

ANEXO III

**ANÁLISIS SOBRE ALUMBRADO
(LIGHTS)**

EN ESPAÑOL

**PARA
TRAIN SIMULATOR**

SEPTIEMBRE 2004

VERSION 1.0

INDICE

1	Anexo III - Análisis sobre alumbrado
2	Lights
2	Light
2	Type
3	Conditions
4	Headlight
5	Unit
5	Penalty
6	Control
6	Service
6	TimeOfDay
6	Weather
7	Coupling
7	FadeIn
7	FadeOut
7	Cycle
8	States
8	State
8	Duration
8	LightColour
9	Position
9	Azimuth
10	Elevation
10	Angle
11	Transition
11	Radius
12	Ejemplos
12	Ejemplo 1: Corrección alumbrado Dash 9
19	Ejemplo 2: Alumbrado "Bidireccional"
20	Ejemplo 3: Luz "Mars" en parada de emergencia "Penalty"
22	Ejemplo 4: Luz en "Vaiven"
22	Ejemplo 5: Luz de sirena
23	Ejemplo 6: Chispas entre pantógrafo y catenaria
25	Ejemplo 7: Efecto del fuego de caldera en cabina para locomotora de vapor
25	Ejemplo 8: Artificial Intelligence (AI)
32	Comentarios Finales
32	Agradecimientos
32	Otras Publicaciones

INTRODUCCION

Con este "Análisis sobre Alumbrado" he de comentaros, ante todo, que no os voy a contar nada nuevo que no sepáis ya, aunque eso sí, lo voy a contar en Español.

Esto que vais a leer va dirigido a: los que ya saben, por que en parte, gracias a ellos he podido cubrir algunos puntos aquí tratados, y, quizás a ellos también se les escapen algunos detalles que han podido quedar aquí reflejados; y a los que comienzan ahora con esto del diseño del alumbrado de las diferentes unidades del MTS, a los que espero llegar con la suficiente claridad en las explicaciones que se van a ir sucediendo. Y ante todo dejar claro que se trata de un trabajo de divulgación y totalmente desinteresado.

Lo cierto es que al poco de tener el Train Simulator comenzaron a asaltarme los "porqués de esto y lo otro, y los cambios aquí y allá" y con lo que primero empecé a tener inquietudes y a hacer experimentos fue precisamente con el capítulo de las luces y como se configuraban.

Las fuentes de las que me he valido para obtener los datos que se van a exponer a continuación han sido con toda seguridad las mismas que vosotros utilizáis o habéis utilizado en alguna ocasión, pero ante todo y sobre todo, la práctica, y el empleo del sistema de "ensayo y error", a veces durante horas, hasta lograr obtener lo que buscaba.

Esto me ha llevado bastante mas tiempo de lo que imaginé en un principio y probablemente algunas cosas todavía se me escapan o simplemente no me he dado cuenta de que están ahí, y no las he visto. Por eso me gustaría... mejor aún, agradecería, que cualquier corrección, indicación o aportación que pudierais hacer lo hicierais con total confianza. Comento esto por que no pretendo que este trabajo "se quede aquí y ya está", sino que deseo que siga avanzando con aportaciones de todos hasta lograr completar toda la información disponible sobre este tema, eso sí, en español, para que cualquier usuario pueda valerse de ella y no tener tantos problemas a la hora de interpretar y valorar tanta y tanta documentación disponible en otras lenguas, con la correspondiente perdida de tiempo.

De momento me he limitado a darle el aspecto que tiene y que podéis observar, pero pretendo seguir profundizando y ampliándolo todo lo posible en sucesivas revisiones, indicando si así lo deseáis, las fuentes o colaboradores que hayan intervenido en el mismo, siempre y cuando esa sea su voluntad y no se manifieste lo contrario.

Para los que deseen hacer traducciones a otras lenguas, no hace falta que pidan permiso ya que desde este mismo momento tienen mi autorización. Eso sí, lo único que les exijo, es que no varíen el sentido del contenido de lo que aquí se expone, y que sean lo mas fieles posibles al texto original en cuanto a su orden y precisión.

En cuanto a su distribución, este trabajo es totalmente libre (Freeware), y su utilización de dominio publico. Estará libre de cualquier tipo de carga económica así como de publicidad del tipo que sea. Su distribución por la red se realizara asimismo, de manera libre, y desde servidores relacionados con el Train Simulator con conexión de uso general para todos los usuarios sin pago de cuotas ni donativos por parte de estos. En caso contrario no podrá ser distribuido.

La justificación del titulo: Anexo III, "Análisis del Alumbrado" se debe a que esta es solo una pequeña parte de otro trabajo mas extenso que comencé hace bastante tiempo y que no he podido concluir aún con cierto rigor y claridad, salvo pequeños fragmentos como este.

Sigo trabajando en completar este proyecto, y al mismo tiempo en otros que espero poder publicar y sean de vuestro interés.

Si quieres contactar conmigo, puedes hacerlo en: smog_21@yahoo.es (cuidado con el guión bajo que hay entre la "g" y el "2"), poniendo como Asunto: "Alumbrado".

Saludos

Smog – Septiembre 2004 – Versión 1.0

ANEXO III – ANALISIS SOBRE ALUMBRADO (LIGHTS)

Este anexo tratará de explicar la manera de diseñar los diferentes tipos de alumbrado presentes en el juego, como pueden ser luces principales, de emergencia, posición, etc. Todas las locomotoras y unidades remolcadas, tanto manejadas por nosotros como manejadas por el ordenador, pueden usar luces, y estas, ser configuradas. Las podemos encontrar fijas, o despidiendo destellos, incluso simulando la rotación de una sirena o las chispas del rozamiento entre patín y catenaria, o los rescoldos que caen por la zona inferior de una locomotora de vapor, etc, y también pueden cambiar automáticamente dependiendo de ciertos parámetros como sería el amanecer o el anochecer, o la activación de los frenos de emergencia. Los parámetros relativos al alumbrado siempre vienen especificados dentro de la sección Wagon, bajo el epígrafe Lights no importando de qué unidad se trate. Como se puede observar, dentro de este apartado, el campo es muy amplio y cada cual puede echarle toda la imaginación que desee en cuanto a su diseño.

Debemos de ser muy cuidadosos con los paréntesis que abramos, por que después, deberán de coincidir con el número de ellos que cerremos y debemos de cerrarlos en el lugar adecuado para no obtener resultados no deseados.

Una buena manera de trabajar es haciendo sangrías usando el tabulador de modo que podamos llevar cierto control y encontrar el error enseguida en caso de cometerlo.

La variable Lights (o conjunto de ellas) presentan generalmente la notación básica siguiente cambiando por supuesto los diferentes valores de configuración:

```
Lights ( 1
  Light (
    comment( luz roja trasera )
    Type ( 0 )
    Conditions (
      Headlight ( 3 )
      Unit ( 3 )
    )
    FadeIn ( 0.5 )
    FadeOut ( 0.5 )
    Cycle ( 0 )
    States ( 2
      State (
        Duration ( 0.35 )
        LightColour ( 80ff0000 )
        Position ( 0.0 3.26 -10.542 )
        Azimuth ( -180 -180 -180 )
        Transition ( 0 )
        Radius ( 1.0 )
      )
      State (
        Duration ( 0.35 )
        LightColour ( 00000000 )
        Position ( 0.0 3.26 -10.542 )
        Azimuth ( -180 -180 -180 )
        Transition ( 0 )
        Radius ( 1.0 )
      )
    )
  )
)
```

En esta notación, solo hemos diseñado una simple bombilla intermitente, en concreto de la parte posterior de un vagón de carga y tan solo para la posición Bright, de las dos posiciones de alumbrado que nos permite el juego: Dim o Bright. La notación anterior nos servirá de referencia para ir viendo paso a paso que es cada cosa y que posibilidades tiene, y ampliaremos algunas más que no aparecen en este esquema. Recordar tan solo que podemos encontrarnos en algunos modelos de

locomotoras varias decenas de notaciones idénticas a esta en su estructura, una por cada luz, pero eso no debe echarnos atrás, sino que nos llevará mas o menos tiempo hacer un apartado Lights como es debido. No solo las luces que lucirán bajo nuestro control, sino también las que lo harán cuando la unidad esté bajo el control del ordenador o lo que se denomina Artificial Intelligence (AI).

De todas maneras también nos servirá de apoyo, en parte, la documentación que suministra el programa, pero sobre todo explicaremos que es cada cosa siguiendo la estructura que podemos encontrar dentro de cada archivo .eng o .wag.

Lights

Esta variable nos abre la sección dedicada al alumbrado en general del tipo que sea. La notación es muy sencilla ya que siempre tendrá la forma:

```
Lights ( n
```

Donde el paréntesis de cierre lo encontraremos al final de la sección englobando todos los eventos luminosos y el término "n" informa al programa de cada elemento Light que tenemos. De todos y cada uno de ellos sea cual sea su finalidad.

Light

Por cada luz que diseñemos habremos de poner una variable Light. Por pequeña que sea la luz o elemento luminoso. Su notación no es complicada, todo lo contrario, se limita a:

```
Light (
```

Pero en su interior encierra un considerable conjunto de subvariables que nos ayudarán a configurar ese punto luminoso y a conseguir el efecto deseado.

Lo primero que nos encontraremos seguramente es el apartado Comment que no es de obligada presencia, pero si recomendable, para saber que luz es la que estamos diseñando o poder localizar con más rapidez una que queramos modificar. La estructura de esta línea es ya conocida: Tan solo, Comment (y aquí dentro viene el comentario, explicación o indicación) y listo, teniendo cuidado con no olvidarse los paréntesis. Podemos utilizar otra opción de identificación ahorrándonos el Comment, que veremos en el siguiente apartado.

Type

Esta variable solo acepta dos valores: 0 y 1. Su notación normal es:

```
Type ( 0 )  
Type ( 1 )
```

Podemos aprovecharla también para indicar que luz estamos diseñando evitando así el Comment del apartado anterior. Debemos hacer la notación siguiente:

```
Type ( 0 #Comentario )
```

La almohadilla # evita que el programa lea lo que viene a continuación, mientras permanezca dentro del paréntesis.

Esta variable nos permite elegir entre los dos tipos de luces pueden ser definidas en el programa:

```
Glow = 0  
Cone = 1
```

La iluminación de tipo Glow es la que se emplea como efecto visual delante de cada proyector que tenga la locomotora o vagón. Su forma es circular y su luminosidad va atenuándose en intensidad a medida que nos alejamos desde su centro. Dependiendo de la configuración de ciertos valores, esa forma circular es plana, pero presentará ciertos ángulos de inclinación según los ejes X e Y. Este tipo de efecto luminoso lo podemos encontrar justo delante de cualquier faro de locomotora que tengamos. Algunos casos son los de las imágenes que se muestran a continuación:



Gölsdorf 310



Class 50



En estos dos ejemplos se pueden ver algunos detalles curiosos. En el de la Gölsdorf 310 vemos como los puntos de luz no salen de los respectivos focos, sino que están en “el aire” por delante de las luminarias (se aprecia mejor en el detalle del margen). Es algo que debemos de ajustar para que el efecto sea lo mas realista posible, y que veremos, aunque con otro modelo de locomotora. De todas formas se aprecian las características del esquema luminoso de tipo Glow y los puntos de luz aparecen mas o menos “concentrados” quizás por el efecto óptico de la imagen mostrada. Los tres son “planos” pero podremos diseñarlos con inclinaciones adecuadas.

