

Tumbar castillos con sensores en Vexcode VR

Grado sugerido: Octavo

Deiber Andrés Aldana Pulido

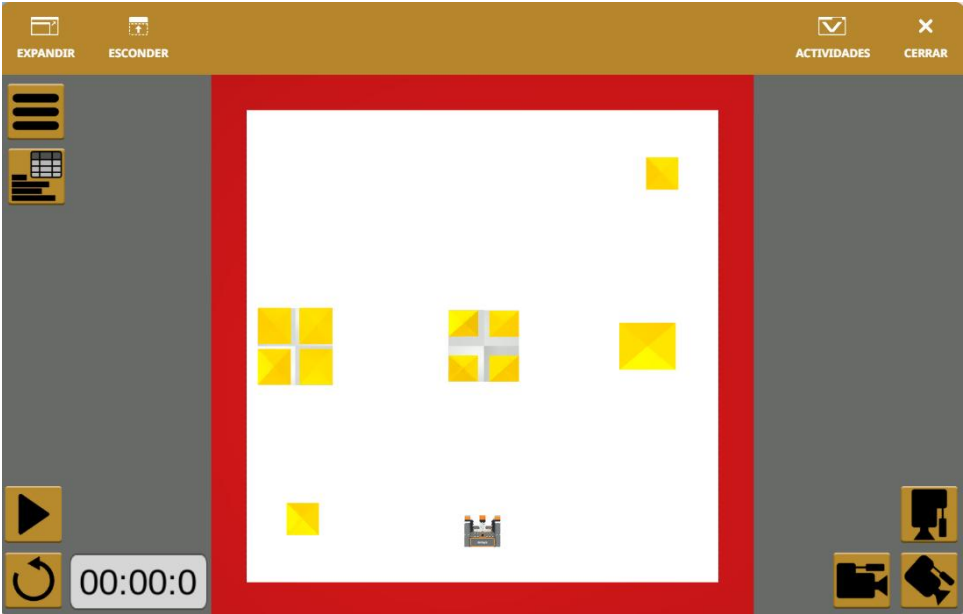
Publicado en el Banco Virtual de Recursos de Colombia Programa en el año 2025.

Este material se comparte bajo la licencia Creative Commons Atribución-NoComercial-SinDerivadas 4.0 Internacional (CC BY-NC-ND 4.0). Puede copiar y redistribuir el material en cualquier medio o formato, siempre que dé el crédito adecuado al autor, no lo use con fines comerciales, y no remezcle, transforme o cree a partir del material.

Para más información, consulte la licencia completa en [Deed - Attribution-NonCommercial-NoDerivatives 4.0 International - Creative Commons](#)

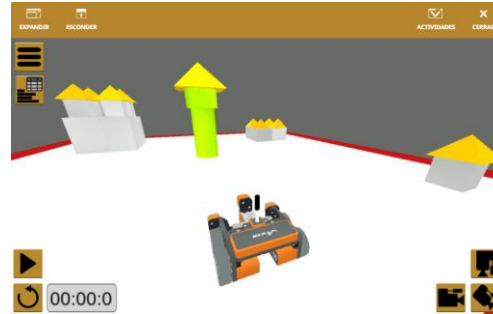
Para contactar al autor/a de este recurso, escriba a: ingaldana569@gmail.com

TUMBAR CASTILLOS CON SENSORES EN VEXcode VR

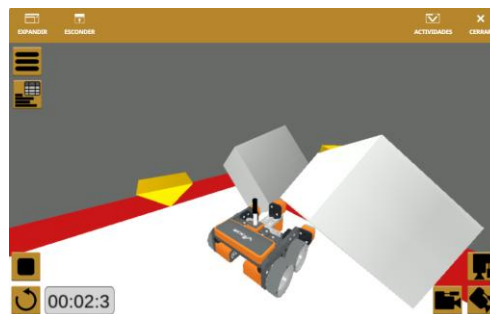
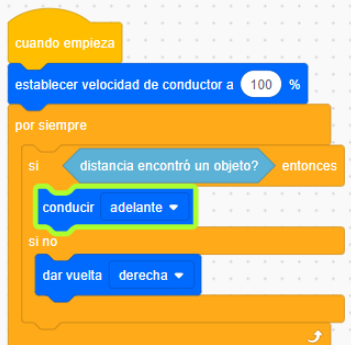
Aprendizaje(s) esperado(s)	<ul style="list-style-type: none"> Comprender cómo combinar sensores de distancia y parachoques en un entorno de navegación. Programar un algoritmo que permita al robot detectar castillos, posicionarse y ejecutar acciones automáticas. Aplicar lógica condicional para tomar decisiones ante obstáculos, colisiones y objetivos.
Materiales requeridos	<ul style="list-style-type: none"> Computador con acceso a internet Plataforma VEXcode VR: https://vr.vex.com Cuaderno de apuntes Lápiz y colores Plantilla impresa de la zona de juegos (para trabajo sin conexión)
Conocimientos previos requeridos	<ul style="list-style-type: none"> Uso de bloques condicionales (si... entonces...) Programación básica de sensores (parachoques y distancia) Control de movimientos (giro, avanzar, retroceder)
Actividad(es) a desarrollar <i>Indique las acciones que realizarán el/la docente y sus estudiantes y las indicaciones si el trabajo se debe realizar de forma individual, en parejas o grupal.</i>	
1. Inicio ✓ Ingresar a https://vr.vex.com/ y seleccionar la zona de juegos “castillo de choques dinámico” en VEXcode VR.	Tiempo estimado <i>Minutos o porcentaje</i> 30 minutos
	

- ✓ Mostrar cómo el robot puede detectar un castillo usando el sensor de distancia y colisionar controladamente con él utilizando el sensor parachoques. Para ello, se recomienda desarrollar junto con los estudiantes los algoritmos por separado (sensor de distancia y sensor parachoques) y socializar su funcionalidad antes de integrarlos en un solo programa.

¿Para qué sirve este algoritmo?



¿Para qué sirve este algoritmo?



- ✓ En cuaderno: escribir qué decisiones puede tomar un robot si detecta un castillo a poca distancia.


