Zycars: Juego de conducción 2D

Ingeniería Técnica en Informática de Sistemas
José Jesús Marente Florín

Universidad de Cádiz

Septiembre 2011

Índice

ÍNDICE

Introducción

Zycars

- Juego de conducción en dos dimensiones con vista cenital
- Varios modo de juego
- Uso de ítem durante las carreras

¿Por qué este proyecto?

- Muy pocos juegos libre con las mismas características
- Interés por el mundo de los videojuegos
- Cursar la asignatura de Diseño de Videojuegos aumentó el interés por el desarrollo de estos
- Contribuir al mundo del software libre

Introducción

Objetivos

- Realizar un juego de coches completamente funcional
- Juego para todo tipo de jugadores
- Dificultad exigente
- Competidores aceptables, que proponga un desafío
- Facilmente ampliable

No es un simulador

- Movimiento básico de los coches
- La colisiones se corrigen de forma sencilla
- Arcade

ÍNDICE

Modos de juego

Carrera rápida

- Jugador contra tres oponentes
- Número de vueltas deseadas
- Una única carrera.

Campeonato

- Jugador contra tres oponentes
- Cuatro circuitos a completar
- Número de vueltas deseadas
- Puntuación

Contrarreloj

- Jugador compite solo
- Sólo ítems de turbo
- Tres vueltas al circuito

Elementos del juego

Personajes

- Variedad
- Distintas características (tamaño, velocidad...)
- Ampliables

Bolas de ítems

- A lo largo de todos los circuitos
- Proporcionan un ítem aleatorio

Tipos de ítems

- Ataques a distancia
- Obstáculos
- Velocidad

Colaboración y Recursos

Diseño Gráfico

Se ha contado con la colaboración de David Nieto Rojas, quien ha realizado todo el apartado gráfico del juego.

Música

Toda la música usada se ha obtenido de Jamendo. Grupos:

- Bob Wizman
- Pirato Ketchup
- Los Cadaver
- The Wavers
- Zamalska

jamendo
open your ears

ÍNDICE

PLANIFICACIÓN

PROPUESTA DEL PFC

A finales de Mayo del 2010 se comenzó a plantear que se podía realizar como proyecto fin de carrera. Tuvo lugar las reuniones con el tutor, con el fin de obtener distintas ideas. Finalmente entre todas las ideas propuestas se decidió realizar este proyecto.

TIEMPO DE DESARROLLO

En septiembre de 2010 se comenzó el desarrollo, acabando aproximadamente a finales de Junio de 2011.

Planificación

Fases

Durante el periodo de desarrollo tuvieron lugar las distintas fases:

- **Fase de análisis**: indentificación de las necesidades del software.
- **Fase de diseño**: diseño de todo el sistema.
- **Fase de aprendizaje**: familiarización con el lenguaje python y la biblioteca pygame.
- **Fase de desarrollo**: implementación de todo los obtenido en la fase de diseño. Fase más larga.
- **Pruebas y correcciones**: pruebas necesarias para comprobar el correcto funcionamiento. En paralelo a la fase de desarrollo
- **Redacción de la memoria**: realización de la memoria final.

Diagrama de Gantt I

WBS	Nombre	Trabajo
1	Planificación	2d
2	**Análisis**	**5d**
2.1	Recopilación de requisitos	3d
2.2	Recursos necesarios	2d
3	**Aprendizaje**	**20d**
3.1	Python	15d
3.2	Pygame	5d
4	**Desarrollo**	**257d**
4.1	Necesidades básicas del mot	30d
4.2	Comportamiento básico del	15d
4.3	Carga de Escenarios	25d
4.4	Colisiones básicas	35d
4.5	**Menús**	**26d**
4.5.1	Principal	5d
4.5.2	Opciones	6d
4.5.3	Selección personaje	6d
4.5.4	Selección circuito	7d
4.5.5	Pausa	2d
4.6	Control de carrera	15d
4.7	**Items**	**21d**
4.7.1	Bola de items	3d
4.7.2	Misil	3d
4.7.3	Bola	3d
4.7.4	3 Misiles	3d

