

Manual para construcción de ROVER

División de Software

12 de julio de 2022

1. Introducción

Según la Administración Nacional de Aeronáutica y el Espacio (NASA), un rover es simplemente un vehículo con la capacidad de explorar la superficie de un planeta o luna. Sin embargo, por sencilla que suene su definición, su modelado y construcción se caracterizan por ser complejos por las adversidades que puede enfrentar lejos de la Tierra y de un encargado de verificar su correcto funcionamiento. Un rover de exploración espacial requiere de distintas ramas de la ingeniería: la mecánica y la electrónica se encargan de darle soporte, resistencia y movimiento, junto con un aprovechamiento óptimo de la energía eléctrica para la realización de sus tareas. De igual manera, es necesario que el robot cuente con un cerebro capaz de tomar una decisión cuando sea requerido, además de una implementación de software óptima que garantice que esta toma de decisiones se efectúe lo más rápido posible y sea comunicada a los módulos pertinentes de nuestro sistema. Por último, pero no menos importante, el desarrollo de cada parte de dicho sistema requiere de una correcta administración del tiempo y de los recursos humanos para realizarse. En conjunto, estas cinco divisiones deberían ser capaces de darle forma a un rover móvil. Por ello, es momento de que ustedes, reclutas de la división de Mecánica, se encarguen de aquello que a la parte mecánica del robot competa. En el presente manual se les proporcionan ciertas consideraciones e indicaciones para su diseño y construcción. En caso de que se les presente alguna dificultad, tendrán la oportunidad de acudir a un asesor perteneciente al equipo.

2. ¿Qué construir?

Dada la versatilidad de nuestra rama consideramos importante que existan una variedad de tipos de desarrollo a elegir, te recomendamos seleccionar el que más se ajuste a tus recursos disponibles, te sea más sencillo y en lo posible te tome el menor tiempo posible, tanto de desarrollo como de armado, si aplica. A continuación te enumeraremos las opciones disponibles aunados a su respectiva rúbrica de evaluación.

2.1. ROVER 2D

En este reto deberás programar un rover, bajo tu representación gráfica deseada, utilizando tu lenguaje de preferencia y este deberá recorrer un terreno (prediseñado o aleatorio) mismo que contenga obstáculos, el rover podrá o no funcionar mediante interacción de usuario, pero debería ser capaz de detectar los obstáculos y evitarlos.

Rúbrica Rover 2D				
Concepto	Malo	Suficiente	Bueno	Destacable
Movilidad	El rover es estático o tiene una ruta básica recta de ida y vuelta	El rover puede rodear el plano y esquivar obstáculos(es interactivo y/o autónomo)	El robot puede recorrer cualquier lugar del plano, ya sea con un orden específico o aleatorio	El rover implementa algoritmos bfs, dfs, dijkstra o alguna variación, misma que le permite conocer el plano, puede o no incorporar algún mecanismo de aprendizaje
Autonomía	No funciona, deja al rover estático o simplemente no esquivar obstáculos	El rover funciona, pero presenta paros o fallos inesperados, choca con obstáculos o en ocasiones no los esquivar	El rover esquivar los obstáculos utiliza alguna regla general para evitarlos no presenta fallos graves	El rover implementa algún algoritmo de reconocimiento, backtracking, búsqueda o trazado de rutas y funciona completamente
Interactividad(si aplica)	Los controles fallan, no puede operarse en totalidad el ROVER	Los controles operan de forma concisa, no son rápidos o ergonómicos ni intuitivos pero cumplen el objetivo	Son de fácil y rápido uso, cumplen el objetivo y son eficientes	Proporcionana retroactividad, son ergonómicos, intuitivos y permiten emocionarse con la operación del ROVER

Rúbrica Rover 2D				
Concepto	Malo	Suficiente	Bueno	Destacable
Documentación	Nula documentación	Escasa, describe mínimamente las funciones y no utiliza la convención propia de su lenguajes	Es correcta y comprensible, describe en su totalidad el programa, funcionamiento, módulos, etc	Puede generarse a partir de algún sistema como Javadoc o pydoc y ser visualizada a través de un navegador o lector especializado
Interfaz	Cuenta con los elementos mínimos pero no es comprensible su utilización a simple vista	Incorpora buenos elementos gráficos, pueden resultar difíciles de utilizar algunas funciones, no es del todo comprensible	La interfaz es comprensible a simple vista, permite utilizarla y es visualmente atractiva	Proporciona retroactividad, cumple con principios de diseño

2.2. Modelado de ROVER 3D

En esta opción deberás desarrollar un ROVER en 3D que igual que la anterior muestra pueda ser interactivo, autónomo o ambos, para ello te recomendamos utilizar el motor de Unity para desarrollo 3d, nuevamente el desarrollo, elección de lenguaje, herramientas y plataforma quedan a tu elección. El rover deberá ser capaz de moverse sobre una superficie, misma que deberá contener obstáculos y estos deberán ser evitados, la rúbrica es la correspondiente a la Rúbrica Rover 2D.