2. Desarrollo

- ✓ Reto:

Proponga a los estudiantes un reto práctico. Indíqueles que deben modificar los algoritmos vistos previamente para que, utilizando el sensor de distancia y el sensor de color, el robot detecte los castillos, tome decisiones y los derribe sin salirse de la pista.

Motíuelos planteando una dinámica competitiva: el equipo que logre tumbar todas las estructuras correctamente, en el menor tiempo y con mayor precisión, será el ganador del desafío. Esta actividad les permitirá aplicar e integrar conocimientos de sensores, trayectorias y lógica condicional en un contexto lúdico y retador.

80 minutos

	
<p>3. Cierre</p> <ul style="list-style-type: none"> ✓ Socializar junto con los estudiantes las estrategias utilizadas: ¿qué distancia configuraron para detectar un castillo? ¿cómo reaccionó el robot tras la colisión? ✓ Reflexión escrita: <ul style="list-style-type: none"> • ¿Qué aprendí al combinar sensores para cumplir un objetivo? • ¿Dónde se aplican estos sistemas de detección y reacción en la vida real? 	<p>10 minutos</p>
<p>Adaptaciones</p> <ul style="list-style-type: none"> • Zonas con baja conectividad: Se puede implementar la actividad utilizando la plantilla impresa de la zona de juegos “castillo de choques dinámico”, junto con recortes de bloques de programación (disponibles en el Anexo B). Los estudiantes construirán secuencias sobre papel, desplazando una ficha que represente el robot y trazando su recorrido con colores, flechas o símbolos. Esta versión permite desarrollar pensamiento lógico y habilidades de secuenciación sin necesidad de dispositivos digitales. • Grupos grandes: Se recomienda organizar el aula por estaciones rotativas donde los equipos pequeños se turnen para diseñar algoritmos, ejecutar simulaciones, analizar estrategias y registrar resultados, optimizando así el uso de recursos disponibles. Cada equipo puede asumir roles rotativos como programador, estrategia, documentador y evaluador, lo que fomenta la participación equitativa • Diversidad cultural y de género: Promover el trabajo en parejas mixtas y asignar roles rotativos (programador, evaluador, diseñador de recorrido). Incluir ejemplos de uso de sensores en contextos culturales diversos: defensa de territorios, automatización de sistemas agrícolas locales, etc. 	

- **Estudiantes con discapacidad visual leve o total:** Los estudiantes con baja visión pueden utilizar lupas manuales, hojas impresas en alto contraste o ampliadas. Para estudiantes con ceguera total, se sugiere incorporar plantillas táctiles, descripciones verbales del recorrido y apoyo de compañeros mediante trabajo colaborativo. También se puede usar un lector de pantalla para navegar en el entorno de programación.

Actividades evaluativas

Para evaluar los aprendizajes esperados, el docente puede apoyarse en tres evidencias clave:

- Ejecución del algoritmo en VEXcode VR: El robot debe detectar castillos, acercarse a ellos y derribarlos mediante colisión. Debe reaccionar si choca con otros obstáculos.
- Registro en el cuaderno: Representar el recorrido, los puntos de detección y colisión, y las acciones tomadas.
- Reflexión escrita: ¿Qué función cumplió cada sensor? ¿Cómo mejoraría la precisión para derribar los castillos?