En el de la Class 50, vista de frente, se puede apreciar el núcleo luminoso mas o menos intenso en el centro y luego un contorno que se va desvaneciendo a medida que nos alejamos.

La iluminación de tipo Cone se da principalmente en la parte frontal de cada locomotora y por delante de esta. No es más que la superficie de vía o área frontal la que queda iluminada al encender las luces.



La intensidad lumínica, color de luz, o tamaño, se irán viendo una vez que sepamos que variables modificar.

Este es por tanto el aspecto “grafico” de estos dos tipos de efectos luminosos, que queramos o no, no son mas que “trucos gráficos” que engañan al ojo, y su utilidad es la que se indica, en ocasiones les daremos otras utilidades que muchas veces dependerán de la imaginación de cada uno para poder conseguir los efectos deseados.

Conditions

Son las condiciones bajo las que está funcionando la unidad. Su notación más habitual es:

Conditions (

Y puede tener en su interior varias subvariables que pueden configurarse a su vez de varias maneras diferentes. Podrán aparecer combinadas, en solitario, o no aparecer. Algunas de ellas solo se emplean en la configuración de unidades AI. Las subvariables a tener en cuenta son:

Headlight
Unit
Penalty
Control
Service
TimeOfDay
Weather
Coupling

Si observamos cualquier archivo .eng o .wag podremos darnos cuenta que las dos primeras subvariables suelen aparecer siempre:

```
Conditions (
  Headlight ( )
  Unit ( )
)
```

Pero otra opción para ver como queda la expresión anterior podría ser:

```
Conditions (
  Headlight ( )
  Unit ( )
  Service ( )
  TimeOfDay ( )
)
```

O cualquier combinación de los componentes del listado anterior no importando el orden. Lo que si puede ocurrir es que se originen “interferencias” entre unas y otras a la hora de especificarlas. Por ejemplo, con TimeOfDay y Weather. Si configuramos la primera para que las luces funcionen solo de noche, y además esta lloviendo, y configuramos la segunda para que el alumbrado se active con lluvia, al amanecer, si sigue lloviendo (que es lo normal ya que el juego no acepta cambios climatológicos a lo largo de una actividad) las luces no se apagarán. Debemos ser cuidadosos a la hora de configurar estas variables ya que los efectos que queremos conseguir pueden quedar anulados por una orden “mal diseñada”, según se mire en algunos casos.

Cada una de las variables anteriores se puede configurar a su vez para darnos un efecto determinado. Veamos de que manera podemos hacerlo:

Headlight

Con esta variable determinamos que luces deberán “lucir” o no, dependiendo de la posición en que tengamos el mando de las luces de la cabina (tecla H). Esta posición podrá ser Off, Dim y Bright. Pero los valores de la variable son cuatro en total:

Headlight (0)	Ignora el control de mando.
Headlight (1)	El control de mando esta en posición Off.
Headlight (2)	El control de mando esta en posición Dim.
Headlight (3)	El control de mando esta en posición Bright.

Esto quiere decir que todas las luces, del tipo que sean, que tengan configurado el valor 0, lucirán en todo momento. Las que tengan configurado el valor 1 solo lucirán cuando el mando se encuentre en posición Off, Las que tengan configurado el valor 2 lucirán únicamente cuando el mando se encuentre en posición Dim, y las que tengan configurado el valor 3 lucirán cuando el mando se encuentre en posición Bright.

Por lo tanto, si tenemos una luz de posición, y queremos que luzca en las posiciones Dim y Bright, deberemos de colocar un apartado Light por cada una de las posiciones disponibles, cambiando tan

solo los valores de Headlight que serán 2 y 3 respectivamente. (Se verá en los ejemplos mas adelante).

Unit

Nos determina la posición en la que va colocada la unidad ya sea vagón o locomotora. Los valores que puede adoptar la variable son cuatro:

Unit (0)	Ignora donde se colocará la unidad.
Unit (1)	La unidad ocupa cualquier lugar intermedio.
Unit (2)	La unidad ocupa el inicio de la composición.
Unit (3)	La unidad ocupa el final de la composición.

Para el caso de una locomotora la configuración mas habitual es el valor 2 ya que van tirando en cabeza del resto de la composición. Y un vagón tendría la configuración 3 por ir a remolque. La configuración dependería también del tipo de luz que estemos definiendo.

Este apartado es interesante bajo el punto de vista de qué luces permanecen encendidas según lo que tengamos enganchado o no por delante o por detrás. Si tenemos dos locomotoras en tracción múltiple no tiene mucho sentido que las dos vayan con "todo encendido". Lo lógico sería que la delantera solo encendiéndose su conjunto luminoso delantero, y la trasera su conjunto posterior. Pero en el momento de desengancharse, automáticamente la delantera encendería su conjunto posterior y la otra locomotora quedaría con todo apagado (si no hemos configurado nada en contra) indicando que está "fuera de uso". Ocurriría lo contrario al engancharlas de nuevo. Pero no solo nos limitamos a la unión entre locomotoras, sino que el comportamiento sería similar con vagones remolcados. En el caso de estos, al configurarlos con valor 3, conseguimos que al ser enganchados se encienda la iluminación posterior del ultimo.

De todas maneras todas las unidades pueden aceptar numerosas combinaciones de alumbrado por "posición" en la que se encuentren dentro de la composición y el mejor sistema para obtener lo que deseamos es hacer varias pruebas.

Lo que si que debemos de tener presente en todo momento, por ejemplo con los vagones, es que cuando formamos un tren con el mismo tipo de vagón, lo hacemos repitiendo "ese" vagón en concreto, que responde siempre a la misma premisa de configuración. Si deseáramos que la respuesta fuera diferente deberíamos crear nuevos vagones (supondría duplicar la carpeta completa y renombrarla con todo lo que ello conlleva) y configurar cada uno de ellos según nuestros deseos. Lo mismo ocurre con las locomotoras en tracción múltiple. Si son iguales, normalmente conseguimos enganchar varias a costa de que siempre sea la misma.

Penalty

Configuraríamos las luces de emergencia con esta variable. Los valores que puede adoptar son tres:

Penalty (0)	Ignora la parada de emergencia.
Penalty (1)	Luces emergencia encendidas sin parada emergencia.
Penalty (2)	Luces emergencia encendidas durante parada emergencia.

Lo cierto es que el programa nos deja efectuar paradas de emergencia pero en lo que se refiere a la cuestión de alumbrado, que indica que se trata de una parada de este tipo, no hay ninguna locomotora o vagón configurado para que se note de manera "externa".

Para el valor 1 las luces de emergencia se encienden cuando la unidad es detenida automáticamente en la vía por alguna causa como puede ser exceso de velocidad si así lo hemos configurado.

Para el valor 2 las luces de emergencia se encienden cuando nosotros efectuamos la parada de emergencia.

En la sección de ejemplos al final de este manual, veremos como configurar alguna respuesta luminosa ante una parada de emergencia.

Control

Nos determina quien controla la unidad, si el jugador o el ordenador. Esta variable viene bien a la hora de configurar AI. Presenta tres valores:

Control (0)	Ignora quien es el que controla la unidad.
Control (1)	El Jugador no controla la unidad.
Control (2)	El jugador controla la unidad.

El caso de la configuración con valor 1, puede darse por ejemplo durante una tracción múltiple, en la que el jugador solo controla la locomotora de cabeza (esta estaría configurada con valor 2), pero no las demás (configuradas con valor 1) que forman la tracción, suponiendo que sean locomotoras diferentes.

Si la tracción se efectúa con la misma locomotora pero repetida (aspecto que el juego acepta sin mas), la configuración por defecto sería la de valor 2. En este caso no suele aparecer reflejada.

Service

Nos indica si la unidad esta funcionando o no. Acepta tres valores de configuración:

Service (0)	Ignora si la unidad esta en servicio o no.
Service (1)	La unidad no esta en servicio.
Service (2)	La unidad esta en servicio.

Esta variable nos viene bien a la hora de tener aparcados trenes en las estaciones. Simplemente están ahí. Apropiaada para configuraciones AI.

TimeOfDay

Nos determina la configuración de encendido o apagado automático del alumbrado dependiendo de la hora del día en que nos encontremos. Tiene tres valores de configuración:

TimeOfDay (0)	Ignora la hora a la que nos encontremos.
TimeOfDay (1)	Solo de día.
TimeOfDay (2)	Solo de noche.

Esta variable es adecuada para AI, aunque también la podemos utilizar sin problema alguno si tenemos nosotros el control de la locomotora.

Para el programa, si consideramos un sistema horario de 24 horas, es de día desde las 6:00 hasta las 19:00, y es de noche desde esta hora hasta alcanzar de nuevo las 6:00.

Significa esto que si configuramos esta variable con el valor 1, tendremos las luces encendidas durante toda la franja horaria que dure el día, y apagadas el resto del tiempo. Si la configuramos con la variable 2, ocurre lo contrario (opción más lógica); tendremos las luces apagadas durante el día y encendidas durante la noche.

Weather

Nos determina la configuración de encendido o apagado automático del alumbrado dependiendo de las condiciones climatológicas. Tiene cuatro valores de configuración:

Weather (0)	Ignora las condiciones climatológicas.
Weather (1)	Solo esta activado si hace bueno.
Weather (2)	Solo esta activado si llueve.
Weather (3)	Solo esta activado si nieva.

Esta variable se aplica principalmente en unidades bajo AI. Démonos cuenta que si seleccionamos "Rain" en el programa, estará lloviendo desde el principio hasta el final. Desconozco si la intervención

de programas de modificación de las condiciones medioambientales tipo Sky Conductor o Kosmo permiten la variación climatológica durante el desarrollo de un recorrido. En el caso de que así fuera, veríamos como el alumbrado funcionaría de manera automática según que condiciones climatológicas reinasen en cada momento y dependiendo del valor de configuración especificado para cada luz.

Coupling

Esta variable determina por que lado es enganchada la unidad. Se emplea en unidades dirigidas por el ordenador AI y tiene cuatro valores de configuración:

Coupling (0)	Ignora por donde se engancha la unidad.
Coupling (1)	La unidad es enganchada por delante.
Coupling (2)	La unidad es enganchada por detrás.
Coupling (3)	La unidad es enganchada por ambos extremos.

