

# ZYCARS: JUEGO DE CONDUCCIÓN 2D

Ingeniería Técnica en Informática de Sistemas  
José Jesús Marente Florín

Universidad de Cádiz

Septiembre 2011

# ÍNDICE

- 1 INTRODUCCIÓN
- 2 DESCRIPCIÓN
- 3 CALENDARIO
- 4 IMPLEMENTACIÓN
- 5 HERRAMIENTAS
- 6 CONCLUSIONES
- 7 BIBLIOGRAFÍA

# ÍNDICE

- 1 INTRODUCCIÓN
- 2 DESCRIPCIÓN
- 3 CALENDARIO
- 4 IMPLEMENTACIÓN
- 5 HERRAMIENTAS
- 6 CONCLUSIONES
- 7 BIBLIOGRAFÍA

# INTRODUCCIÓN

## JUEGOS DE CONDUCCIÓN

- Objetivo: llegar a la meta
- Zonas diferenciadas
- Adictivos
- Para todo tipo de jugadores
- Variados modos de juego
- No pasan de moda



Super Sprint - Atari (1986)



Micromachines - NES (1991)

# INTRODUCCIÓN

## ZYCARS

- Juego de conducción en 2D con vista cenital
- Tres modo de juego
- Uso de ítem durante las carreras



Toy Cars - Xbox 360 (2011)

## ¿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
- Dificultad progresiva (adicción por aprendizaje)
- Competidores aceptables, que proponga un desafío superable
- Fácilmente ampliable

## NO ES UN SIMULADOR

- Arcade, prima la diversión
- Coches fáciles de manejar
- Colisiones sólo paran a los coches (estrategia)



Super Mario Kart -  
SNES (1992)

# ÍNDICE

- 1 INTRODUCCIÓN
- 2 DESCRIPCIÓN
- 3 CALENDARIO
- 4 IMPLEMENTACIÓN
- 5 HERRAMIENTAS
- 6 CONCLUSIONES
- 7 BIBLIOGRAFÍA

# 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 colaborado en el apartado gráfico del juego.

### MÚSICA

Se usa música libre adecuada para el videojuego. Grupos:

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



# ÍNDICE

- 1 INTRODUCCIÓN
- 2 DESCRIPCIÓN
- 3 CALENDARIO
- 4 IMPLEMENTACIÓN
- 5 HERRAMIENTAS
- 6 CONCLUSIONES
- 7 BIBLIOGRAFÍA

# PLANIFICACIÓN

## TIEMPO DE DESARROLLO

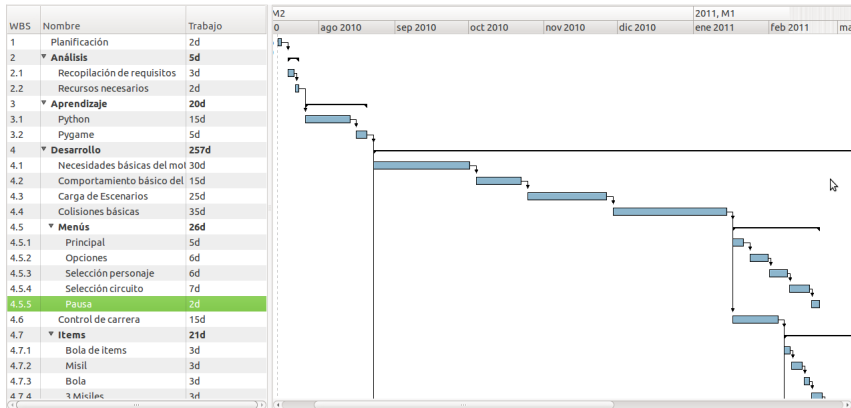
De septiembre de 2010 a septiembre de 2011.

