

## ¿Y qué es realmente Funny Box?

Yo prefiero siempre hacer distinción entre lo que es Funny Box como **juego** y lo que es, o quiere ser, como **proyecto**.

Si hablamos de él como juego, no estaremos refiriendo principalmente a lo que vemos cuando hacemos doble clic sobre el archivo ejecutable de la aplicación. Pero si hablamos de Funny Box como proyecto hacemos referencia a los planes de futuro que para él se han planeado, todo aquello que de momento no es, pero que el tiempo y la ayuda de los futuros implicados en el proyecto puede conseguir. Este aspecto, al contrario que el anterior, se podría decir que ya no depende de mí, al menos no en su totalidad. Tendrá mucho que ver la acogida que este proyecto tenga entre las diversas comunidades de programadores/desarrolladores, no solamente la de Fénix (lenguaje de inicio del proyecto), si no la de cualquier comunidad afín a determinado lenguaje de programación que desee versionar la aplicación en su lenguaje preferido. Podría decirse que la consideración como proyecto es algo menos tangible y para representar el proyecto está la aplicación o el juego.

Hecha esta distinción empezaré definiendo al concepto que quizá resulte menos abstracto:

### ¿Qué es Funny Box como juego?

Simplificando, este juego es un *tres en raya*. Le he añadido una dimensión más y ha surgido lo que podríamos llamar **un tres en raya 3D**. Es muy posible que nunca antes hayáis visto algo así, yo no lo había visto antes. Sin embargo no es tan extraño ni imaginárselo ni la manera de jugar. La novedad principal es que ya no tenemos las típicas nueve casillas, ahora tenemos 27 y además, podemos ganar alineando las fichas en cualquier dirección posible: horizontal, vertical, diagonal, pasando por el centro del cubo, etc. Ahora serán muchísimas las maneras de ganar la partida, y muy pocas las partidas que acaban en tablas.

El *tradicional tres en raya* o también llamado *tic-tac-toe*, tiene un gran problema en el hecho de que normalmente las partidas terminan en tablas si los dos jugadores saben jugar y no cometen un descuido. Este hecho hace que jugar contra una máquina (que no comete descuidos) sea muy aburrido.

Todo lo contrario ocurre con este juego, de todas las partidas que he jugado, las partidas que han acabado en tablas pueden contarse con los dedos de las manos. Además, no es posible crear una inteligencia artificial para la máquina que le asegure la victoria o la no derrota. Y si la descubris estaré encantado en escucharos, pero aviso que computacionalmente, en jerga de IA, este problema es *NP-Completo*, en otras palabras, imposible de resolver con los habituales árboles de búsqueda, no se puede resolver en un tiempo razonable (no queremos esperar el movimiento de la máquina más que unos pocos segundos).

Otra importante novedad es que no serán dos los jugadores que se disputen la partida, serán 3. Esto aparte de complicar los cálculos para la máquina que calcule las mejores jugadas, también añade más dificultad de cara al jugador que tendrá que evitar que sus dos rivales no consigan su objetivo. Conseguir que ellos no logren ganar la partida te dificultará las cosas mucho para alinear tus fichas y ganar tú la partida. Es más, cuando estés amenazando con ganar al siguiente movimiento, los rivales tendrán dos oportunidades de bloquear tu amenaza o amenazas. Es necesario que este juego sea para tres jugadores para equilibrar la balanza, ya que con tan solo dos jugadores el cubo se llenaría de casillas de uno u otro jugador y prácticamente eligiendo una casilla al azar en mitad del juego podrías sin darte cuenta ganar la partida sin que el otro jugador pudiera evitarlo.

