



# Compilador Modular

MATERIA: COMPILADORES

RENÉ HAGI BOTELLO CASAS

LUIS ALBERTO PÉREZ CONTRERAS

## Índice

---

- Objetivo.
- Introducción.
- Herramientas.
- Visión general del compilador.
- Desarrollo.
  - Analizador Léxico.
  - Analizador Sintáctico.
    - Tipos de errores sintácticos y sus mensajes.
  - Analizador Semántico.
    - Tipos de errores sintácticos y sus mensajes.
- Interfaz Gráfica.
  - Características.
- Errores Lógicos.
- Lenguaje Dalek.
  - Introducción.
  - Instrucciones.
- Casos de Prueba.
- Instalación.
- Requisitos del Sistema.

## **Compilador Modular**

### **Objetivo:**

- Aplicar los conocimientos vistos en clase por medio de la realización de un compilador modular.
- Crear un lenguaje de programación simple y por módulos para que los niños puedan desarrollar una mentalidad lógica.

### **Introducción:**

Un compilador modular lo podemos tomar como que, cada una de sus palabras reservadas, son objetos visibles en la pantalla y que realiza una tarea única y, que en algunos casos, puede contener modificaciones simples de sus acciones. Con esto obtenemos una gran diversificación de acciones que se pueden realizar en el compilador, limitadas solo por la imaginación del usuario.

Para la realización de este proyecto se tomó como referencia la aplicación de nombre "Scratch", el cual se basa en un lenguaje de programación gratuito para la creación de historias, animaciones y juegos para niños (Lab L. K.-M., 2015).

### **Herramientas:**

- Diseño del compilador: Unity 3D
- Lenguaje para programar el compilador: C# y/o Javascript.

### **Visión general del compilador:**

El compilador se verá como un videojuego simple para niños donde puedan desarrollar su mentalidad lógica para resolver problemas simples; esto se hará por medio de un personaje principal llamado "Dalek" que quiere llegar de un punto 'A' a un punto 'B' y nosotros le daremos las instrucciones en forma de bloques que queremos que realice y, así, cumplir el objetivo.



De igual manera aunque cuenta con las acciones simples de un juego (avanzar, retroceder, girar a la derecha, girar a la izquierda, etcétera) se proveen varias tareas simples que, aunque algunas no ayudan en lo absoluto al terminar el nivel, harán movimientos o sonidos divertidos para ser más ameno la aplicación.

La interfaz gráfica contará con un “canvas” o contenedor de gráficos donde se mostrarán todas las posibles acciones que puede realizar el personaje; de ahí se pueden arrastrar al área de ejecución para colocarse en el orden deseado por el usuario para cumplir el objetivo.

Una vez dadas todas las instrucciones que se desean (o se puedan) se procede a la compilación y ejecución donde el personaje realizará los pasos en el mismo orden en que fueron dados.

Si el personaje llega al punto B el objetivo se cumplió, en caso contrario se retorna un mensaje de error de término del nivel.

### **Desarrollo:**













Como cualquier compilador lo dividiremos en sus secciones de Front-End:

- Análisis Léxico.
- Análisis Sintáctico.
- Análisis Semántico.

## Análisis Léxico (Entregable 28/04/2015)

Consta de la identificación de 100 palabras reservadas como token de acuerdo a las siguientes tablas:

1		El personaje salta hacia el frente		El personaje baila	11
2		El personaje hace girar su helice		El personaje suelta una bola de energia	12
3		El personaje enciende su luz		El personaje recoge una bola de energia	13
4		El personaje apaga su luz		El personaje emite un ladrido constante	14
5		El personaje avanza hacia el frente		El personaje emite un sonido de alarma	15
6		El personaje avanza hacia la derecha		Detiene el baile del personaje	16
7		El personaje avanza hacia la izquierda		El personaje emite un sonido chillante	17
8		El personaje avanza hacia atrás		Hace que se escuche la música de fondo	18
9		El personaje gira hacia la izquierda		Mute - Silencia la música de fondo	19
10		El personaje gira hacia la derecha		Detiene el baile del personaje	20

21		Detiene el movimiento de helice del personaje		El personaje genera 31 fuego
22		Detiene el sonido de alarma		El personaje genera 32 agua
23		Detiene el ladrido del personaje		El personaje genera 33 viento
24		Hace sonar un sonido de tambores una vez		El personaje se clona en un holograma 34
25		El personaje se despide de ti		El personaje te saluda 35
26		El personaje se duerme en su posición		
27		El personaje se despierta de su sueño		

PINTA DE COLOR EL SUELO EN				45	20	20	100
FORMATO RGB:				46	20	10	70
ID	R	G	B	47	40	80	205
36		0	0	48	40	70	190
37		0	80	49	40	50	150
38		0	70	50	40	20	130
39		0	50	51	40	10	100
40		0	20	52	40	0	70
41		0	10	53	60	80	205
42		20	80	54	60	70	190
43		20	70	55	60	50	150
44		20	50	56	60	20	130