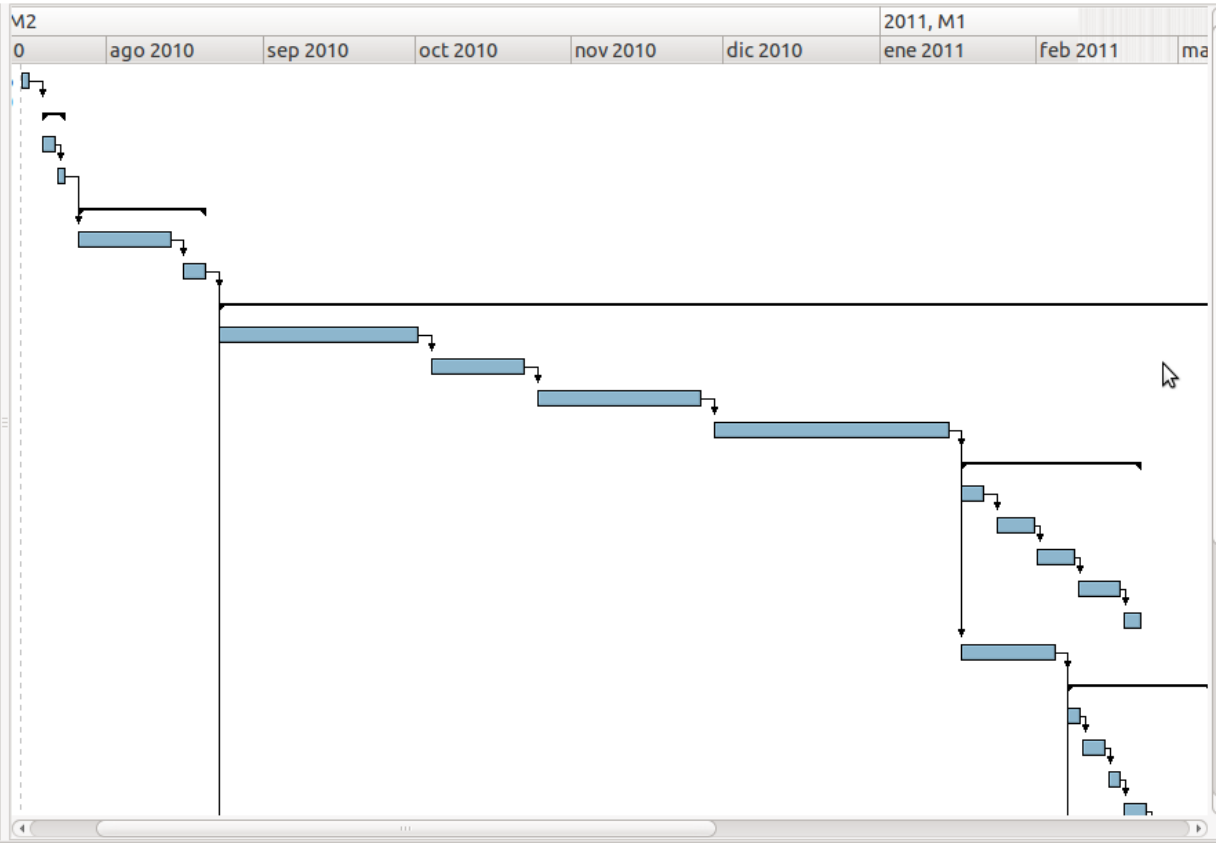
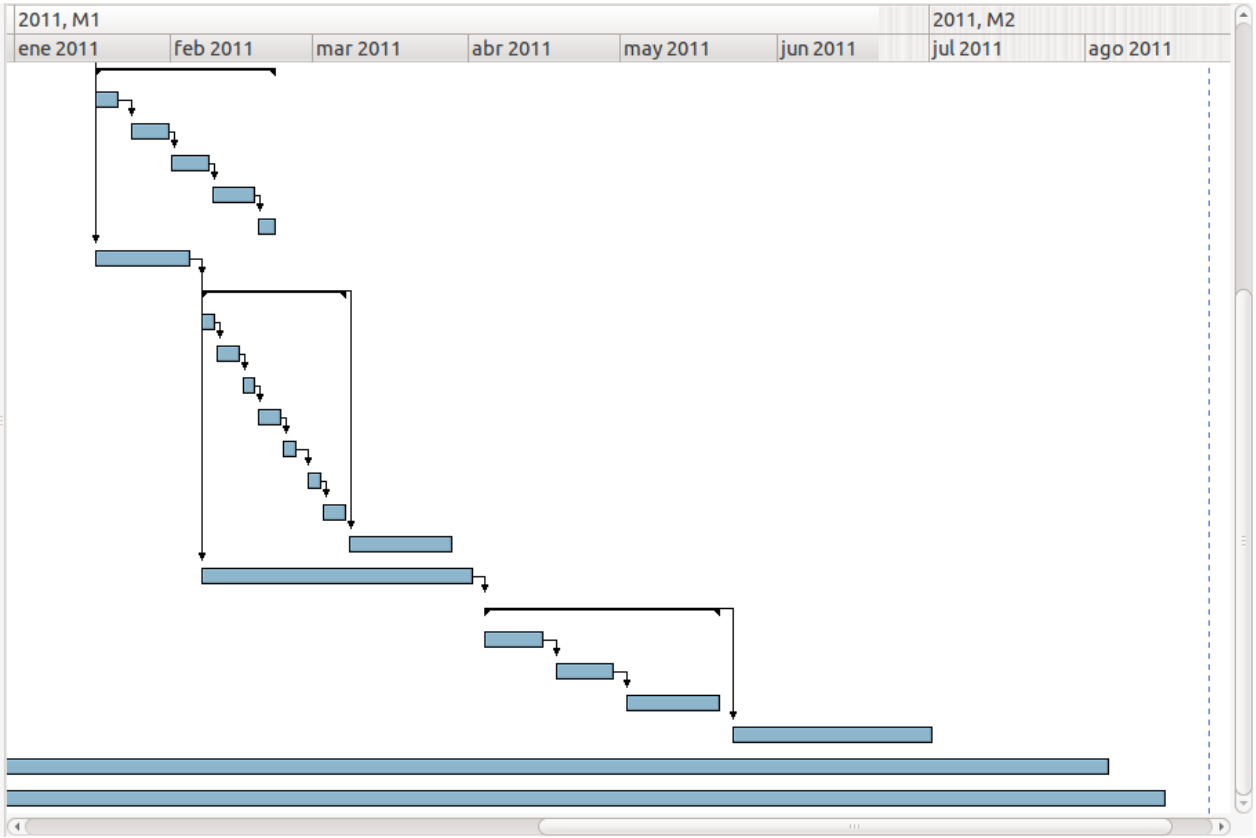


