

# Jornadas de Automática

## Gamificación en robótica móvil usando ROS2 y CoppeliaSim

Pérez-Montes, J.J., González-Monroy, J., Galindo, C. \*

Dpto. de Ingeniería de Sistemas y Automática, Universidad de Málaga, C/ Arquitecto Francisco Peñalosa, nº 6, 29071, Málaga, España.

**To cite this article:** Pérez-Montes J.J., González-Monroy J., Galindo C. Gamification for mobile robotics through ROS2 and Coppeliasim. Jornadas de Automática, 45. <https://doi.org/10.17979/ja-cea.2024.45.10754>

### Resumen

La robótica móvil se está imponiendo como una de las disciplinas con mayor relevancia para los futuros ingenieros. Entre sus aplicaciones, encontramos el diseño e implementación de vehículos inteligentes y autónomos que requieren sensores específicos como, por ejemplo, sensores láser, y técnicas y algoritmos que permitan el correcto control, navegación y localización de los vehículos. El aprendizaje de estas técnicas por parte de estudiantes de grado suele verse limitado debido a su nivel de complejidad y a la falta de estimulación al no vislumbrar la ejecución de sus implementaciones en robots reales y/o en situaciones realistas. Con este propósito, se presenta en este trabajo un gestor de juegos basado en ROS2 y CoppeliaSim que permita a los alumnos implementar el control de robots móviles en un entorno simulado de competición. La simulación realista de los algoritmos implementados por los alumnos sumado a la competitividad entre ellos hace aumentar su motivación y dedicación a la robótica móvil.

**Palabras clave:** Desarrollos curriculares para la enseñanza de control y otros ingenieros, Sistemas autónomos robóticos, Robotica Inteligente, Navegación de Robots, Simulación.

### Gamification for mobile robotics through ROS2 and Coppeliasim

#### Abstract

Mobile robotics is emerging as one of the most relevant disciplines for future engineers. Among its applications, we find the design and implementation of intelligent and autonomous vehicles that require specific sensors such as laser sensors, and techniques and algorithms that enable proper control, navigation, and localization of the vehicles. Learning these techniques by undergraduate students is often limited due to their complexity and the lack of stimulation from not seeing the execution of their implementations in real robots and/or realistic situations. To this end, this work presents a game manager based on ROS2 and CoppeliaSim that allows students to implement the control of mobile robots in a simulated competition environment. The realistic simulation of the algorithms implemented by the students, combined with the competitiveness among them, increases their motivation and dedication to mobile robotics.

**Keywords:** Teaching curricula developments for control and other engineers, Autonomous robotic systems, Intelligent robotics, Robot Navigation, Simulation.

### 1. Introducción

La creciente expansión de la robótica, y en particular de la robótica móvil, requiere la formación de profesionales capaces de ampliar los límites actuales y contribuir a la creación de nuevas aplicaciones robóticas que beneficien a la sociedad. La robótica es un campo donde concurren

múltiples tecnologías, como la electrónica, la informática o las telecomunicaciones, lo que la convierte en una disciplina de gran complejidad (Cañas et al., 2018).

Es en este contexto donde la *gamificación* pasa a ser un factor clave. Se entiende por gamificación el conjunto de técnicas de aprendizaje que trasladan la mecánica de los

juegos al ámbito académico con el fin de alcanzar resultados educativos más efectivos (Educativa, 2023). El éxito de la gamificación estriba en la motivación del alumnado cuando se les presenta como reto batir a sus compañeros en el desarrollo de una aplicación de los conceptos teóricos impartidos en clase.