<b>57</b>	60	10	100	<b>79</b>	140	50	150
<b>58</b>	60	0	70	<b>80</b>	140	20	130
<b>59</b>	80	80	205	<b>81</b>	140	10	100
<b>60</b>	80	70	190	<b>82</b>	140	0	70
<b>61</b>	80	50	150	<b>83</b>	150	0	0
<b>62</b>	80	20	130	<b>84</b>	160	0	70
<b>63</b>	80	10	100	<b>85</b>	160	80	205
<b>64</b>	80	0	70	<b>86</b>	160	70	190
<b>65</b>	100	80	205	<b>87</b>	160	50	150
<b>66</b>	100	70	190	<b>88</b>	160	20	130
<b>67</b>	100	50	150	<b>89</b>	160	10	100
<b>68</b>	100	20	130	<b>90</b>	180	0	70
<b>69</b>	100	10	100	<b>91</b>	180	80	205
<b>70</b>	100	0	70	<b>92</b>	180	70	190
<b>71</b>	120	80	205	<b>93</b>	180	50	150
<b>72</b>	120	70	190	<b>94</b>	180	20	130
<b>73</b>	120	50	150	<b>95</b>	180	10	100
<b>74</b>	120	20	130	<b>96</b>	200	80	205
<b>75</b>	120	10	100	<b>97</b>	200	70	190
<b>76</b>	120	0	70	<b>98</b>	200	50	150
<b>77</b>	140	80	205	<b>99</b>	200	20	130
<b>78</b>	140	70	190	<b>100</b>	200	10	100

Como vemos los primeros 35 módulos o palabras reservadas son acciones directas que puede hacer nuestro personaje o el ambiente mientras que las otras 65 corresponden al cambio de color del suelo en que se encuentre el personaje.

Si es posible, por aceptación del profesor, pondremos un "Color Picker" que acepte cualquier combinación de color RGB para que sea más dinámico el compilador, pudiendo llegar a tener 16'777,216 colores + 35 acciones, un total de 16'777,251 palabras reservadas que puede manejar.

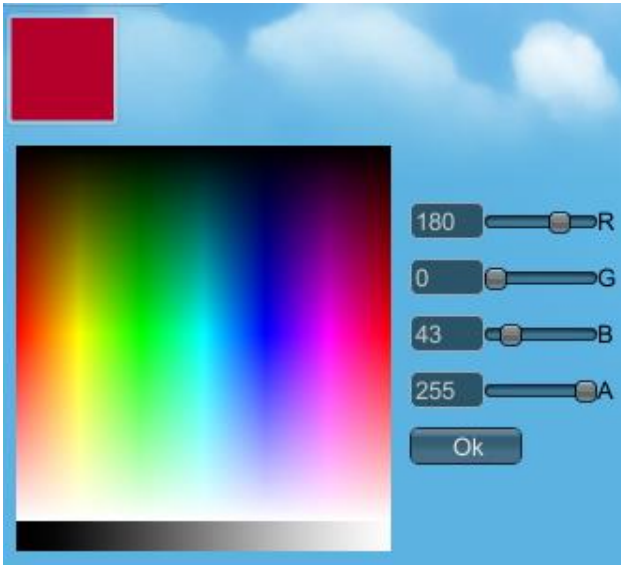


## Entregable final (después de 28/04/2015)

Una vez que se aceptó la propuesta se desarrolló el sistema completo de selección de color tal como se muestra a continuación:



El icono a la izquierda despliega un menú completo para la selección de color.



El recuadro superior es el bloque que se puede arrastrar al área de ejecución, mientras que el recuadro RGB permite escoger un color específico con poner el cursor encima o al escribir los valores RGBA (Rojo, Verde, Azul, Alfa) y presionar el botón "Ok".

## Análisis Sintáctico:

Por el tipo de compilador que tenemos es difícil de ver este análisis pero lo tomaremos en base a que nuestra palabra reservada de cambio de color necesita un ingreso correcto de valores en sus sobrecargas RGBA y, en caso de tener un error este mismo se debe detener el resto de la compilación y ejecución del programa.

## Tipos de errores sintácticos y sus mensajes:

- No ingresar nada (null): "No puede estar vacío el campo R."
- Ingresar una letra: "Se esperaba un número en el campo R."
- Ingresar un flotante: "Se esperaba un entero en el campo R."
- Ingresar un valor superior a 255: "Error en campo R: Solo se pueden valores entre 0 y 255."

\*Como vemos cada error es específico, por lo que se podrá saber exactamente qué campo es el que cuenta con un problema.