DIAGRAMA DE GANTT II

WBS	Nombre	Trabajo
4.5	**Menus**	**26d**
4.5.1	Principal	5d
4.5.2	Opciones	6d
4.5.3	Selección personaje	6d
4.5.4	Selección circuito	7d
4.5.5	Pausa	2d
4.6	Control de carrera	15d
4.7	**Items**	**21d**
4.7.1	Bola de items	3d
4.7.2	Misil	3d
4.7.3	Bola	3d
4.7.4	3 Misiles	3d
4.7.5	Aceite	3d
4.7.6	Chicle	3d
4.7.7	Turbo	3d
4.8	HUD Carrera	15d
4.9	Inteligencia Artificial	40d
4.10	**Modos de juego**	**35d**
4.10.1	Carrera rápida	10d
4.10.2	Contrarreloj	10d
4.10.3	Campeonato	15d
4.11	Creación de circuitos	30d
5	Pruebas y soluciones de bugs	250d
6	Redacción de la memoria	257d



ÍNDICE

Separar datos del código

En todo momento se ha procurado separar todos los datos de los personajes, circuitos y menús, del código fuente.

Ventajas

- No es necesario saber programar para realizar cambios sobre cualquier parámetro.
- Cualquier persona puede ampliar el juego con nuevos personajes y nuevos circuitos, siguiendo los manuales creados para ello.

Solución

- Todo se lee de ficheros XML

Formato de circuitos

Mapas de tiles

Tile: imagen cuadrada, rectangular o hexagonal, utilizada para generar imágenes de mayor complejidad.

Editor de mapas: Tiled

Proporcionaba todas las necesidades básicas, como una sencilla edición y creación de niveles, así como la gestión de capas, para poder poner elementos en el circuito a un nivel superior o inferior.
Para ello se debía crear una imagen con todos los tiles que compondrían un circuito (tileset).
Genera como resultado un XML.

Inconveniente

No permitia indicar de forma sencilla que tiles eran atravesables, colisionables o de cualquier otro tipo.