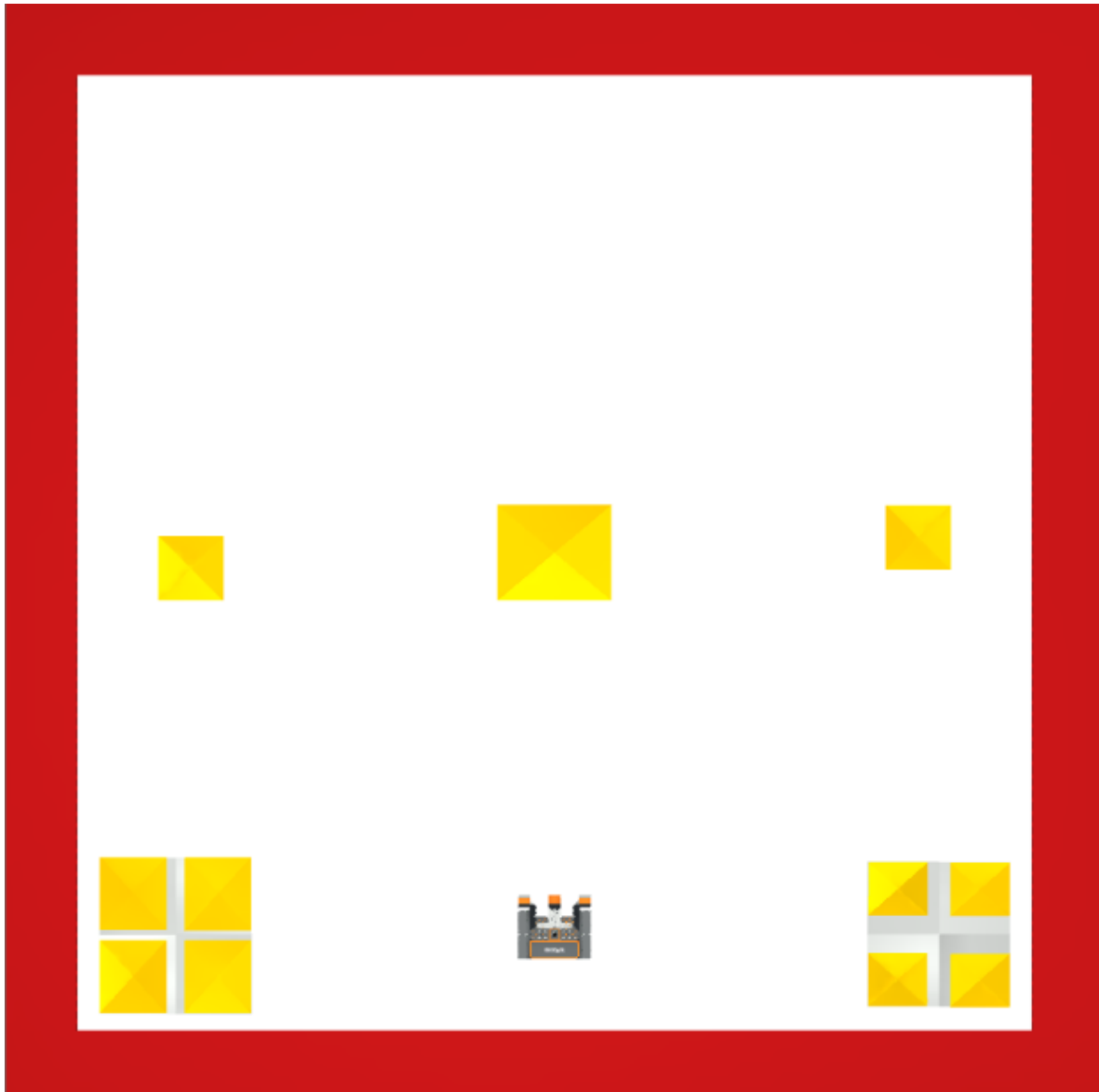
Referencias

VEX Robotics. (2024). *VEXcode VR Activities*.
<https://education.vex.com/vr/>
 VEX Robotics. (2025). *VEXcode VR sin conexión*.
<https://kb.vex.com/hc/es/articles/8608511437076-Usando-VEXcode-VR-sin-conexi%C3%B3n>

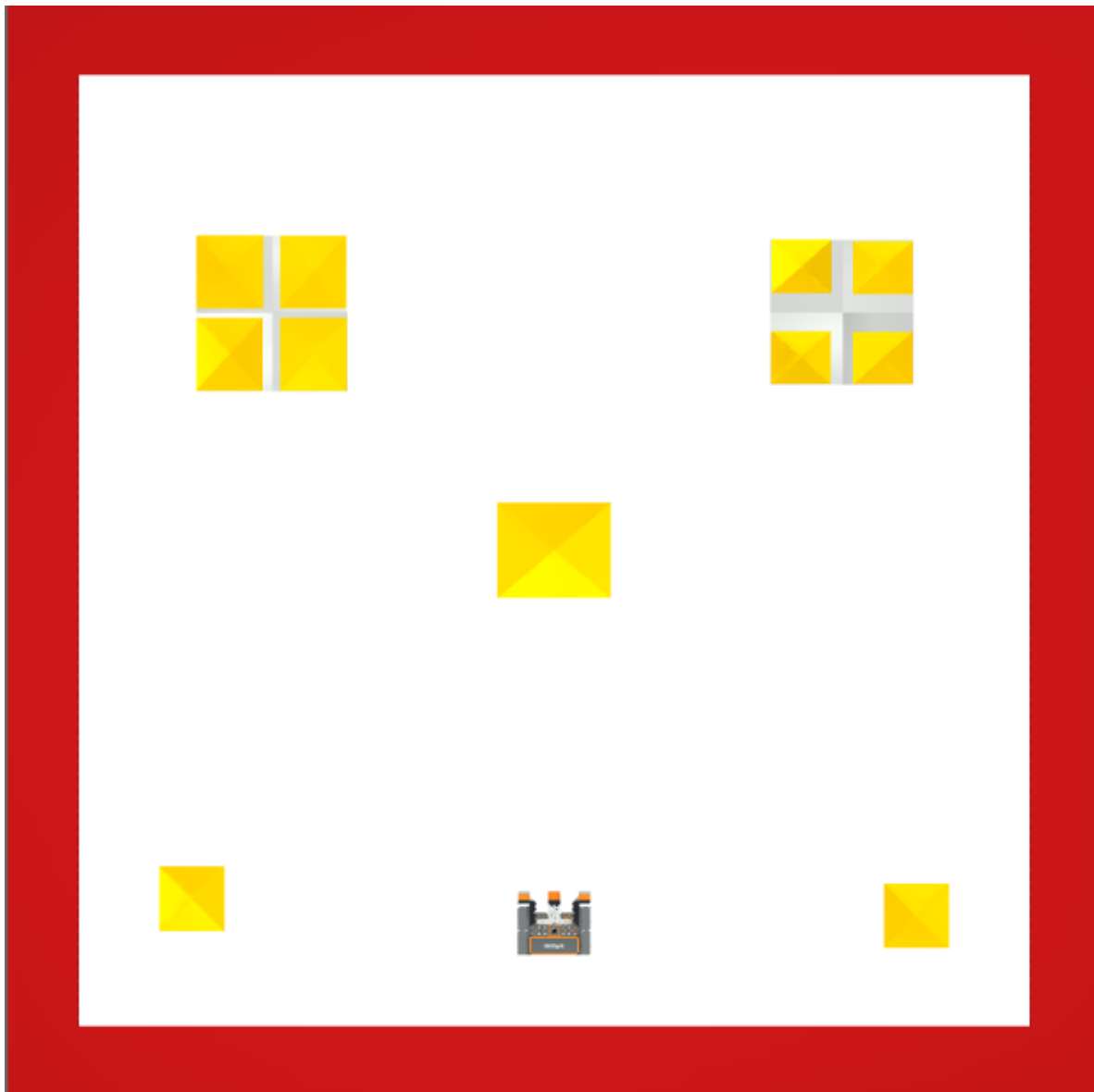
ANEXOS

ANEXO A. Plantillas zonas de juego “castillo de choques dinámico” para actividades desconectadas