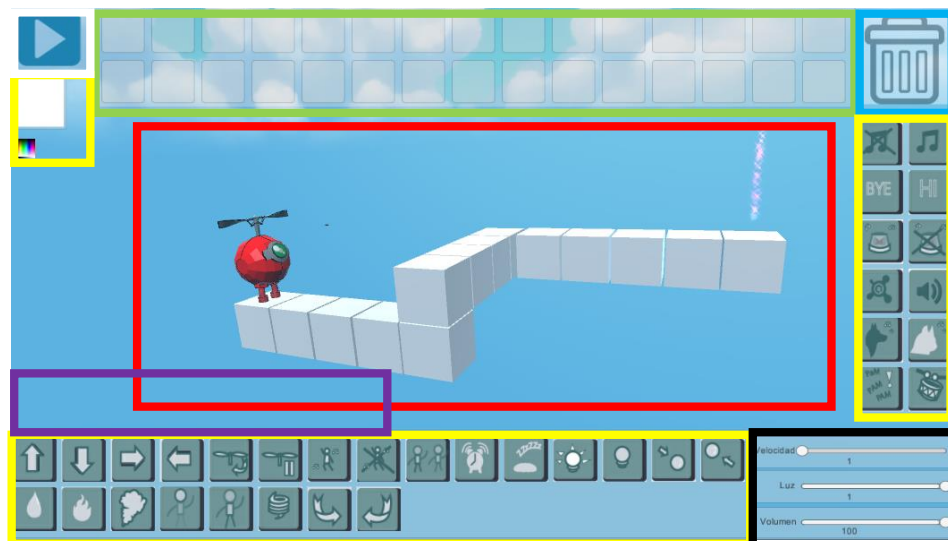
## Análisis Semántico:

Sabiendo que la semántica es el sentido que le damos a las palabras entonces nos damos cuenta que en nuestro lenguaje tiene algunas palabras de activación y desactivación de acciones por lo que entra el problema de intentar cancelar un comando si este no se ha activado antes; para esta situación entra el análisis semántico.

## Tipos de errores semánticos y sus mensajes:

- Desactivar las hélices sin haberlas prendido antes: "Las hélices no pueden apagarse antes de prenderse."
- Desactivar el baile sin haber empezado a bailar: "No puedes parar de bailar sino has empezado."
- Apagar la luz sin haberla prendido: "No puedes apagar la luz sino la has prendido."
- Hacerse visible cuando sin estar invisible: "No puedes aparecer si eres visible."
- Apagar la música si está no se prendió antes: "No puedes quitar la música si no está encendida."
- Intentar despertar si no estaba dormido: "No puedes despertar si ya lo estas."
- Apagar la alarma sin que esté rendida: "No puedes detener la alarma si no está sonando."
- Parar de ladrar sin estar ladrando: "No puedes parar de ladrar si no estás ladrando."

## Interfaz Gráfica:



## Características:




Corresponde al área de comandos donde se encuentran todas las posibles instrucciones de nuestro lenguaje; se subdividen en 3 tipos: acciones (en la parte inferior de la pantalla) que controlan habilidades específicas de Dalek (nuestro personaje principal) como el moverse, generar un objeto o elemento, girar, desaparecer, rotar, brincar, etcétera; sonidos (en la parte derecha de la pantalla) los cuales generan sonidos o audios completos que pueden sonar una vez o de manera constante hasta que se detenga con otro comando; y el de color, que permite modificar el color del suelo por medio del selector RGBA que se explicó anteriormente.



Son los sliders de configuración en tiempo real, los cambios en estas variables permite un ajuste rápido y sencillo de las acciones que pueden realizarse; sus características son las siguientes:

**Velocidad:** cambia la velocidad (delay) de ejecución, en otras palabras, es el retraso que existe entre que se ejecute un comando y otro; puede variar entre 1 y 5 segundos. En caso de tener un procesador más lento y que las acciones toman más tiempo de lo normal es recomendable aumentar el retraso puesto que si una acción tarda más y se superponen con otra puede resultar en una ejecución incorrecta del código (en caso del salto produce un error lógico).

**Luz:** cambia la intensidad de la luz que emite Dalek, su efecto sólo puede ser visto si se activa el comando de "prender luz" . Si se establece el valor en 0 y se activa el comando produce un error lógico que se muestra.

**Volumen:** como su nombre lo dice permite el cambio del volumen de los sonidos que produzca Dalek, en caso de tener un valor de 0 y activar alguno produce un error lógico.



Es el eliminador de comandos o instrucciones, en caso de tener un error y querer eliminar una instrucción simplemente se debe de arrastrar y soltar en esta área.



Main o área de ejecución, en estas casillas se ponen las instrucciones en el orden que se quieren ejecutar empezando en la esquina superior izquierda y yendo hacia la derecha como en la escritura occidental. Pueden intercambiarse las casillas de manera libre y los espacios en blanco son aceptados por el compilador aunque no es óptimo. Una consideración importante es que un espacio vacío no consume tiempo de ejecución por lo que no se pueden tomar como retraso.



(Contorno blanco) es el botón de compilación, una vez que se hayan corregido los todos los errores sintácticos la compilación inicia, en caso de encontrar un error semántico se detiene y lo muestra, si no hay ningún error prosigue a la ejecución de comandos. En caso de terminar la ejecución y que el programa no lleve a Dalek a su destino, si se vuelve a presionar el nivel se reiniciará para ingresar un código nuevo que funcione.