Formato de circuitos

Solución

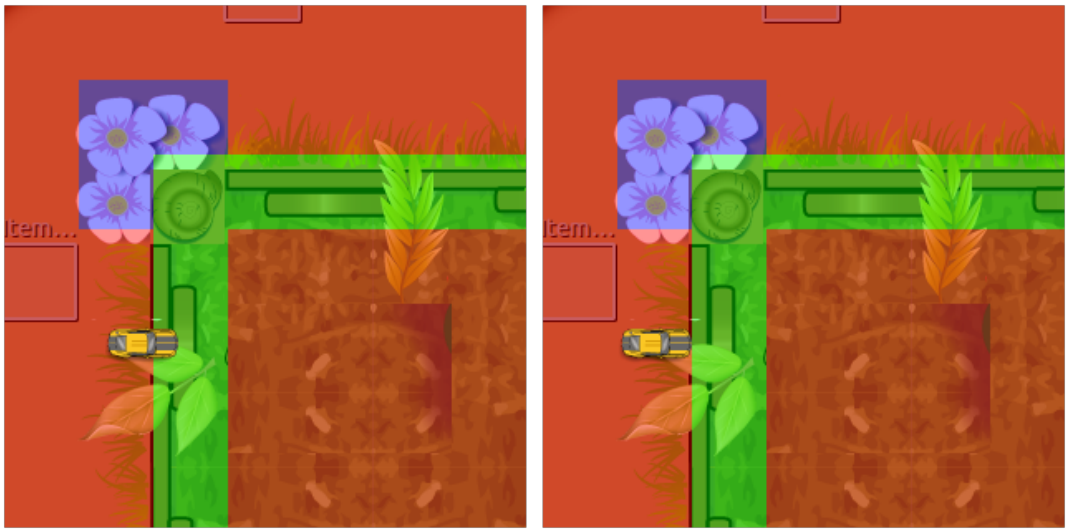
Una imagen extra con las mismas características, donde los tiles sera de un único color, en función del tipo que estos sean.

Colisiones

Una de las cosas más básicas en cualquier tipo de juego.

Colisión con el escenario

- Detectamos si atravesamos algún tile no atravesable
- Si es así corregimos la posición del coche en según la dirección, sentido y lado del tile por el que colisione
- En el caso de que el tile sea de tipo realentizador, diminuimos la velocidad del coche



Colisiones

Colisión entre vehículos

- De forma similar a la colisión con el escenario
- Cuando se detecta la colisión se corrige la posición de los vehículos, en función la dirección, sentido y lado del tile por el que colisionen

Colisión entre vehículos e ítems

- Si es un ítem de ataque a distancia, destruiremos dicho ítem y cambiaremos el estado del coche con el que colisione
- Si el ítem es un obstaculo, cambiaremos el estado del coche en función del tipo de ítem

Inteligencia artificial

Otro de los aspectos más importante de un videojuego de las características de Zycars, es la inteligencia artificial, ya que en dos de los tres modos de juegos disponibles el objetivo es obtener la mejor clasificación posible, por delante de los demás coches controlados por el ordenador

Habilidades

- Realización del recorrido: debe ser capaz de realizar los recorridos de los circuitos.
- Lanzamiento de ítems: también debe poder usar los ítems que reciba de las bolas de ítems.

Realización del recorrido. Algoritmo A*

Aprovechando que tenemos un circuito creado por tiles y que podemos saber en todo momento en el tile actual que se puede encontrar cualquiera de los competidores, se decidió implementar el algoritmo de búsqueda A*.

Objetivo

Buscar el camino más corto y óptimo, en el caso de que exista, desde un nodo origen, hasta un nodo destino. A la hora de buscar dicho camino se tienen en cuenta factores como, el valor heurístico que poseen cada uno de los nodos, así como el coste real del recorrido.

Parámetros

Los parametros que se tienen en cuenta en cada uno de los nodos.