## FASES

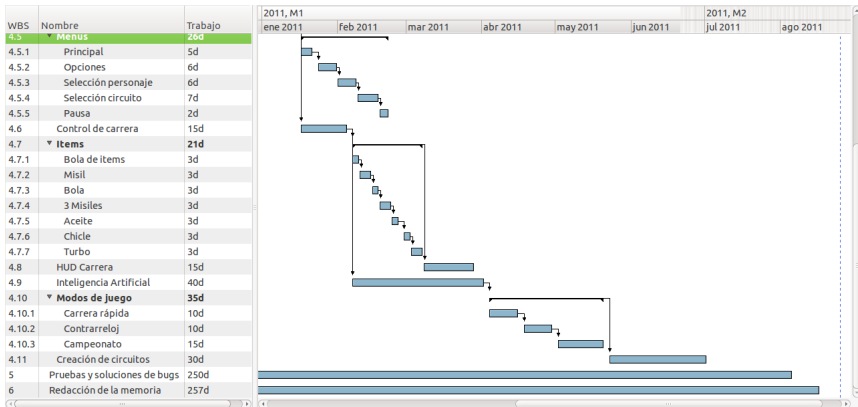
Durante el periodo de desarrollo tuvieron lugar las distintas fases:

- **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 lo 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



# DIAGRAMA DE GANTT II



# ÍNDICE

- 1 INTRODUCCIÓN
- 2 DESCRIPCIÓN
- 3 CALENDARIO
- 4 IMPLEMENTACIÓN**
- 5 HERRAMIENTAS
- 6 CONCLUSIONES
- 7 BIBLIOGRAFÍA

## SEPARAR DATOS DEL CÓDIGO

Desacople código / datos (personajes, circuitos, menús, etc).

### 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

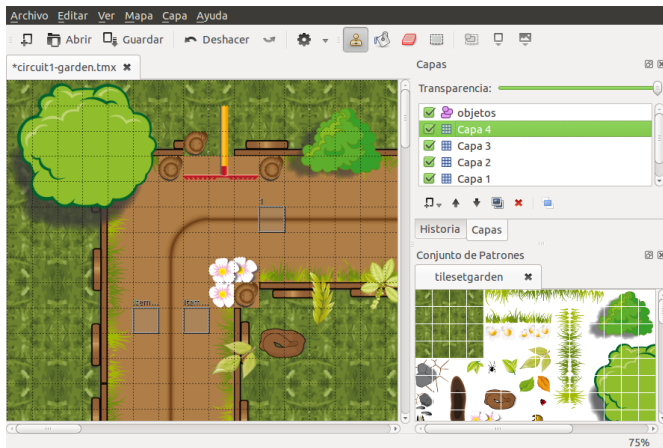
**<?xml?>**



# FORMATO DE CIRCUITOS

## MAPAS DE TILES

Tile: imagen cuadrada usada para generar imágenes de mayor complejidad.  
Usor del editor de mapas Tiled.



# FORMATO DE CIRCUITOS

## INCONVENIENTE

No permite indicar los tipos de los tiles.

## SOLUCIÓN

Usar una imagen extra.

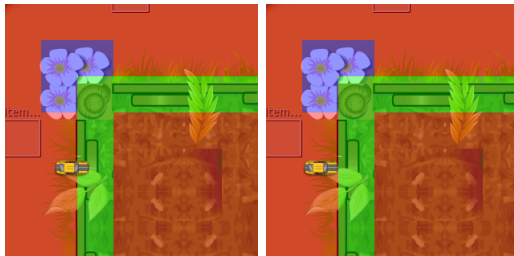


# COLISIONES

Una de los aspectos más importantes en este tipo de juegos.

## 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 ralentizador, disminuimos 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 por el que colisionen
- Podemos usar nuestro coche para evitar adelantamientos

## COLISIÓN CON ÍTEMS DE ATAQUE

- Se destruye el ítem y se cambia el estado del coche que colisiona

## COLISIÓN CON OBSTÁCULOS

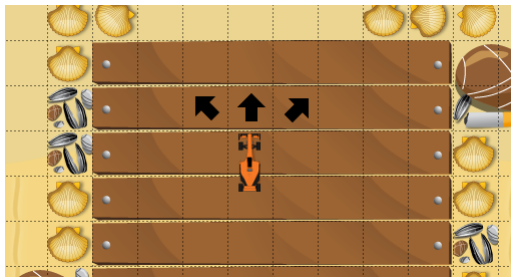
- Cambia el estado del coche en función del tipo de obstáculo (atravesable o no)

# INTELIGENCIA ARTIFICIAL

Aspectos muy importante en videojuego de las características de Zycars: interviene en en dos de los tres modos de juegos.

## 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\*

Teniendo un circuito de tiles podemos realizar búsquedas de caminos a través de estos.

### OBJETIVO

Buscar el camino más corto y óptimo desde un nodo origen, hasta un nodo destino. Se tienen en cuenta factores como el valor heurístico de los nodos, así como el coste real del recorrido.