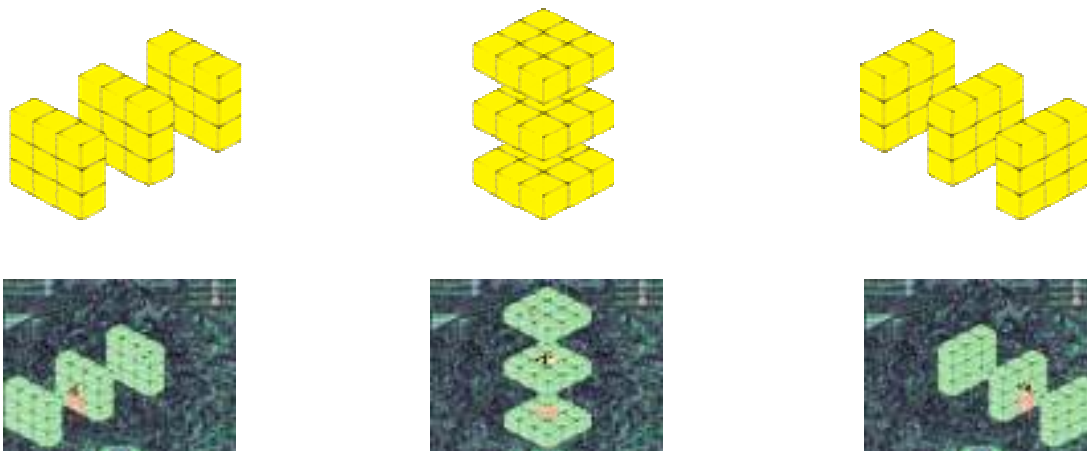
Cada partida forma parte de un torneo a x rondas, al llegar al número máximo de rondas el jugador que más victorias haya conseguido será el vencedor y se podrá comenzar un nuevo torneo. De esta manera se pueden jugar tandas de partidas en lugar de una y volver a comenzar otra.

Los que tengan en mente el cubo de juego se estarán haciendo la siguiente pregunta: ¿Y cómo vemos o seleccionamos las casillas que quedan ocultas en la parte no visible del cubo?, es normal hacerse esa lógica pregunta, yo me la hice durante un tiempo y barajaba dos soluciones:

- *Hacer el cubo en 3D real y rotarlo para mostrar la cara deseada.* Pero esto no era una solución porque la casilla del centro del cubo seguiría sin ser visible por muchas vueltas que le diéramos. Habría que resolver esta cuestión usando transparencias para visionar el interior. Los que conocen fénix saben que se puede hacer con librerías 3D elementales.
- *Construir un cubo desmontable que pueda expandirse en cualquier dirección.* Creación del 'tablero' constituido por entidades independientes que pueden desplazarse teniendo en cuenta su posición global, representados por pequeños cubos en perspectiva isométrica. Expandiendo el cubo de manera que todas las casillas queden visibles. El problema de la visibilidad, sobre todo de la casilla central del cubo, queda así resuelto.

Opté finalmente por la segunda opción, que de cara al jugador tan solo requiere un poco de visión espacial para imaginarse el cubo replegado, enseguida logras acostumbrarte.

Estos son los gráficos correspondientes a las tres formas de expansión, en amarillo los bocetos, los otros gráficos son las capturas de cómo ha quedado finalmente.



Con el ratón se selecciona la casilla a marcar. Si tras pulsar el ratón sobre una casilla decidimos no hacer ese movimiento finalmente, podemos soltar el ratón fuera de la casilla y no se seleccionará.

El apartado gráfico ha sido todo un reto para mí. Nunca antes había creado yo la totalidad de los gráficos en uno de mis juegos. En este aspecto creo que he sabido defenderme bien. Aun así, esta implementación está abierta a nuevos aportes gráficos y remodelación parcial o total de los gráficos por parte de futuros colaboradores.

El proyecto se compone de **dos modos de juego**:

- *Juego Multijugador*: modo de juego para tres personas desde distintos ordenadores. Uno de los tres jugadores actuará de servidor y creará una partida en su dirección IP, éste comunicará su dirección IP al resto de jugadores (mediante algún programa de mensajería instantánea) que se unirán a la partida introduciendo la dirección de la dirección IP del jugador servidor. Una vez que los dos jugadores se hayan conectado a la partida creada dará comienzo la partida.  
La comunicación entre los tres jugadores se apoyará en el protocolo de transporte de paquetes TCP/IP, que ofrece confiabilidad en las transmisiones.
- *Juego Individual*: modo de juego para un solo jugador, no se necesita conexión a Internet como en el modo de juego anterior. Este modo te enfrenta a la máquina, que controlará al jugador 2 y al jugador 3, cada uno de los jugadores, en este caso virtuales, irá contra el resto de jugadores y no se aliarán contra el jugador humano. Este modo de juego puede servir de entrenamiento para jugar contra jugadores humanos, también permite jugar pese a que no dispongas de conexión a Internet o no consigas reunir a dos personas para jugar. La cuidada inteligencia artificial pondrá las cosas muy difíciles e incluso sin haber cometido un descuido aparente, podrás encontrarte en una situación perdedora que la cpu sabrá aprovechar. Pese a las limitaciones inherentes de crear un algoritmo solvente para jugar a este juego, se ha creado una implementación usando árboles de búsqueda con profundidad limitada que sin acaparar en exceso al procesador resulta altamente competitivo en su opción de dificultad '*Normal*'. En este aspecto, la IA, espero que se animen colaboradores a ofrecer soluciones mejores a la existente, aunque sea en pseudocódigo o simplemente ideas.