- h'(n) es el valor heurístico del nodo actual n, hasta el final
- g(n) el coste real del camino desde el origen al nodo actual
- Función de evaluación: f(n) = g(n) + h'(n)

REALIZACIÓN DEL RECORRIDO. ALGORITMO A*

ESTRUCTURAS DIFERENCIADAS

- Lista de abiertos: nodos por los que aún no se han pasado
- Lista de cerrados: nodos por los que ya se han pasado

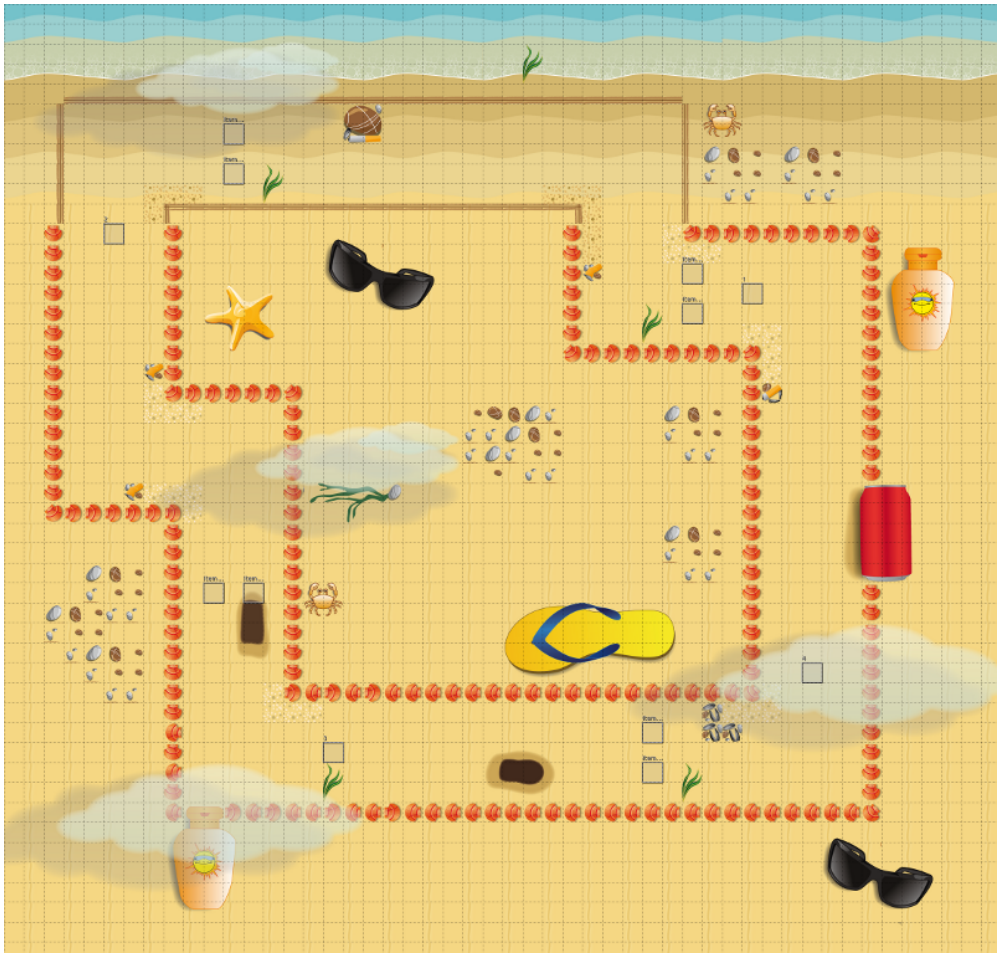
FUNCIONAMIENTO

Partiendo de un nodo en el que nos encontramos actualmente:

1. Obtenemos vecinos
2. Comprobamos que no esten en abiertos ni cerrados
3. Si alguno esta en abiertos, comprobamos su f(n), si es menor lo sustituiremos
4. Introducimos en abiertos los que cumplan las condiciones
5. Obtener de abiertos el nodo que tenga un f(n) menor y comenzamos de nuevo todo el proceso.
6. Una vez lleguemos al nodo objetivo, detenemos la búsqueda y devolvemos el camino completo.

REALIZACIÓN DEL RECORRIDO. ALGORITMO A*

Aplicación en Zycars

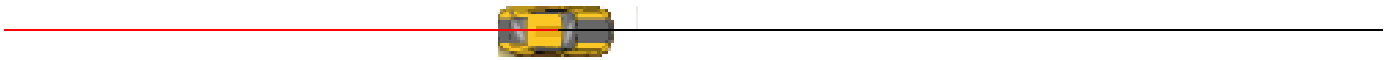


Lanzamientos de ítems

Capaz de lanzar los ítems disponibles a los largo del juego, según las distintas situaciones en la que se encuentre.