Dependiendo por donde sea enganchada, el alumbrado lucirá o no. Por ejemplo las luces posteriores se apagarán de modo que sea la última unidad la que encienda las suyas si se trata de otra locomotora. Los vagones suelen tener las luces posteriores activadas.

FadeIn

Esta variable permite definir el tiempo, en segundos, que tarda la luz en encenderse desde que se acciona el mando hasta que toma su brillo máximo. La notación típica es:

FadeIn (n)

Donde “n” es el número de segundos pudiendo ser un numero decimal. Si “n” = 0 el encendido es instantáneo.

FadeOut

Esta variable permite definir el tiempo, en segundos, que tarda la luz en apagarse desde que se acciona el mando hasta que finalmente deja de lucir y se apaga. La notación típica es:

FadeOut (n)

Donde “n” es el número de segundos pudiendo ser un numero decimal. Si “n” = 0 el apagado es instantáneo.

Cycle

Esta variable acepta dos valores y su notación típica es:

Cycle (0)
Cycle (1)

Si el valor adoptado es el 0, el programa efectuará un ciclo “circular” empezando por el primer estado de luz que hayamos definido (“State”, que veremos dentro de dos apartados), leyendo en orden el resto, y al llegar al último volverá a comenzar de nuevo por el principio. Si el valor es 1 el programa efectuará un ciclo de “vaivén”. Empezando por el primer estado, leyendo el resto en orden y al llegar al último estado, leerá de nuevo los estados anteriores en orden inverso hasta llegar al primero y comenzar de nuevo la secuencia. Estas lecturas se harán de manera indefinida para ambos valores, 0 y 1 mientras la luz esté activada.

Para el caso particular de que tuviéramos cinco State definidos, las secuencias de lectura quedarían como se indica a continuación:

Valor 0, secuencia: 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 y así sucesivamente.
Valor 1, secuencia: 1 2 3 4 5 4 3 2 1 2 3 4 5 4 3 2 1 2 3 4 5 4 3 2 1 y así sucesivamente.

Este parámetro se combina en cierto modo con el siguiente.

NOTA: Lo cierto es que con esta variable, Cycle, lo que debería de suceder es lo que se ha expuesto en las líneas anteriores, pero no es así. Con el valor 0 configurado el programa no da problemas, pero con el valor 1, sencillamente se bloquea. La documentación original comenta muy escuetamente la manera correcta de configuración que es la comentada antes.

Por más pruebas y combinaciones que he realizado no he logrado que el valor 1 tenga efecto visual o aplicación alguna, y la mayoría de las ocasiones se ha bloqueado o ha funcionado pero sin tener en cuenta la orden. De todas maneras, el efecto que se debería lograr con el valor de configuración 1, también se logra con la variable "State" configurada en una secuencia adecuada y que se verá dentro de un par de apartados.

States

Una luz está formada por un número de estados (State) que son la manera en que dicha luz se encuentra en un momento de tiempo determinado. Estos estados pueden repetirse de forma cíclica o repetitiva (como se indicó en la variable del apartado anterior).

Si queremos diseñar una luz parpadeante, necesitaremos dos estados de luz. Un estado con la luz encendida y otro con ella apagada. Al hacerlos funcionar de manera cíclica, nos dará la sensación de parpadeo y si además variamos el tiempo de encendido y el brillo podremos acentuar o atenuar dicho efecto (Flash).

Pero no solo podemos buscar el efecto parpadeante, también podemos hacer que parezca un movimiento de vaivén o incluso circular como el provocado por una sirena.

Esta variable tiene la siguiente notación típica:

```
States ( n
```

Donde "n" es el número de subapartados "State" que vamos a definir dentro de esta variable. Debemos de cuidar siempre de que "n" coincida con el número de "State" diseñados. Es muy importante.

State

Es cada uno de los subapartados que deberemos crear y configurar para lograr el efecto luminoso deseado. Esta variable tiene a su vez varios subapartados más y su notación típica es:

```
State (
```

(Observar que hay una "s" de diferencia entre esta variable y la del apartado anterior). Vamos a ver ahora cada una de las variables que componen un State.

Duration

Determina en segundos la duración de la luz de este State en concreto. Cada State creado podrá tener idéntica Duration o diferente. La notación típica es:

```
Duration ( n )
```

Donde "n" puede ser un valor entero o decimal ambos positivos, incluido el 0 que nos proporciona luz permanente.

LightColour

Determina el color de la luz así como su transparencia u opacidad. La variable viene dada en base 16 (hexadecimal) y tiene la forma:

```
LightColour ( ααrrvvaa )
```

Como se determina el color, es, más que nada, cuestión de práctica y alguna que otra noción teórica. Tan solo indicar que muchos colores y tonos se obtienen por la mezcla de los tres colores básicos (para los monitores) que son el rojo, verde y azul, además de la transparencia u opacidad dada por el valor de α :

- α : Nos define la transparencia u opacidad de la luz.
- rr: Nos define la cantidad de rojo que interviene en el color final.
- vv: Nos define la cantidad de verde que interviene en el color final.
- aa: Nos define la cantidad de azul que interviene en el color final.

En el sistema hexadecimal tenemos valores entre 00 y FF para cada una de estas cuatro parejas de datos. Aunque todas vayan unidas en un mismo bloque, pueden determinarse separadamente y al unir las ver el resultado final de la mezcla.

Para entender un poco mejor todo esto puede consultarse el [Anexo X](#).

La transparencia u opacidad de la luz afecta principalmente a las luces de tipo Glow indicadas en el apartado Type. (Con las de tipo Cone no hay interacción de tipo alguno).

Position

Esta variable tiene tres valores que se corresponden con los ejes cartesianos x, y, z. Respectivamente nos indican la posición del foco con respecto al centro de la unidad especificado en el archivo .s. La notación típica es:

Position (anchura altura longitud)

- Anchura: Es la longitud a la derecha o izquierda con respecto al eje de simetría de la unidad.
- Altura: Es la altura a la que se encuentra con respecto a la unidad.
- Longitud: Es el alejamiento frontal o posterior al que se encuentra de la unidad.

Pudiendo ser los valores x, y, z negativos y decimales. Todos están expresados en metros.

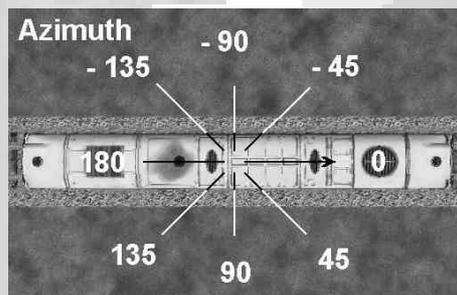
Para alumbrado de tipo "Cone" lo que si parecen obvias son las configuraciones de los valores de anchura y Longitud, pero no tanto el de altura. Lo cierto es que si especificamos una altura demasiado elevada, no podremos ver el campo de luz que se debería desplegar por delante de la locomotora ya que a mayor altura la proyección luminosa se va desvaneciendo. Y si tiene valores negativos, tampoco, ya que quedaría "enterrado" bajo la vía. Este valor va configurado por lógica, a la misma altura en que se encuentra el sistema de alumbrado principal de la locomotora.

Darse cuenta también que esta variable se emplea para posicionar también tipos de luz "Glow".

Mientras que la primera irá normalmente colocada por delante de la locomotora en claro avance, la segunda irá más o menos pegada al emisor de la luz.

Azimuth

Determina la inclinación o rotación con respecto al **eje Y** de las luces de tipo Glow. En el caso más simple, los tres valores que definen esta variable son iguales y comprendidos entre -180 y 180 cada uno, ya que cifras inferiores o superiores respectivamente, provocarán que se genere un error en el programa. La figura especifica la distribución de dichos ángulos:



En casos mas complejos donde se busca definir el arco de la luz se puede configurar con valores en las formas (-45 0 45) para crear una luz con un arco de 90 grados, o también (0 -15 -15). Hay que intentar siempre que las luces de tipo Glow queden orientadas hacia el exterior, pero están limitadas por los valores del Azimuth y Elevation. La mejor manera de configurarlas es probando y viendo el resultado obtenido. Diseñamos una luz y luego la observamos con la cámara desde varios ángulos para ver si el efecto logrado es el que buscamos.

La notación típica de esta variable es:

Azimuth (mínimo centro máximo)

Unidades en grados.

Elevation

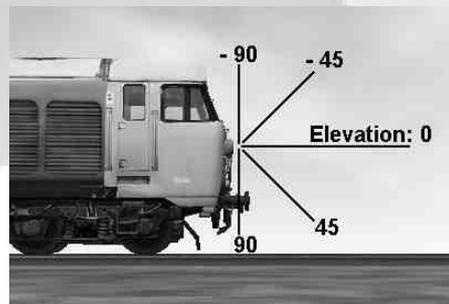
Determina la inclinación o rotación respecto del **eje X** de las luces de tipo Glow. Su función es similar que la efectuada por Azimuth y su configuración presenta las mismas particularidades que en el apartado anterior.

Los tres valores que podemos configurar estarán comprendidos entre -90 y 90 cada uno, ya que cantidades inferiores o superiores respectivamente, provocarán que se genere un error en el programa.

Su notación habitual es:

Elevation (mínimo centro máximo)

Unidades en grados.



La luz se encuentra situada en el punto donde confluyen las líneas negras sobre la superficie frontal de la locomotora. Las líneas negras indican la dirección en la que queda orientada la luz al darle el valor del ángulo indicado. Entre los valores de la figura pueden ponerse todos los valores intermedios que se desee. En este caso se han considerado iguales los tres valores que componen la variable:

Elevation (0 0 0), Elevation (45 45 45), etc

Para valores diferentes, con notaciones del tipo: Elevation (0 45 45) la cosa cambia y es mejor utilizar el sistema de “prueba y error” hasta lograr el efecto deseado.

(NOTA: Para estas dos variables, y valores de los ángulos diferentes, no he sido capaz de encontrar una pauta de comportamiento estándar para obtener resultados predecibles a la hora de poner los valores, sobre todo en Azimuth. Lo que a veces por un lado sale como esperaba, por el lado contrario no se parece en nada a la prueba anterior. Tan solo cuando los tres valores son iguales, se cumple lo que se espera).

Angle

Es el ángulo de apertura barrido por delante de la locomotora en el caso de luces de tipo Cone. La notación habitual es:

Angle (n)

Donde “n” es el valor del Angulo en grados. Suele ser positivo, aunque si se introducen grados negativos los interpreta igual con el mismo resultado gráfico. (Ver gráfico siguiente: Tipo “Cone”).

Transition

Esta variable especifica como se va a producir el paso de un State al siguiente. Su notación especifica es:

Transition (n)

Si n = 0 el cambio es instantáneo.

Si n = 1 el cambio es mediante interpolación de un State al siguiente.

En el caso de n=1, el paso de un State al siguiente es muy suave dependiendo de los valores de tiempo definidos en la variable Duration.