Un ejemplo de aplicación de gamificación se puso a prueba en un estudio realizado con estudiantes residentes en el Centro Médico Beth Israel Deaconess, en Boston (Nakamoto, 2023). En él, se utilizó la herramienta SimNow (Rojas et al., 2023), un simulador robótico de última generación que utiliza la consola quirúrgica del robot da Vinci. Se siguió un modelo de estudio ABAB (Hayes, 1983), caracterizado por contemplar dos periodos de control con dos periodos de intervención intercalados, todos ellos de cuatro semanas de duración. Los periodos de control tienen como objetivo evaluar la iniciativa de los estudiantes a utilizar el simulador en su tiempo libre, mientras que, en los periodos de intervención, se incluyen técnicas de gamificación para observar si se producía algún cambio en el uso del simulador. La gamificación se aplicó mediante la asignación de diez ejercicios de simulación a los participantes, quienes debían alcanzar una puntuación superior a 90 para superarlos. Como resultado, los estudiantes identificaron que su motivación para utilizar el simulador se debió al aspecto competitivo de los periodos de intervención, pero sobre todo al componente de juego (gamificación). También destacaron que la visibilidad de las puntuaciones incentivó a utilizar más el simulador, especialmente cuando observaban a compañeros obteniendo puntuaciones altas en la clasificación. Este enfoque gamificado y competitivo se reveló como un impulsor efectivo para el uso del simulador en el entorno educativo.

Evidencias así respaldan cómo esta forma de trabajar las asignaturas da lugar a aprendizajes más efectivos y funcionales, con una serie de beneficios notables entre los que se incluyen la mejora de la transferencia de conocimientos a situaciones del mundo real, la contribución a la retención de los aprendizajes en la memoria a largo plazo y el estímulo e incremento de la motivación (Serrano et al., 2019).

En términos generales, la programación de robots móviles ha sido tradicionalmente considerada una disciplina de enseñanza ardua, dada su complejidad y la exigencia de que los estudiantes asimilen conceptos y algoritmos altamente abstractos. En este sentido, se vuelve necesario adoptar algún acercamiento innovador que facilite la comprensión de estos conceptos, transformándolos en actividades prácticas que fomenten un conocimiento más comprensivo (Zhan et al., 2022).

En este trabajo se presenta un gestor de juego que utilizando ROS2 y CoppeliaSim que sirve como aliciente a los estudiantes de grado para mejorar su aprendizaje de las técnicas y algoritmos necesarios para la programación de robots móviles, así como mejorar sus habilidades en este

campo. Este trabajo está basado en (Pérez-Montes et al., 2024) y será implantado en la asignatura de Control y Programación de Robots del Grado de Ingeniería Electrónica, Robótica y Mecatrónica de la Universidad de Málaga el próximo curso 2024/2025.

## 2. ROS2 y Coppelasim

El gestor de juegos desarrollado está basado en el framework de programación robótica ROS2<sup>1</sup> y el simulador Coppelasim<sup>2</sup>, ambos ampliamente utilizados en aplicaciones robóticas tanto a nivel educativo como en aplicaciones reales.

### 2.1. ROS2

ROS (Robot Operating System) se ha convertido de facto en el estándar de programación en la comunidad robótica. Se trata de un framework de código abierto que surge a principios de 2007 en la Universidad de Stanford para abordar mecanismos estandarizados de comunicación entre componentes software en el ámbito de la robótica (Koubaa et al., 2016). ROS considera, entre otros, dos mecanismos de comunicación entre las unidades de ejecución que son los denominadas nodos. En concreto los *topics* que establecen una comunicación 1:n de forma que todos los nodos suscritos a un topic particular recibirán la información enviada. Otro mecanismo es el denominado *servicio*, que implementa una comunicación 1:1 entre un nodo cliente y un nodo servidor.

Con el tiempo, ROS ha ido evolucionando e incluyendo potentes herramientas y librerías que facilitan la labor de comunicación y programación de arquitecturas robóticas. La última versión, ROS2, mejora la anterior empleando un protocolo de comunicación estandarizado basado en el Servicio de Distribución de Datos (DDS) para sistemas en tiempo real, incluyendo mejoras significativas en cuanto a eficiencia, seguridad, escalabilidad y calidad del servicio.

### 2.2. CoppeliaSim

El número de herramientas disponibles para la simulación de robots ha aumentado considerablemente en los últimos años. Podemos encontrar por ejemplo herramientas como Webots (Olivier 2004), MORSE (Echeverria et al., 2011) o el simulador Gazebo (Koenig and Howard, 2004) integrado en ROS.