Siendo la parte central de la pantalla es donde se ve a nuestro personaje en un cierto nivel que tiene que superar dependiendo de nuestro programa; en tiempo de ejecución se van viendo sus acciones paso por paso. Si Dalek alcanza a llegar a donde se encuentra la estela el nivel es superado sino se manda un error lógico.



En esta área se desplegarán los errores que se vayan encontrando durante la compilación (sintácticos y semánticos) y lo de ejecución (lógicos).

### Errores lógicos:

Este compilador muestra algunos errores lógicos simples que pueden llegar a cometerse fácilmente:

- **Salto en el aire:** en caso de que nuestro personaje este en el aire (saltando o cayendo) e intente brincar mandará un mensaje de que no es posible para él: "No puedes saltar en el aire."

- **Salir del camino:** en caso de que nuestro personaje caiga al vacío por cualquier situación mandará el siguiente mensaje: "Caíste del camino, no puedes llegar a tu destino."
- **Fin del programa sin éxito:** en caso de terminar la ejecución del programa y Dalek no esté en su destino mostrará el mensaje: "Programa terminado, Dalek no llegó a su destino."
- **Luz con intensidad de 0:** si es activada la luz pero se tiene una intensidad de 0 manda el mensaje: "No podrás ver la luz si está al mínimo de intensidad."
- **Música con volumen en 0:** si se activa un sonido o audio pero el volumen está en el mínimo manda un mensaje: "No podrás escuchar la música o los sonidos sin volumen."



## Lenguaje "Dalek"









### Introducción:








Es un lenguaje para niños o inexpertos en la programación creado por 2 estudiantes de la Universidad Marista de Guadalajara para la clase de Compiladores del 6to semestre.









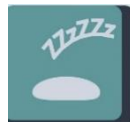
### Instrucciones:









Cuenta con 36 instrucciones simples para crear los programas que aunque ya se mostraron anteriormente se explicarán con más detalle:

id	Símbolo	Características
1		Dalek salta hacia el frente una distancia de un bloque de altura por un bloque hacia adelante (un bloque corresponde directamente a las dimensiones de los bloques donde está parado Dalek). No puede ejecutarse esta instrucción en el aire ni cuando está cayendo.
2		Dalek mueve sus hélices como si intentara volar, esta acción se repite hasta que le llegue el comando de detenerse. Si se quisiera realizar otra acción que interfiera con el movimiento de sus hélices (bailar, dormir, despertar, etcétera) esta última no tendrá efecto alguno.
3		Dalek prende su ojo como un foco para alumbrar su camino, aunque el escenario normalmente ya está alumbrado le da un



		divertido efecto al personaje; se puede controlar la intensidad de la luz desde el panel de configuración anteriormente mencionado. Su efecto se mantendrá hasta que se reciba el comando de apagado de luz.
4		Apaga la luz del ojo de Dalek, en caso de que no lo tenga prendido manda el correspondiente error semántico.
5		Dalek avanza en dirección donde esté mirando, si se encuentra con una pared la intentará atravesar pero normalmente no tiene suerte y se queda dónde estaba. Si se encuentra en el aire podrá dar un impulso pequeño para avanzar.
6		Dalek avanza hacia la derecha de donde esté mirando, si se encuentra con una pared la intentará atravesar pero normalmente no tiene suerte y se queda dónde estaba. Si se encuentra en el aire podrá dar un impulso pequeño para avanzar.
7		Dalek avanza hacia la izquierda de donde esté mirando, si se encuentra con una pared la intentará atravesar pero normalmente no tiene suerte y se queda dónde estaba. Si se encuentra en el aire podrá dar un impulso pequeño para avanzar.
8		Dalek avanza en dirección contraria a donde esté mirando, si se encuentra con una pared la intentará atravesar pero normalmente no tiene suerte y se queda dónde estaba. Si se encuentra en el aire podrá dar un impulso pequeño para avanzar.
9		Dalek rota 90 grados en el sentido de las manecillas del reloj, al momento en que modifica su vista hay que recordar que los comandos de movimiento cambian a como estaban antes. En caso de activar una animación esta se mantiene sin problemas.
10		Dalek rota 90 grados en el sentido contrario de las manecillas del reloj, al momento en que modifica su vista hay que recordar que los comandos de movimiento cambian a como estaban antes. En caso de activar una animación esta se mantiene sin problemas.

