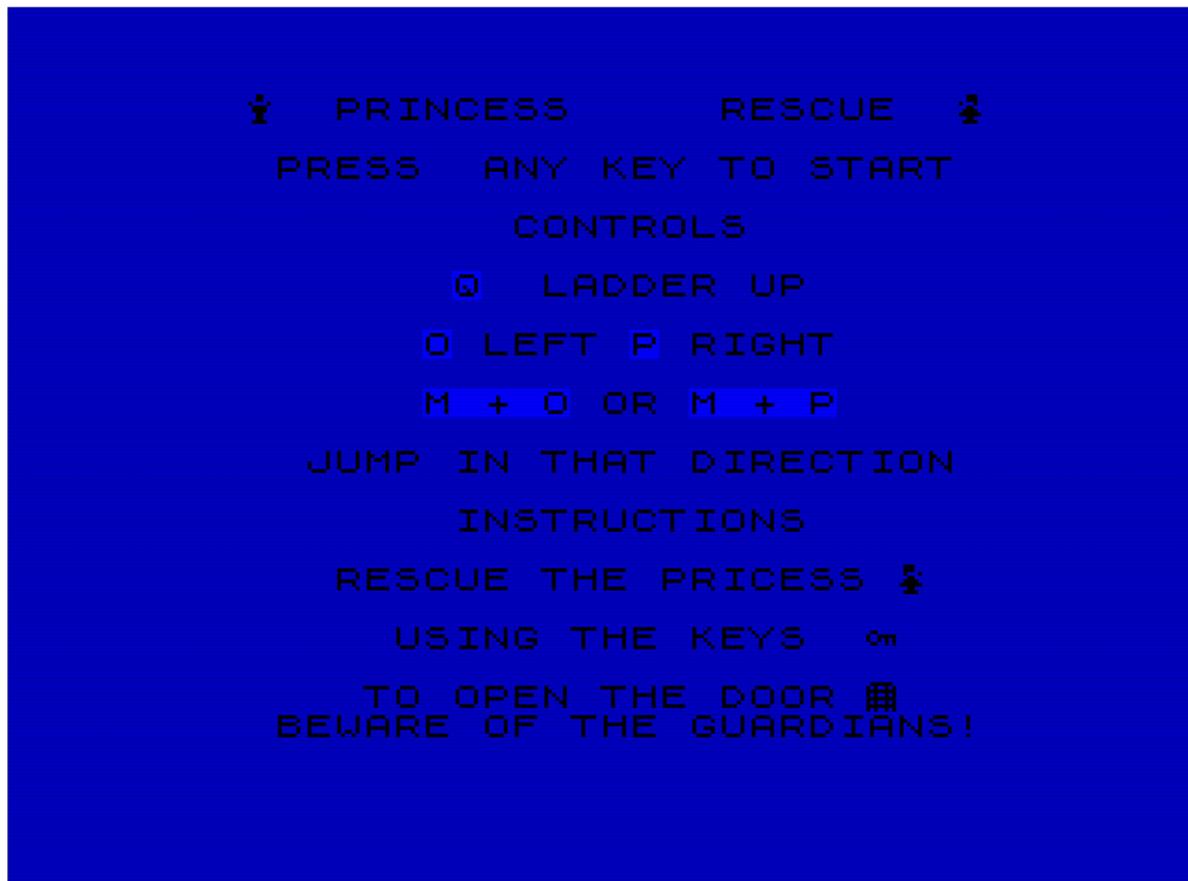
El juego debe de funcionar correctamente para las versiones de 48K y 128K. De hecho ha sido probado tanto en el emulador *Fuse* como en máquinas reales y no se han presentado problemas.

Más en particular ha sido probado en un +2 español y un 48K.

## Motivaciones

Bueno, pues el caso es que con la excusa del concurso, y como nunca había desarrollado ningún juego en Basic para el Spectrum, me puse manos a la obra.

Ha sido un viaje entretenido no exento de frustraciones, pero que me ha permitido aprender muchas cosas que ignoraba de mí amado Spectrum y del compilador de Boriel y sus posibilidades. Toda una pasada y eso que no he podido explotar al máximo su potencia.



## Objetivos de programación

Uno de mis objetivos cuando comencé a darle vueltas al juego, era conseguir un juego con apariencia de aquellos juegos de primera hornada que tanto me fascinaron, pero con dos premisas.

- Que la velocidad de movimiento, aun siendo Basic fuera rápida.
- Que pudiera realizar el diseño de niveles de una forma gráfica desde dentro del propio código.

Ambos objetivos, pienso que se han cumplido con creces, aunque sacrificando bastante memoria en el caso de los niveles, y la velocidad gracias al compilador de Boriel, sino imposible. Nada más que hay que ver a qué velocidad se ejecuta desde Basic interpretado.

Handwritten musical score for "Bar" by Héctor Martínez Echeveste. The score is for a Spectrum game and includes parts for Flute, Crotales, Tambourine, and Harpsichord. It features handwritten annotations such as "114", "DATA 1", "DATA 2", "DATA 3", "DATA 4", "DATA 5", and "DATA 6" with corresponding note numbers. The title "Bar" is written in a decorative font, and the composer's name "Héctor Martínez Echeveste" is at the top right. The score is in 6/8 time and includes a key signature of one flat (Bb).