En opciones, aparte de cambiar de pantalla completa a modo ventana, también es posible modificar la inteligencia artificial por la que se guiará la máquina. Es decir, podemos decidir cuales son el tipo de algoritmos que utiliza el programa cuando juega contra nosotros.

**Fácil**: algoritmo completamente aleatorio. Está a modo de ejemplo de algoritmo trivial. Lo común es ganar con el tercero de nuestros movimientos.

**Normal**: búsqueda en anchura a profundidad 1. (opción recomendada)

**Difícil**: todavía por implementar. Se reserva esta opción por si surgiera una nueva implementación que haga a la cpu casi invencible.



Profundicemos un poco más, **las reglas de juego o sistema de juego** es el siguiente:

Empieza la partida uno de los tres jugadores aleatoriamente. Esta determinación aleatoria del turno lo hace el servidor y lo envía a los demás jugadores en el caso de tratarse del modo multijugador. Aquel que comienza la partida podrá escoger la casilla central lo que le da una evidente ventaja.

No obstante, tendrá al elegir esta casilla un pequeño inconveniente porque de acabar la partida en empate el jugador que partía con ventaja será el perdedor y los otros dos jugadores, pese a no haber hecho tres en raya se proclamarán ambos vencedores de la partida.

Es decir, ocupar la casilla central te da una gran ventaja pero conlleva la responsabilidad de saber aprovechar esta ventaja ganando la partida haciendo tres en raya. El resto de jugadores se sentirán más estresados intentando que el jugador de la casilla del centro (o su otro rival) no haga tres en raya, pero si consiguen aguantar hasta el final habrán ganado ambos la partida.

El jugador o jugadores que hayan ganado la partida sumarán un punto, el resto de jugadores no sumarán puntos. Cada partida será realmente parte de un juego, llamémoslo torneo, en el que hay que alcanzar un número determinado de puntos, aquel que logre alcanzarlos primero será el ganador de ese "juego a x puntos", pudiéndose también producirse un empate. En una sola partida podría realmente ganar cualquiera, con un juego a x partidas hay más posibilidades de que sea el jugador de más valía el que se alce con la victoria final.

*Detalles a mencionar:*

- Tras cada jugada efectuada por cualquier jugador, se muestra con un reborde amarillo ese último movimiento para saber siempre cual ha sido la última jugada de los rivales.
- En cada movimiento que la cpu calcula aparece información de que se está realizando ese cálculo. Realmente el tiempo de aparición del mensaje es bastante mayor que el tiempo que realmente está utilizando para calcular la mejor jugada. En el caso de ser necesario por introducir cálculos más complejos de IA, se puede alargar fácilmente este tiempo, incluso se puede poner un contador. (para colaboradores)
- Pulsando sobre el botón de ayuda en la parte inferior central de la pantalla aparece información de utilidad para aquellos que jueguen por primera vez. A lo lado izquierdo y derecho de este botón aparecen también mensajes que orientan a los novatos.
- En ningún caso se permite el "reenganche" de un jugador que ha abandonado la partida en curso en el modo multijugador. En ese caso, se deberá volver a crear la partida.
- Ejecutando simultáneamente tres veces la aplicación es posible jugar una partida multijugador en el mismo equipo. Tan solo hay que crear la partida con una de las tres instancias del programa y con el resto unirse a la partida mediante esa dirección IP o la dirección por defecto: **127.0.0.1** que es una dirección de pruebas que actúa como si fuéramos *LocalHost*.
- Las pruebas del modo online se han probado de la manera descrita en el punto anterior. Aunque en teoría no debería de haber problemas, no lo puedo afirmar con seguridad, así que si descubren errores importantes de conexión o comunicación en el modo online ruego que me lo comuniquen.

## ¿Y qué es Funny Box como **proyecto**?

Lo que ha continuación detallo podría llamarse una declaración de intenciones respecto a la continuidad del proyecto. Son una serie de principios y normas generales en torno a las que propongo que giren las acciones y decisiones que se tomen en relación al desarrollo. En caso de que alguna cláusula no es muy acertada se admitirá cualquier sugerencia de rectificación o retoque.

