UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática

TFG del Grado en Ingeniería Informática

Seguimiento de líneas basado en openCV para AGVs

Presentado por Antonio de los Mozos Alonso
en Universidad de Burgos — 21 de noviembre de 2017
Tutor: Jesús Enrique Sierra García

UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática

D. Jesús Enrique Sierra García, profesor del departamento de Ingeniería Civil, área de lenguajes y sistemas informáticos.

Expone:

Que el alumno D. Antonio de los Mozos Alonso, con DNI 71705119C, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado Seguimiento de líneas basado en openCV para AGVs.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 21 de noviembre de 2017

Vº. Bº. del Tutor:

D. Jesús Enrique Sierra García

Resumen

En este primer apartado se hace una **breve** presentación del tema que se aborda en el proyecto.

Descriptores

Palabras separadas por comas que identifiquen el contenido del proyecto Ej: servidor web, buscador de vuelos, android ...

Abstract

A **brief** presentation of the topic addressed in the project.

Keywords

keywords separated by commas.

Índice general

Índice de figuras

Índice de tablas

Introducción

Descripción del contenido del trabajo y del estrucutra de la memoria y del resto de materiales entregados.

Objetivos del proyecto

Este apartado explica de forma precisa y concisa cuales son los objetivos que se persiguen con la realización del proyecto. Se puede distinguir entre los objetivos marcados por los requisitos del software a construir y los objetivos de carácter técnico que plantea a la hora de llevar a la práctica el proyecto.

Conceptos teóricos

En aquellos proyectos que necesiten para su comprensión y desarrollo de unos conceptos teóricos de una determinada materia o de un determinado dominio de conocimiento, debe existir un apartado que sintetice dichos conceptos.

Algunos conceptos teóricos de LaTeX[1].

3.1. Secciones

Las secciones se incluyen con el comando section.

Subsecciones

Además de secciones tenemos subsecciones.

Subsubsecciones

Y subsecciones.

3.2. Referencias

Las referencias se incluyen en el texto usando cite [2]. Para citar webs, artículos o libros [1].

[1] Créditos a los proyectos de Álvaro López Cantero: Configurador de Presupuestos y Roberto Izquierdo Amo: PLQuiz

3.3. Imágenes

Se pueden incluir imágenes con los comandos standard de LaTeX, pero esta plantilla dispone de comandos propios como por ejemplo el siguiente:

Figura 3.1: Autómata para una expresión vacía

3.4. Listas de items

Existen tres posibilidades:

- primer item.
- segundo item.

1. primer item.
2. segundo item.

Primer item más información sobre el primer item.

Segundo item más información sobre el segundo item.

-

3.5. Tablas

Igualmente se pueden usar los comandos específicos de LaTeXo bien usar alguno de los comandos de la plantilla.

Herramientas	App AngularJS	API REST	BD	Memoria
HTML5	X			
CSS3	X			
BOOTSTRAP	X			
JavaScript	X			
AngularJS	X			
Bower	X			
PHP		X		
Karma + Jasmine	X			
Slim framework		X		
Idiorm		X		
Composer		X		
JSON	X	X		
PhpStorm	X	X		
MySQL			X	
PhpMyAdmin			X	
Git + BitBucket	X	X	X	X
MikTEX				X
TEXMaker				X
Astah				X
Balsamiq Mockups	X			
VersionOne	X	X	X	X

Tabla 3.1: Herramientas y tecnologías utilizadas en cada parte del proyecto

Técnicas y herramientas

4.1. Técnicas de desarrollo

Metodología ágil

Este proyecto se ha resuelto mediante metodología ágil.

Cabe destacar que este proyecto no es un proyecto software común en el que tenemos unos requisitos iniciales mas o menos claros, y tenemos alguna idea del final. En este caso lo que buscamos es probar diferentes códigos, herramientas, librerías para satisfacer un primer requisito esencial, la detección de lineas en el suelo. Si conseguimos satisfacer ese requisito, podremos proponer nuevos requisitos en base al resultado obtenido.

Entonces, no hay un objetivo final claro, si bien, lo ideal seria avanzar lo máximo posible, también se puede dar el caso de probar y probar diferentes elementos, y no llegar a obtener un resultado esperado.