2.3. Robot Móvil

Para este caso te recomendamos contar con un Arduino UNO o alguna de sus variantes, si deseas utilizar algún otro microcontrolador te solicitamos nos envíes un correo, así como con algunos sensores(selección libre), el objetivo es que tu robot pueda utilizar los sensores en función con algún propósito específico, deberás armar una estructura del material que desees y deberá contar con 4 o 6 ruedas, así mismo el robot puede ser autónomo o controlado a través de alguna interfaz, sin embargo es recomendable contar con alguna interfaz a través de la cual se puedan monitorear los datos sustraídos

y que está sea inalámbrica. (Recomendamos un sensor y un actuador, tal que el actuador trabaje en función del sensor)

Rúbrica Robot Rover				
Concepto	Malo	Suficiente	Bueno	Destacable
Objetivo	Los sensores no monitorean y los actuadores no funcionan	Los sensores monitorean de forma pobre, existe mucho ruido y los actuadores tienen paros inesperados	Los sensores monitorean de forma adecuada, con ligeros errores y los actuadores funcionan sin paros fatales	Sensores y actuadores corren de forma continúa sin ninguna especie de paro
Código y Documentación	El código está desordenado, no modulado y muy acoplado, la documentación es nula y el código es incomprensible a simple vista	La documentación es escasa pero el código está ordenado y modulado, es funcional	El código está modulado, bien documentado de acuerdo al lenguaje que fue utilizado	El código es legible, modulado, utilizo POO o semejante y correspondiente al lenguaje seleccionado, es clara la definición de objetivos además cuenta con pruebas unitarias para diversos segmentos del código útil.

Rúbrica Robot Rover				
Concepto	Malo	Suficiente	Bueno	Destacable
Interfaz	Cuenta con los elementos mínimos pero no es comprensible su utilización a simple vista	Incorpora buenos elementos gráficos, pueden resultar difíciles de utilizar algunas funciones, no es del todo comprensible	La interfaz es comprensible a simple vista, permite utilizarla y es visualmente atractiva	Proporciona retroactividad, cumple con principios de diseño

3. Sobre la entrega

Para entregarlo se realizará una breve exposición donde se muestre el código, el objetivo planteado, problemas durante el desarrollo, así como el producto funcional. Todos deberán participar y cualquiera deberá ser capaz de responder preguntas sobre el desarrollo y producción del mismo.

4. Consejos para el desarrollo

Ya que has llegado hasta aquí, lo que esperamos ver es tu resiliencia, adaptación, capacidad colaborativa y por supuesto tus habilidades y cualidades que te distinguen de los demás así como tu capacidad de unirlos a la de los demás.

Para tener un proceso lo más productivo posible te recomendamos:

- Utilicen sus recursos locales y creatividad, material reciclado material de algún proyecto anterior o incluso componentes de un artefacto descompuesto.
- Menos es más. Planteen objetivos alcanzables para el tiempo proporcionado, la parte importante es que cumplan con ellos en lugar de tener diversos fallidos
- Realiza una investigación previa y rápida, que consolide y argumente tus decisiones, no es necesario presentar algún tipo de bibliografía sino que la experiencia lo demuestre. Siempre mantenla accesible, incluso cuando ya no te parezca útil.
- Utilicen versionadores como Git, github o gitlab que permitan el desarrollo colaborativo, y también les permita crear respaldos en casos de errores, no recomendamos utilizar OneDrive, Google Drive entre otras, debido a su mal manejo de código, así podrás evitarte rehacer código que fue descompuesto por el codificador de la nube
- Tu asesor puede guiarte y mostrarte opciones, pero de ninguna forma tomará decisiones por ti o tu equipo ni podrá ayudarte a desarrollar, solicítale ayuda cuando lo requieras.
- ¡Éxito futuro cosmonauta! UNAM Space confía en ti.