11		Dalek empieza a bailar sin parar hasta que se le mande el comando de detenerse, si se invierten los comandos se manda un error semántico. Aun cuando Dalek baila, él puede realizar todas las acciones que se le mande sin problemas aunque se vea ridículo moviéndose y bailando a la vez. De preferencia se debe de activar la música antes (sólo es una recomendación). En caso de activar otro efecto de movimiento repetitivo este último no tendrá efecto.
12		Dalek suelta una bola de energía y la deja suspendida donde se encuentra, no tiene ningún fin para esta versión de compilador y lenguaje pero puede ayudar a futuras mejoras del software para complicar los niveles.
13		Dalek recoge una bola de energía, en caso de no encontrar alguna se manda un error lógico; no tiene ningún fin para esta versión de compilador y lenguaje pero puede ayudar a futuras mejoras del software para complicar los niveles.
14		Dalek empieza a ladrar como un perro y no para hasta que se le mande la orden; este como la energía y los sonidos son para que en futuras versiones haya más dinamismo y poder mejorar los niveles.
15		Dalek suena una alarma como un minion de "Despicable Me 2" y no para hasta que se le mande la orden; este como la energía y los sonidos son para que en futuras versiones haya más dinamismo y poder mejorar los niveles.
16		Dalek suena la clásica alarma de Avast: "La base de datos de virus, ha sido actualizada", una vez por cada que se mande llamar; este como la energía y los sonidos son para que en futuras versiones haya más dinamismo y poder mejorar los niveles.
17		Dalek suena el sonido del juego Flappy Bird cuando pasa por un obstáculo, una vez por cada que se mande llamar; este como la energía y los sonidos son para que en futuras versiones haya más dinamismo y poder mejorar los niveles.
18		Dalek suena los primeros 20 segundos de la canción de "Celebrate" de Empire of the Sun y no para hasta que se le

		mande la orden; este como la energía y los sonidos son para que en futuras versiones haya más dinamismo y poder mejorar los niveles.
19		Dalek detiene la canción de "Celebrate" de Empire of the Sun.
20		Dalek detiene su baile.
21		Dalek detiene el movimiento de sus hélices.
22		Dalek detiene el sonido de alarma.
23		Dalek para de ladrar.
24		Dalek suena unos tambores como antes de una sorpresa, una vez por cada que se mande llamar; este como la energía y los sonidos son para que en futuras versiones haya más dinamismo y poder mejorar los niveles.
25		Dalek se despide del usuario con una voz divertida, una vez por cada que se mande llamar; este como la energía y los sonidos son para que en futuras versiones haya más dinamismo y poder mejorar los niveles.
26		Dalek cierra su ojo, baja sus hélices y se pone a dormir hasta que le manden el comando de despertarse. Aunque esté dormido, Dalek es sonámbulo y puede realizar las acciones que se le indiquen siempre y cuando no requieran sacarlo de su sueño como (bailar, mover hélices, etcétera); saltar, girar y moverse está permitido.

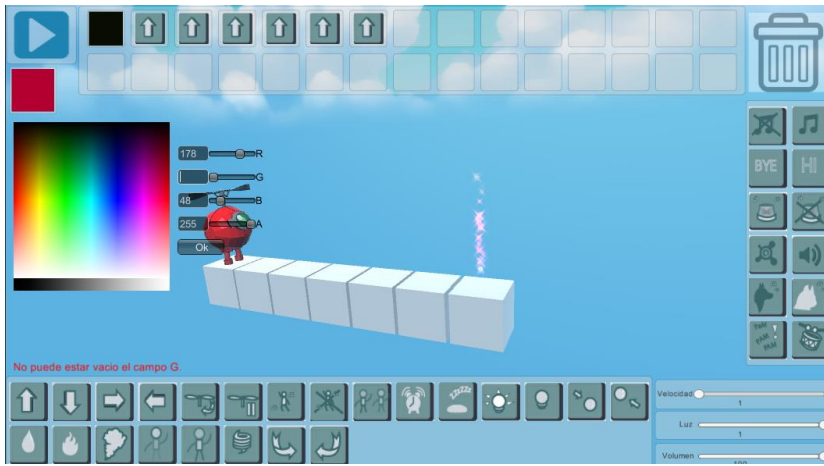
27		Dalek se despierta de su sueño y regresa a su estado natural despierto.
28		Dalek se hace invisible pero puede realizar cualquier acción aunque no lo veamos. Esta es otra de las acciones para complicar los niveles en un futuro.
29		Dalek se hace visible, en caso de ya serlo manda el mensaje de error semántico.
30		Dalek suena igual que el perezoso de la película de "Los Croods", una vez por cada que se mande llamar; este como la energía y los sonidos son para que en futuras versiones haya más dinamismo y poder mejorar los niveles.
31		Dalek produce una ilusión de fuego sobre sus hélices, no quema en lo absoluto pero podría servir para futuras versiones; las animaciones y movimientos solo producen un movimiento diferente de las partículas. El generar otro tipo de partículas no cancela su efecto.
32		Dalek produce una ilusión de agua sobre sus hélices, no moja en lo absoluto pero podría servir para futuras versiones; las animaciones y movimientos solo producen un movimiento diferente de las partículas. El generar otro tipo de partículas no cancela su efecto.
33		Dalek produce una ilusión de humo sobre sus hélices, no daña en lo absoluto pero podría servir para futuras versiones; las animaciones y movimientos solo producen un movimiento diferente de las partículas. El generar otro tipo de partículas no cancela su efecto.
34		Dalek se clona a sí mismo para distraer al enemigo (si existieran) no realizará ninguna acción futura del dalek original y las animaciones que tenga no surtirán efecto pero si el haber generado algún tipo de partícula.



