

ETSID

# Clip-Factory

Informática Industrial y Comunicaciones.

Antonio Arco Torres | John Jairo Marín | Daniel Rivero Paz

11-6-2019

## INTRODUCCIÓN

---

El juego que se ha desarrollado consiste en un tablero de 5 por 5 casillas en la fase inicial, que representa la planta de una fábrica. En las distintas casillas se pueden colocar diferentes tipos de máquinas con la intención de que el jugador actúe como el gerente de una fábrica de clips.

Hay tres tipos de máquinas, las cuales se explicarán en detalle más adelante, que controlan la producción de clips, la obtención de materia prima para fabricar los mismos y el precio de venta de los clips.

**El objetivo del juego es conseguir que la fábrica sea viable, optimizando la producción de clips lo máximo posible combinando las distintas máquinas, sin llegar a alcanzar la bancarrota.**

## INSTRUCCIONES

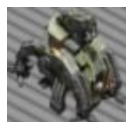
---

Al abrir el juego nos encontramos en la pantalla de inicio, podremos elegir entre empezar una partida nueva o cargar una anterior. El juego guarda la partida automáticamente con cada cambio de día, por lo que no será necesario que nos encarguemos de guardar manualmente.

Una vez estamos dentro de una partida, nos encontraremos en el tablero, a la derecha tenemos la información relevante de nuestra fábrica:

- **Fondos:** El capital del que disponemos para comprar y mejorar máquinas.
- **Producción:** La cantidad de clips que se producen al día.
- **Precio:** La cantidad que se nos paga por clip vendido.
- **Prod. Alambre:** la cantidad de alambre que producimos, necesaria para que las máquinas encargadas de fabricar clips puedan funcionar.
- **Cons. Alambre:** la cantidad de alambre que consumen nuestras máquinas en total.
- **Gastos:** Todas las máquinas tienen un coste diario, este dato es la suma de todos los costes.
- **Beneficio:** Los beneficios que se obtienen por día.
- **Días:** Los días que han transcurrido desde que se empezó a gestionar la fábrica.

A la izquierda del tablero nos encontramos las tres máquinas de las que podemos disponer:



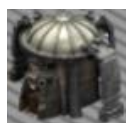
**Trefiladora**

Es la encargada de producir alambre.  
Consume dinero.



**AutoClipper**

Es la encargada de producir clips.  
Consume dinero y alambre.