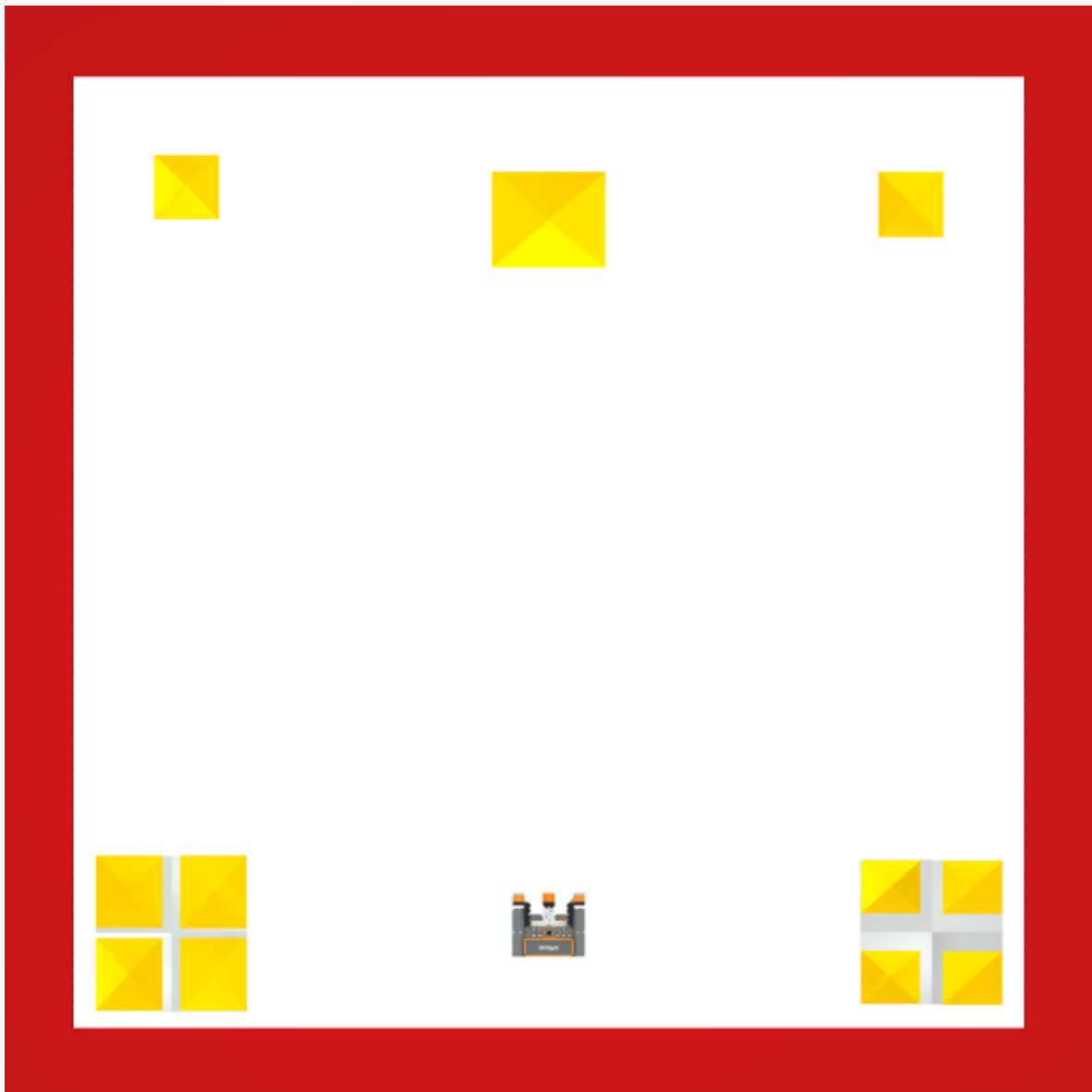
1.



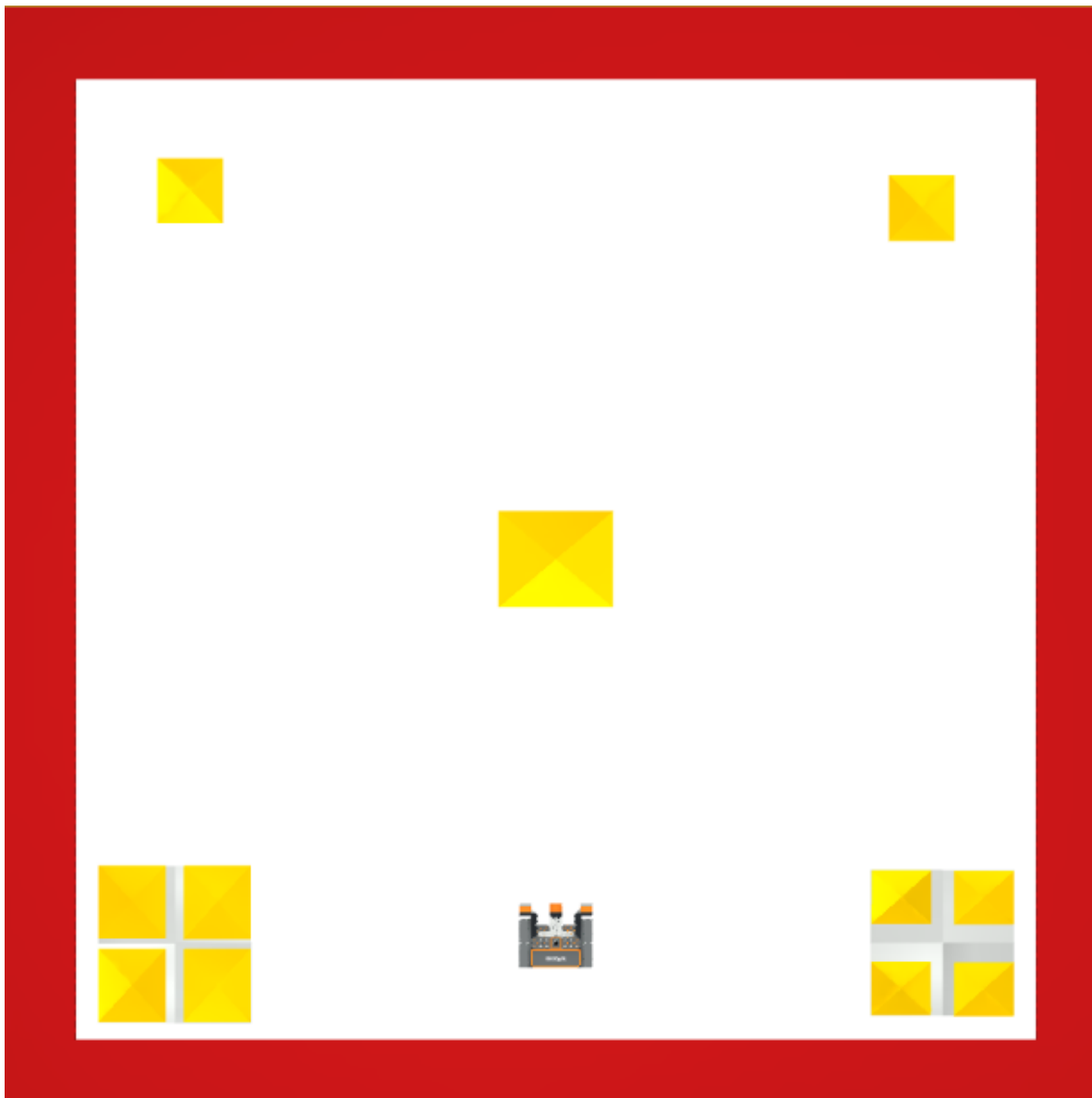
2.