Solución

Se eligió una forma muy sencilla y eficiente a la hora de realizarlo. Para ello cada vehículo controlado por la inteligencia artificial, tiene tanto un segmento que va desde el centro del coche hacia unos píxeles por delante de la posición actual del vehículo, como otro segmento que también va desde el centro pero uno píxeles atrás de la posición del vehículo.



ÍNDICE

Herramientas

Lenguaje de programación: Python

Oportunidad perfecta para aprender un nuevo lenguaje de programación. Durante el curso 2009/2010, se conoció bastante bien dicho lenguaje. Entre sus principales características:

- Lenguaje interpretado de alto nivel
- Tipado dinámico y multiplataforma
- Multiparadigma, soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional

Destacar que se han obtenido unos resultado muy satisfactorios y ha cumplido todas las expectativas esperadas.

HERRAMIENTAS

BIBLIOTECA GRÁFICA: PYGAME

Wrapper de la biblioteca SDL, de C/C++, para Python, por lo que tiene todas las virtudes de dicha biblioteca:

- Multiplataforma compatible con Microsoft Windows, GNU/Linux, Mac OS y QNX.
- Muy completa, manupilacion de imágenes 2D, y gestión de sonido, música y la entrada estándar del sistema.
- Usada durante el desarrollo de la asignatura de Diseño de Videojuegos, se conocen todas sus características muy bien.

Herramientas

Analizador de código: Pylint

El código implementado debía seguir un estándar uniforme y que estuviera exento de errores o signos de mala calidad. Para ello se usó la herramientaPylint.
Analiza el código Python en busca de errores y señales de mala calidad.

Sistema de control de versiones: Subversion

Todo el código y recursos de Zycars está alojado en el sistema que proporciona Google Code, bajo el sistema de control de versiones subversion.
Nos permite:

- Control de todas las versiones
- Visualizar todos los cambios
- Volver a versiones anteriores

Herramientas

Documentación del código: Doxygen

- Permite la documentación sencilla y legible de todo el código
- Generando la documentación en varios formatos como puede ser HTML o PDF.

Para python existe la herramienta Doxypy, que nos permite usar la convención de comentarios de Python y adaptarlos a Doxygen.

Editor de mapas: Tiled

Editor de mapas de tiles de propósito general, escrito en C++, usando la librerías gráficas QT.

- Fácil de usar
- Flexible para trabajar con distintos motores de juegos
- Software libre

ÍNDICE

Conclusiones

Cumplimiento de objetivos

- Todos los objetivos marcados al inicio del PFC han sido cumplidos.
- Más duración de la esperada
- Contribución al mundo del software libre
- Juego de coches en 2D totalmente funcional

Valoración personal

- Enfrentamiento a un proyecto complejo en solitario
- Aprendizaje de nuevas herramientas
- Puesta en práctica de conocimientos adquiridos

Índice

Mejoras y Ampliaciones

Posibles mejoras y ampliaciones:

- **Modo dos jugadores**: nuevo modo de juego que nos permitiera jugar contra otra persona en el mismo ordenador. Pantalla quedaría dividida en dos.
- **Modo en red**: modo de juego para jugar en red contra otros oponentes. Más conveniente que el modo de dos jugadores, ya que dos personas jugando en un mismo ordenador puede llegar a ser incomodo.
- **Soporte para varias resoluciones**: cómodo para personas con pantalla muy pequeñas, como usuarios de netbooks, o también para persona con grandes resoluciones que desean una ventana de juego mayor.
- **Grabación de las mejores vueltas**: opción que permitiera grabar la vuelta más rápida de cada uno de los circuitos, almacenándolas en un fichero, y poder visualizarlas posteriormente.

ÍNDICE

BIBLIOGRAFÍA RECOMENDADA


Página de *Python*
http://www.python.org/


Página oficial sobre *Pygame*
http://www.pygame.org/


Larman, Craig
Applying UML and Patterns, 3ª Edición. Prentice Hall, 2004.


Pilgrim, Mark
Dive into Python. Appress, 2004.

DEMOSTRACIÓN

Demostración de Zycars

Esto es todo

Gracias por su atención

¿Preguntas?

http://code.google.com/p/zycars/