En este trabajo se ha optado por CoppeliaSim ya que destaca por su versatilidad y facilidad de uso, brindando a los usuarios la capacidad de simular una amplia gama de aplicaciones en robótica.

CoppeliaSim, antiguamente conocido como V-REP, es un potente simulador robótico con amplias posibilidades en el mundo educativo. Se basa en una arquitectura de control distribuida, donde cada objeto o modelo puede controlarse

<sup>1</sup> ROS homepage: <https://www.ros.org/>

<sup>2</sup> Coppelia robotics homepage: <https://www.coppeliarobotics.com/>

individualmente mediante diversos métodos, como scripts embebidos implementados en Python o LUA.

Esta versátil plataforma de simulación permite la creación de entornos virtuales realistas incluyendo motores de física y mecanismos de comunicación con otros componentes software como por ejemplo con nodos de ROS2.

### 3. Gamificación

En este trabajo se implementa la gamificación en forma de un torneo de supervivencia disputado en una escena simulada en CoppeliaSim. Cada estudiante tendrá que implementar el control de un robot móvil, mediante un nodo de ROS2, que será desplegado en dicha escena. Los diferentes componentes software implementados se muestran en la Figura 1. Como se observa, el software implementado por los alumnos no se comunica directamente con el simulador, sino que lo hacen a través de un nodo intermedio denominado gestor de juegos que funciona como puente entre cada jugador y el simulador. Este gestor que actúa como *man-in-the-middle* (MITM) es el encargado de traducir los mensajes de los alumnos hacia el simulador y a filtrar la información de la escena.

El desafío que se presenta no se limita a la navegación libre sin restricciones, sino que el entorno cuenta con una serie de obstáculos y objetivos que requirieren que el jugador implemente algoritmos de evitación de colisiones y lógica de decisión.

En concreto, la escena distribuye de forma aleatoria un número determinado de puntos de recarga de batería a los que los jugadores deben dirigirse para evitar la descarga total de la batería del robot, lo que supondría la finalización del juego.

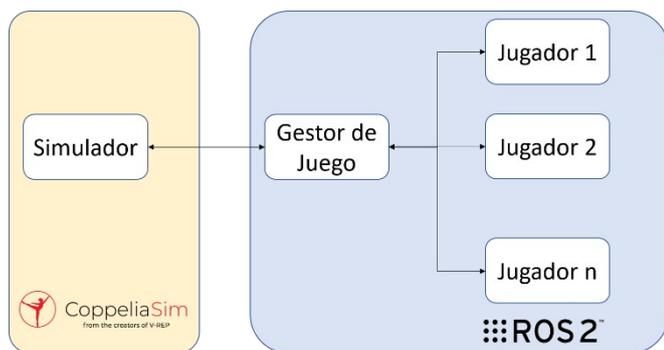


Figura 1: Esquema de comunicaciones entre los jugadores y el simulador a través del gestor de juego.

Adicionalmente se incluyen bloques de habilidad que dotan al robot que se aproxime de habilidades especiales por un período de tiempo concreto, como por ejemplo un escudo de protección, un arma para atacar a oponentes, o la habilidad de navegación automática a un destino. Los estudiantes deberán implementar la lógica necesaria para desplazarse hasta los lugares donde se pueden adquirir las habilidades disponibles y utilizarlas para batir a sus rivales y así mantenerse el mayor tiempo posible en el juego.

El objetivo final del juego es sobrevivir al resto de adversarios, estableciéndose una tabla de puntuación basada en el tiempo de supervivencia y la actividad de cada jugador, como por ejemplo el número de bloques adquiridos, el número de adversarios derrotados, etc.

#### 3.1. Gestor de juego

La función del nodo gestor de juego es la de filtrar la información y comandos desde y hacia el simulador, evitando que un jugador pueda interferir en el movimiento de un robot rival.