Por ser un proyecto software, las ventajas de usar una metodología ágil son claras:

- Las personas que participan en el desarrollo adquieren roles diferenciados.
- El desarrollo es incremental, por etapas. Al final de cada etapa se entrega una parte del software funcional al cliente.
- Partimos de unos requisitos básicos y podemos ir añadiendo más a lo largo del desarrollo, o modificar alguno de los existentes.
- La comunicación con el cliente se hace a lo largo de todo el desarrollo, favoreciendo que el producto final sea lo que esperaba.

Una de las metodologías ágiles más usadas es Scrum, cuyo principio clave es que los clientes pueden cambiar de idea sobre lo que quieren y necesitan. En Scrum hay 3 roles diferenciados:

- Product Owner: es el cliente. Propietario de la idea o proyecto a desarrollar.
- ScrumMaster: experto en Scrum, ayuda tanto al Product Owner como al Equipo Scrum a alcanzar los objetivos finales del proyecto.
- Equipo Scrum: equipo de personas que va a realizar el proyecto. Es conveniente que sea un equipo multidisciplinario, en el que cada persona sea experta o conocedora de un campo diferente a las demás. El equipo debe de ser de 5 a 9 personas, para favorecer la comunicación y fluidez del desarrollo.

Se basa en 3 principios básicos, en comparación con metodologías antiguas:

- Favorecer la comunicación del equipo en lugar de la excesiva documentación.
- Buscar la calidad del resultado en los conocimientos de las personas que han participado en el desarrollo, y no en los procesos empleados para hacerlo.
- Solapar las fases de desarrollo en lugar de hacerlas de forma secuencial o en cascada.

Para este proyecto en concreto, el alumno tomara el rol de Equipo Scrum y ScrumManager. El profesor actuará como cliente.

Como antes se ha explicado, este no es un proyecto software común, por lo que partiremos desde un requisito inicial, y cuando el cliente lo consideré valido y completado, propondrá mas requisitos.

En cada sprint se explorarán diferentes herramientas en busca de un objetivo, se harán pruebas con alguna de esas herramientas, estableciendo los parámetros iniciales de cada prueba, el resultado esperado y obtenido. El software evolucionará con cada prueba.

4.2. Herramientas de documentación

LaTeX

Latex es un sistema de composición de texto, que busca la creación de documentos con una alta calidad tipográfica.

Para la documentacion de este proyecto se ha usado TexMaker, que es gratuito bajo la licencia GPL.

Página oficial editor Latex: http://www.xm1math.net/texmaker/

También se ha usado MikTex como implementación de Latex para windows. En su página dicen que por ser un conjunto de paquetes, no tienen una licencia concreta como en otros casos, aun así siguen las directrices de la FSF, y establecen una serie de pautas de uso distribuición y modificación.

Página oficial MikTex: https://miktex.org/

4.3. Herramientas de gestión

Trello

Trello es una herramienta online de gestión de tareas intuitiva y simple. Podemos crear diferentes tableros y crear tareas dentro de estos. La vamos a usar principalmente para comunicarnos entre profesor y alumno. Crearemos 3 tableros, tareas por hacer, tareas en desarrollo y tareas completadas.

Página de trello: https://trello.com/

GitHub

Github es una plataforma online para albergar proyectos desarrollados mediante un sistema de control de versiones git. Es una herramienta mundialmente conocida en el mundo del desarrollo de software, y una de las más usadas para este tipo de propósitos.

Principalmente tiene 3 tipos de elementos:

- Milestones: Son puntos temporales de importancia dentro de la planificación del proyecto.
- Issues: Tareas a realizar. Pueden ser asignadas a um Milestone.

- Commits: Son el elemento que permite ir realizando cambios en el repositorio: añadir elementos, modificar, eliminar... Sirven para ir resolviendo las tareas que vayan surgiendo.

La usaremos para albergar el código del proyecto, y para aplicar metodologías ágiles y scrum. Los Sprints los representaremos con Milestones. Las tareas que vayan surgiendo en cada Sprint como Issues, y la resolución de las diferentes tareas se harán con commits.

