EUROBOT 2014

PREHISTOBOT

Pilot study of

ARDUCOMPUGLOBALHYPERMEGABOT	MARVIN

From

team.

Index

Questionnaire 3
General description 4
The team 4
The robots 4
The budget 7
Strategy 8
Technical description 8
Finding and tracking targets: Computer Vision 8
Navigation 10
Beacons 10
Avoiding the enemy 10
Project Organization 12

Questionnaire

Name of the team: MammothBusters

1. Number of people involved on the project: 7
2. Is it the first participation to the contest for the team: Yes.
3. Did some of the team members take part in the contest before? Yes, one of us took part in Eurobot 2010 Feed the world.
4. Do you wish the visit of a Eurobot volunteer to help/assist during the year? Yes.
5. Provisional budget of the project: 350€
6. Do you accept the technical information contained in this pilot study to be published? Yes.
7. If yes, do you allow the publication of this information before the contest? No.

General description

At the beginning we started with the idea of beating the rest of the teams, at least in Spain, but quickly we would realized that it's not an easy task: our team has a strongly component of software developers, but we have an important lack in mechanicals and a bit in electronics.

We had to deal with the different personal situations like exams, works, a member that moved to another city, new members in january… It had become a quite difficult task, but when we saw our robots moving in the field, when we saw our application recognizing forms, we realized it was worth.

One of the most important thing we are obtaining taking part in this contest (besides enjoying and having fun with our friends) it's that we are learning loads of things: our members are closing a lot of engineering gaps they have, such as learning Java, discovering Arduino, learning the first steps in digital signal processing, and one very important thing: planning a project by ourselves.

The team

We are a team mostly formed by engineering students from the University of Alcalá de Henares, but we also have a junior researcher in computer vision, a network engineer and a java developer. We have the following roles and tasks:

- Ricardo Guerrero: coordination of the team, general strategy, digital image processing and hardware.
- Adriana Huerta: hardware and software of the secondary robot, building elements of the scenario.
- Dmytro Radchenko: central system of the main robot, building elements of the scenario.
- Pedro Gómez: hardware and software of the secondary robot, economic management.
- Iker Burguera: android development, digital image processing.
- Fernando: mechanics and hardware of the main robot.
- Javier: android development, digital image processing.

The robots

Due to the allowance of using two robots, and because of we already had one small robot built from last years robotic's week in Alcalá de Henares University, we have decided to split the scoring into two robots: the main, that is called Arducompuglobalhypermegabot and the secondary, called Marvin.

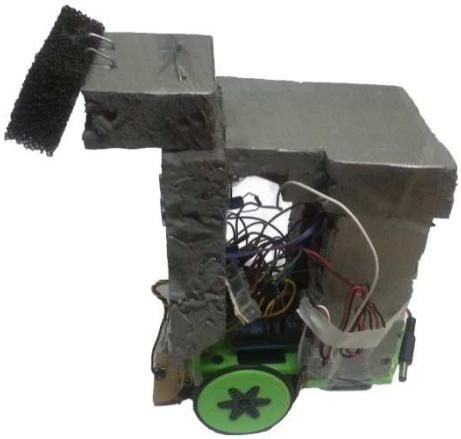


Fig 1: Marvin in its first stages.

Marvin, as can be seen in Figure 1, was a very simple robot that has optoreflexive sensors for being able to follow lines, an ultrasound sensor to avoid obstacles and uses two simple modified servos to move. It's not so fast, it's no strong and it's not "clever" but it's suitable more than enough to fulfill some easy tasks like following the black line and put the paintings on the fresco. After some test [FIGURE 2], we have seen that Marvin can be used for more things than just the paintings, so we have decide to coupling a cannon for throwing the spears to the mammoths and also tuning-up it for the funny action using the same cannon.

FOTO MARVIN y FOTO MARVIN TEST FRESCO

Arducompuglobalhypermegabot, in Figure 3, was thinked as the main robot, capable of achieving the hardest tasks like get all the fires without regarding the location, position and configuration and also to pick up the fruits avoiding the toxic ones. It has a powerful navigation system based in computer vision that allows it to dynamically finds all the targets and, when it's required, come back to the depot to release all game elements, thanks to the visual beacons we will put in the scenario.