Este nodo implementa un servicio de registro que debe ser llamado por cada participante. Tras el correcto registro, el gestor crea los topics necesarios para comunicarse tanto con el jugador como con el simulador. Estos topics incluyen un código de usuario único (UID) que deberá usar cada participante para enviar sus comandos de control. En concreto para cada jugador se consideran los topics de la tabla 1.

Como se observa el gestor de juego se comporta como un puente entre los jugadores y el simulador controlando el acceso de cada jugador, evitando el uso de habilidades no disponibles, o el movimiento de un robot con su batería agotada.

Tabla 1: Comunicación entre jugadores y gestor de juegos

<code>/UID_jugador1/cmd_vel</code>	Publicado por los jugadores con los comandos de velocidad lineal y angular para el control de su robot
<code>/UID_jugador1/goal_xy</code>	Publicado por los jugadores que dispongan de la habilidad de navegación automática
<code>/UID_jugador1/laser_scan</code>	Publicado por el gestor de juego con la información proporcionada por el sensor laser simulado
<code>/UID_jugador1/scene_info</code>	Publicado por el gestor de juego con información del juego, como las habilidades disponibles, el nivel de batería, etc.

#### 3.2. Simulador

La escena implementada en el simulador es la encargada de mantener la visualización del juego, así como de aplicar los comandos de velocidad (lineal y angular) a cada uno de los robots participantes, detectar colisiones, simular la información sensorial, etc.

El escenario donde se disputa el juego es una escena de CoppeliaSim en la que se disponen de forma aleatoria los diferentes elementos dentro de un área cuadrada. La creación del escenario es configurable, estableciéndose el número de objetos de cada tipo. Se ha prestado especial atención a la distribución uniforme de los diferentes elementos para no favorecer a ningún participante. La Figura 2 muestra los objetos considerados y la Figura 3 un ejemplo de escena.



Figura 2: Representación de los diferentes elementos de la simulación. El robot móvil está basado en un Pioneer 3-DX dotado de un sensor láser. En la figura se muestra la situación en la que dispone de la habilidad de escudo y arma de ataque.

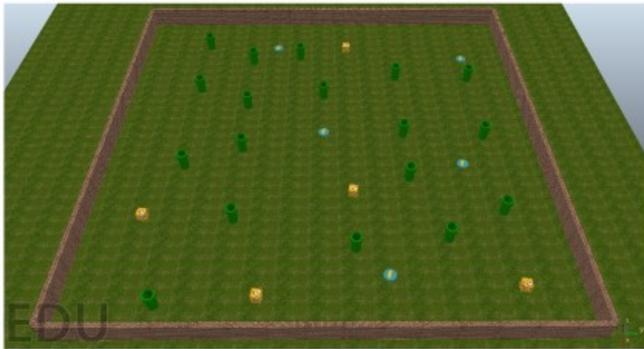


Figura 3: Ejemplo de escena. Se disponen los diferentes elementos de forma aleatoria y se queda a la espera de la llegada de participantes.

La escena implementa mecanismos de comunicación con el framework de ROS2. En concreto la simulación de Coppelia se comunica con el nodo gestor de juego mediante el topic `/create_robot` que permite el despliegue de un nuevo jugador en la escena. Asimismo, para cada robot desplegado, el simulador se comunica con el gestor mediante los mismos topics de la tabla 1 añadiendo un prefijo preestablecido para cada jugador. La Figura 4 muestra el grafo de comunicaciones para el caso de dos jugadores.

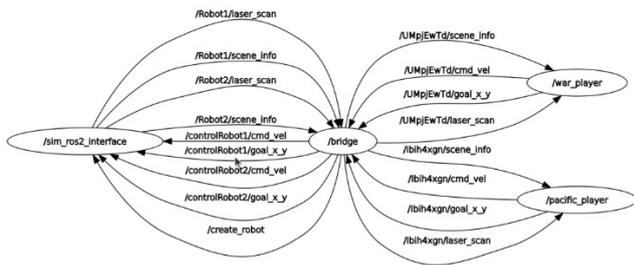


Figura 4: Grafo de comunicaciones para el caso de dos jugadores.