35		Dalek le dice "hola" al usuario con una voz divertida, una vez por cada que se mande llamar; este como la energía y los sonidos son para que en futuras versiones haya más dinamismo y poder mejorar los niveles.
36		Dalek cambia el color del bloque donde se mantiene en pie dependiendo del que se haya seleccionado en el panel RGBA; si bien se cuentan con millones de colores, nuestros bloques no contienen todos por lo que el color se adapta lo mejor posible a la selección del usuario. Esta acción es meramente estética.

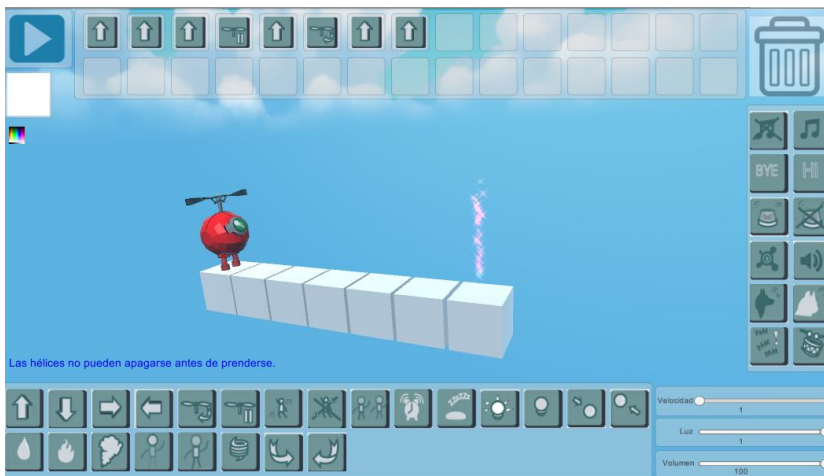
## Casos de prueba:

Caso 1: (erróneo) (Nivel 1) Error sintáctico en la elección de color.



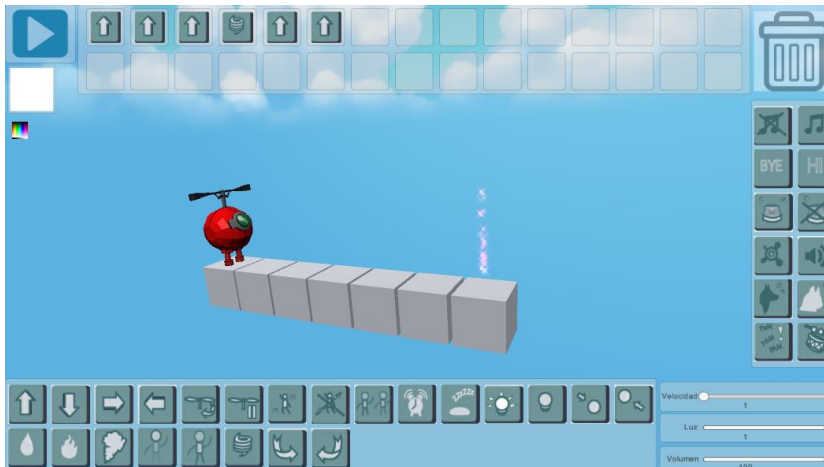
A pesar que el programa correría sin problemas el haber escogido un color inexistente manda el error en rojo.

Caso 2: (erróneo) (Nivel 1) Error semántico al mover las hélices.



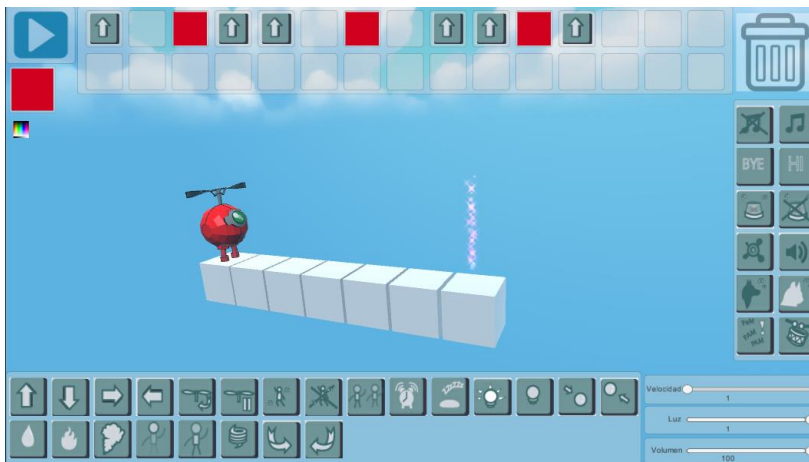
El programa igual que el anterior puede funcionar pero al intentar parar las hélices antes de encenderlas manda el error semántico y no pasa a la ejecución.

Caso 3: (correcto) (Nivel 1) Funcionamiento normal.