## Música

Quiero dedicar un apartado especial al tema de la música de inicio del juego, que he realizado transcribiendo con ciertas modificaciones la partitura de "Bar" de Héctor Martínez Echeveste.

Finalmente he visto que mediante el comando Beep, se pueden conseguir cosas bastante aceptables, pero desde luego hay que estar retocando bastante los tiempos de cada una de las notas para adecuarlos a la partitura original. Además, perdemos por su puesto lo de tener más de un canal para el sonido. Pero pienso que finalmente el resultado no ha sido tan malo.

## Sprites

Como el aire del juego tenía que ser de los primeros juegos que salieron para el Spectrum en Basic, no he querido complicar en exceso el diseño de los mismos. Lo que si he querido añadir ha sido animaciones de desplazamiento para el personaje. De todas formas, queda claro que no me voy a ganar la vida como diseñador gráfico.



## Pantalla de carga

La pantalla de carga ha sido realizada mediante el uso de `bmp2scr` para el escenario del castillo y luego posteriormente mediante `SevenUp`, se han dibujado tanto el caballero como la princesa, títulos, etc.

Como ya he mencionado, no creo que me mis virtudes de grafista hagan que me gane la vida con ello.

## Entorno de Desarrollo

Todo el juego ha sido realizado mediante el uso de *Sublime Text* al cual se le ha configurado el Plug-in para poder compilar directamente con *Boriel* con las siguientes opciones:

```
{
  "working_dir": "$file_path",
  "cmd": ["zxbc --sinclair --tap -a $file_name"],
  "selector" : "source.bas",
  "shell": "bash"
}
```

El equipo de desarrollo ha sido un viejo MacBook2,1 con Debian GNU/Linux 10.

La verdad es que bastante contento por darle una segunda vida a tan bello portátil.

## Generación de Tap sin Boriel

Para la generación del fichero Tap intentado que vaya todo lo rápido que pueda, he usado *Zxbasicus* con todas las opciones permitidas para acelerar la ejecución.

Sin embargo, el uso de del array de los enemigos me ha penalizado bastante, pero como el objetivo era usar Boriel, no he buscado ninguna solución para el mismo.

Además el sistema de colisiones y la que me he montado con las dichas escaleras, también influyen en la velocidad de ejecución del juego no compilado.

Comando para zxbasicus:

Linux:

```
wine ./zxbasicus.exe -t --subslvar --shortenv --mergelines --delunusedv --delrem all --alloptim --strlitwchr -i ./rescueprin.bas -o ./basicus.bas
```

Windows:

```
zxbasicus.exe -t --subslvar --shortenv --mergelines --delunusedv --delrem all --alloptim --strlitwchr -i ./rescueprin.bas -o ./basicus.bas
```

## Cosas que se han quedado en el tintero

Pues ha habido varias cosas que finalmente han sido descartadas o no abordadas.

El tema de las escaleras ha sido un quebradero de cabeza, por la forma que me he montado para gestionar las detecciones de Sprites, y de hecho en las primeras versiones se podía bajar por las mismas, pero finalmente se eliminó dicha opción ya que no aportaba nada a nivel de jugabilidad, y me complicaba en exceso la gestión de sprites y detecciones.

También la variedad de enemigos se ha resentido, pero tras ver que más de cuatro enemigos en pantalla ralentizaban el juego, abandone añadir más tipos de enemigos. Quizá hubiese sido interesante añadir enemigos voladores.

Inicialmente el personaje contaba con un salto vertical que finalmente se eliminó en virtud de simplificar la gestión de colisiones y porque no le veía mucha utilidad más allá de poder coger llaves en sitios inaccesibles mediante el salto direccional.

## Jugabilidad y dificultad

Mucho tiempo se ha pasado depurando cada uno de los niveles. Si bien inicialmente la dificultad de los mismos era muy superior a la que finalmente se ha presentado, pienso que es positiva para no desmotivar en exceso al jugador. Hay que ser realista con el tipo de juego y después de usar a mis hijas como betatesters, creo que el nivel es el adecuado para pasar un buen rato e incluso poder terminar el juego sin mucha complicación. Es más, hasta el número de vidas ha tenido que ser aumentado en pos de lo comentado.

## **Generando Tap Final**

Para generar el Tap final simplemente realizamos la copia en orden. Cargador+Pantalla de Carga+Juego.

Para ello tan solo usamos el comando cat en Linux:

```
cat Loader.tap screen.tap rescueprin.tap > Princess_Rescue.tap
```

En Windows podríamos hacer algo similar con type:

```
type Loader.tap screen.tap rescueprin.tap > Princess_Rescue.tap
```

## **Conclusiones**

Pues para una primera aproximación, finalmente ha quedado más o menos decente el resultado. Si bien hay muchas cosas que cambiaría en un proyecto futuro, pienso que el objetivo marcado ha sido alcanzado.

Gracias por probar el juego y si has llegado hasta aquí, gracias también por leer mis reflexiones casi más filosóficas que técnicas.

¡Espero verte en el próximo juego!

RADICAL COMPUTERS COMPANY

BY NEXTRIC