4.4. Herramientas de desarrollo

Python

El lenguaje de programación usado para este proyecto es python. Ha sido elegido, primero porque las librerias de OpenCV están para este lenguaje, lo cual es un requisito imprescindible; y segundo por la facilidad de uso, la versatilidad y la gran cantidad de librerias externas con las que cuenta.

Concretamente se ha usado python en su versión 2.7.5. Nos hubiera gustado dar el salto a python 3, pero no hay librerías de OpenCV de forma oficial para esta versión, solo librerías traducidas de python 2 a python 3, por usuarios, por lo que no contamos con un soporte oficial y nos podemos encontrar con errores y sorpresas desagradables.

PyLint

PyLint es una herramienta para analizar y encontrar errores en códigos python. Se puede añadir a IDEs de forma sencilla, y se pueden personalizar los distintos avisos que proporciona. Además permite refactorizar código y cuenta con los estándares para el desarrollo de códigos en python.

OpenCV

OpenCV es una librería software de vision aritificial y machine learning. Nos proporcionará todos los elementos software necesarios para poder obtener y procesar imágenes. OpenCV se distribuye bajo licencia BSD.

Visual Studio Code

Visual Studio Coce es el IDE elegido para este proyecto. Es un IDE sencillo, compatible con una grán variedad de lenguajes de programación y

compatible con Linux, Windows y MacOS.

Se pueden añadir extensiones de una forma muy sencilla dentro del propio IDE, de hecho es el propio IDE el que nos sugiere instalar algunas extensiones dependiendo del lenguaje de programación que estemos usando.

En mi caso, en cuanto detecto que en mi repositorio había ficheros python, me recomendó instalar la extension de python, la cual incluye pylint, el debugger de python y alguna otra herramienta más.

Además de todo esto, es compatible con git de una forma muy sencilla, simplemente abrimos la carpeta de nuestro repositorio y ya entra en funcionamiento el control de versiones.

Su propietario es Microsoft y sus licencias son:

- Codigo fuente, licencia del MIT.
- Binarios, licencia FreeWare.

Aspectos relevantes del desarrollo del proyecto

Este apartado pretende recoger los aspectos más interesantes del desarrollo del proyecto, comentados por los autores del mismo. Debe incluir desde la exposición del ciclo de vida utilizado, hasta los detalles de mayor relevancia de las fases de análisis, diseño e implementación. Se busca que no sea una mera operación de copiar y pegar diagramas y extractos del código fuente, sino que realmente se justifiquen los caminos de solución que se han tomado, especialmente aquellos que no sean triviales. Puede ser el lugar más adecuado para documentar los aspectos más interesantes del diseño y de la implementación, con un mayor hincapié en aspectos tales como el tipo de arquitectura elegido, los índices de las tablas de la base de datos, normalización y desnormalización, distribución en ficheros3, reglas de negocio dentro de las bases de datos (EDVHV GH GDWRV DFWLYDV), aspectos de desarrollo relacionados con el WWW... Este apartado, debe convertirse en el resumen de la experiencia práctica del proyecto, y por sí mismo justifica que la memoria se convierta en un documento útil, fuente de referencia para los autores, los tutores y futuros alumnos.

Trabajos relacionados

Este apartado sería parecido a un estado del arte de una tesis o tesina. En un trabajo final grado no parece obligada su presencia, aunque se puede dejar a juicio del tutor el incluir un pequeño resumen comentado de los trabajos y proyectos ya realizados en el campo del proyecto en curso.

Conclusiones y Líneas de trabajo futuras

Todo proyecto debe incluir las conclusiones que se derivan de su desarrollo. Éstas pueden ser de diferente índole, dependiendo de la tipología del proyecto, pero normalmente van a estar presentes un conjunto de conclusiones relacionadas con los resultados del proyecto y un conjunto de conclusiones técnicas. Además, resulta muy útil realizar un informe crítico indicando cómo se puede mejorar el proyecto, o cómo se puede continuar trabajando en la línea del proyecto realizado.

Bibliografía

[1] John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* MIT Press, 1992.

[2] Wikipedia. Latex — wikipedia, la enciclopedia libre, 2015. [Internet; descargado 30-septiembre-2015].