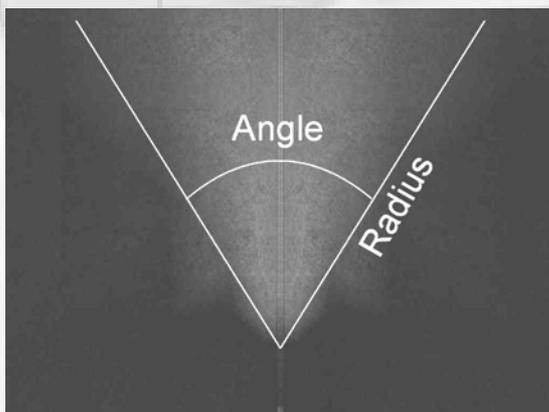
Radius

En luces de tipo “Cone” es la longitud del haz de luz que se proyecta por delante de la unidad para alumbrar la vía.

En luces de tipo “Glow” es el radio del disco que compone dicha luz. La notación habitual es:

Radius (n)

Donde “n” es el valor del radio en metros.



Tipo “Cone”



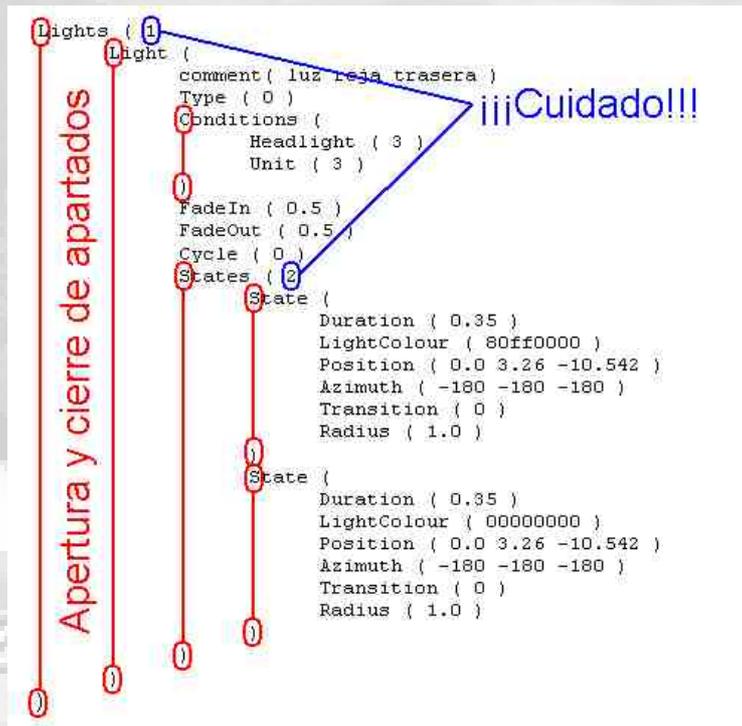
Tipo “Glow”

En los dos esquemas anteriores se pueden observar algunos de los parámetros de configuración de ambos tipos de luz. Véase que en el tipo “Glow” no se especifica Angle, y la variable Radius indica medidas totalmente diferentes para ambos tipos de alumbrado. Mientras en “Cone” el valor Radius puede alcanzar varias decenas de metros, en “Glow” apenas llegaremos en muchas ocasiones al medio metro.

Llegados a este punto hemos completado y expuesto todas las variables que intervienen en el diseño del alumbrado de una unidad. Recordar de nuevo que hay que cerrar todos los paréntesis que hemos ido abriendo para que todo funcione sin problemas.

Una buena práctica es usar el tabulador de modo que tengamos un esquema como el de la figura siguiente.

En él se puede apreciar como cada paréntesis que se cierra en la parte inferior coincide en la parte superior con el apartado que se abre, quedando ambos en la misma vertical. Acordarse de configurar



correctamente los valores adecuados que van tras los paréntesis de State y Lights, que para este caso concreto son 1 y 2 respectivamente:

“1” por que solo hay definido un apartado Light dentro de Lights.

“2” por que solo hay definidos dos apartados State dentro del apartado States.

Pero en caso de confundirnos en esta configuración, que sea por exceso ya que el programa no lo notará.

Ahora solo nos queda ver mediante algunos ejemplos como se diseñan y configuran algunos efectos de alumbrado, pudiendo haber todos aquellos como nuestra imaginación nos permita.

Comentar que para locomotoras de vapor la configuración de las luces siempre está en posición de apagado o encendido, es decir, Off o Bright. Si configuramos la variable HeadLight con el valor 2 correspondiente a Dim el programa ignorará la orden, por lo que deberemos de configurar esta variable siempre con valores de 1 para apagado y 3 para encendido.

Antes de entrar en el apartado de ejemplos, recordemos que al principio señalamos un error de diseño en el modelo Gölsdorf 310. No es el único, por desgracia, ya que si observamos la unidad GP38, dentro del apartado Lights define las luces posteriores de la unidad, pero en el juego no se ven por ninguna parte, ya que quedan ocultas dentro de la carrocería. La unidad Dash 9 tampoco se libra de estos errores. Si observamos con detenimiento, vemos que el conjunto principal de alumbrado de su parte delantera tiene dos faros. Al encender las luces solo se enciende uno y además el punto de luz que se enciende presenta una leve desviación, no coincidiendo con ninguno de los focos principales. Hay más ejemplos de estos que cada uno podrá solucionar. De todas maneras el primer ejemplo que veremos, solucionará el problema de la Dash 9, y también nos da alguna variante de utilización.

EJEMPLOS

Ejemplo 1: Corrección alumbrado Dash 9.

En este primer ejemplo, partiremos de la unidad Dash 9 que viene con el juego. El trabajo que vamos a realizar será ajustar adecuadamente el único punto de luz que se enciende en el frontal y crear otro

completamente nuevo. Además deberemos hacer que cada luz luzca según la posición del mando de cabina en sus dos opciones: Dim y Bright. Efectuaremos algún ajuste más pero de menor importancia como es aumentar el haz de luz delantero para mejorar la visibilidad. El apartado que nos define los parámetros de alumbrado completo es el siguiente, tal cual viene en el juego:

```
Lights ( 10
  Light (
    comment( Sphere of light )
    Type ( 1 )
    Conditions (
      Headlight ( 3 )
      Unit ( 2 )
    )
    Cycle ( 0 )
    FadeIn ( 0.5 )
    FadeOut ( 0.5 )
    States ( 1
      State (
        Duration ( 0.0 )
        LightColour ( ffffffff )
        Position ( 0.0 3.5 18 )
        Transition ( 0 )
        Radius ( 400.0 )
        Angle ( 15.0 )
      )
    )
  )
  Light (
    comment( Head light dim )
    Type ( 0 )
    Conditions (
      Headlight ( 2 )
      Unit ( 2 )
    )
    FadeIn ( 0.5 )
    FadeOut ( 0.5 )
    Cycle ( 0 )
    States ( 1
      State (
        Duration ( 0.0 )
        LightColour ( 80ffffff )
        Position ( 0.15 3.20 9.679 )
        Azimuth ( 0.0 -5.0 5.0 )
        Transition ( 0 )
        Radius ( 1.0 )
      )
    )
  )
  Light (
    comment( Head light bright )
    Type ( 0 )
    Conditions (
      Headlight ( 3 )
      Unit ( 2 )
    )
    FadeIn ( 0.5 )
    FadeOut ( 0.5 )
    Cycle ( 0 )
    States ( 1
      State (
        Duration ( 0.0 )
        LightColour ( ffffffff )
        Position ( 0.15 3.20 9.679 )
        Azimuth ( 0.0 -5.0 5.0 )
        Transition ( 0 )
        Radius ( 1.0 )
      )
    )
  )
  Light (
    comment( Head light bright )
    Type ( 0 )
    Conditions (
      Headlight ( 3 )
      Unit ( 2 )
    )
```

```

)
FadeIn ( 0.5 )
FadeOut ( 0.5 )
Cycle ( 0 )
States ( 1
    State (
        Duration ( 0.0 )
        LightColour ( ffffffff )
        Position ( 0.15 3.20 9.679 )
        Azimuth ( 0.0 -5.0 5.0 )
        Transition ( 0 )
        Radius ( 1.0 )
    )
)
)
Light (
    comment( Front right flashing light )
    Type ( 0 )
    Conditions (
        Headlight ( 3 )
        Unit ( 2 )
    )
    Cycle ( 0 )
    FadeIn ( 0.5 )
    FadeOut ( 0.5 )
    States ( 2
        State (
            Duration ( 0.3 )
            LightColour ( ffffffff )
            Position ( -0.60932 1.98713 10.600 )
            Azimuth ( 0 -17.0 -17.0 )
            Transition ( 0 )
            Radius ( 1.5 )
        )
        State (
            Duration ( 0.3 )
            LightColour ( 00000000 )
            Position ( -0.60932 1.98713 10.600 )
            Azimuth ( 0 -17.0 -17.0 )
            Transition ( 0 )
            Radius ( 1.5 )
        )
    )
)
Light (
    comment( Front left flashing light )
    Type ( 0 )
    Conditions (
        Headlight ( 3 )
        Unit ( 2 )
    )
    Cycle ( 0 )
    FadeIn ( 0.5 )
    FadeOut ( 0.5 )
    States ( 2
        State (
            Duration ( 0.3 )
            LightColour ( 00000000 )
            Position ( 0.60123 1.98713 10.600 )
            Azimuth ( 0 17.0 17.0 )
            Transition ( 0 )
            Radius ( 1.5 )
        )
        State (
            Duration ( 0.3 )
            LightColour ( ffffffff )
            Position ( 0.60123 1.98713 10.600 )
            Azimuth ( 0 17.0 17.0 )
            Transition ( 0 )
            Radius ( 1.5 )
        )
    )
)
Light (
    comment( Rear bottom light )
    Type ( 0 )
    Conditions (

```


Bloque Light 1. Dentro del Comment, lo denominaremos "Proyección luz delantera". Este nos define el haz de luz delantero o la parte de vía y paisaje que quedará iluminada por la locomotora. Para ampliarlo un poco lateralmente solo modificaremos Angle y pondremos el valor (30). Con eso será suficiente.

Bloque Light 2: Dentro del Comment lo denominaremos "Luz delantera inferior principal Dim". Este nos define la luz "corta" del conjunto principal delantero. Hemos observado que no está donde debería, así que vamos a desplazarla haciendo que se corresponda con el faro principal inferior. Para ello modificaremos la variable Position y adoptaremos los valores (0.123 3.04 9.72). También vamos a aumentar el valor Radius a (1.3).

Bloque Light 3: Dentro del Comment lo denominaremos "Luz delantera superior principal Bright". Este nos define la luz "larga" del conjunto principal delantero cuando el mando de cabina se encuentra en posición Bright. Las coordenadas de situación son erróneas. Vamos a colocarlo de modo que ocupe la posición del foco superior. Para ello modificaremos la variable Position adoptando los valores (0.123 3.245 9.721). También vamos a aumentar el valor Radius pero esta vez a (1.5).