### PARÁMETROS

Los parámetros que se tienen en cuenta en la búsqueda:

- $h'(n)$  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:  $f(n) = g(n) + h'(n)$

### ESTRUCTURAS

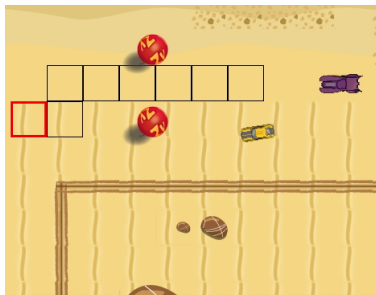
- Lista de abiertos: nodos por los que aún no se han pasado
- Lista de cerrados: nodos por los que ya se han pasado

# REALIZACIÓN DEL RECORRIDO. ALGORITMO A\*

## FUNCIONAMIENTO

Partiendo del nodo actual:

- 1 Obtenemos vecinos
- 2 Si no están en abiertos ni cerrados y los metemos en abiertos
- 3 Obtenemos de abiertos el nodo con menor  $f(n)$  y comenzamos de nuevo
- 4 Una vez en el nodo objetivo, detenemos la búsqueda y devolvemos el camino



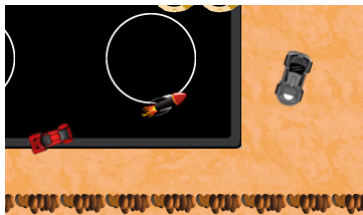
## LANZAMIENTOS DE ÍTEMS

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

### SOLUCIÓN

Cada vehículo tiene dos segmentos:

- Delantero: comprueba si algún oponente está delante para lanzar ítem
- Trasero: verifica la parte trasera en busca de algún oponente para dejar un obstáculo





# RECOPILACIÓN

## ZYCARS

- Multiplataforma
- 12 circuitos
- 3 modos de juego
- 3 campeonatos
- 7 personajes
- 6 ítems
- Circuitos ampliables con tiled
- Personajes ampliables
- Dificultad adecuada



# ÍNDICE

- 1 INTRODUCCIÓN
- 2 DESCRIPCIÓN
- 3 CALENDARIO
- 4 IMPLEMENTACIÓN
- 5 HERRAMIENTAS**
- 6 CONCLUSIONES
- 7 BIBLIOGRAFÍA

# HERRAMIENTAS

## LENGUAJE DE PROGRAMACIÓN: PYTHON

Entre sus principales características:

- Sintaxis limpia y que favorece un código legible.
- Multiplataforma



## BIBLIOTECA GRÁFICA: PYGAME

Conjunto de módulos de Python para la creación de videojuegos:

- Multiplataforma
- Muy completa (imágenes 2D, sonido, música y entrada estándar)



# HERRAMIENTAS

## ANALIZADOR DE CÓDIGO: PYLINT

Analiza el código Python en busca de errores y señales de mala calidad. La nota obtenida en el código del proyecto es de 8.25 sobre 10.

## FORJA DEL PROYECTO

Alojado en el sistema que proporciona Google Code, bajo el sistema de control de versiones subversion.

- Pública
- Descargas para windows, linux y código fuente
- Página inicial (vídeos de demos, descripción, capturas, etc)

## DOCUMENTACIÓN DEL CÓDIGO: DOXYGEN

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

Para python existe la herramienta Doxypy.

# ÍNDICE

- 1 INTRODUCCIÓN
- 2 DESCRIPCIÓN
- 3 CALENDARIO
- 4 IMPLEMENTACIÓN
- 5 HERRAMIENTAS
- 6 CONCLUSIONES
- 7 BIBLIOGRAFÍA

# 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

## POSIBLES MEJORAS Y AMPLIACIONES:

- Modo dos jugadores (pantalla dividida, juego en red)
- Soporte para varias resoluciones
- Grabación de las mejores vueltas

## CONCLUSIONES

Zycars se incluirá en la próxima versión de  
Guadalinux



# ÍNDICE

- 1 INTRODUCCIÓN
- 2 DESCRIPCIÓN
- 3 CALENDARIO
- 4 IMPLEMENTACIÓN
- 5 HERRAMIENTAS
- 6 CONCLUSIONES
- 7 BIBLIOGRAFÍA



## 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/>