Contar algo tipo: como se puede ver en la figura 3, tendría un soporte en la parte superior para colocar el móvil Android. 2 zonas de almacenamiento, la superior de fruta y la inferior de fuego. Delante el brazo robotico y 2 trampillas de descarga en la parte de atrás

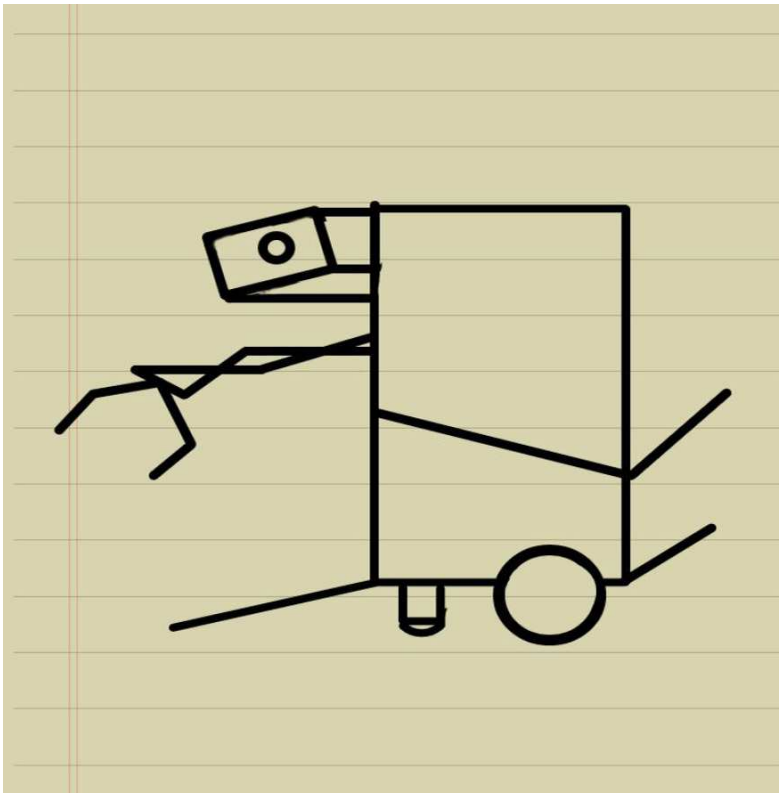


Fig. 3: The first draft of Arducompuglobalhypermegabot.

Fig 4: a robotic toy arm that we have hacked to incorporate in our design.

The budget

One of the main points decided when the team was founded was that we will not spend more than 50€ each one.

Por eso reutilizamos cosas que ya teniamos tipo movil android, marvin, brazo robotico que nos han prestado,e tc c

Strategy

Coger todos los fuegos, coger todas las frutas, depositar 3 fuegos al final del partido en nuestra esquina, los demas cerca para que no haya problemas (y solo al final para que no nos los quiten), depositar las frutas.

Marvin coloca los dos frescos, se dirige al mamut a una posición que le permita disparar pero no obstaculice al adversario. Dispara todas las flechas, reorienta el cañón, se prepara para la funny action. Al llegar a los 90s dispara la red

Technical description

Finding and tracking targets: Computer Vision

Although all the elements are located at fixed positions at the beginning, thus making it perfect for using odometry, we decided to rely in computer vision to find the objects. Why? In spite of the initial loss of speed, using computer vision allows our robots to behave dynamically avoiding problems like torches moved, fallen fires, elements already captured by the other teams, etc. Also makes it great to recognize which color is in the top face of the fires so we can move the less parts of our robot to achieve the correct storage or to recognize the poisonous fruits to not pick them.

Because of computer vision is computationally expensive and we are talking about an embedded device, our solutions were reduced to boards like Raspberry Pi or Beagle Bone, but we guess that maybe they gives a frames-per-second rate that couldn't be suitable for our purposes, so we find another solution: Android. One of our members have a powerful Galaxy Note II that has 4 cores of 1.6 GHz and using it we achieve two objectives: keeping a low budget and having a powerful enough platform that allows our robot have a very fast sensor to detect the environment.