### 3.3. Jugadores

Una vez establecida la comunicación entre el simulador y el nodo gestor de juego se pueden ejecutar los nodos de los jugadores quienes disponen de la información listada en la tabla 2, la cual les es transmitida por el topic `/scene_info`. Como se observa, los jugadores conocen su nivel de batería,

el cual disminuye proporcionalmente con respecto a la velocidad y distancia recorrida, su posición y orientación<sup>3</sup>, las habilidades que tienen disponible y la lista de objetos que están dentro de su campo de visión. Esta información es enviada a través del gestor de juego a los jugadores, pudiendo éste ocultar aleatoriamente algún valor para incrementar la dificultad del juego.

Dada esta información, además de la información provista por el sensor láser a través del topic `/laser_scan`, los jugadores deben implementar la lógica para buscar los bloques de habilidad y una vez en posesión de alguna de ellas utilizarlas para su provecho. Por ejemplo, la Figura 5 muestra un caso en el que el jugador aprovecha la habilidad de navegación automática para llegar a un punto de interés.

Tabla 2: Información accesible por los jugadores

<b>Nivel de batería</b>	Valor entre 0 y 100.
<b>Pose del robot</b>	Posición y orientación del robot $x, y, \theta$
<b>Habilidades activas</b>	Entre las tres consideradas en el juego: 1) Navegación automática, 2) Arma de ataque, 3) Escudo de protección.
<b>Objetos detectados</b>	Se informa a los jugadores la posición de los obstáculos, bloques de habilidad y oponentes detectados en el campo de visión.



Figura 5: Ejemplo de navegación autónoma. Una vez adquirida esta habilidad, el jugador puede seleccionar un destino a través del topic `/goal_x_y`

## 4. Pruebas realizadas

El presente trabajo de gamificación aún no ha podido ser implantado por tanto no se cuentan con resultados con alumnos reales.

Como prueba de concepto, se han implementado dos jugadores que revelan la utilidad y eficacia de todos y cada uno de los aspectos mencionados anteriormente. En concreto, a modo de ilustración, se ha implementado un jugador con intenciones bélicas y otro con intenciones pacíficas. Ambos tienen en común la estrategia de fijar un objetivo de entre los objetos detectados, ya sea un bloque de habilidad o un punto de recarga, y comandar el robot en términos de velocidad lineal y angular hasta alcanzarlo. Para ello se utiliza la información del sensor láser para evitar obstáculos.

<sup>3</sup> En esta versión, por simplicidad, se considera que los jugadores no tienen que resolver el problema de localización.

El jugador pacífico muestra su interés por únicamente por las estaciones de recarga para mantener un buen nivel de batería y por la habilidad de escudo para mantenerse protegido. No seleccionara bloques de habilidad de arma o navegación automática.

Por el contrario, el jugador bélico primará la obtención del arma y una vez está en posesión de la misma, fijará como objetivo cualquier oponente en su radio de acción. Igualmente, si dispone de la habilidad de navegación automática, la activará para alcanzar con rapidez su objetivo.

La Figura 6 muestra un ejemplo en el que el jugador bélico, en posesión del arma alcanza y destruye al robot pacífico.



Figura 6: Ataque de un jugador en posesión de la habilidad de arma. En este caso el contrincante no dispone de escudo protector, por tanto, es derrotado.

## 5. Conclusiones

Este trabajo ha puesto de manifiesto la importancia de la gamificación para el estudio de asignaturas del campo de la robótica móvil. Se presenta como resultado preliminar que aún no ha podido ser implantado en una asignatura con alumnos reales, aunque se prevé un gran potencial y tasa de éxito en base a experiencias previas. Se plantea su implantación para el curso 2024/2025 de la asignatura de Control y Programación de Robots de Ingeniería Electrónica, Robótica y Mecatrónica de la Universidad de Málaga.