1. El proyecto FunnyBox no debe vincular exclusivamente su desarrollo a una única implementación o lenguaje de programación, así como tampoco lo hará hacia un sistema operativo exclusivo, deberá buscarse la máxima compatibilidad posible entre SS.OO..
2. En su primera versión se ha empleado Fénix como también podría haberse empleado casi cualquier otro lenguaje. Sería una gran noticia el que aparecieran nuevas versiones de la aplicación (que no del proyecto, ya que realmente formaría parte del mismo proyecto, pues éste apoyará siempre la aparición de nuevas implementaciones) apoyándose en otros lenguajes distintos a Fénix.
3. Basándose en este proyecto podrían aparecer otras implementaciones en otros lenguajes, pero de no aparecer implementación alguna no supondría un problema pues se continuaría mejorando la versión en Fénix igualmente.
4. Es preferible que estas versiones que pudieran aparecer sean coordinadas por comunidades de usuarios, o por usuarios que quieran implicarse a un nivel importante. La implementación en un nuevo lenguaje no dependerá de la implementación que ya exista en otro lenguaje La implementación y coordinación puede delegar en diversas comunidades, asociaciones, agrupaciones, particulares, etc.
  - o Por ejemplo, si surgiera una versión en Java, el autor o autores podrían ser los coordinadores de la implementación en Java del proyecto FunnyBox. Serían los responsables de esa versión, encargados de su mantenimiento, su adecuación a los principios del proyecto y de fomentar su desarrollo lo mejor posible, entre otras tareas.
5. La aplicación deberá ser software libre y permitir colaboración de los desarrolladores que estén interesados, de lo contrario no se considerará esa aplicación parte del proyecto FunnyBox.
6. No es obligatorio emplear software libre para futuras implementaciones pese a que lo que se obtenga sí que ha de ser software libre. Obligar a usar software libre para implementar futuras versiones sería discriminar a los demás lenguajes y no sería justo.

Cualquier aplicación paralela a la/s ya creada/s deberá implementarse teniendo en cuenta una serie de importantes directrices.

- o **Se deberá conseguir la compatibilidad entre las aplicaciones del proyecto FunnyBox.** Será necesario que un usuario que utilice una versión en diferente lenguaje a otras versiones pueda jugar con los usuarios con aplicaciones de distinta implementación.  
Ejemplo: Jugador 1: aplicación Java  
Jugador 2: aplicación Fénix  
Jugador 3: aplicación Flash

Esto no es difícil de conseguir usando las mismas librerías TCP/IP o compatibles, en general enviar una jugada (un paquete IP), esperarla, recibirla, etc, es muy parecido en muchos lenguajes.

La nueva aplicación deberá adaptarse a las ya existentes, pero si se justifica la razón se podrá pedir que algún aspecto de la aplicación ya existente se rediseñe parcialmente para

un mejor funcionamiento o mayor funcionalidad. Los responsables de una implementación siempre intentarán ayudar a los del resto de implementaciones, sobre todo en esta tarea.

- Las aplicaciones resultantes no tienen por que ser “calcos” de ninguna versión, será muy positivo que cada versión intente mejorar aspectos que otras implementaciones no han conseguido abordar del todo, o que aporten mejoras para hacer que la versión en su lenguaje elegido sea la mejor, mejoras que perfectamente podrán posteriormente ser introducidas en las implementaciones ya existentes. Pero he de insistir en que la compatibilidad, sobre todo en el modo multijugador, debe de mantenerse. En el caso de que una implementación tan solo ofrezca jugar de modo individual y no multijugador, se aceptará su inclusión al proyecto FunnyBox puesto que no hay compatibilidad posible al no haber modo multijugador.
- **Se debe lograr que la aplicación funcione en cualquier sistema operativo.** Si finalmente no se consigue que no sea por no haberlo intentado, se deberá procurar la máxima compatibilidad entre sistemas operativos.
- Aquellos que se encarguen del proyecto bajo determinado lenguaje no tienen por qué saber programar en el resto de implementaciones existentes, por ello **se fomentará la cooperación entre comunidades de programadores** para explicar a otros los algoritmos escritos en un lenguaje y que deben de trasladarse a otro.

Después de definir los principios que serán la base fundamental del proyecto, vamos a detallar de qué formas se puede colaborar en FunnyBox. Nos referimos a continuación a la versión Fénix por estar ya implementada, pero todo es extensible a otras versiones.