Si hemos seguido las instrucciones hasta aquí, el resultado es el siguiente: Al pulsar la tecla H la primera vez, Dim, lucirá una tenue luz "corta" en el foco central inferior. Al pulsar H una segunda vez, lucirá otra luz más potente en el foco central superior. La inferior se habrá apagado.

Bloque Light 4: Dentro del Comment lo denominaremos "Luz delantera inferior principal Bright". Estaba repetido con el Bloque Light 3 (y su efecto superpuesto en el juego) antes de modificarlo.

Lo ajustaremos con el foco inferior y que luzca en posición Bright también. Hagamos que el foco inferior luzca con la misma intensidad que el superior para la posición Bright. Para ello modificamos la variable Position adoptando los valores (0.123 3.04 9.72). Aumentamos el valor Radius a (1.5).

Antes de proseguir, detengámonos unos instantes para hacer un análisis. Si nos fijamos, en los valores "z" de ambos focos centrales en posición Bright vemos que el superior tiene el valor 9.721 y el inferior el valor 9.72. ¿Y esta diferencia a que se debe? ¿No deberían ser iguales? En los valores "x" son iguales, ambos se desplazan "gráficamente" con respecto al eje de simetría de la locomotora. En el eje "y" tienen que ser distintos, por que los focos están en distintas alturas. ¿Por qué en "z" hay esta leve diferencia? La respuesta es sencilla. Si ambos tuvieran el mismo valor, al encender los focos observaríamos algunos pequeños "errores" gráficos en forma de barras o "red" negras durante el desarrollo del juego, o cuando moviéramos la cámara por la zona delantera en cualquier dirección. Esos leves errores se deben a la superposición de gráficos que experimentan ambas luces. Si damos una leve diferencia de separación entre uno y otro, modificando el valor "z" de Position, de forma que no se interfieran gráficamente y así queden superpuestos solucionaremos el problema. Pero hay otro inconveniente. ¿Qué foco adelantamos? El superior o el inferior. No es lo mismo adelantar uno u otro. No es lo mismo por una razón. Si movemos la cámara en sentido ascendente y hemos elegido mal la posición de los focos veremos que a partir de un momento determinado uno de los focos se "apaga", desaparece. En cambio si los hemos colocado convenientemente no observaremos ningún contratiempo más. Por lo tanto adelantaremos el foco superior, es decir, daremos algo más de valor al término "z" de la variable Position del foco superior. Si no fuera suficiente, aumentaríamos ese valor ya que como podemos observar en este caso, estamos trabajando con milésimas.

El resultado que obtendríamos ahora sería una tenue luz en el foco inferior al pulsar H y dos potentes luces sobre cada foco al volver a pulsar H.

Si observamos a plena luz del día y con los faros encendidos, desde uno de los laterales, podemos comprobar que con estos valores que se han dado, existe una pequeña "rendija" entre la posición gráfica de las luces y el gráfico de los faros en el frontal de la locomotora. Podemos disminuir los valores "z" en Position hasta dejar las luces "pegadas" a la superficie de los focos, pero aparecerán unas sombras o "ángulos" en la parte inferior de los focos debida a que la superficie de la locomotora no es vertical, sino que tiene una ligera inclinación en su frontal. No debemos olvidar que el efecto de "luz" no es más que un truco gráfico superpuesto al gráfico de la locomotora, y como tales gráficos que son ambos, si se interfieren, la luz penetra a través de la "chapa" del frontal desapareciendo y formando así esas "sombras". Por tanto los valores que se han adoptado son para evitar este

pequeño inconveniente que probablemente en otras ocasiones pueda servirnos como ventaja dependiendo de los casos que se nos planteen.

Bloque Light 5: Dentro del Comment la denominaremos “Luz delantera derecha intermitente Bright”. Vamos ahora con las luces parpadeantes. Eso es lo que se define en este bloque. Si nos fijamos, hasta ahora, en todos los bloques vistos, solo había un elemento State dentro de States. En este bloque, aparecen 2 State que son los responsables de que las luces parpadeen. Este bloque en concreto se encarga del foco derecho si nos colocamos sobre la vía, frente a la locomotora.

El primer State, nos indica que la luz está encendida por medio de LightColour (fffffff) que es la expresión máxima en código hexadecimal de blanco opaco. Y el Segundo State (0000000) nos indica apagado. La Variable Duration en cada State nos da el tiempo que permanece la luz encendida o apagada en este caso. Si variamos Duration (no lo vamos a hacer, aunque si alguien lo desea, es muy libre) y ponemos para el primer State; Duration (0.01) y para el segundo State; Duration (1) veremos que el resultado se convierte en un rápido destello seguido de un segundo de reposo.

Otra variable presente es Transition. Está configurada para ambos State como (0). Si la configuráramos como (1) veríamos como el paso de encendido a apagado y viceversa, es mas “lento”, pero esto se apreciaría aún mejor si hacemos Duration (2) en ambos State. Veremos como la luz se “toma su tiempo” para ir encendiéndose hasta alcanzar el máximo de luminosidad y para ir apagándose hasta desaparecer. Podemos poner Duration diferentes en cada State y obtener así otros efectos, e incluso agregar mas State siempre y cuando en States especifiquemos que los hemos agregado.

Si a Transition le damos otro valor que no sean el 0 o el 1, obtenemos un error y el programa se detiene.

Algo curioso es el valor de Azimuth. Lo cierto es que en el primer State tiene su razón de ser hasta cierto punto. La luz va levemente girada sobre su eje vertical hacia la parte derecha de la vía; hacia la cuneta derecha si nos colocamos delante de la locomotora. Probablemente, tal y como se explicó anteriormente para no interferir gráficamente con la plataforma inferior. Pero dicho giro, aparte de lo comentado no está justificado por que en realidad, el destello iría dirigido a quien se encontrase sobre la vía, es decir, de frente. (Quizás tenga alguna otra razón que se nos escapa). Pero donde si que no tiene ninguna razón de ser es en el segundo State. Sencillamente en este State la luz está apagada y no se aprecia ningún efecto, salvo que se justifique por la comodidad del diseñador y no “romperse la cabeza” a la hora de poner los valores (copiar y pegar o algo similar).

Los únicos valores que podríamos modificar son los de Radius. El valor original puede parecer algo grande para el tamaño de la luminaria. De acuerdo que son destellos, pero con un valor de 0.5, incluso de 1 sería suficiente. Adoptemos Radius (0.8), aunque cada cual puede adoptar el valor que desee. Y también podemos modificar en Position el valor “z” para acercar un poco más la luz a su emisor: Position (-0.60932 1.98713 10.548)

En el segundo State podríamos eliminar las variables de Position Azimuth y Radius sin problema, aunque su única razón de estar seguramente sea por comodidad a la hora de diseñar el efecto luminoso. Las dejaremos como están.

Bloque Light 6: Dentro del Comment la denominaremos “Luz delantera izquierda intermitente Bright”. Este bloque nos define los parámetros de funcionamiento de la luz parpadeante izquierda. Todo lo que se ha comentado en el Bloque anterior es válido para este bloque salvo algunos pequeños ajustes. El más importante es ver que los dos estades de que consta, están invertidos con respecto al anterior. Primero tenemos el State “apagado” y luego el State “encendido” de esta manera es como se logra alternar ambos destellos.

Por lo demás variaremos Radius a (0.8) y Position a (0.61 1.98713 10.55). Y nos llevamos otra pequeña sorpresa, y es que a la vista de los valores Position, salvo el correspondiente a “y” que coincide el lado derecho con el lado izquierdo, el resto no coincide dándonos a entender que el diseño de la locomotora no es completamente simétrico. Este aspecto no es que tenga excesiva importancia, pero es un dato a tener en cuenta.

Azimuth en este caso va orientado hacia la parte izquierda o cuneta izquierda si nos colocamos delante de la locomotora.

Tras haber desgranado estos dos Bloques luminosos, estamos, o, deberíamos estar en condiciones de diseñar cualquier tipo de efecto parpadeante, como sería el caso tan vistoso de los “chispazos” que se producen en las locomotoras eléctricas entre el patín y la catenaria. De todas maneras se dará un ejemplo de diseño mas adelante pero sin apenas comentario. ¿Y las chispas que suelta una

locomotora de vapor sobre la vía? Sería este mismo efecto, pero cambiando de color a rojizo, o anaranjado.

Vayamos ahora a las luces posteriores de la locomotora. Gráficamente tenemos un faro principal compuesto por dos focos situados en la parte superior central, y un pequeño foco colocado sobre la plataforma, a la derecha. Solo configuraremos ese ahora, dejando los otros dos “apagados” de momento.

Bloque Light 7: Lo ignoramos ya que es el que nos regula el foco inferior de la luz principal posterior. Para ello lo encerramos en un Comment ().

Bloque Light 8: Lo ignoramos ya que es el que nos regula el foco superior de la luz principal posterior. Para ello lo encerramos en un Comment ().

Bloque Light 9: Dentro del Comment la denominaremos “Luz posterior roja Bright”. Con este bloque configuramos la luz inferior trasera que se encuentra sobre la plataforma a la derecha según miramos de frente o en el sentido de la marcha. Tal y como está diseñada es obvio que no se encuentra donde debería. Además está diseñada para que funcione solo cuando el mando de encendido se encuentre en posición Bright: Variable Headlight (3). Ajustaremos su posición y diseñaremos un nuevo Bloque para que luzca también con el mando en posición Dim.

La variable LightColour (80ff0000), si la convertimos en **80 ff 00 00** para una mejor interpretación, nos dice lo siguiente: El valor 80 es un valor aproximadamente intermedio entre la opacidad y la transparencia total. (En este caso ff = opaco, 00 = transparente), por lo tanto la luz será mas bien débil o de poca fuerza. El valor ff nos indica un rojo en su máxima expresión. El primer valor 00 nos indica que el color verde se encuentra totalmente ausente, e igualmente el segundo valor 00 para el azul. En resumen, tendremos un color rojo algo “flojillo” para la luz trasera.

La variable Azimuth como se puede apreciar ha variado, y bastante. No la vamos a tocar, pero vamos a comentarla. Si tuviera como valores el 0, la luz estaría mirando hacia delante y nosotros desde nuestra posición “por detrás” de la locomotora no podríamos verla. Tendríamos que mover la cámara a una posición mas adelantada para ver que efectivamente la luz se encuentra ahí. Si queremos que alumbre hacia atrás deberemos de dar a Azimuth los valores que ya tiene o también acepta el valor 180 negativo para las tres entradas. Si quisiéramos que mirase hacia la derecha, introduciríamos Azimuth (90 90 90) y para el lado izquierdo, Azimuth (-90 -90 -90) así sucesivamente para orientaciones intermedias. La verdad es que Azimuth (al igual que Elevation) se comportan de una forma predecible cuando sus tres entradas son iguales. Otra cosa es cuando no lo son. No queda más remedio que probar hasta lograr el efecto deseado, ya que no solo nos da la orientación de la luz, sino que también desde qué posiciones de la cámara vamos a poder ver esa luz.