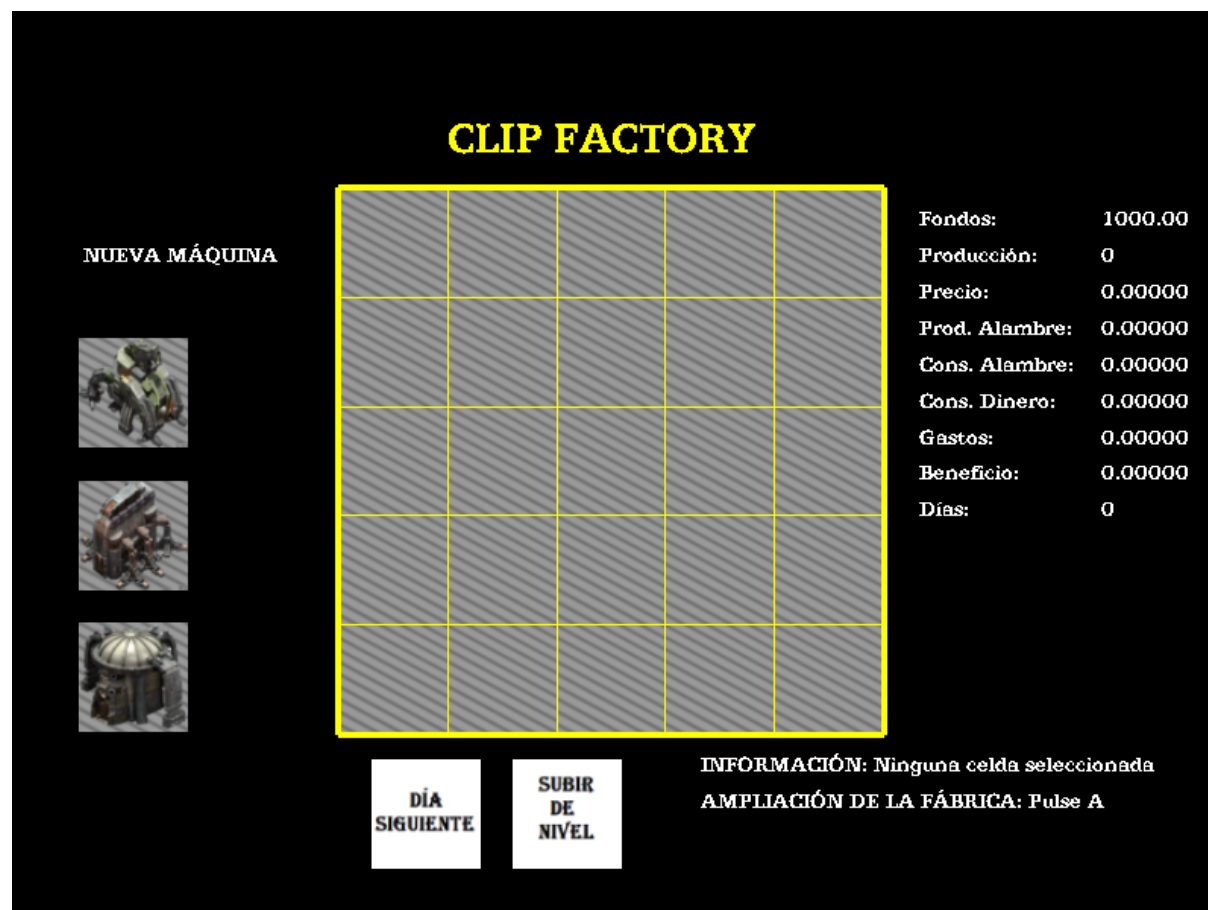
**Marketing**

Controlan el precio de venta de los clips.  
Consume dinero.

Debajo del tablero encontraremos dos botones, uno para subir de nivel las maquinas, lo que mejorara sus propiedades y otro para pasar de día. Siempre que se ejecute una acción, ya sea comprar una maquina nueva o mejorar una ya existente, pasara un día en el juego.

Para comprar una maquina nueva debemos seleccionar una casilla vacía y a continuación la maquina que se quiere crear.

Para subir una máquina de nivel primero seleccionamos la máquina que se quiere mejorar y a continuación se selecciona el botón subir de nivel.



## CÓDIGO

Para desarrollar este juego se han desarrollado clases propias además de usar la clase gltools para el funcionamiento de la parte gráfica, las clases que se han desarrollado han sido las siguientes:

### PIEZA:

Es una clase abstracta que sirve como estructura para las maquinas del juego, que tiene cada una clase propia dedicada. Esta clase contiene las variables básicas comunes a cada máquina. También cuenta con datos que serán propios de cada y que será el constructor de la maquina el encargado de rellenarlo.

### AUTOCLIPPER:

Esta clase es una herencia publica de pieza, será la encargada de gestionar los consumos y aportes de los autoclippers así como de su impresión. Gracias al uso de virtual podremos imprimirla.

### MARKETING:

Esta clase determina el comportamiento de los diferentes parámetros de esta máquina, al igual que la clase anterior para los AutoClipper.

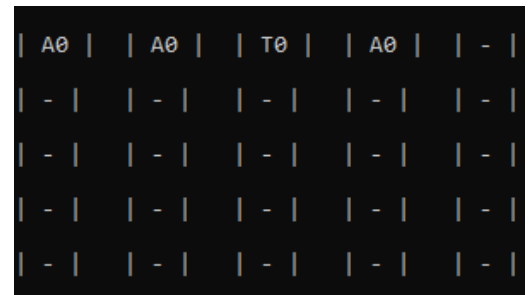
### TREFILADORA:

Al igual que las dos clases anteriores, determina los parámetros necesarios para definir estas máquinas.

### FABRICA:

Esta es la clase en la que se contiene toda la fábrica, en ella se declaran todas las variables que componen el juego como los fondos que hay disponibles, la cantidad de alambre que se produce y todos los parámetros que se han explicado en las instrucciones.

La planta de la fabrica estará representada **por una matriz dinámica**, lo que nos permitirá actualizar la fabrica y aumentar su tamaño. A la izquierda podemos ver una imagen de la matriz en el modo consola del juego usado para la depuración del código.



A0	A0	T0	A0	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-

Los métodos más importantes con los que cuenta esta clase son:

**UpdateFabric**, que realiza la función que se encarga de leer las celdas una a una para calcular los costes totales de la fabrica en ese momento, así como la cantidad de alambre que se produce y todas las demás variables. Una vez calculados todos esos valores actualiza en juego y aparecen los datos en la pantalla.

**lightsOn**, llamando a este método el juego crea una matriz dinámica que será la base de la fábrica, es decir, el tablero, inicializando todas las casillas sin ninguna máquina.

**CambiaTamaño**, sirve para darle al jugador la posibilidad de ampliar la fábrica, inicializando esta nueva matriz más grande de la misma manera que el método anterior.

Para añadir nuevas máquinas en el tablero existe el método **new\_maquina**, al que se llama cuando el jugador quiere añadir una máquina nueva. Este método lee el tipo de máquina que se quiere añadir y donde y a continuación modifica la matriz. En el caso de que la función devuelva un 0 algo no ha funcionado bien.

**UpdateMachine** se encarga de aumentar el nivel de una máquina cuando el jugador lo solicita.

**imprimirEstado** y **Control** se utilizan cuando se juega utilizando la consola de Windows y no la parte gráfica. El primero se encarga de imprimir los parámetros que en la parte gráfica se encuentran a la derecha del tablero y el otro de leer la entrada por el teclado.

**Update\_dias** se encarga de hacer pasar un día en el mundo de nuestro juego.

**SaveGame** se encarga de guardar los parámetros relevantes de nuestra partida en un archivo CSV, posteriormente, el método **LoadGame**, se encargará de cargar esta partida salvada para poder reanudar un juego anterior. Como mejora se podría codificar este CSV con una llave incluida en el código, ya que el usuario no tendrá acceso al mismo. El objetivo de codificar el CSV es que este no pueda hacer trampas y modificar el archivo que contiene la partida guardada para aumentar sus fondos o añadir máquinas sin seguir los parámetros del juego.

El .csv que se usa tiene la siguiente estructura interna.

	A	B	C	D	E	F
1	-	-	-	-	-	-
2	-	-	T0	-	-	-
3	A0	-	T0	-	-	-
4	-	M2	-	-	-	-
5	-	-	-	-	-	-
6	-	-	-	-	-	-
7	f	-813				
8	d	21				

## COORDINADOR:

Esta clase es la máquina de estados del juego, se encargará de decidir si se tiene que mostrar la pantalla de inicio, si se está jugando o si se ha terminado la partida. Usando la librería ETSIDI.h se consigue añadir gráficos y sonido de una manera sencilla.

Cuenta con el método **Dibuja**, que será el encargado de dibujar los gráficos de nuestro juego.

También se encarga de controlar las entradas del usuario con los métodos **MouseButton** y **KeyDown**, que se encargaran de registrar las acciones que el usuario realiza con el ratón y el teclado respectivamente.

## TABLERO:

Será la encargada de gestionar lo que se muestra en pantalla y la interacción del usuario con ello. Dibuja la fábrica y los datos necesarios para que el jugador pueda tomar decisiones. Para ello dispone de métodos que calculan coordenadas y tabulan correctamente los datos. También, por ejemplo, dispone de un método que indica qué casilla se ha seleccionado con el ratón. Contiene los métodos que gestionan la interacción mediante el ratón y el teclado cuando el estado es el de "JUEGO". Se sirve de

los métodos de Fábrica, ya que el tablero es solo la representación de esta, y lo que se modifica al interactuar con el tablero es la fábrica.

Por ultimo como punto de interés gustaría mencionar que el juego cuenta con una interfaz de consola

```
| A0 | | A0 | | A0 | | A0 | | - |
| M0 | | - | | - | | - | | - |
| T0 | | - | | - | | - | | - |
| - | | - | | - | | - | | - |
| - | | - | | - | | - | | - |

Fondos: 780
Produccion: 2000
Precio: 0.01
Produccion alambre: 100
Consumo alambre: 40
Consumo dinero: 60
Gastos: 0
Beneficios: 0
Dias transcurridos: 3
JuegoGuardado
```