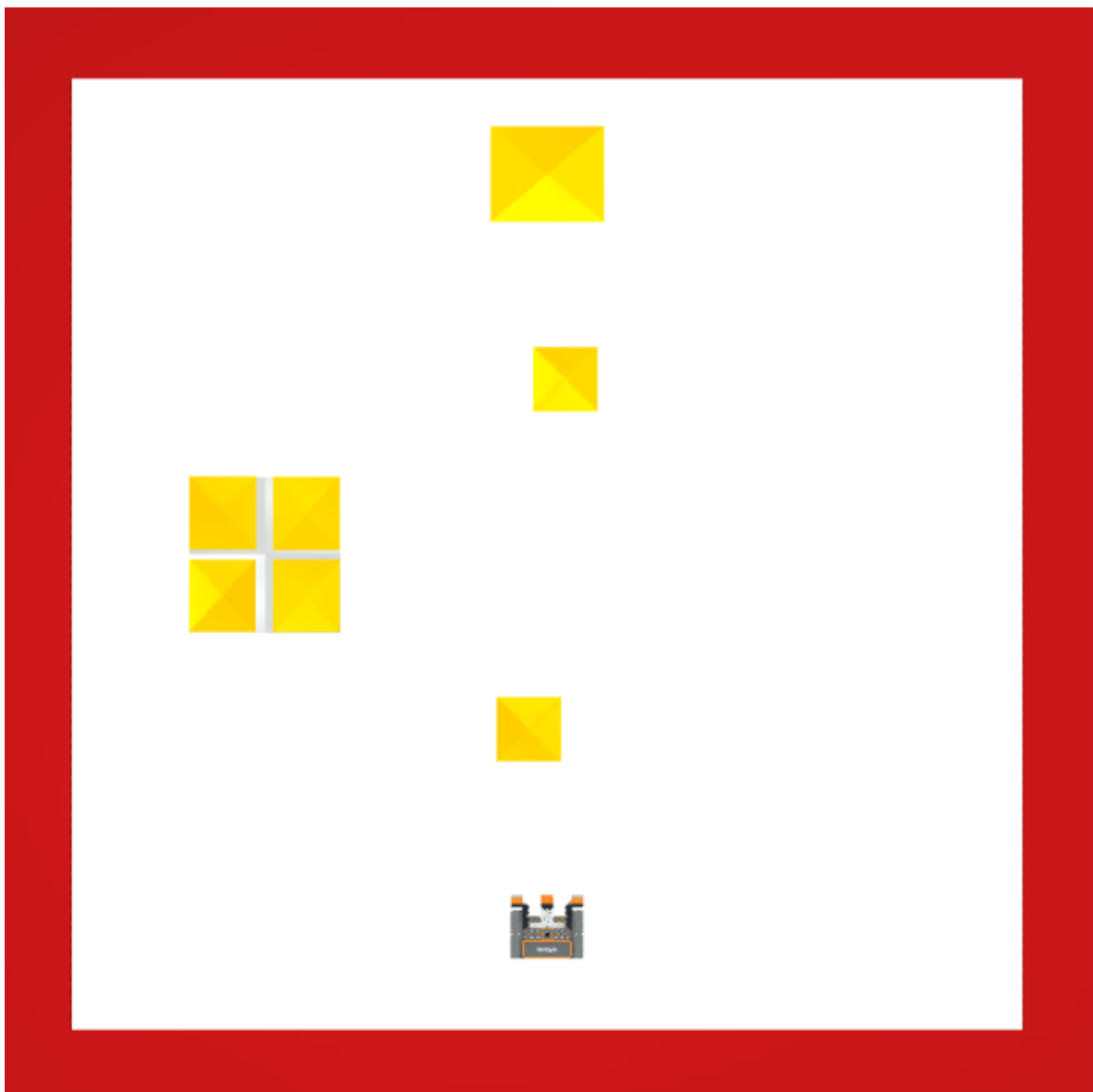
3.