Our first approach, relying in the fact that colours of the elements are known, is to perform a color segmentation followed by morphological operations (dilation an erosion)

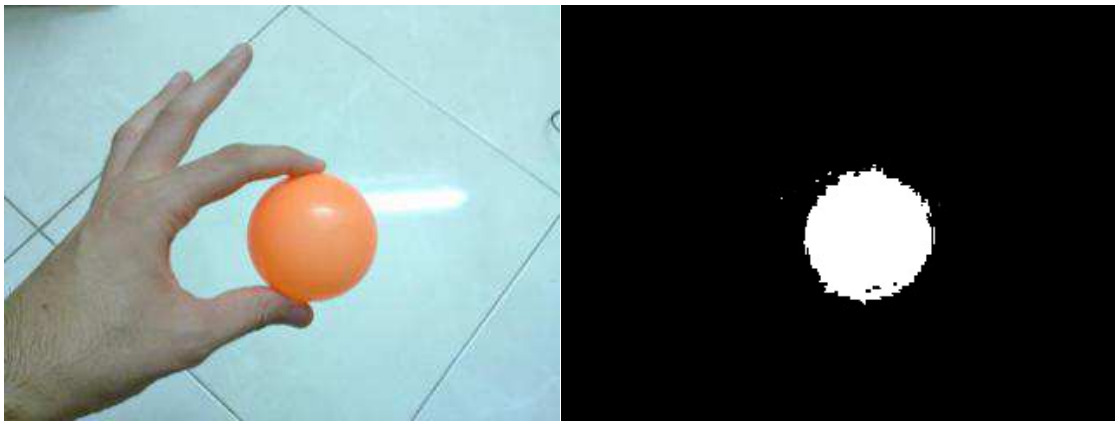


Fig. 5: shows a color segmentation using orange color as a target. The shape could be more accurate using an erosion that will filter the smalls white regions followed by a dilation that will fill the little gaps inside the circle.

that allows to find all the posible targets of the same class like is shown in Figure 5. Because of can be more than one class of the same color, i.e. fires and arrows, hearts of fires and torches, the following step is to execute a shape matching that filters the classes we are not looking for. For doing that, we retrieve the external contour of each blob and and use the Douglas-Peucker algorithm to simplify the contour approximating it with another polygon with less vertices and them perform some computations with them: edge counting and angles checking, convexity analysis, etc. After some test, we have realized that when we are too close to the target, e.g. a fire, we don't see the full shape, i.e. we don't see a triangle, so we have decided to incorporate the computation of the Seven Hu's Momments wich allows us to measures propierties like eccentricity or angle rotation wich increase the confidence for classifying each object.

One of the bigger issues when dealing with computer vision are related with lightning conditions. To overcome this problem, we have developed an application that will measure the colors of each class rather than hardcoding them into the system. But color segmentation is still quite difficult using the standard RGB color system, so we have decided to use a much more suitable color system called HSV that allows in a very easy way to segment each color including those variances produced by lightning changes.

Fig. 6: our Android application for target recognition and cost calculation to each target.

Our application (see Figure 6) is still in alpha version, but it's capable of finding all objects needed. In a future when testing in the final scenario with all the focus and illumination, if we find is not

Si hay problemas de fiabilidad metemos el cany y fuera, pero dará menos fps

Contar los problemas de la luz y que los hemos resuelto con nuestra aplicacion para coger los colores en el momento, pero que tendríamos un plan b que pondríamos en ejecución si más adelante, tras las pruebas vemos que es necesario, pero que en principio no lo haríamos porque son algo más lentas y queremos mantener el máximo fps posible

Contar algo de los hu moments para tirarnos el rollo

Hablar primero del tracking por segmentacion de color + operaciones morfológicas + shape matching para evitar "flechas", luego decir que si hubiera problemas en la detección derivado de problemas con la iluminación podríamos pasar a usar el plan b: estamos pensando en un canny edge detector o una transformada de hough para lineas seguido de un structural analysis para detectar formas. Calculando los hu moments + el número de aristas, podríamos distinguir un objeto de otro e incluso decir si está tumbado, de pie, boca arriba o boca abajo

Navigation

La aplicación móvil devuelve el coste de cada elemento (en función de la región de la pantalla en la que aparece y el área medida, con lo que el sistema central empleará un algoritmo dinámico de selección de objetivos basándose en minimizar una función de coste. No sabremos donde estamos en el escenario, ni lo necesitamos, al principio.

Hablar del plan b de los encoders

Beacons

Avoiding the enemy

In the case of *Marvin*, avoiding an enemy is simple: everything in a distance below a threshold is considered as an enemy, so it will wait until the obstacle moves. There is only one exception to this: since the beginning of the match, Marvin is taking into account the distance travelled. When Marvin detects an obstacle and the distance measured by the encoders is inside a

predefined range, we can consider that It has reached to the fresco, so we ignore the ultrasound sensor and we continue forward to place the paintings.