Por lo tanto, y resumiendo, siempre que queramos ver una luz posterior, o lateral, debemos configurar Azimuth adecuadamente.

Pasemos a colocar la luz sobre el foco. Antes debemos aclarar algo: Desconocemos si el gráfico del foco de la locomotora que aparece en la parte posterior donde colocaremos nuestro efecto de luz es realmente para eso; es para albergar la luz de emergencia en caso de una parada de emergencia; o realmente tiene otra finalidad. Por lo pronto nosotros aprovecharemos dicho grafico para posicionar nuestra luz posterior.



Modificamos la variable correspondiente dejándola como Position (0.93 1.97 -10.5054). Y con esto queda configurada esta luz.

Vamos a crear ahora otra igual para la posición Dim. Para ello copiamos todo el Bloque Light 9 y lo pegamos a continuación. Solo modificamos la variable Headlight de valor (3) a valor (2). No es necesario ningún ajuste más. Tan solo denominarla en Comment como “Luz posterior roja Dim”.

Con esto queda configurada completamente la cuestión de alumbrado para la Dash 9. Tenemos alumbrado delantero y posterior para el sentido normal de marcha de dicha unidad. Solo nos queda ajustar la variable Lights del principio para terminar el trabajo y deberá tener el valor 8:

Lights (8

Podríamos seguir añadiendo efectos luminosos como por ejemplo que en posición Dim la parte delantera de la vía se ilumine apenas. Sería cuestión de crear un nuevo haz de luz variando los valores de las variables del primero. O que ese mismo efecto lo hicieran las luces parpadeantes, es decir, alumbrar alternativamente los laterales de la vía. Esto ya queda como ejercicios o prácticas para cada uno. Tan solo acordarse de incrementar el valor Lights del principio, aunque si lo definimos con un valor digamos de 50, no nos causará ningún problema y podremos diseñar todos los efectos que queramos siempre y cuando no superemos dicho valor. Si lo superamos, lo incrementaremos también.

Una variable que aparece en todos los Bloques y que no hemos comentado es "Unit". Si nos fijamos, solo aparecen dos valores de los cuatro posibles: el 2 y el 3. Y además si observamos con un poco mas de detenimiento, los valores 2 aparecen en bloques de alumbrado delantero, y el 3 en bloques traseros. Esto quiere decir que si se engancha algo por la parte posterior, la luz dejará de lucir automáticamente hasta que desenganchemos esa parte. No ocurrirá lo mismo por la parte delantera, que si enganchamos algo, las luces seguirán tal cual, si estaban encendidas.

Ejemplo 2: Alumbrado "bidireccional".

No es el caso de la locomotora del ejemplo anterior, pero hay muchos otros modelos de locomotoras denominados "bicabina" es decir, con dos cabinas, una en cada extremo, que las permite circular en ambos sentidos, o hacer tracciones con la locomotora en circulación invertida con lo que necesitaremos ver que es lo que hay por "delante". Eso requiere que prescindamos de ciertos "lujos" como es tener alumbrado de "cortas" y "largas", ya que podemos emplear la posición Dim para definir todo el alumbrado delantero, y el Bright para el trasero, dejando fijas las luces de posición.

Vamos a efectuar este ejemplo aprovechando la locomotora anterior tal y como quedó configurada al final del ejemplo 1.

Muchos de los Bloques Light se pueden aprovechar tal y como están, haciendo tan solo pequeños ajustes.

Los Bloques Light 1, 3, 4, 5 y 6 ajustamos a Headlight (2). Bloqueamos con Comment () el Bloque Light 2, y el Bloque Light 9 que corresponda a la posición Bright. Con ello hemos resuelto el alumbrado de circulación "normal" para la Dash 9.

En "circulación invertida" debemos configurar los bloques Light 7 y Light 8 que quedaron inactivos anteriormente, y también el punto de luz que hay en la parte delantera derecha según nos colocamos frente a la locomotora por su parte delantera. Podemos usar el punto de luz posterior como parpadeante. Eso sí, en todos estos bloques, la variable Headlight deberá tener el valor (3).

Bloque Light 7: Foco principal inferior de la parte trasera: Position (0.01007 4.10572 -9.598). La verdad es que está ya bien configurado de origen, pero se ha preferido efectuar un leve ajuste por la cuestión de las interferencias entre gráficos de luz. Observar también la configuración de la variable Azimuth para orientar el faro hacia atrás.

Bloque Light 8: Foco principal superior de la parte trasera: Position (0.01007 4.26114 -9.5982). Como se puede comprobar ahora comparando con el Bloque anterior, se aprecia el ajuste en el valor "z", que nos evita interferencias. Esta luz "sobresale" un poco mas que la otra y así de esta manera no nos dará problemas.

Crearemos ahora dos Bloques nuevos, correspondientes a la luz de parpadeante y a la luz posterior respectivamente.

Luz Parpadeante: Sencillamente podemos copiar y pegar cualquiera de las dos de la parte delantera y modificar la variable Position adoptando las coordenadas de la luz roja posterior. Modificaremos también Azimuth en los dos apartados State dando el valor 180 a todas sus entradas.

Luz Delantera Roja: Copiamos la luz trasera roja y modificamos Azimuth a valores 0 en todas sus entradas. Damos valores de Position (-0.85 1.935 10.448).

Con esto terminamos los ajustes de las luces en la locomotora, pero todavía queda un pequeño detalle que tocar. Se trata de la zona iluminada que deberemos ver por la parte posterior. En esto el programa se queda algo limitado por que no contempla la posibilidad de alumbrar hacia atrás. Pero mediante un pequeño “truquillo” podemos solventar dignamente este inconveniente.

Haz de Luz Posterior: Se trata de crear un haz de luz que puede tener las mismas características que el delantero y mediante las coordenadas de Position lo desplazamos por detrás de la locomotora la distancia que sea precisa, variando el valor “z” (Normalmente adopta números negativos). Como solución no es precisamente una maravilla, por que dicha iluminación se comporta de manera un tanto peculiar, pero al menos tendremos visibilidad. Lo cierto es que estaremos viendo en la parte mas próxima a la locomotora lo que normalmente debería quedar mas alejado y viceversa. El programa no nos permite girar esta zona de luz. Si no lo hacemos excesivamente ancho, no se notará en demasía este “arreglillo”. Ajustaremos a Position (0.0 4.18 -180) y a Radius (150), Angle podríamos reducirlo a 10, aunque no está mal con el valor original.

El valor “y” de Position se obtiene de hacer la media aproximada de altura entre los dos focos posteriores.

Tan solo recordar ajustar el valor de Lights al principio, que ahora deberá ser 11.

Hemos obtenido así el alumbrado para una locomotora “bicabina” y con ello se da por concluido este segundo ejemplo.

Ejemplo 3: Luz “Mars” en parada de emergencia “Penalty”.

En el ejemplo que se expone a continuación se puede ver el efecto rotatorio y al mismo tiempo el aumento y disminución de la intensidad luminosa del foco superior principal cuando accionamos la tecla “Return” que corresponde a la parada de emergencia o “Penalty”. (Los valores están adaptados a la Dash 9).

```
Light (
  Type( 0 #Luz "Mars" Penalty)
  Conditions (
    Unit ( 2 )
    Penalty ( 2 )
  )
  FadeIn ( 0.5 )
  FadeOut ( 0.5 )
  Cycle ( 0 )
  States ( 6
    State (
      Duration ( 0.20 )
      LightColour ( ffffffff )
      Position ( 0.223 3.3 9.721 )
      Transition ( 1 )
      Radius ( 0.75 )
    )
    State (
      Duration ( 0.20 )
      LightColour ( ffffffff )
      Position ( 0.123 3.3 9.721 )
      Transition ( 1 )
      Radius ( 1.5 )
    )
    State (
      Duration ( 0.20 )
      LightColour ( ffffffff )
      Position ( 0.023 3.3 9.721 )
      Transition ( 1 )
      Radius ( 0.75 )
    )
    State (
      Duration ( 0.20 )
      LightColour ( ffffffff )
      Position ( 0.023 3.2 9.721 )
      Transition ( 1 )
      Radius ( 0.75 )
    )
    State (
      Duration ( 0.20 )
```


Con algunas modificaciones podemos conseguir el mismo efecto bajo la posición Bright. Para ello eliminaríamos la variable Penalty (2) y pondríamos Headlight (3).

En realidad, nuestras locomotoras no tienen un sistema de señalización visual tan sofisticado por lo que hacer el diseño de uno más sencillo, del estilo de una luz más o menos potente y parpadeante, no debería resultar complicado ahora.

Ejemplo 4: Luz en “Vaiven”.

Para este tipo de luz podemos ayudarnos del ejemplo anterior, y modificando los valores de Position convenientemente lograr que la luz haga el efecto de ir de derecha a izquierda y viceversa (O de arriba abajo). No debería de resultar complicado si se ha examinado con detenimiento y comprendido la configuración del ejemplo anterior. Logramos así el efecto que no podíamos conseguir con la variable Cycle (debido al no funcionamiento de esta).

Ejemplo 5: Luz de sirena.