Si bien no ha sido implantada aún, los autores vislumbran aspectos a añadir al gestor de juegos propuesto que sin duda incrementará la motivación de los alumnos. Algunas de las ampliaciones propuestas son:

- Implementación de una funcionalidad en línea. Esta idea ofrece a los estudiantes la posibilidad de participar en el torneo de forma remota. Cada jugador podría cargar su nodo desde una página web habilitada para esta función, lo que no solo aumenta la accesibilidad al juego, sino que también ampliaría la comunidad de jugadores al permitir que

participantes de diversas ubicaciones se enfrenten entre sí.

- Sistema de clasificación. La introducción de un sistema de clasificación brinda a los estudiantes la oportunidad de competir y comparar sus implementaciones con el resto de participantes, promoviendo la competencia y la mejora constante.
- Ampliación del número de habilidades. La incorporación de más habilidades no solo aumenta la emoción del torneo, sino que también requiere que los jugadores tomen decisiones estratégicas. Algunas que pueden ser consideradas son la de invisibilidad, evitando la detección de un jugador, restauración de la salud del robot, inmovilización de contrincantes, inclusión de jugadores bots en el juego, etc.

## Referencias

- Cañas Plaza, J.M., Martín Florido A., Perdices García E., Rivas Montero F., and Calvo Palomino, R., 2018. "Entorno docente universitario para la programación de los robots," *Revista iberoamericana de automática e informática industrial*, vol. 15, no. 4, pp. 404–415.
- Echeverría G., Lassabe N., Degroote A., and Lemaignan S., 2011, "Modular open robots simulation engine: MORSE," 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 2011, pp. 46-51, doi: 10.1109/ICRA.2011.5980252
- Educativa, "Gamificación: el aprendizaje divertido," línea]. Accedido el: 23/10/2023. [Online]. Available: <https://www.educativa.com/blog-articulos/gamificacion-el-aprendizaje-divertido/>
- Hayes, Steven., 1983. Single-case research designs: Methods for clinical and applied settings by A.E. Kazdin. *Journal of Behavior Therapy and Experimental Psychiatry – Journal of Behavior Therapy and Experimental Psychiatry* 14. 10.1016/0005-7916(83)90018-6.
- Koenig N., and Howard A., 2004, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Sendai, Japan, 2004, pp. 2149-2154 vol.3, doi: 10.1109/IROS.2004.1389727.
- Koubaa A. (2016), *Robot Operating System (ROS): The Complete Reference*, 1st ed., ser. *Studies in Computational Intelligence*, volume 625, 707, 778. Cham: Springer Nature, 2016, vol. 625.
- Nakamoto K., Jones D. B., and Adra S. W., 2023. "Gamification of robotic simulation to train general surgery residents," *Surgical endoscopy*, vol. 37, no. 4, pp. 3136–3144.
- Olivier M., 2004. *WebotsTM: Professional Mobile Robot Simulation*. *International Journal of Advanced Robotic Systems*. 1. 10.5772/5618.
- Pérez-Montes J., Galindo C., and González-Monroy, J. (2024). Trabajo fin de grado "Gamificación en robótica móvil usando ROS2 y CoppeliaSim". Universidad de Málaga
- Rojas, A., Paterakos P., Ramirez-Barriga, M., Thrastardottir T., Talamonti, M., and Hogg M., 2023. A Simulation-Based Robotic Curriculum for General Surgery Residents. *Videoscopy*, vol. 33 n.1. DOI: 10.1089/vor.2023.0022
- Serrano-Díaz N., del Río C. R., and Noguera E. C., 2019. "Educational innovation with the use of gamification and robotics," *INFAD (Barcelona)*, vol. 3, no. 1, pp. 545–552.
- Zhan Z., He L., Tong Y., Liang X., Guo S., and Lan X., (2022) "The effectiveness of gamification in programming education: Evidence from a meta-analysis," *Computers and Education: Artificial Intelligence*, vol. 3, p. 100096, DOI:10.1016/j.caeai.2022.100096