The case of *Arducompuglobalhypermegabot* is much more complex because of the movements and the task it's going to do. Our intention is to use a proximity sensor that can be detected only by the front side of our robot. We only need to know if the enemy is in our way or not. We know, from the last experiences of us and the people we talk with, that using infrared beacons is not such a good idea despite of many teams use it. Tal tal tal, pensamos en un sensor hall pero no tiene suficiente distancia, asi que hemos pensado en dos maneras: 1) modo radar con ultrasonidos, 2) ultrasonidos + camara

Beacons: color template + ultrasounds for Marvin

The scoring

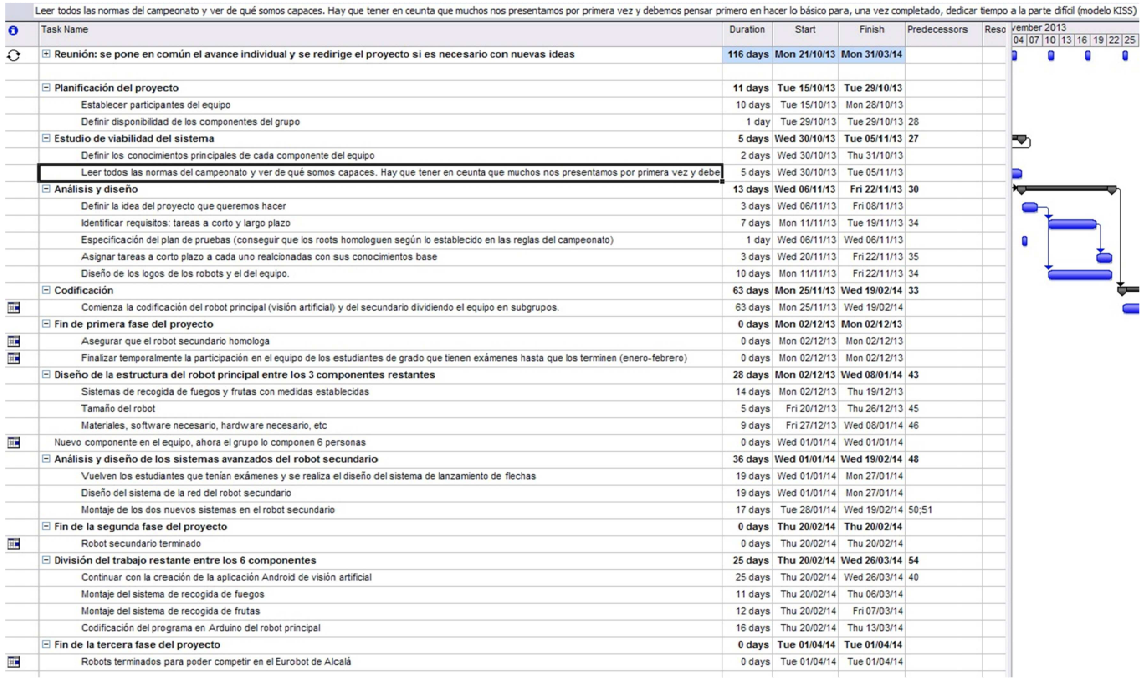
Project Organization

In order to organize this big project we have employed several tools:

For team communication and in order to centralize all the information we have, we are using a Google group, that is a mix of forum and mailing list. It combined with a record of every meeting the group has, allows us be coordinated in the difficult situation we have where the most of the meetings there is only the half of the group. Also, the records are useful for have a track of the tasks assigned to each one.

For documents like code, test images, tutorials, graphical designs, etc we have decided to use Dropbox, because is the perfect tool for sharing all files among the members and keeping them updated.

We have also to take into account the budget management and time planning. For those tasks we have considered the best solution the use of Excel and Microsoft Project respectively. In Figure 7 and 8 there is our time planning for this project.