La luz de una sirena puede ser simulada tanto si esta se encuentra en posición horizontal como en vertical. Lo más habitual es que se encuentren sobre techos de locomotoras o vagones. Un ejemplo típico de este caso es el que se muestra a continuación con seis State, aunque si se quiere obtener una sirena aún más detallada no tendríamos más que duplicar los State y ajustar algunos valores internos de las variables (ángulos, Duration y Transition principalmente) para obtener una sensación de “suavidad y rapidez o lentitud” en el giro del haz de luz. Lo que ya es más complicado de obtener, por no decir imposible es el reflejo de esa luz en los alrededores del entorno.

```
Light (
  Type ( 0 #Luz de sirena )
  Conditions (
    Unit ( 2 )
    Headlight ( 0 )
  )
  FadeIn ( 0 )
  FadeOut ( 0 )
  Cycle ( 0 )
  States ( 9
    State (
      Duration ( 0.075 )
      LightColour ( fffffff0 )
      Position ( 0 4.85 7.13 )
      Azimuth ( 180 180 180 )
      Elevation ( -15 -15 -15 )
      Transition ( 1 )
      Radius ( 0.4 )
    )
    State (
      Duration ( 0.15 )
      LightColour ( fffffff0 )
      Position ( 0.02 4.85 7.14 )
      Azimuth ( 135 135 135 )
      Elevation ( -15 -15 -15 )
      Transition ( 1 )
      Radius ( 0.4 )
    )
    State (
      Duration ( 0.15 )
      LightColour ( fffffff0 )
      Position ( 0.04 4.85 7.16 )
      Azimuth ( 90 90 90 )
      Elevation ( -15 -15 -15 )
      Transition ( 1 )
      Radius ( 0.4 )
    )
    State (
      Duration ( 0.15 )
      LightColour ( fffffff0 )
      Position ( 0.02 4.85 7.18 )
      Azimuth ( 45 45 45 )
    )
  )
)
```



```

Headlight ( 0 )
Unit ( 3 )
)
FadeIn ( 0 )
FadeOut ( 0 )
Cycle ( 0 )
States ( 8
State (
Duration ( 4.0 )
LightColour ( 11000000 )
Position ( 0 5.5 -8.31 )
Azimuth ( 0 -180 180 )
Transition ( 0 )
Radius ( 0.90 )
Elevation ( 0 -90 90 )
)
State (
Duration ( 0.05 )
LightColour ( ffaaaaff )
Position ( 0 5.5 -8.31 )
Azimuth ( 0 -180 180 )
Transition ( 0 )
Radius ( 1.30 )
Elevation ( 0 -90 90 )
)
State (
Duration ( 15.0 )
LightColour ( 11000000 )
Position ( 0 5.5 -8.31 )
Azimuth ( 0 -180 180 )
Transition ( 0 )
Radius ( 0.90 )
Elevation ( 0 -90 90 )
)
State (
Duration ( 0.05 )
LightColour ( ffaaaaff )
Position ( 0 5.5 -8.31 )
Azimuth ( 0 -180 180 )
Transition ( 0 )
Radius ( 0.90 )
Elevation ( 0 -90 90 )
)
State (
Duration ( 7.0 )
LightColour ( 11000000 )
Position ( 0 5.5 -8.31 )
Azimuth ( 0 -180 180 )
Transition ( 0 )
Radius ( 0.90 )
Elevation ( 0 -90 90 )
)
State (
Duration ( 0.05 )
LightColour ( ffaaaaff )
Position ( 0 5.5 -8.31 )
Azimuth ( 0 -180 180 )
Transition ( 0 )
Radius ( 1.1 )
Elevation ( 0 -90 90 )
)
State (
Duration ( 0.05 )
LightColour ( 11000000 )
Position ( 0 5.5 -8.31 )
Azimuth ( 0 -180 180 )
Transition ( 0 )
Radius ( 0.90 )
Elevation ( 0 -90 90 )
)
State (
Duration ( 0.05 )
LightColour ( ffaaaaff )
Position ( 0 5.5 -8.31 )
Azimuth ( 0 -180 180 )
Transition ( 0 )
Radius ( 0.90 )
)

```

```
Elevation ( 0 -90 90 )
```

Observamos en este ejemplo como la variable Headlight se configura con el valor 0 ya que es preciso que el efecto funcione todo el tiempo. El único inconveniente es que también funciona con la locomotora detenida. ¡¡¡Que le vamos a hacer, el programa... no es perfecto!!!. Aunque el efecto queda bonito, eso hay que reconocerlo.

Ejemplo 7: Efecto del fuego de caldera en cabina para locomotora de vapor.

En este otro ejemplo, se puede ver la configuración para que parezca que se ve el interior de la caldera de una locomotora de vapor cuando la compuerta de carga de carbón se encuentra abierta. Las coordenadas están configuradas para aplicarla en la Gölsdorf 380. Si conducimos dicha locomotora en horario nocturno el efecto queda más acentuado que en horario diurno. Para lograr que funcione, tan solo es cuestión de copiar las líneas siguientes dentro del apartado Lights y aumentar el valor de 5 a 6. Si el color no es el adecuado o aparece algo “desvaído” puede variarse modificando el apartado LightColour, o también aumentar el número de State.

```
Light (
  Type ( 0 # Fuego de caldera )
  FadeIn ( 0.5 )
  FadeOut ( 0.5 )
  Cycle ( 0 )
  States ( 2
    State (
      Duration ( 0.05 )
      LightColour ( 80ff7200 )
      Position ( 0 2.2 -4.1 )
      Azimuth ( -180 -180 -180 )
      Transition ( 0 )
      Radius ( 1.4 )
    )
    State(
      Duration ( 0.05 )
      LightColour ( 80ffe610 )
      Position ( 0 2.2 -4.1 )
      Azimuth ( -180 -180 -180 )
      Transition ( 0 )
      Radius ( 1.4 )
    )
  )
)
```

Ejemplo 8: Artificial Intelligence (AI)

Aquí se configurarán las luces para que sea el ordenador el que gestione el alumbrado de las unidades que no están bajo nuestro control. Desde el mismo momento en que se diseña un archivo .eng, en el apartado de alumbrado, Lights, el diseñador debería de añadir las luces necesarias para cubrir este capítulo. Realmente no se trata más que duplicar algunas de las luces, Light, que ya tenemos y modificar los valores de algunas de las variables que hemos expuesto mas arriba. Con eso sería suficiente para que, por ejemplo, una locomotora encendiese su alumbrado al caer la noche y lo apagase al amanecer. O para que lo mantuviera encendido durante una nevada (realmente sería toda la actividad ya que no hay cambios climáticos a lo largo de ella). Pero esto iría mas en ayuda de los diseñadores de actividades que en los del usuario que solo va a jugar. El diseñador tan solo se preocuparía de diseñar su actividad con una nevada o con lluvia y sabría que ese alumbrado en concreto luciría.

Lo que si que queda un poco limitado en el programa es que una vez diseñado el elemento de alumbrado, este permanecerá fijo. ¿Qué quiere decir esto? Muy sencillo. Nosotros en las unidades que manejamos podemos modificar el alumbrado según las posiciones del mando correspondiente: Off, Dim y Bright. Es decir, podemos alternar entre varios modos de alumbrado a voluntad. Las unidades manejadas por el ordenador no tienen esa capacidad. Si se ha diseñado el alumbrado en modo Bright para que actúe en caso de nevada, el programa es lo que hará: Las luces se encenderán en modo Bright y estarán en ese modo mientras dure la nevada. O lo mismo para el alumbrado diurno

nocturno. Si hemos configurado las variables para que el alumbrado luzca en modo Dim durante la noche, todas las unidades así configuradas que nos crucemos durante las horas nocturnas lucirán alumbrado Dim. Y cuando lleguen las horas diurnas se apagará.

Como ya se comentó antes, podemos hacer que varias variables actúen en combinación: Alumbrado de noche y lloviendo. Si el diseñador de la actividad pretende que veamos un amanecer lluvioso, sabrá que las luces irán encendidas durante la noche y que justo cuando llegue la hora que el ordenador considera como día, las luces... ¡¡¡No se apagarán!!!. Lógico. El que no se apaguen lo determina la variable que nos indica que lucirán mientras llueva. Este es un caso claro de interferencia entre órdenes. Y como este caso hay muchos mas que deberemos saber solucionar.

Hay, por el contrario, algunos que no se pueden llevar a la practica como es el caso que la locomotora de cabeza encienda su alumbrado al entrar en un túnel. Si esa locomotora tenía el alumbrado apagado antes de entrar, seguirá con él apagado dentro del túnel, y viceversa.

Para no perder la costumbre vamos a diseñar el alumbrado AI para la Dash 9. Algo sencillo, como que por ejemplo luzca en horario diurno nocturno. Para ello partiremos de las modificaciones efectuadas en el ejemplo 1. Si dicho ejemplo lo seguisteis como se explicó en su momento, deberiais de haber obtenido lo siguiente:

```
Lights ( 8 )
  Light (
    comment( Proyeccion luz delantera )
    Type ( 1 )
    Conditions (
      Headlight ( 3 )
      Unit ( 2 )
    )
    Cycle ( 0 )
    FadeIn ( 0.5 )
    FadeOut ( 0.5 )
    States ( 1 )
      State (
        Duration ( 0.0 )
        LightColour ( ffffffff )
        Position ( 0.0 3.5 18 )
        Transition ( 0 )
        Radius ( 400.0 )
        Angle ( 15.0 )
      )
    )
  )
  Light (
    comment( Luz delantera inferior principal Dim )
    Type ( 0 )
    Conditions (
      Headlight ( 2 )
      Unit ( 2 )
    )
    FadeIn ( 0.5 )
    FadeOut ( 0.5 )
    Cycle ( 0 )
    States ( 1 )
      State (
        Duration ( 0.0 )
        LightColour ( 80ffffff )
        Position ( 0.123 3.04 9.72 )
        Azimuth ( 0.0 -5.0 5.0 )
        Transition ( 0 )
        Radius ( 1.3 )
      )
    )
  )
  Light (
    comment( Luz delantera superior principal Bright )
    Type ( 0 )
    Conditions (
      Headlight ( 3 )
      Unit ( 2 )
    )
    FadeIn ( 0.5 )
```

```

FadeOut ( 0.5 )
Cycle ( 0 )
States ( 1
  State (
    Duration ( 0.0 )
    LightColour ( ffffffff )
    Position ( 0.123 3.245 9.721 )
    Azimuth ( 0.0 -5.0 5.0 )
    Transition ( 0 )
    Radius ( 1.5 )
  )
)
)
Light (
  comment( Luz delantera inferior principal Bright )
  Type ( 0 )
  Conditions (
    Headlight ( 3 )
    Unit ( 2 )
  )
  FadeIn ( 0.5 )
  FadeOut ( 0.5 )
  Cycle ( 0 )
  States ( 1
    State (
      Duration ( 0.0 )
      LightColour ( ffffffff )
      Position ( 0.123 3.04 9.72 )
      Azimuth ( 0.0 -5.0 5.0 )
      Transition ( 0 )
      Radius ( 1.5 )
    )
  )
)
Light (
  comment( Luz delantera derecha intermitente Bright )
  Type ( 0 )
  Conditions (
    Headlight ( 3 )
    Unit ( 2 )
  )
  Cycle ( 0 )
  FadeIn ( 0.5 )
  FadeOut ( 0.5 )
  States ( 2
    State (
      Duration ( 0.3 )
      LightColour ( ffffffff )
      Position ( -0.60932 1.98713 10.548 )
      Azimuth ( 0 -17.0 -17.0 )
      Transition ( 0 )
      Radius ( 0.8 )
    )
    State (
      Duration ( 0.3 )
      LightColour ( 00000000 )
      Position ( -0.60932 1.98713 10.548 )
      Azimuth ( 0 -17.0 -17.0 )
      Transition ( 0 )
      Radius ( 0.8 )
    )
  )
)
Light (
  comment( Luz delantera izquierda intermitente Bright )
  Type ( 0 )
  Conditions (
    Headlight ( 3 )
    Unit ( 2 )
  )
  Cycle ( 0 )
  FadeIn ( 0.5 )
  FadeOut ( 0.5 )
  States ( 2
    State (
      Duration ( 0.3 )
      LightColour ( 00000000 )