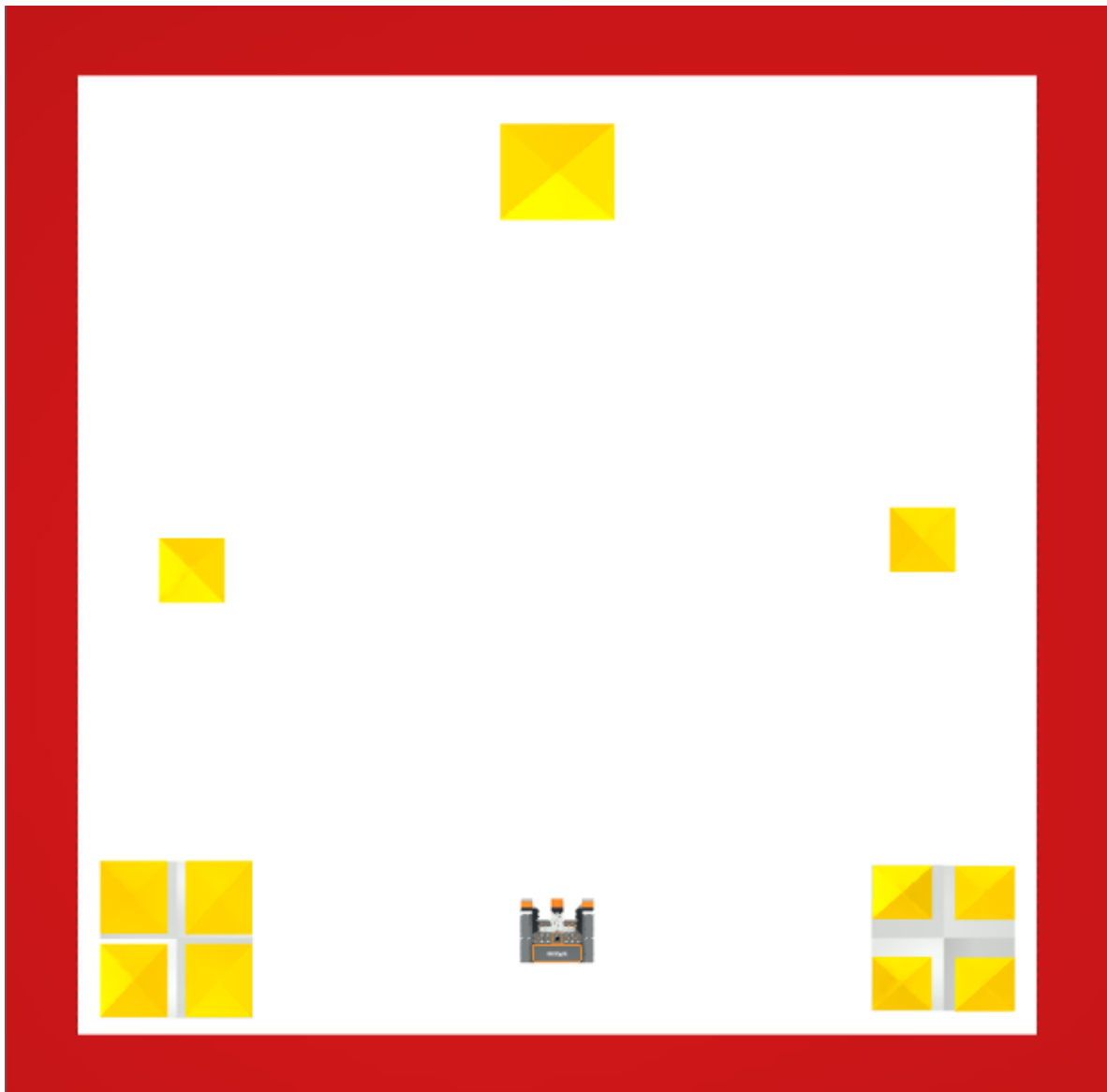
4.



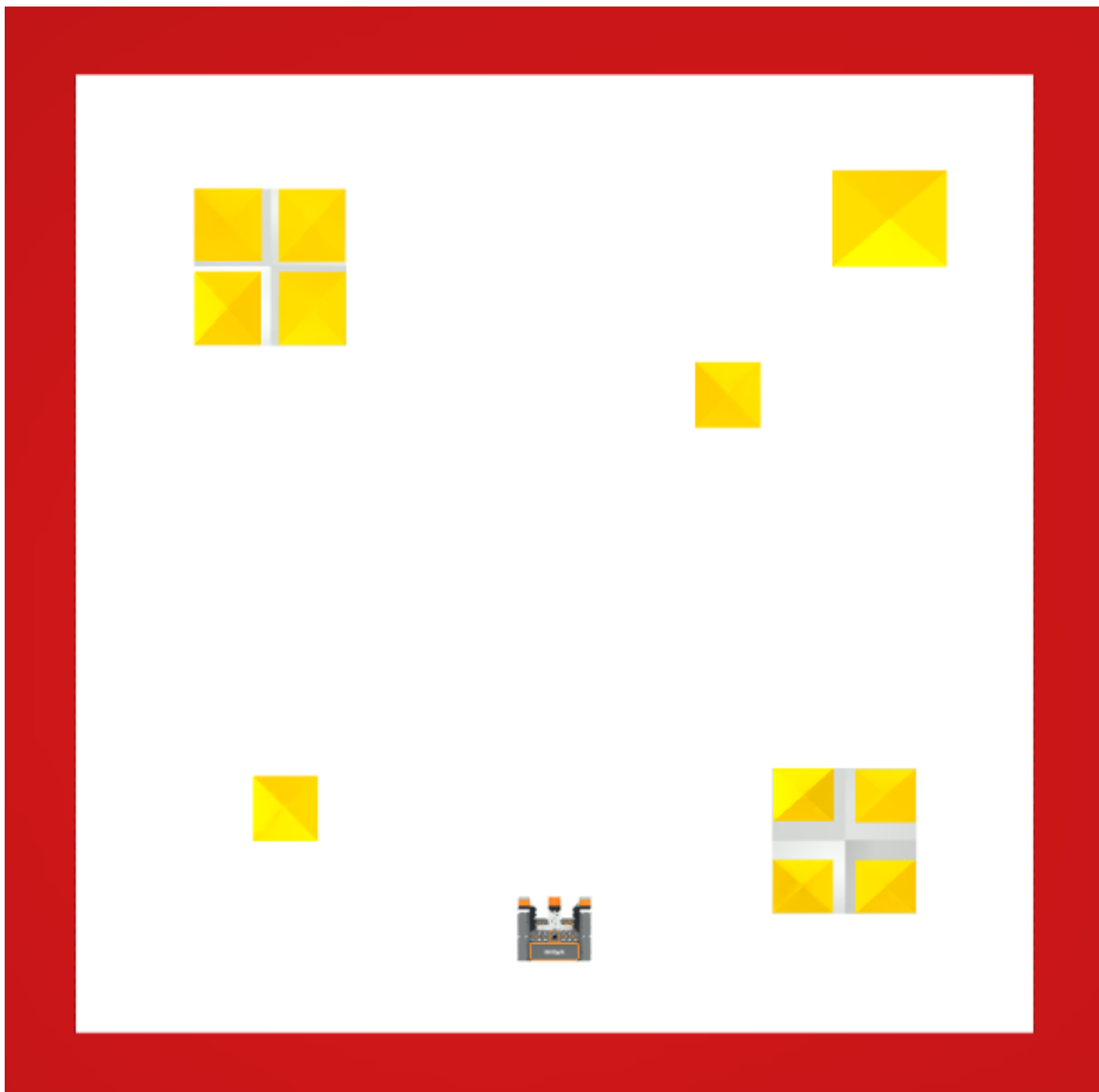
5.



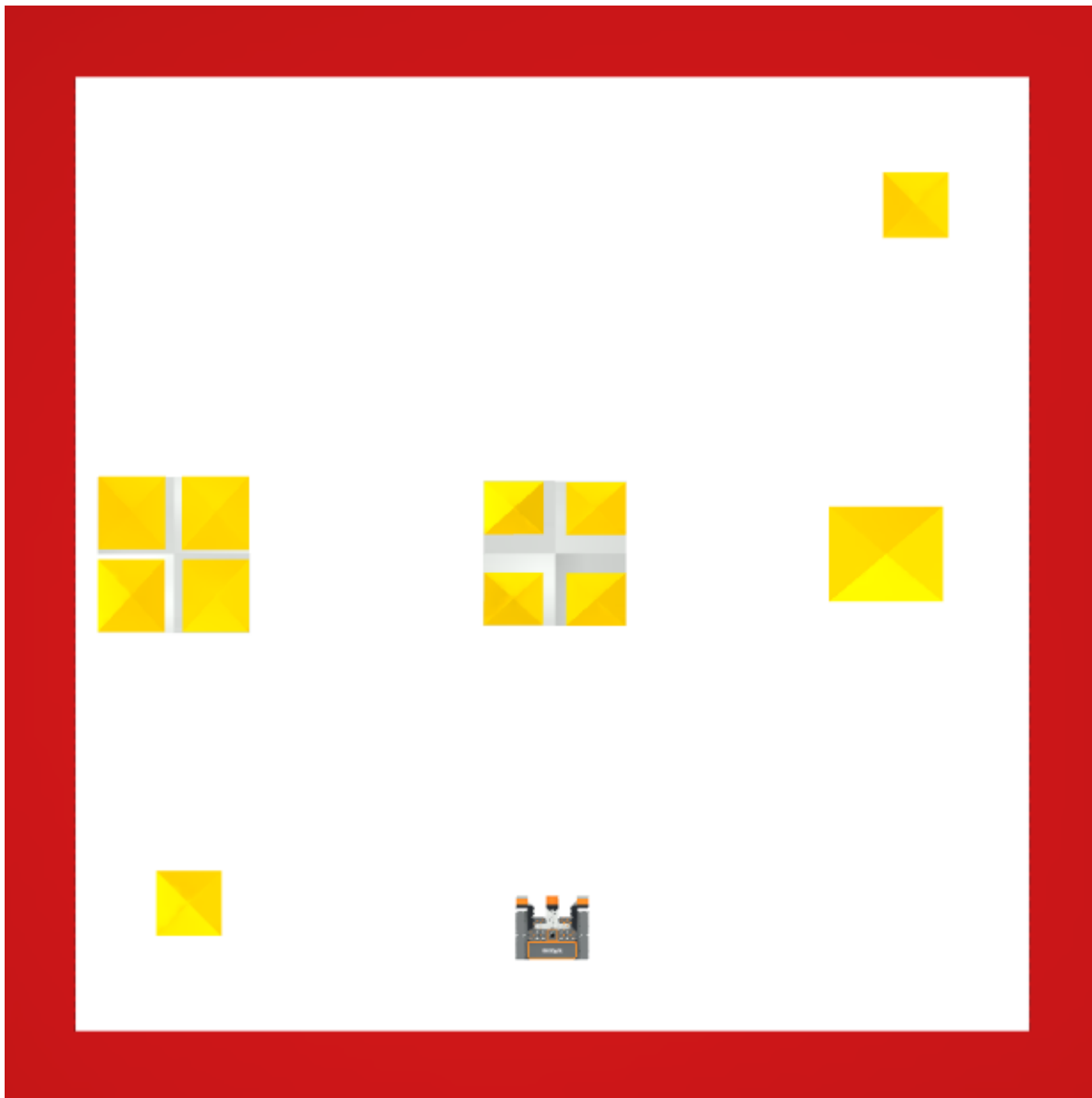
6.



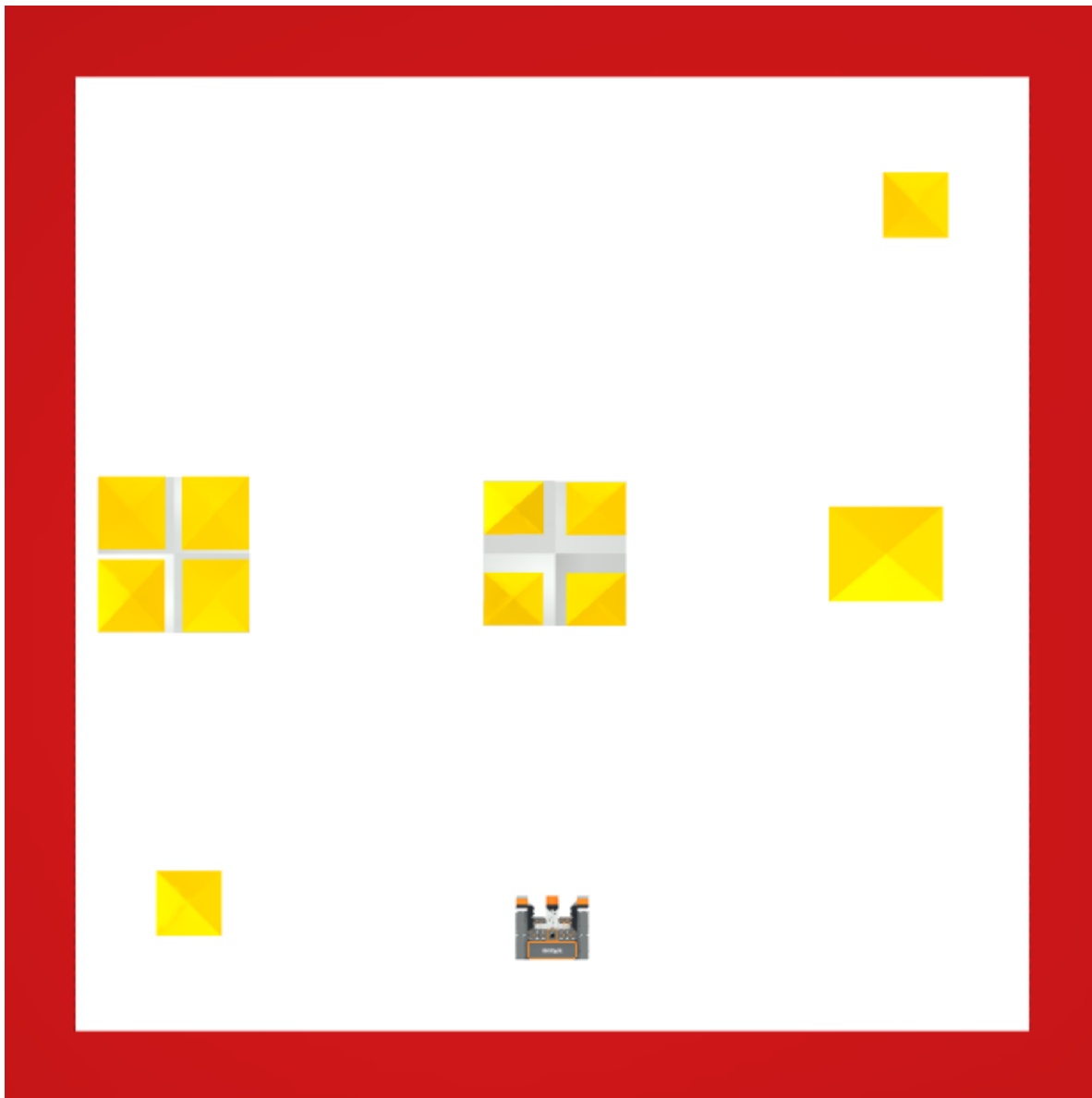
7.



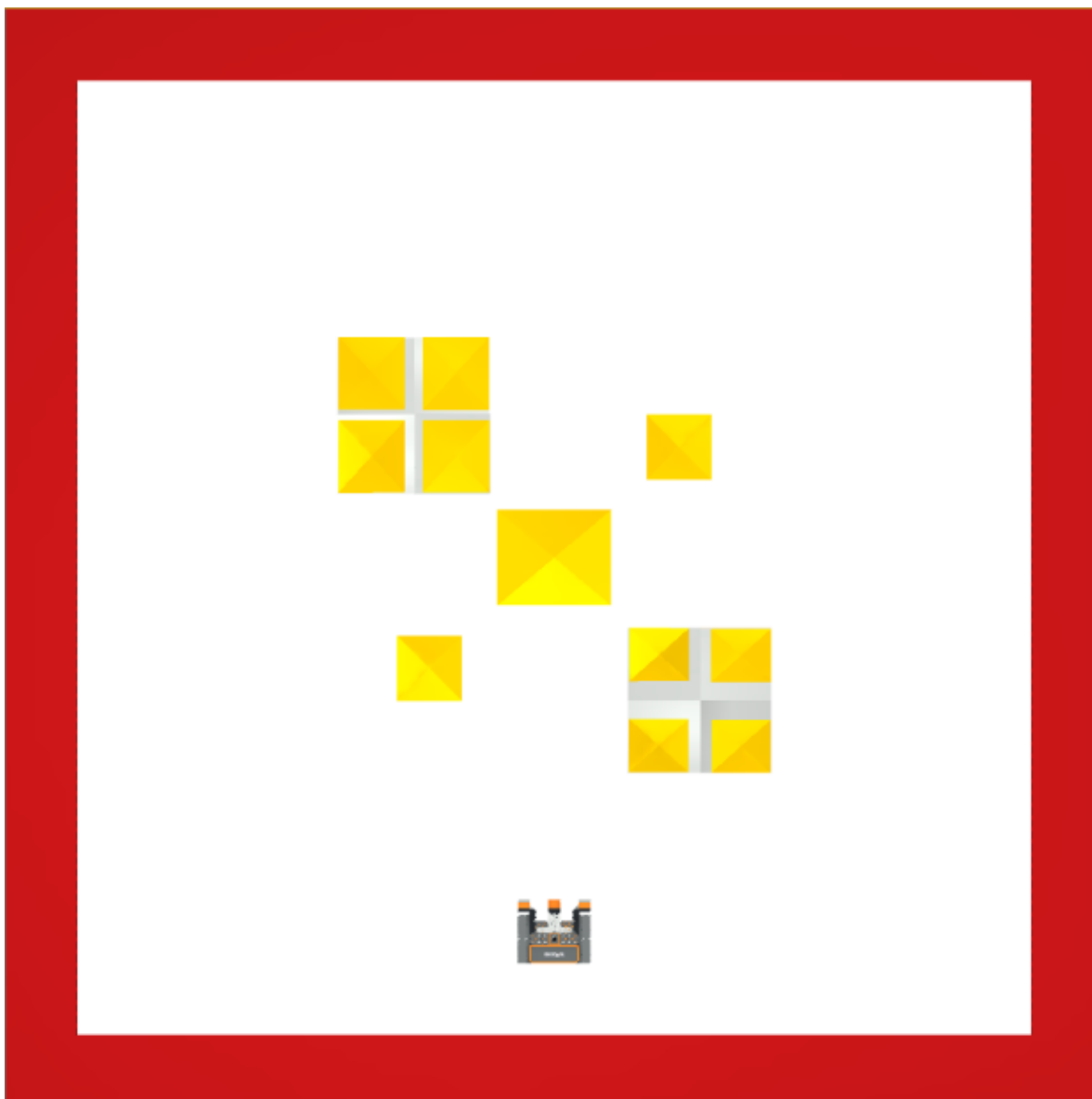
8.



9.



10.



ANEXO B. Bloques para imprimir

Bloques



ANEXO C. Solución al Reto