Leer todos las normas del campeonato y ver de qué somos capaces. Hay que tener en ceunta que muchos nos presentamos por primera vez y debemos pensar primero en hacer lo básico para, una vez completado, dedicar tiempo a la parte difícil (modelo KISS)

Task Name	Duration	Start	Finish	Predecessors	Reso
Reunión: se pone en común el avance individual y se redirige el proyecto si es necesario con nuevas ideas	**116 days**	**Mon 21/10/13**	**Mon 31/03/14**		
Planificación del proyecto	**11 days**	**Tue 15/10/13**	**Tue 29/10/13**		
Establecer participantes del equipo	10 days	Tue 15/10/13	Mon 28/10/13		
Definir disponibilidad de los componentes del grupo	1 day	Tue 29/10/13	Tue 29/10/13	28	
Estudio de viabilidad del sistema	**5 days**	**Wed 30/10/13**	**Tue 05/11/13**	**27**	
Definir los conocimientos principales de cada componente del equipo	2 days	Wed 30/10/13	Thu 31/10/13		
Leer todos las normas del campeonato y ver de qué somos capaces. Hay que tener en ceunta que muchos nos presentamos por primera vez y debe	5 days	Wed 30/10/13	Tue 05/11/13		
Análisis y diseño	**13 days**	**Wed 06/11/13**	**Fri 22/11/13**	**30**	
Definir la idea del proyecto que queremos hacer	3 days	Wed 06/11/13	Fri 08/11/13		
Identificar requisitos: tareas a corto y largo plazo	7 days	Mon 11/11/13	Tue 19/11/13	34	
Especificación del plan de pruebas (conseguir que los roots homologuen según lo establecido en las reglas del campeonato)	1 day	Wed 06/11/13	Wed 06/11/13		
Asignar tareas a corto plazo a cada uno realcionadas con sus conocimientos base	3 days	Wed 20/11/13	Fri 22/11/13	35	
Diseño de los logos de los robots y el del equipo.	10 days	Mon 11/11/13	Fri 22/11/13	34	
Codificación	**63 days**	**Mon 25/11/13**	**Wed 19/02/14**	**33**	
Comienza la codificación del robot principal (visión artificial) y del secundario dividiendo el equipo en subgrupos.	63 days	Mon 25/11/13	Wed 19/02/14		
Fin de primera fase del proyecto	**0 days**	**Mon 02/12/13**	**Mon 02/12/13**		
Asegurar que el robot secundario homologa	0 days	Mon 02/12/13	Mon 02/12/13		
Finalizar temporalmente la participación en el equipo de los estudiantes de grado que tienen exámenes hasta que los terminen (enero-febrero)	0 days	Mon 02/12/13	Mon 02/12/13		
Diseño de la estructura del robot principal entre los 3 componentes restantes	**28 days**	**Mon 02/12/13**	**Wed 08/01/14**	**43**	
Sistemas de recogida de fuegos y frutas con medidas establecidas	14 days	Mon 02/12/13	Thu 19/12/13		
Tamaño del robot	5 days	Fri 20/12/13	Thu 26/12/13	45	
Materiales, software necesario, hardware necesario, etc	9 days	Fri 27/12/13	Wed 08/01/14	46	
Nuevo componente en el equipo, ahora el grupo lo componen 6 personas	0 days	Wed 01/01/14	Wed 01/01/14		
Análisis y diseño de los sistemas avanzados del robot secundario	**36 days**	**Wed 01/01/14**	**Wed 19/02/14**	**48**	
Vuelven los estudiantes que tenían exámenes y se realiza el diseño del sistema de lanzamiento de flechas	19 days	Wed 01/01/14	Mon 27/01/14		
Diseño del sistema de la red del robot secundario	19 days	Wed 01/01/14	Mon 27/01/14		
Montaje de los dos nuevos sistemas en el robot secundario	17 days	Tue 28/01/14	Wed 19/02/14	50;51	
Fin de la segunda fase del proyecto	**0 days**	**Thu 20/02/14**	**Thu 20/02/14**		
Robot secundario terminado	0 days	Thu 20/02/14	Thu 20/02/14		
División del trabajo restante entre los 6 componentes	**25 days**	**Thu 20/02/14**	**Wed 26/03/14**	**54**	
Continuar con la creación de la aplicación Android de visión artificial	25 days	Thu 20/02/14	Wed 26/03/14	40	
Montaje del sistema de recogida de fuegos	11 days	Thu 20/02/14	Thu 06/03/14		
Montaje del sistema de recogida de frutas	12 days	Thu 20/02/14	Fri 07/03/14		
Codificación del programa en Arduino del robot principal	18 days	Thu 20/02/14	Thu 13/03/14		
Fin de la tercera fase del proyecto	**0 days**	**Tue 01/04/14**	**Tue 01/04/14**		
Robots terminados para poder competir en el Eurobot de Alcalá	0 days	Tue 01/04/14	Tue 01/04/14		

Fig. 7: time planning using Microsoft Project

Fig. 8: time planning using Microsoft Project