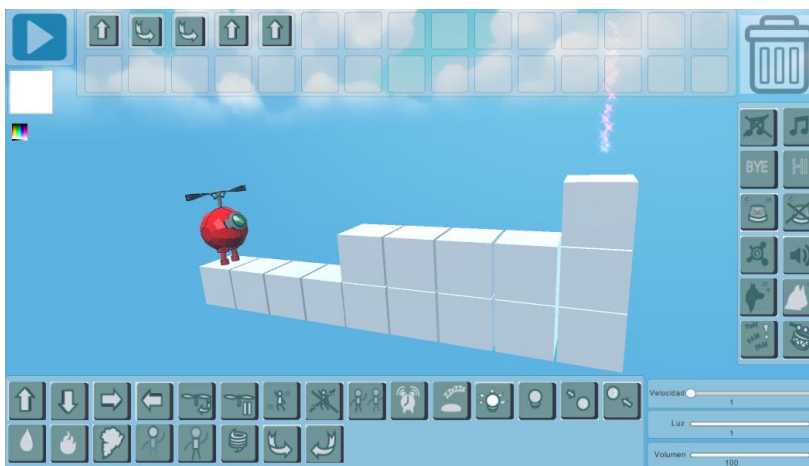
El programa funciona sin problemas y llegamos al final del camino correctamente para avanzar de nivel. Sólo se necesita avanzar 6 cuadros para finalizar.

Caso 4: (correcto) (Nivel 1) Funcionamiento normal.



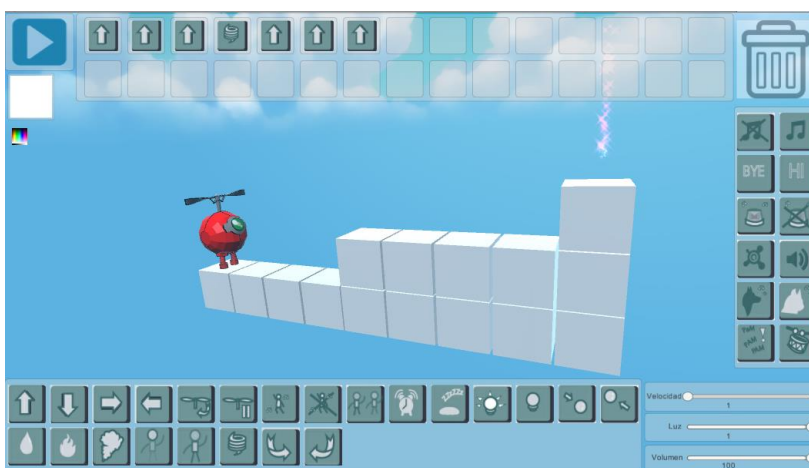
El programa funciona y termina bien, avanza las 6 casillas que se necesitan para terminar el nivel pero va pintando de rojo el suelo cada 2 casillas.

Caso 5: (erróneo) (Nivel 2) Error lógico por caída.



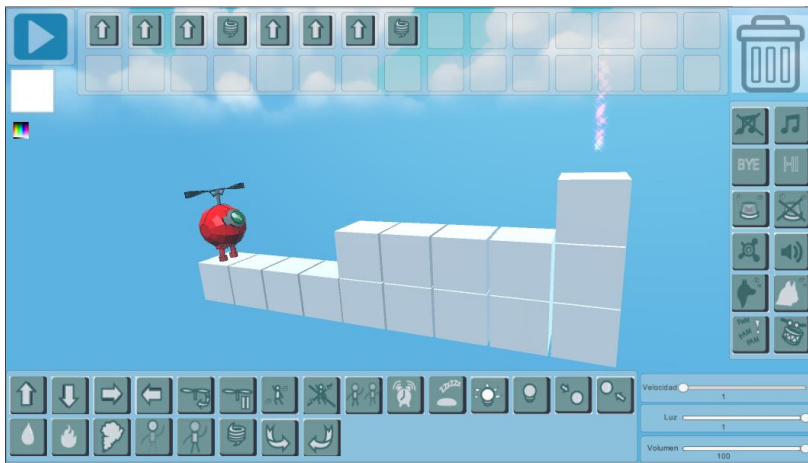
En el último paso Dalek caerá por el vacío y no llegará a su destino. Avanzará un paso, dará media vuelta y avanzará dos espacios hasta caer.

Caso 6: (erróneo) (Nivel 2) Error lógico por falta de pasos para llegar al destino.



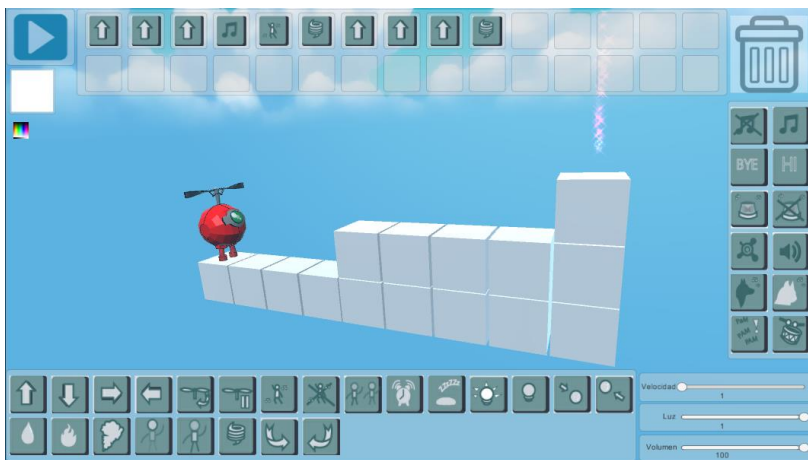
Como vemos el programa aparenta estar bien pero le falta un salto para poder llegar al destino, al no llegar despliega un error lógico

Caso 7: (correcto) (Nivel 2)



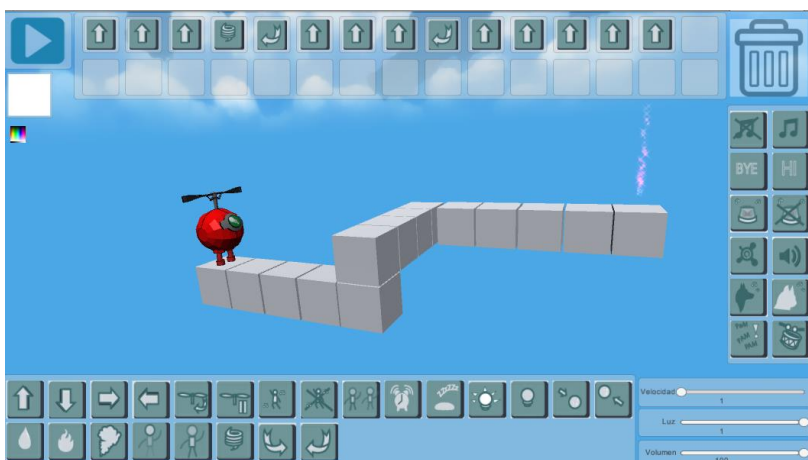
Corrigiendo el caso anterior se le agrega el salto para llegar al destino.

Caso 8: (correcto) (Nivel 2)



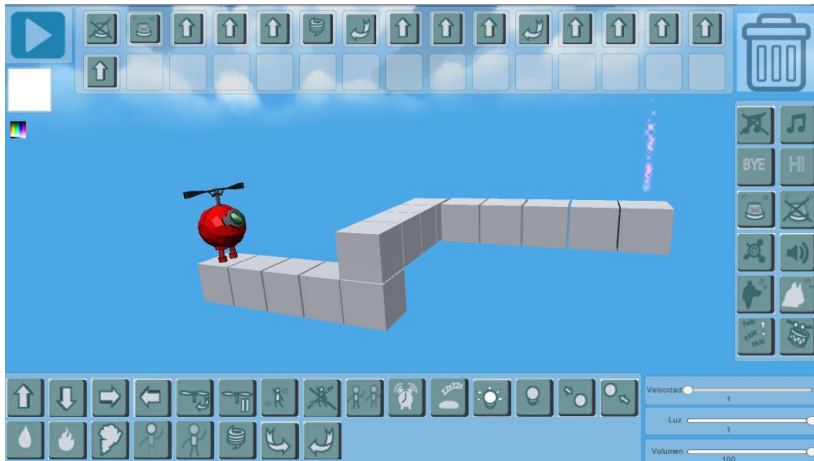
En este caso se cumple el objetivo pero con el extra de que baila y se oye la música al fondo después del tercer paso.

Caso 9: (erróneo) (Nivel 3) Error lógico por caída.



Siendo el nivel más complicado la primera vuelta que se da es incorrecta y lleva a Dalek al vacío.

Caso 10: (erróneo) (Nivel 3) Error semántico por alarma.



En los primeros pasos se invirtieron los símbolos de alarma por lo que muestra un error semántico y de todos modos no se ha resuelto el problema de la vuelta.

Caso 11: (correcto) (Nivel 3).



Se resuelve el problema de la vuelta del caso 9 y 10 para completar el nivel correctamente.

Caso 12: (correcto) (Nivel 3).



Se resuelve el problema de la alarma y aunque hay un espacio vacío se sabe que no hay ningún problema.

**Instalación:**

Ejecutar el archivo "Dalek Compiler.exe";

**Nota: Mantener en la misma carpeta el ejecutable "Dalek Compiler.exe" y la carpeta "Dalek Compiler\_Data".**

**Requisitos del sistema:**

- SO: Windows 7+ 32bits.
- Tarjeta de vídeo: con capacidades para DirectX 9 y con soporte para resoluciones 16:9 (1280x720, 1360x768 o 1366x768).
- Memoria RAM: mínimo 1 GB.
- Procesador CPU: compatible con el conjunto de instrucciones SSE2. Probado y funcionando en Intel i3 y AMD E2.