- **Remodelación gráfica:** el aporte gráfico que pueden prestar va desde la sustitución de las 4 posibles fichas o casillas del cubo, los fondos, los botones del menú, los mensajes emergentes, etc. Puede plantearse como si se tratara de algo similar a los típicos “skins”, es decir, se puede cargar un lote de gráficos que sin eliminar los existentes pueden activarse en “opciones”. En la versión Fénix tan sólo hay que guardar los gráficos, respetando completamente las dimensiones, en otro fpg (fichero para gráficos) conservando el número de gráfico dentro del fpg para así cargar un lote de gráficos con solo seleccionar el fichero de gráficos a utilizar.
- **Mejora en IA:** programadores con conocimientos avanzados en inteligencia artificial pueden aportar nuevos algoritmos de búsqueda en árboles, heurísticas, análisis teóricos sobre posibles soluciones, o quizá una brillante solución a este problema cuya alta dificultad quedó latente en el documento “Análisis de complejidad”.
- **Mejora en código fuente:** puede ser retocado el código de cara a que quede más claro para los demás programadores. Puede también ser depurado para lograr más rapidez en la ejecución o una mejor interactividad con el jugador, corrección de posibles bugs, etc.
- **Mejora en modo Multijugador:** cualquier aporte que ayude al control de flujo y errores entre los mensajes que se envían los ordenadores que juegan una partida. Esta es la primera vez que uso una librería de comunicación TCP/IP y sin duda, habrá mucho que mejorar.
- **Remodelación del interfaz de menú:** cambiar parcial o totalmente el menú, por estética, usabilidad, ampliación de opciones, etc. Cualquier cambio que no sea puramente gráfico y que afecte al menú.

- **Desarrollo y mantenimiento de documentación del proyecto:** siendo un proyecto colaborativo será de gran importancia tanto crear documentos que ayuden a orientar a nuevos colaboradores como mantener actualizados los documentos tras cualquier cambio relevante dentro del proyecto. Si cada uno hace una determinada tarea y no explica lo que ha hecho, como lo ha hecho y como se usa, difícilmente podrán coordinarse debidamente todas las aportaciones que surjan.
- **Mantenimiento de la forja del proyecto:** esta forja se encuentra en <https://forja.rediris.es/projects/csl2-funnybox/> y servirá de gran ayuda para la administración del proyecto. Por su planificación de tareas, foro, control de versiones, subida/bajada de archivos, etc.
- **Creación/mantenimiento de una web para el proyecto:** estaría bien que aparte de la forja contáramos con una web en exclusiva para el proyecto FunnyBox, tanto para los involucrados con el rol de colaboradores como para aquellos que únicamente jueguen en cualquiera de las versiones existentes. Sería el portal oficial, que englobaría todas las implementaciones existentes y haría de nexo de unión para todos los que tengan relación en este proyecto. Podrían ser muchas las funcionalidades de este portal, ya se debatirán detenidamente mas adelante.
- **Aportaciones de jugabilidad:** nuevas funcionalidades que ayuden a hacer la experiencia de juego más divertida o gratificante. Por ejemplo, actualmente solo puede jugar una única persona o bien tres personas, pero no es posible por el momento comenzar una partida dos jugadores humanos y la cpu. Otro ejemplo de aporte perteneciente a esta categoría sería crear un ranking global para todas las partidas que se hayan jugado desde la instalación en la maquina. Tan sólo son un par de ideas de entre todas las propuestas que pudieran surgir. El sistema de juego actual fue debidamente meditado desde su concepción y por ello no deberá variar drásticamente tras cualquier aporte.

Con cualquier aportación que se desee hacer será altamente recomendable poner en conocimiento de la comunidad ese aporte que se desea hacer. De esa manera todos los demás estarán al corriente y no tendremos por ejemplo a varias personas programando exactamente la misma funcionalidad sin que lo sepan entre ellos. También servirá para hacer recomendaciones y sugerencias al programador que desee contribuir de esa manera.

Con esta parte relativa a la visión de FunnyBox como proyecto no quería más que por el momento dar a conocer los objetivos perseguidos. Ya he detallado muchas cosas y creo que no es momento de extenderse más con un proyecto que en el momento en que se escriben estas líneas aun no ha salido a la luz.

Jose Luis López Sánchez  
Coordinador del proyecto Funny Box