```

```

        Position ( 0.61 1.98713 10.600 )
        Azimuth ( 0 17.0 17.0 )
        Transition ( 0 )
        Radius ( 0.8 )
    )
    State (
        Duration ( 0.3 )
        LightColour ( ffffffff )
        Position ( 0.61 1.98713 10.55 )
        Azimuth ( 0 17.0 17.0 )
        Transition ( 0 )
        Radius ( 0.8 )
    )
)
)
Light (
    comment( Luz posterior roja Bright )
    Type ( 0 )
    Conditions (
        Headlight ( 3 )
        Unit ( 3 )
    )
    FadeIn ( 0.5 )
    FadeOut ( 0.5 )
    Cycle ( 0 )
    States ( 1
        State (
            Duration ( 0.0 )
            LightColour ( 80ff0000 )
            Position ( 0.93 1.97 -10.5054 )
            Azimuth ( -180 -180 -180 )
            Transition ( 0 )
            Radius ( 0.5 )
        )
    )
)
)
Light (
    comment( Luz posterior roja Dim )
    Type ( 0 )
    Conditions (
        Headlight ( 2 )
        Unit ( 3 )
    )
    FadeIn ( 0.5 )
    FadeOut ( 0.5 )
    Cycle ( 0 )
    States ( 1
        State (
            Duration ( 0.0 )
            LightColour ( 80ff0000 )
            Position ( 0.93 1.97 -10.5054 )
            Azimuth ( -180 -180 -180 )
            Transition ( 0 )
            Radius ( 0.5 )
        )
    )
)
)
)
Comment(Solo usaremos las luces posteriores en caso de unidad reversible.
)
Light (
    comment( Luz posterior principal inferior Bright )
    Type ( 0 )
    Conditions (
        Headlight ( 3 )
        Unit ( 3 )
    )
    FadeIn ( 0.5 )
    FadeOut ( 0.5 )
    Cycle ( 0 )
    States ( 1
        State (
            Duration ( 0.0 )
            LightColour ( ffffffff )
            Position ( 0.01007 4.10572 -9.598 )
            Azimuth ( -180 -180 -180 )
            Transition ( 0 )
            Radius ( 0.50 )
        )
    )
)
)
)
)

```

```

    )
  )
)
Light (
  comment( Luz posterior principal superior Bright )
  Type ( 0 )
  Conditions (
    Headlight ( 3 )
    Unit ( 3 )
  )
  FadeIn ( 0.5 )
  FadeOut ( 0.5 )
  Cycle ( 0 )
  States ( 1
    State (
      Duration ( 0.0 )
      LightColour ( ffffffff )
      Position ( 0.01007 4.26114 -9.5982 )
      Azimuth ( -180 -180 -180 )
      Transition ( 0 )
      Radius ( 0.50 )
    )
  )
)
)Aquí se termina el bloqueo para luces posteriores )
)

```

Deseamos que el ordenador haga lucir los siguientes elementos para funcionamiento AI:

```

Proyeccion luz delantera
Luz delantera superior principal Bright
Luz delantera derecha intermitente Bright
Luz posterior roja Bright

```

Con esto será suficiente. La “Luz delantera izquierda intermitente Bright” se le ha “fundido” a la unidad dirigida por el ordenador. Si alguien desea “cambiar la bombilla”... ya lo sabe: Que opere de idéntica manera a como se va a explicar para la derecha. Y en el doble foco principal solo lucirá el superior.

Duplicaremos estos cuatro elementos “Light” a continuación del ultimo que tengamos definido y haremos las modificaciones que se muestran:

```

Light (
  comment( Proyeccion luz delantera )
  Type ( 1 )
  Conditions (
    Unit ( 2 )
    Control ( 1 )
    Service ( 2 )
    TimeOfDay ( 2 )
  )
  Cycle ( 0 )
  FadeIn ( 0.5 )
  FadeOut ( 0.5 )
  States ( 1
    State (
      Duration ( 0.0 )
      LightColour ( ffffffff )
      Position ( 0.0 3.5 18 )
      Transition ( 0 )
      Radius ( 400.0 )
      Angle ( 15.0 )
    )
  )
)
)
Light (
  comment( Luz delantera superior principal Bright )
  Type ( 0 )
  Conditions (
    Unit ( 2 )
    Control ( 1 )
    Service ( 2 )
    TimeOfDay ( 2 )
  )
  FadeIn ( 0.5 )
)

```



```

Conditions (
    Headlight ( )
    Unit ( )
    Penalty ( )
    Control ( )
    Service ( )
    TimeOfDay ( )
    Weather ( )
    Coupling ( )
)

```

Si no lo hacemos, es por comodidad de diseño y para ahorrarnos trabajo. Es habitual coger tan solo solemos la parte que nos interesa. Eso nos lleva a algunos detalles del estilo, por ejemplo, que si no definimos apartado para alumbrado AI, el programa cogerá por defecto todas las luces configuradas como Bright para los trenes dirigidos por el ordenador.

Las unidades dirigidas por nosotros deberíamos configurarlas como Control (2), pero no lo hacemos. En realidad no configuramos nada en este aspecto. Aceptamos los valores por defecto y lo cierto es que el programa funciona bien.

Podéis hacer una sencilla prueba vosotros mismos con la Dash en cualquier actividad nocturna. Si os fijáis el apartado Lights dentro del archivo .eng no trae definición de alumbrado para AI, y la locomotora alumbra a pesar de ello en modo AI. El alumbrado que nos muestra es el que corresponde a la posición Bright. Eliminar o anular mediante Comment's todos los elementos Light que estén configurados para alumbrado Bright. Veréis como ahora cualquier unidad Dash en funcionamiento AI va con todo apagado.

Con esto quiero llamar la atención sobre aspectos de configuración que aparentemente parecen lógicos pero que en el resultado final no obtenemos lo que deseamos, por lo cual para algunas cosas no tendremos más remedio que experimentar. Lo que hayamos podido configurar para AI, en ocasiones puede interferir en nuestra unidad y nosotros no tener el control real. Soy consciente que en el caso puesto como ejemplo anterior, el de especificar en todas nuestras unidades la variable Control (2) conlleva un trabajo laborioso (imaginaros tener que modificar todos los .eng y .wag con esta variable uno a uno), y si además, tal y como está funciona bien, razón de mas para no tocarlo. Pero en ocasiones tendremos que reparar en este detalle para lograr seguir teniendo nosotros el control de nuestra unidad y el ordenador el control de la suyas sin interferencias.

Por esta razón personalmente, el apartado Conditions, y bajo mi punto de vista, debería de llevar en cada momento el mínimo numero de valores a configurar en su interior, si con ello conseguimos el efecto deseado tanto para nuestras unidades como para las que maneja el ordenador.

Por este motivo, y en este caso, va a ser necesario configurar ambas luces intermitentes, así como también los dos focos principales delanteros de nuestra unidad con la variable Control (2) que la agregaremos en los cuatro apartados Light dentro de Conditions (No importa el orden en que vayan colocadas las ordenes dentro del apartado):

```

Conditions (
    Headlight ( 3 )
    Unit ( 2 )
    Control ( 2 )
)

```

En caso de no hacerlo, habrá una interferencia de ordenes entre nuestra unidad y la unidad AI ya que esta última no obedecerá correctamente la orden dada (No habrá "luz fundida" y además lucirán los dos focos centrales). Esto se debe a que el programa, si nosotros no especificamos quien maneja la unidad, va a leer para AI el valor en modo Bright de dicha luz intermitente. Si no tuviéramos definidas luces en modo Bright no se nos daría el problema anterior.

Hay que contar con que la Dash 9 que utilizamos para nuestro control, es la misma que usa el ordenador para su control. Si fueran diferentes locomotoras, estos inconvenientes desaparecerían ya que nuestra locomotora tendría una configuración y la dedicada a AI tendría otra distinta y específica para AI.

En esto el programa, por decirlo de alguna manera, presenta una ventaja (no todo tienen que ser inconvenientes ni defectos), que no es ni mas ni menos que en una misma locomotora poder obtener soluciones de alumbrado totalmente distintas dentro de la misma actividad.

La otra variable AI que consideramos en nuestro ejemplo es la de TimeOfDay, que hemos configurado con el valor 2 para que las luces AI estén encendidas durante la noche y se apaguen automáticamente durante el día.

Muchas de las variables como es el caso de TimeOfDay podemos usarlas también configuradas para unidades que se encuentran bajo nuestro control. El efecto que tienen es idéntico que cuando se emplean en unidades AI solo que de esta manera hemos “automatizado” el funcionamiento del alumbrado.

Como se puede observar hay muchas combinaciones, unas posibles y otras no tanto debido a las interferencias entre órdenes que se pueden producir. A partir de aquí ya entra en juego la imaginación de cada cual para sacar más rendimiento de este tema sobre alumbrado.

Comentarios Finales

Deseo haber os arrojado alguna luz, y nunca mejor dicho, sobre este tema del que sin duda compartimos una apasionante afición.

Cuando yo comencé con los trenes virtuales, la verdad es que desconocía por completo que hubiera un mundo tan amplio y tanta gente detrás. La mayor parte de la información venía y sigue viniendo en inglés, y muy poca o poquísima en español, salvo algunos estupendos trabajos, bajo mi punto de vista, y que la mayoría ya conoceréis a través de los diferentes foros.

También me he pasado muchas horas leyendo y probando una y otra vez, lo que me ha valido para ir sacando cosillas aquí y allá y las he recopilado en lo que habéis visto. Por eso deseo que este trabajo sea útil para todo aquel que lo consulte, ya que en cierto modo, una parte la habéis hecho vosotros sin daros cuenta y por eso espero vuestra colaboración para intentar completarlo con todo lo que no haya salido reflejado o se haya quedado en el tintero, y para corregir los errores que hubiera podido cometer.

Para finalizar, pedir disculpas si en algún sitio, sobre todo en la sección de ejemplos, “he metido la pata” al diseñar los sistemas de alumbrado. Si algo he logrado encontrar todavía, es la reglamentación, o el código español por el que se rigen nuestros maquinistas a la hora de hacer funcionar el alumbrado y hacerlo así más real en este mundo virtual.

No tengo más que comentar de momento, salvo recordaros que hagáis copias de seguridad de todos los archivos que vayáis a modificar por si es necesario dar marcha atrás y comenzar de nuevo. (Alguno pensará: ¡¡¡JO, y lo avisa ahora que ya lo tengo todo liado, y no hay por donde cogerlo!!!. Ya conocéis los viejos dichos: “El que avisa... no es traidor” y “Mas vale tarde... que nunca”. Sencillamente, si esta vez habéis metido la pata en algo, tened la seguridad que la próxima vez lo haréis como es debido y adoptareis las precauciones necesarias. Eso fijo).

Un saludo

Agradecimientos:

En general:

Por vuestro interés a todos los que habéis logrado llevar vuestra lectura hasta estas líneas y sacar algo en claro de todo lo expuesto anteriormente. Quizás, esta lectura haya resultado pesada y demasiado técnica, espero que no, por que no era mi intención.

En particular:

Otras Publicaciones: