وزارة التربية
Ministry of Education
State of Kuwait | دولـــة الكويت

دليل المعلم

الروبوت التعليمي ■ الفصل الأول

دليل المعلم

الروبوت التعليمي

إشـراف

أ/ منى سـالم عـوض سـالم (رئيس اللجنة)

إعـــداد

د/ أشرف رضوان رضوان سليمان

أ/ ســاره مبـارك الحيـس العازمـي

أ/ نوريـه حسـين علي بوحـمد

أ/ هنـاء دغـش بويتـل المطـيري

تصميم وإخـراج

أ/ ساره ياسين عبدالله الأمير

أ/ نوريـه حسـين علي بوحـمد

المـــحـــتـــــوى

	مقدمة	1
1	الاستراتيجية التعليمية	2
	استراتيجيات الصف السادس	4
	استراتيجيات الصف السابع	5
	استراتيجيات الصف الثامن	6
	استراتيجيات الصف التاسع	7
2	استراتيجيات مقترحة للتقويم المستمر	8
3	نـظام STEAM	12
4	الروبوت في التعليم	14
5	استخدام منصة ORL	18
	التعريف بواجهة ORL	19
	التعرف على بيئة العمل RCJ Rescue Online Sim	20
	أساسيات البرمجة بالمنصة	25
	استخدام المحاكي في ORL	29
6	دروس الصف السادس	31
	- وصف الوحدة	32
	- الدرس الأول	34
	- الدرس الثاني	39
	- الدرس الثالث	44
7	دروس الصف السابع	49
	- وصف الوحدة	50
	- الدرس الأول	52
	- الدرس الثاني	56
	- الدرس الثالث	60
8	دروس الصف الثامن	65
	- وصف الوحدة	66
	- الدرس الأول	68

- الدرس الثاني 73
- الدرس الثالث 77

9 دروس الصف التاسع 82
- وصف الوحدة 83
- الدرس الأول 85
- الدرس الثاني 90
- الدرس الثالث 95

10 خطة الدرس 100
- مقدمة عن خطة الدرس 101
- سجل الإعداد الكتابي لخطة الدرس 102
- خطوات اعداد خطة الدرس 103
- أخطاء شائعة يقع فيها المعلم عند إعداد خطة الدرس 105
- نموذج خطة الدرس 106
- مقترح خطة درس روبوت 108
- ملاحظات عامة للمعلم 111
- إرشادات للمعلم عند التدريس باستخدام المحاكي 112

11 دليل المعلم لحل مشاكل ORL 114

12 دروس المواءمة للسنة الدراسية 2026/2025 118

درس الصف السابع 119

درس الصف الثامن 129

درس الصف التاسع 140

13 تحدي الأبطال 153

تحدي الأبطال Free Space 155

تحدي الأبطال Maze 157

تحدي الأبطال Line Following 160

تحدي الأبطال Sumo Battle Arena 164

14 المصادر 167

المقـــــدمة

عزيزي المعلم

يسعدنا أن نقدم لك هذا الدليل الشامل الذي يهدف إلى دعمك في تقديم مادة الحاسوب لمتعلمي المرحلة المتوسطة، بما يتوافق مع أحدث التوجّهات التربوية، ويعكس تطورات العصر الرقمي، ويلبّي احتياجات المتعلمين في مجال التقنية الرقمية ويواكب احتياجات المتعلمين في عالم الروبوتات والبرمجة.

يركّز هذا المنهج على تمكين المتعلمين من أساسيات التفكير الحاسوبي، وتنمية مهارات البرمجة الحركية، وتطوير قدراتهم على الابتكار وحل المشكلات من خلال أنشطة وتجارب عملية باستخدام الروبوتات التعليمية وبرامج المحاكاة. كما يسعى إلى تعزيز العمل التعاوني، وتنمية مهارات التواصل الفعّال والتفكير النقدي عبر مشروعات وتطبيقات ممتعة.

لقد رُوعي في تصميم هذا الدليل أن تتكامل المفاهيم التقنية مع القيم التربوية والوطنية، حيث يُسهم المنهج في ترسيخ الهوية الوطنية الكويتية في نفوس المتعلمين، وتعزيز المواطنة الرقمية، من خلال التوعية بالاستخدام المسؤول للتقنية، والالتزام بأخلاقيات التعامل مع الروبوتات والبرمجيات، والتفكير الواعي في بيئة رقمية سريعة التغير.

كما يتناغم هذا المنهج مع توجهات رؤية الكويت 2035، التي تسعى إلى إعداد جيل قادر على التعامل بفعالية مع التكنولوجيا الحديثة، وتسخيرها في خدمة المجتمع والتنمية المستدامة. ويقدّم محتوى هذا الدليل فرصًا تعليمية متنوعة قائمة على التعلم النشط، والمشروعات، والتطبيقات العملية، التي تتيح للمعلم خيارات متعددة لتنفيذ الدروس وفق طبيعة الفصل واحتياجات المتعلمين.

يوفر الدليل لك خطة واضحة لكل وحدة دراسية، تشمل: الأهداف، والمهارات المستهدفة، والاستراتيجيات المقترحة، والأنشطة الختامية، مع نماذج لتقييم الأداء. كما يحتوي على مصادر إثرائية وأمثلة توضيحية تساعدك على تقديم المحتوى بطريقة جذابة وعملية.

نحن على يقين أن هذا الدليل سيكون رفيقك الأمثل في رحلتك التعليمية، ويساعدك على تقديم دروس ممتعة، تسهم في تطوير مهارات المتعلمين التقنية، وتعزيز ثقتهم في التعامل مع الروبوتات والتقنيات الحديثة بكفاءة وفاعلية.

مع أطيب التمنيات

01

الاستراتيجيات التعليمية

تتوافق مع مناهج الروبوت من الصف السادس وحتى التاسع
تراعي أسلوب STEAM وتدرج الصعوبة.

عـزيزي المعـــلم

في ظل التحولات المتسارعة في ميدان التعليم وتكنولوجيا المعلومات، أصبح من الضروري تبنّي استراتيجيات تعليمية حديثة تسهم في جعل العملية التربوية أكثر حيوية وتفاعلية، وتحوّل بيئة الصف إلى مساحة نشطة للتجريب والاكتشاف. إن الاعتماد على طرق التدريس التقليدية القائمة على التلقين والإلقاء لم يعد كافيًا لإكساب المتعلمين المهارات والمعارف المطلوبة في هذا العصر، بل بات لزامًا على المعلم أن ينوع أساليبه بما يتلاءم مع الفروق الفردية بين الطلاب، ويستجيب لميولهم واهتماماتهم المختلفة.

فالاستراتيجية التعليمية تمثل الخطة التي ينظم من خلالها المعلم الموقف التعليمي، من حيث اختيار الطريقة المناسبة، والأدوات والوسائل، وطبيعة النشاط، والتوقيت الملائم، بما يضمن تحقيق الأهداف المنشودة. وقد روعي في تصميم استراتيجيات تعليم الروبوت للمرحلة المتوسطة أن تنسجم مع نظام STEAM، الذي يدمج العلوم والتقنية والهندسة والرياضيات في سياق واحد متكامل، ويعتمد على التعلم القائم على المشكلات والمشروعات. كما أخذت هذه الاستراتيجيات بعين الاعتبار تدرج الصعوبة من صف لآخر، بحيث ينتقل الطالب من الأساسيات البسيطة في الصف السادس، إلى تطبيقات أكثر عمقًا وتعقيدًا في الصف التاسع، مرورًا بأنشطة عملية في الصف السابع ومشروعات مصغّرة في الصف الثامن.

إن الاستخدام المدروس للاستراتيجيات التعليمية يسهم في رفع دافعية الطلاب، وتنمية مهاراتهم في التفكير النقدي والإبداعي، وتعزيز قدراتهم على حل المشكلات بصورة منهجية. كما أن الدمج بين أكثر من استراتيجية في الدرس الواحد يثري الموقف التعليمي، ويفتح المجال أمام أنماط تعلم متنوعة، ويساعد على

إتاحة فرص متوازنة لجميع الطلاب للمشاركة والتفاعل وبناء المعرفة الذاتية.

ومع اتساع دائرة التحول الرقمي وتنوّع الأدوات التقنية المتاحة، يبرز دور المعلم في اختيار الاستراتيجية الأنسب وتوظيفها بطريقة تدعم نواتج التعلم المستهدفة، وتواكب مهارات القرن الحادي والعشرين، ليصبح المتعلم قادرًا على استخدام الروبوتات والتقنيات الذكية بوعي وإبداع، وتسخيرها في خدمة مجتمعه وتنميته المستدامة.

استراتيجيات الصف السادس

مرحلة التعرف والاستكشاف

هدف المرحلة: الاستكشاف وفهم المبادئ الأساسية للروبوت والبرمجة.

التعلم القائم على الاستكشاف

أنشطة بسيطة للتعرف على مكونات الروبوت والمستشعرات (إن وجد).

مثال

- فتح مشروع جديد في محاكي Open Roberta Lab، وتجربة أوامر تشغيل المحركات والتحكم بسرعة الروبوت، ثم ملاحظة الفرق عند تغيير القيم. الهدف هو أن يكتشف المتعلم بنفسه كيف تؤثر الكتل البرمجية على حركة الروبوت.

التعلم التعاوني

تقسيم المتعلمين إلى مجموعات صغيرة لتجميع روبوت بسيط أو برمجته لتنفيذ مهمة سهلة.

مثال

- تقسيم المتعلمين إلى مجموعات من 3–4 متعلمين، وكل مجموعة تبرمج الروبوت في المحاكي ليتحرك في مسار مربع باستخدام مستشعر الدوران أو مؤقت الوقت، بعد ذلك تتبادل المجموعات برامجها لاختبارها ومناقشة التحسينات.

التعلم بالألعاب التعليمي (Gamification)

مسابقات خفيفة تعتمد على تنفيذ أوامر برمجية صحيحة.

مثال

- تنظيم مسابقة على المحاكي بعنوان "سباق الروبوت"، حيث يُطلب من المتعلمين برمجة الروبوت ليصل إلى نقطة محددة بأقل وقت ممكن، مع الالتزام بقواعد المسار. تمنح نقاط للسرعة، والدقة، ووضوح فكرة البرنامج.

التجارب العملية القصيرة

ربط مفاهيم العلوم (مثل الحركة والطاقة) بمشاريع روبوتية صغيرة.

مثال

- ربط درس العلوم عن "الطاقة الحركية" بأن يبرمج المتعلمين الروبوت في المحاكي للتحرك بسرعات مختلفة، ثم يلاحظون المسافة التي يقطعها قبل التوقف. يتم مناقشة كيف تؤثر زيادة السرعة على المسافة والحركة.

استراتيجيات الصف السابع

مرحلة التطبيق الموجه

هدف المرحلة: تنمية القدرة على تصميم برامج وتنفيذ مشاريع روبوتية بسيطة.

التعلم القائم على حل المشكلات

تقديم مشكلات محدودة النطاق تتطلب برمجة الروبوت لحلها.

مثال

- يواجه الروبوت في المحاكي متاهة بها منعطفات وزوايا قائمة. يطلب من المتعلمين برمجته لاجتياز المتاهة باستخدام مستشعر المسافة لتجنب الاصطدام، مع تعديل الأكواد حتى يصل إلى الهدف بنجاح.

العصف الذهني

لتوليد أفكار لتطوير أداء الروبوت أو تحسين تصميمه.

مثال

- عرض برنامج أولي يجعل الروبوت يتراجع ويلمس العائق الخلفي مرة واحدة فقط. يقوم المتعلمين بالعصف الذهني لإضافة تحسينات، مثل: جعل الروبوت يكرر العملية عدة مرات، أو تغيير سرعة الحركة للخلف، أو إضافة شرط باستخدام مستشعر المسافة لتفادي عقبات أمامية قبل التحرك مجدداً.

التعلم القائم على النمذجة (Modeling)

استخدام محاكي مثل ORL لمحاكاة التجارب قبل تنفيذها.

مثال

- قبل تنفيذ تجربة على الروبوت الحقيقي، يستخدم المتعلمين المحاكي لمحاكاة سيناريو "اختبار المستشعرات"، حيث يتراجع الروبوت حتى يلمس العائق الخلفي، ثم يتحرك للأمام ويتوقف عند بُعد معيّن باستخدام مستشعر المسافة. يتم تعديل البرنامج وضبط أوقات التوقف داخل حلقة تكرار حتى يعمل بسلاسة.

التعلم التكاملي (STEAM Integration)

ربط البرمجة بالرياضيات (مثل الحسابات الهندسية) وبالعلوم (مثل قياس المسافة والسرعة).

مثال

- ربط الرياضيات بالحركة من خلال حساب الزمن اللازم لقطع المسافة الأمامية بعد ملامسة العائق الخلفي. يبرمج المتعلمين الروبوت ليتراجع للخلف حتى يضغط مستشعر اللمس، ثم يتحرك للأمام بسرعة محددة ويتوقف عند مسافة 15 سم من جسم أمامه (باستخدام مستشعر المسافة)مع وضع الكود داخل حلقة تكرار لمراقبة الأداء باستمرار.

استراتيجيات الصف الثامن

مرحلة الإبداع والتطوير

هدف المرحلة: تطوير مهارات التفكير المنطقي والإبداعي وتنفيذ مشروعات متقدمة.

التعلم القائم على المشاريع

تنفيذ مشروع روبوت لحل مشكلة من واقع الحياة المدرسية أو المحلية.

مثال

- تنفيذ مشروع "روبوت متتبع للخط" لحل مشكلة تنظيم السير في ممرات المدرسة. يستخدم المتعلمين مستشعر اللون لاكتشاف الخط الأسود على الأرضية، Gyroscope sensorg للحفاظ على الاتجاه في المنعطفات، ودوال مخصصة لتقسيم البرنامج إلى مهام مثل (اتبع الخط – توقف عند اللون الأحمر – عدّل الاتجاه).

الصف المقلوب

دراسة المفاهيم البرمجية المتقدمة في المنزل عبر فيديوهات، ثم تطبيقها عملياً في الصف.

مثال

- يشاهد المتعلمين في المنزل فيديو يشرح كيفية استخدام Gyroscope sensor لقياس الزوايا بدقة، وكيفية إنشاء دوال لإعادة استخدام الكود. في الصف، يبرمجون الروبوت في المحاكي ليدور 90° بدقة باستخدام Gyroscope sensor، مع وضع هذا الكود في دالة اسمها "Turn_90"، ثم استدعائها عدة مرات داخل البرنامج الرئيسي..

التعلم بالأقران

تبادل الخبرات بين المتعلمين ومراجعة أكواد بعضهم البعض لتحسينها.

مثال

- بعد برمجة مشروع "روبوت الفرز اللوني" باستخدام مستشعر اللون لتحديد لون المكعبات، ينشئ المتعلمين دوال. يقوم المتعلمين بتبادل الأكواد ومراجعتها مع زملائهم لتحسين تنظيم الدوال وزيادة كفاءة البرنامج.

دمج الهندسة والتصميم

تصميم هياكل روبوتية أكثر تعقيدا تراعي الجوانب الميكانيكية.

مثال

- تصميم روبوت يقوم بجولة تفقدية للممرات، يستخدم Gyroscope sensor للحفاظ على المسار المستقيم، ومستشعر اللون للكشف عن علامات أرضية تشير للتوقف أو الانعطاف. يضع المتعلمين أوامر التحكم في الحركة داخل دوال، مثل "MoveForward" أو "Turn_On_White"، ويختبرون البرنامج على المحاكي قبل بناء النموذج الواقعي.

استراتيجيات الصف التاسع

مرحلة الابتكار والتحدي

هدف المرحلة: تمكين المتعلمين من تصميم حلول روبوتية مبتكرة متكاملة.

التعلم القائم على الابتكار (Innovation-Based Learning)

ابتكار فكرة مشروع جديدة وتنفيذها من الصفر.

مثال

- ابتكار روبوت دفاعي لتحدي السومو في المحاكي، يستخدم خوارزمية PID للتحكم في سرعته واتجاهه بدقة، ويعتمد على قائمة تخزن مواقع الخصم التي تم رصدها عبر مستشعر المسافة. يقوم المتعلمين ببرمجة الروبوت بحيث يتنقل بين هذه المواقع ويعدل استراتيجيته تلقائياً.

التعلم القائم على البحث

دراسة تكنولوجيا أو مستشعرات جديدة وكيفية دمجها في الروبوت.

مثال

- يكلف المتعلمين بالبحث عن تطبيقات خوارزمية PID في الروبوتات الصناعية، ثم استكشاف كيف يمكن دمجها في محاكي Open Roberta Lab لتحسين أداء روبوت السومو أو متتبع الخط. بعد ذلك، يقومون بإنشاء قائمة لتسجيل القراءات اللحظية للمستشعر وتحليلها لضبط معاملات ال PID .

المنافسات الروبوتية المصغرة

محاكاة مسابقات مثل السومو أو تتبع الخط مع وضع شروط ومتغيرات.

مثال

- تنظيم مباراة "سومو افتراضي" على المحاكي، حيث يبرمج كل فريق الروبوت لاكتشاف حدود الحلبة باستخدام مستشعر اللون وتجنب الخروج منها، مع ملاحقة الخصم باستخدام مستشعر المسافة. تُستخدم قائمة لتخزين مواقع العدو المحتملة، ويطبّق المتعلمين منطق برمجي لاتخاذ القرار المناسب بناءً على البيانات المخزنة.

التعلم التكاملي (STEAM Integration)

ربط العلوم (فيزياء الحركة)، التكنولوجيا (برمجة المستشعرات)، الهندسة (تصميم الهياكل)، والرياضيات (تحليل البيانات) في مشروع نهائي.

مثال

- تنفيذ مشروع نهائي يجمع بين: العلوم: تطبيق قوانين الاحتكاك والقوة لتحسين قدرة الروبوت على دفع الخصم. التكنولوجيا: برمجة المستشعرات مع خوارزمية PID للتحكم في السرعة والدقة. الهندسة: تصميم هيكل روبوت يتوازن بين القوة والسرعة. الرياضيات: تحليل بيانات السرعة والمسافة المخزنة في قائمة لحساب المسار الأمثل للهجوم.

02

استراتيجيات مقترحة للتقويم المستمر

في إطار تطبيق مناهج الروبوت للمرحلة المتوسطة، يُعدّ التقييم المستمر عنصرًا أساسيًا في متابعة تقدم المتعلمين، وضمان اكتسابهم للمهارات المستهدفة بصورة تدريجية وفعّالة. ويأتي استخدام مقياس Rubric كأداة عملية ومنظّمة تساعد المعلم على تقويم أداء المتعلمين في مختلف الأنشطة والمشروعات الروبوتية، بطريقة عادلة وموضوعية، بعيدًا عن التقديرات العامة أو الانطباعات الشخصية.

لقد صُمّم مقياس Rubric ليتماشى مع طبيعة دروس الروبوت، حيث يركز على تتبّع مهارات التفكير الحاسوبي، وإتقان خطوات البرمجة الحركية، والقدرة على توظيف بيئة المحاكاة بفاعلية.

كما يُبرز جانب الإبداع في تصميم الحلول، ويشجع على تنمية روح التعاون والعمل الجماعي من خلال مهام مشتركة وتحديات عملية.

ويقوم مبدأ Rubric على تحديد معايير واضحة للأداء، مثل الدقة، والإبداع، وحل المشكلات، والانضباط أثناء التنفيذ، ثم ربط هذه المعايير بمستويات متدرجة من الأداء (ممتاز – جيد جدًا – جيد – يحتاج تحسين)، مع أوصاف دقيقة تتيح للمتعلم فهم نقاط قوته والجوانب التي تحتاج إلى تطوير. وبذلك يصبح التقييم أداة للتعلم بقدر ما هو وسيلة للقياس.

إن اعتماد التقييم المستمر وفق مقياس Rubric يسهم في رفع دافعية المتعلمين نحو التعلم، ويمنحهم تغذية راجعة بنّاءة تساعدهم على تحسين أدائهم أولًا بأول. كما يمكّن المعلم من متابعة نمو مهارات المتعلمين على مدار الفصل الدراسي، وضمان تدرّجهم في استيعاب المفاهيم وتنفيذ التطبيقات العملية بطريقة تواكب أهداف المنهج وتدعم فلسفة STEM في التعليم.

مقياس روبريك (Rubric)

هو أداة مهمة للتقويم المستمر لأنها تعطي المعلم والمتعلم معايير واضحة للأداء المتوقع. عند استخدامها في دروس الروبوت سواء باستخدام LEGO أو Open Roberta Lab أو محاكيات أخرى يمكن أن تكون أكثر فاعلية إذا صيغت بطريقة عملية مرتبطة بالأنشطة.

استراتيجيات التقويم المستمر باستخدام مقياس Rubric

1. المعايير الواضحة (Clear Criteria)

تحديد ما يجب على المتعلم إتقانه البرمجة الصحيحة، توظيف الحساسات، حل المشكلة بطريقة إبداعية، التعاون مع الفريق.

مثال:

"البرنامج يعمل بدون أخطاء" أو "الروبوت ينجز المهمة في الوقت المحدد".

2. مستويات الأداء (Performance Levels)

تقسيم الأداء إلى مستويات (ممتاز – جيد – مقبول – يحتاج إلى تحسين).

مثال:

الروبوت ينجز المهمة 100% من المطلوب" مقابل "الروبوت ينجز نصف المطلوب.

3. التغذية الراجعة الفورية(Immediate Feedback)

إعطاء ملاحظات مباشرة مرتبطة بالبند في مقياس Rubric.

مثال:

الكود صحيح لكن ينقصك استخدام كتلة التكرار لتقليل التكرار اليدوي.

4. التقويم التكويني (Formative Assessment)

استخدام مقياس Rubric في منتصف النشاط لمساعدة المتعلم على تحسين عمله قبل التقييم النهائي.

مثال:

أثناء مهمة برمجة روبوت حارس، يراجع المتعلم القائمة مع معلمه ليعرف أين يطور.

5. مشاركة المتعلمين (Student Involvement)

إشراك المتعلمين في بناء مقياس Rubric أو تقييم أنفسهم وزملائهم به.

مثال:

كل فريق يقيّم فريقاً آخر وفقاً للمعايير: التنظيم – الإبداع – فعالية الحل.

كيفية تطبيق مقياس Rubric على دروس الروبوت

1. في التمارين النظرية

تقييم وضوح الإجابة، توظيف المفاهيم (AND, OR) ، التكرار، القوائم.

مثال :

سؤال يطلب من المتعلم توقع نتيجة برنامج، مقياس Rubric يقيم الدقة + التفسير.

2. في المهمات العملية (على المحاكي أو الروبوت الحقيقي)

المعيار 1: تشغيل الروبوت وإنجاز المهمة.

المعيار 2: استخدام برمجة نظيفة (بدون تكرار غير ضروري).

المعيار 3: التوظيف الصحيح للحساسات أو المنطق.

المعيار 4: التعاون والعمل ضمن الفريق.

3. أثناء المشاريع

مقياس Rubric يقيس الإبداع (حل غير تقليدي) + التوثيق (شرح الكود).

مثال:

مشروع "الروبوت الحارس" يمكن أن يُقيّم على:

1. دقة الكود (40%)
2. نجاح المهمة على المحاكي (30%)
3. الابتكار (20%)
4. العرض والتفسير(10%)

يُعَدّ مقياس Rubric إطارًا مرجعيًا للتقويم المستمر في دروس الروبوت بمختلف المراحل الدراسية، حيث يوفر للمعلم معايير واضحة ومحددة تساعده في قياس مدى تحقق نواتج التعلّم لدى المتعلمين. كما يساهم في متابعة تقدمهم في اكتساب المهارات البرمجية والتقنية، وتعزيز قدراتهم على التفكير المنطقي والإبداعي، بما يضمن جودة العملية التعليمية وتكاملها عبر جميع الصفوف.

03

نظام STEAM

يرمز مصطلح STEAM إلى مجالات: العلوم (Science) ، التكنولوجيا(Technology) ، الهندسة(Engineering) ، الفنون(Arts) ، والرياضيات (Mathematics) ويعتمد هذا النهج التعليمي على الدمج بين هذه المجالات في مواقف تعليمية عملية قائمة على المشاريع، بحيث يكتسب المتعلم المعرفة والمهارات من خلال التجريب، الاكتشاف والتطبيق الواقعي. يساعد هذا الأسلوب المتعلمين على تطوير مهارات القرن الحادي والعشرين مثل: التفكير النقدي، الإبداع، التعاون، حل المشكلات، واستخدام التكنولوجيا بفعالية. كما يُعدّ التعليم عبر STEAM مدخلًا مهمًا لجعل التعلّم أكثر متعة وارتباطاً بحياة المتعلم اليومية، حيث يمكن دمج البرمجة والروبوتات بالعلوم والفنون لإنتاج أنشطةً ثريةً وهادفةً.

أمثلة تطبيقية لدمج STEAM مع درس (الروبوت الحارس):

- العلوم (Science): مناقشة كيفية عمل الموجات فوق الصوتية في مستشعر المسافة، وربطها بمفهوم انتقال الموجات الصوتية وانعكاسها.
- التكنولوجيا (Technology): توظيف منصة Open Roberta Lab لمحاكاة سلوك الروبوت الحارس وتوضيح دور البرمجيات في التحكم بالأنظمة الذكية.
- الهندسة (Engineering): تصميم منطق برمجي يعتمد على القوائم والشروط المركبة (AND/OR) لبناء نظام مراقبة يحاكي أنظمة الأمن الذكية.
- الفنون (Arts): ابتكار سيناريو واقعي للروبوت الحارس (مثل حراسة متحف أو حماية غرفة كنوز)، مع إضافة مؤثرات ضوئية وألوان مختلفة للإنذار.
- الرياضيات (Mathematics): استخدام القوائم لتخزين بيانات المسافة وإجراء مقارنات رقمية، وحساب الزمن باستخدام المؤقت لتنظيم الاستجابة.

من خلال هذه الأمثلة، يدرك المتعلم أن الروبوت ليس مجرد جهاز ميكانيكي، بل هو نموذج متكامل يدمج بين مجالات STEAM ليعكس تطبيقات حقيقية في حياتنا اليومية.

04

الروبوت في التـعليم

منصة Open Roberta Lab

ما المقصود بالروبوت؟

الروبوت هو جهاز مبرمج يمكنه تنفيذ أوامر معينة بشكل تلقائي، مثل الحركة أو الاستجابة للمؤثرات الخارجية من خلال المستشعرات، في مجال التعليم لا يُنظر إلى الروبوت كآلة معقدة، بل كوسيلة عملية لتعليم المتعلمين التفكير المنطقي، حل المشكلات، والعمل الجماعي.

أهمية الروبوت في التعليم؟

يساعد تعليم الروبوتات المتعلمين على:

1. فهم أساسيات البرمجة بطريقة تفاعلية وممتعة.
2. تنمية مهارات التفكير النقدي والإبداع.
3. ربط العلوم والرياضيات بتطبيقات عملية قريبة من حياتهم اليومية.

المحاكيات ودورها في التعليم والتدريب

غالباً ما تحتاج برمجة الروبوتات إلى قطع إلكترونية وأجهزة حقيقية، لكن قد لا تتوفر هذه الموارد دائماً. وهنا تأتي أهمية المحاكيات، وهي بيئات افتراضية تمثل الروبوت على شاشة الحاسوب.

مزايا المحاكيات:

1. التكلفة المادية: لا حاجة لشراء عدة روبوتات لكل متعلم.
2. سهولة التدريب: يمكن للمتعلمين تجربة برامجهم عدة مرات بلا خوف من إتلاف الأجهزة.
3. تشجيع الاستكشاف: يمنح المتعلمين مساحة للتجريب والتعلم من الأخطاء بسرعة.

منصة Open Roberta Lab

ما هي؟

منصة تعليمية مجانية على الإنترنت، طُورت ضمن مشروع Open Roberta من معهد(Fraunhofer IAIS) الألماني بدعم من مؤسسة Google.org .

الغرض منها

تسهيل تعلم البرمجة والروبوتات للأطفال والمتعلمين والمعلمين حول العالم عبر واجهة بسيطة مبنية على الكتل .

ما الذي توفره؟

1. بيئة برمجة بالكتل شبيهة بلغة Scratch.
2. محاكٍ افتراضي لتجربة البرامج على روبوت افتراضي.
3. دعم لأنواع متعددة من الروبوتات مثل: LEGO Mindstorms ، LEGO SPIKE، mBot، Calliope، Micro:bit.
4. إمكانية الحفظ، التصدير، والمشاركة عبر الإنترنت.

اللغة

على الرغم من أن واجهة منصة Open Roberta Lab لا تتوفر باللغة العربية، إلا أنه يمكن للمتعلمين الاستفادة من كتلة "إظهار النص" لبرمجة الروبوت كي يعرض عبارات مكتوبة بالعربية على شاشة المعالج. يتيح ذلك دمج اللغة العربية في الأنشطة العملية، مما يسهم في تعزيز الفهم والتفاعل لدى المتعلمين داخل الصفوف العربية .

ما المهارات التي سيتعلمها المعلم والمتعلمين؟

المعلم:

- التعرف على واجهة منصة Open Roberta Lab وكيفية إدارتها أثناء الحصة.
- اكتساب مهارات تصميم برامج بسيطة بالكتل وتشغيلها على المحاكي.
- القدرة على تنظيم أنشطة صفية عملية باستخدام المنصة دون الحاجة لروبوت حقيقي.
- اكتساب استراتيجيات لشرح البرمجة بطريقة مبسطة وجاذبة للمتعلمين.

المتعلم:

- فهم أساسيات البرمجة بالكتل(BlocKs-based programming) .
- التعرف على مفاهيم برمجية أساسية (الحركة، التكرار، الشروط، المنطق).
- تنمية مهارات التفكير المنطقي وحل المشكلات من خلال تصميم برامج.
- بناء الثقة في استكشاف الأفكار وتجربة الحلول وتعديلها ذاتياً.

كيف تساعد المنصة في تدريس التفكير المنطقي والبرمجة؟

- توفر واجهة مرئية مبسطة تجعل المتعلم يرى خطوات الحل بوضوح.
- الكتل البرمجية مصممة لتقليل الأخطاء النحوية (Syntax Errors) ، مما يسمح للمتعلم بالتركيز على الفكرة والمنطق.
- المحاكي يعطي تغذية راجعة فورية عند تشغيل البرنامج، فيتعلم المتعلم من نتائج تجربته بسرعة.
- الأنشطة الصفية المبنية على المنصة تشجع المتعلمين على استخدام التفكير الحسابي(Computational ThinKing) .
  - التجزيء: تقسيم المشكلة الكبيرة إلى خطوات صغيرة.
  - التجريد: التركيز على ما هو مهم وترك التفاصيل الثانوية.
  - التعرف على الأنماط: ملاحظة تكرار الحلول وتعميمها.
  - الخوارزميات: صياغة خطوات دقيقة لحل المشكلة.

ربط المهارات بأهداف المنهج الدراسي

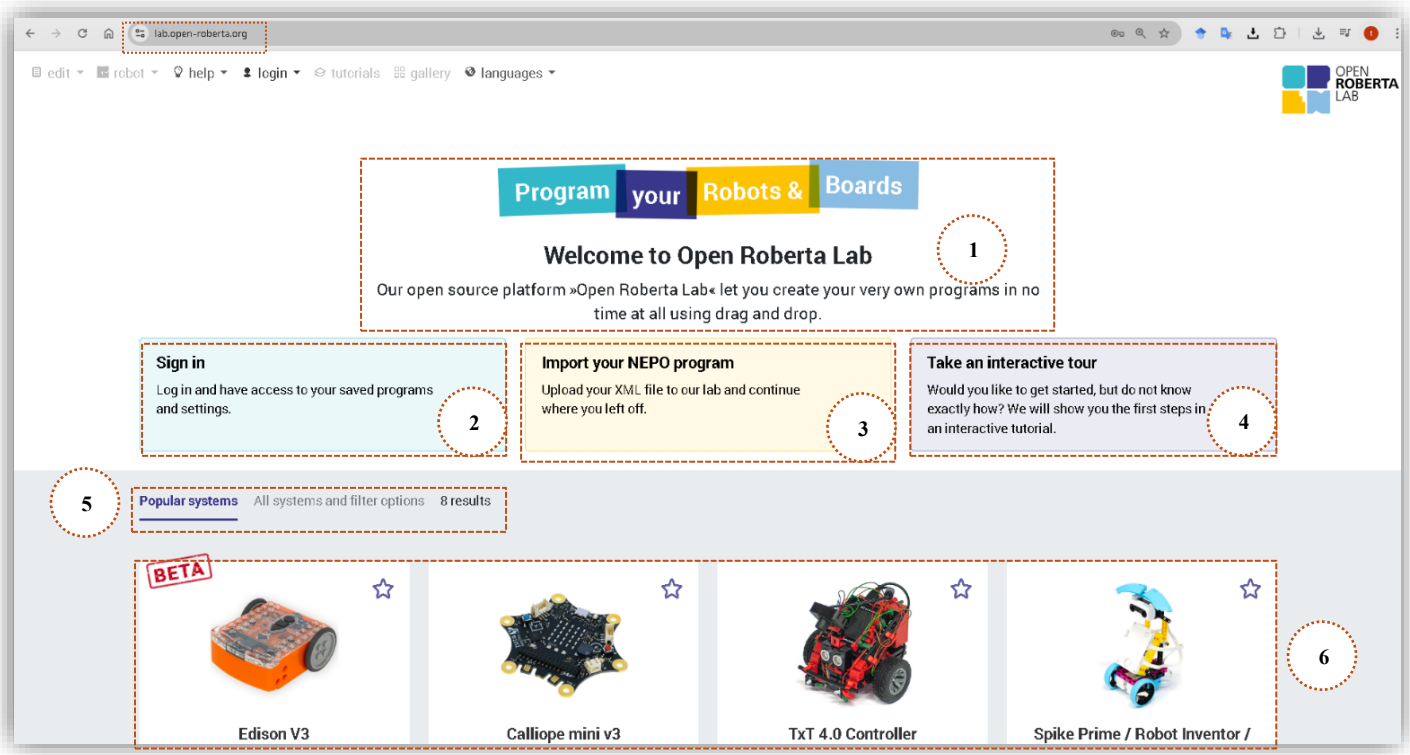
- العلوم: ربط مفاهيم الحركة والقوى ببرمجة الروبوت.
- الرياضيات: تعزيز فهم القياس، المسافات، الزوايا، والتكرار.
- اللغة: تنمية مهارات القراءة التحليلية من خلال متابعة التعليمات.
- التربية التكنولوجية: إعداد المتعلمين لعصر الرقمنة والذكاء الاصطناعي.
- المهارات الحياتية: تعزيز التعاون، العمل الجماعي، والتفكير الإبداعي.

لا تقتصر الأهداف التعليمية لهذا الدليل على تعليم المتعلمين كيفية تحريك روبوت افتراضي فحسب، بل تمتد لتشمل تنمية عقلية برمجية لدى كل من المعلم والمتعلمين، بما يربط بين المنهج الدراسي والمهارات المستقبلية في مجالي البرمجة والتكنولوجيا. وتُعد الروبوتات أداة تعليمية فعالة، فيما تتيح المحاكيات مثلOpen Roberta Lab فرصة تعلمها للجميع دون عوائق مادية أو تقنية، وهو ما يمنح المعلم ثقة أكبر في إدارة الدروس حتى وإن كان مبتدئًا في هذا المجال.

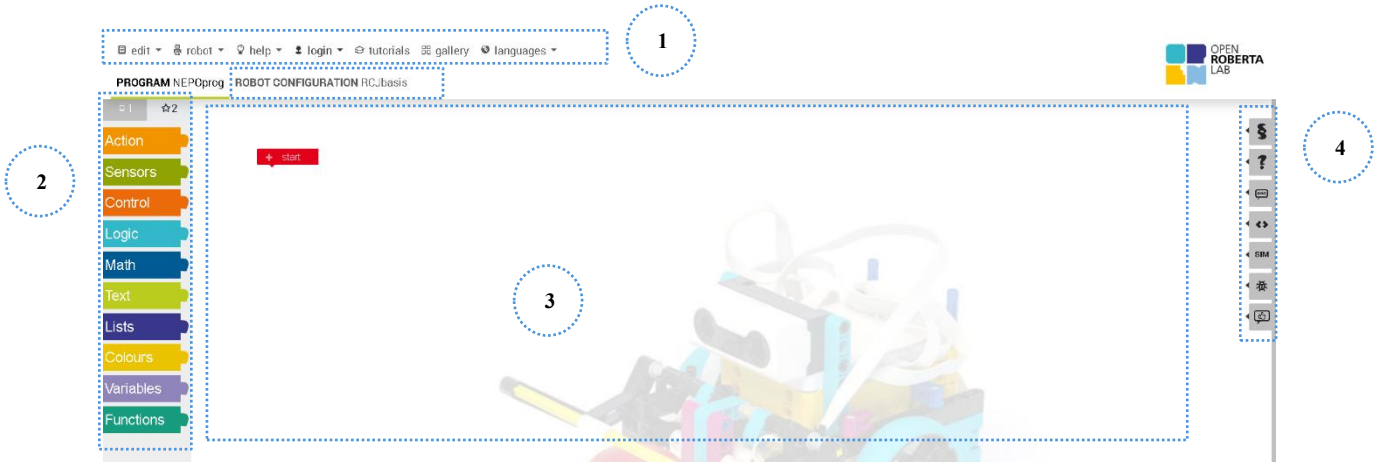
05

استخدام منصة ORL

عند الدخول إلى المنصة عبر الرابط https://lab.open-roberta.org تظهر شاشة البداية



القسم	الوصف	الوظيفة
1- القسم العلوي (الترحيب)	شعار المنصة مع عبارة Welcome to Open Roberta Lab ووصف موجز	تعريف بالمنصة كمصدر مفتوح لتعليم البرمجة بالسحب والإفلات دون الحاجة لكتابة كود نصي.
2- (تسجيل الدخول) Sign in	مربع أزرق على اليسار	يتيح للمعلم/المتعلم الدخول إلى حسابه والوصول للبرامج والإعدادات المحفوظة.
3- (استيراد برنامج) Import your NEPO program	مربع أصفر في المنتصف	رفع ملفات XML لمتابعة البرامج السابقة وتعديلها
4- (جولة تفاعلية) TaKe an interactive tour	مربع بنفسجي على اليمين	يقدّم شرحًا تفاعليًا مبسطًا للخطوات الأولى في استخدام المنصة
5- التبويبان (Systems)	Popular systems g All systems and filter options	تصفح الأنظمة المدعومة واختيار نوع الروبوت/المتحكم المناسب
6- قائمة الروبوتات الشائعة	صور وأسماء الروبوتات (Edison V3 – Calliope mini – TxT 4.0 – SpiKe Prime...)	اختيار النظام المراد برمجته وتجربة المحاكاة أو تحميل البرامج إليه



1. شريط القوائم العلوي (Top Menu Bar)

التحكم العام في المشروع، إعدادات الروبوت، الوصول إلى الدروس والمعرض، وتغيير اللغة Edit (تحرير) / Robot (إعدادات الروبوت/المحاكي) / Help (مساعدة) / Login (تسجيل الدخول) / Tutorials (دروس تفاعلية) / Gallery (معرض المشاريع) / Languages (اختيار اللغة)

2. منطقة الكتل البرمجية (Blocks Palette)

في Open Roberta Lab هناك مظهران لمنطقة الكتل التي يعمل فيها المستخدم على بناء البرنامج (1☐ و 2☆)

هي أزرار التبديل بين المظهرين:

الرمز	اسم الوضع	شكل العرض	الأنسب لـ
1☐	Categories View	عرض الفئات الرئيسية فقط	المبتدئين، البرامج المبسطة
2☆	Toolbox View	عرض الفئات + تقسيمات فرعية	المتقدمين، البرامج التفصيلية

تظهر فيها مجموعات الكتل، مصنفة بالألوان:

- Action **(الإجراءات)**: حركات وأفعال الروبوت.
- Sensors **(المستشعرات)**: قراءة بيانات الحساسات.
- Control **(التحكم)**: التكرار، الشروط، الانتظار.
- Logic **(المنطق)**: AND, OR, NOT، المقارنات.
- Math **(الرياضيات)**: العمليات الحسابية.
- Text **(النصوص)**: التعامل مع النصوص.
- Lists **(القوائم)**: معالجة البيانات على شكل قائمة.
- Colours **(الألوان)**: استخدام الألوان في مستشعر الألوان.
- Variables **(المتغيرات)**: تخزين القيم.
- Functions **(الدوال)**: إنشاء إجراءات خاصة.

3. منطقة العمل (Programming WorKspace)

- هي اللوحة البيضاء حيث توضع الكتل لبناء البرنامج.
- تبدأ دائمًا بكتلة Start الحمراء (نقطة البداية).
- يمكن للمستخدم سحب الكتل من اليسار وربطها في كتلة البداية الموجودة في هذه المنطقة لبناء الخوارزمية.

4. منطقة الاعدادات (Robot Configuration)

كل روبوت أو محاكي يحتاج إلى تكوين (Configuration) يحدد:

- عدد المحركات (Motors).
- نوع الحساسات (Sensors).
- أماكن التوصيل (Ports).

الزر	الاسم	الوظيفة
	Legal information	يفتح/يغلق صفحة المعلومات القانونية (Imprint) و(Privacy Policy).
	(Help)	يفتح/يغلق نافذة المساعدة العامة الخاصة بالمنصة
	Program documentation	يفتح / يغلق لوحة توثيق البرنامج على اليمين لكتابة وصف البرنامج وإضافة الوسوم.
	(Source Code View)	يفتح / يغلق عرض الشيفرة النصية المكافئة لكتل NEPO في لوحة يمين.
	(Simulation view)	يفتح / يغلق واجهة المحاكاة لتجربة البرنامج على البيئة الافتراضية
	(Simulation debug mode)	يفتح / يغلق المحاكاة بوضع التنقيح (Debug) لعرض تتبّع الحساسات والخطوات أثناء التشغيل.
	(FeedbacK)	يفتح نافذة إرسال ملاحظات للمنصة (Please rate the ORL)

القوائم الأساسية

Edit : إنشاء البرامج، حفظها، استيراد/تصدير، عرض المحاكاة أو الكود.

Robot: اختيار النظام/الروبوت، الاتصال بالجهاز الحقيقي من خلال كيبل USB أو البلوتوث إن وجد في النظام.

help : المساعدة، الأسئلة الشائعة، نبذة وملاحظات وسياسة الخصوصية والاختصارات.

login : تسجيل الدخول/الخروج، إدارة المجموعات، تغيير بيانات المستخدم.

tutorials : دروس تفاعلية للبدء خطوة بخطوة، لمهارات متعددة قد تختلف و تتغير من وقت إلى آخر.

gallery : معرض لمشاركة/استعراض برامج المجتمع.

languages : تغيير لغة واجهة المنصة.

الأمر	الوظيفة
(run on)	يبيّن النظام/المحاكي النشط حاليًا مثل(RCJ RescueOnlineSim) .
(open/close simulation view)	فتح/إغلاق لوحة المحاكاة على اليمين لتجربة البرنامج.
(new)	إنشاء برنامج جديد باسم افتراضي (NEPOprog).
(my programs)	فتح قائمة برامجك المحفوظة على الحساب (تكون غير مفعّلة إذا لم تسجّل الدخول).
(multiple robot simulation)	تشغيل محاكاة متعددة الروبوتات في بيئة واحدة.
(example programs)	أمثلة جاهزة للتعلّم السريع أو البدء منها.
(Save)	حفظ البرنامج الحالي (تكون غير مفعّلة إذا لم تسجّل الدخول أو لم يحدث تعديل).
(save as)	حفظ باسم جديد مفيد لتنظيم نسخ الدروس (تكون غير مفعّلة إذا لم تسجّل الدخول).
(open/close source code view)	عرض/إخفاء الشيفرة النصية المكافئة لكتل NEPO في لوحة جانبية.
(open source code editor)	فتح محرّر كود في مساحة مستقلة لقراءة/نسخ الكود (للمستخدمين المتقدمين)

قائمة (edit)

الامر	الوظيفة
(create program linK)	إنشاء رابط مشاركة يفتح البرنامج مباشرة لدى الآخرين
(export program)	تنزيل البرنامج الحالي كملف بامتداد (.xml)
(export all programs)	تنزيل جميع برامجك كأرشيف مضغوط (Zip) للاحتفاظ بنسخة احتياطية.
(import program)	استيراد ملف xml. سبق تصديره للعمل عليه
(NEPO-BlocKs beginner / expert)	التبديل بين مستوى الكتل (مبتدئ/خبير). في وضع expertتظهر كتل وإعدادات أكثر تقدمًا (قد يظهر الخيار باهتًا إذا كان غير متاح للسياق الحالي)

قائمة (robot)

الأمر	الوظيفة
(connect)	الاتصال بجهاز روبوت حقيقي (Bluetooth / USB) عندما يدعم النظام ذلك.
(info)	معلومات عن تهيئة الروبوت النشط (المحرّكات، المستشعرات، المنافذ).
(backK to robot/system selection)	الرجوع لصفحة اختيار النظام/الروبوت لتبديله.

قائمة (help)

الأمر	الوظيفة
general help	دليل عام لاستخدام الواجهة.
FAQ	الأسئلة الشائعة وحلول المشكلات المتكررة.
about the Open Roberta Lab	معلومات عن المختبر نفسه.
about the Open Roberta Project	نبذة عن المشروع والمؤسسة المطوِّرة.
backK to the start page	العودة للصفحة الرئيسية الترحيبية.
logging	سجلّ أحداث مفيد عند التشخيص/الإبلاغ عن أخطاء.
privacy policy	سياسة الخصوصية.
publishing notes	معلومات النشر والإصدارات.
Keyboard shortcuts	قائمة اختصارات لوحة المفاتيح.

قائمة login:

الأمر	الوظيفة
Login	تسجيل الدخول بحسابك الفردي.
Log in with user group	تسجيل الدخول باستخدام مجموعة مستخدمين/صف (كود جماعي).
Logout	تسجيل الخروج.
Manage user groups	إدارة مجموعات المتعلمين (صلاحية المعلم).
Change	تعديل بيانات حسابك (مثل كلمة المرور).
delete user	حذف الحساب.
state information	معلومات حالة الخادم (تشخيص).

ملاحظة:

- يفضل أن يكون اسم المستخدم لكل متعلم هو اسمه ورقم المستخدم 123456 حتى يتمكن المعلم من تذكير المتعلم بالبيانات اللازمة في حال نسيانه (اختياري).
- لا يمكن إنشاء مجموعة أو استعادة كلمة مرور إلا للحسابات التي تم تسجيل بريد الكتروني لها أثناء انشائها.

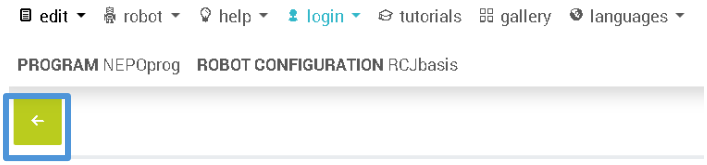
قائمة Tutorials: فتح دروسًا تفاعلية تقودك خطوة بخطوة داخل الواجهة (مفيد جدًا للمبتدئين).

قائمة Gallery: استعراض معرض البرامج المنشورة من المجتمع، وإمكانية مشاركة برنامجك.

قائمة Languages: تغيير لغة واجهة الاستخدام.

ملاحظة:

- لا تتوافر العربية كواجهة حتى الآن، لكن يمكن كتابة نصوص عربية داخل بعض الكتل/العرض على شاشة الروبوت.
- عند دخولك لشاشة إحدى القوائم في حال أردت العودة للصفحة السابقة اضغط على السهم الأخضر الظاهر لك جهة اليسار أعلى الشاشة.



مفهوم البرمجة بالكتل

البرمجة في منصة Open Roberta Lab تعتمد على الكتل المرئية (BlocKs) بدلاً من كتابة الأكواد النصية.

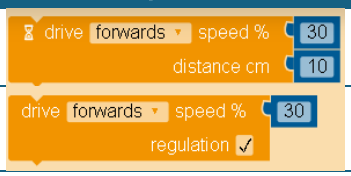
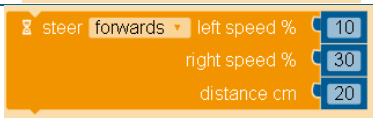
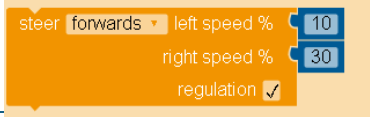
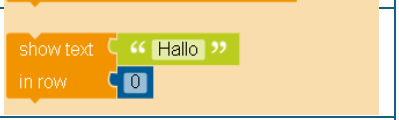
- يقوم المستخدم بسحب الكتل من قائمة الفئات وتجميعها في مساحة العمل من خلال ربطها بكتلة start.
- كل كتلة تمثل أمراً برمجياً محدداً مثل الحركة، الاستدارة، التكرار .... الخ.
- عندما يتم تركيب الكتل معاً، يشكل البرنامج سلسلة من التعليمات ينفذها الروبوت خطوة بخطوة (الخوارزمية).

هذه الطريقة تجعل البرمجة أكثر سهولة للمتعلمين الجدد، حيث تركز على الفكرة والمنطق بدلاً من القواعد اللغوية المعقدة.

الفئات الأساسية للكتل في Open Roberta Lab

Action BlocKs

تحتوي على كتل التحكم بحركة الروبوت (إلى الأمام، إلى الخلف، الدوران، التوقف)

الأمر	الوظيفة
	يجعل الروبوت يتحرك للأمام بسرعة 30% لمسافة 10 سم (يمكن تغيير السرعة والمسافة).
	تحريك الروبوت للأمام بسرعة ثابتة (مع تنظيم السرعة)
	إيقاف الروبوت عند النقطة الحالية.
	دوران الروبوت 20 درجة نحو اليمين بسرعة 30% (يمكن تغيير سرعة وزاوية الدوران)
	دوران مستمر لليمين بسرعة 30% (بدون تحديد زاوية)، (يمكن تغيير السرعة).
	قيادة للأمام مع اختلاف سرعة العجلات (اليسار 10%، اليمين 30%) لمسافة 20 سم، وهذا يجعل الروبوت يسير في منحنى (الانعطاف).
	قيادة للأمام مع اختلاف سرعة العجلات (اليسار 10%، اليمين 30%) بدون تحديد المسافة، وهذا يجعل الروبوت يسير في منحنى (الانعطاف).
	عرض كلمة "Hallo" على شاشة الروبوت(show text) في الصف الأول (in row) (يمكن تغيير النص لعرض رسائل مخصصة ويمكن تغيير رقم الصف الذي ستظهر فيه الرسالة).
	مسح الشاشة وإزالة أي نص أو رسومات معروضة.
	تشغيل نغمة موسيقية قصيرة عند تنفيذ البرنامج.

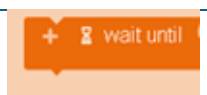
Sensors BlocKs

تُستخدم هذه الكتل للحصول على بيانات من البيئة المحيطة بالروبوت، ثم توظيفها في اتخاذ القرارات داخل البرنامج.

الأمر	الوظيفة
	يتحقق ما إذا كان الزر الأيسر مضغوطًا أم لا (يعيد النتيجة بشكل منطقي صح أو خطأ). مثال: برمجة الروبوت بحيث يبدأ بالتحرك فقط عند الضغط على الزر الأيسر.
	يقرأ اللون الذي يراه مستشعر اللون. مثال: إذا كان اللون أصفر توقف عن الحركة
	يقيس المسافة بين الروبوت وأي جسم أمامه بالسنتيمتر. مثال: إذا كانت المسافة أقل من 15 سم استدر جهة اليمين بمقدار 90 درجة.
	يعطي الوقت الذي مرّ منذ تشغيل المؤقت بالميلي ثانية (كل 1000 ms يساوي 1 ثانية). مثال: جعل الروبوت يسير للأمام لمدة 5 ثوانٍ باستخدام المؤقت بدلًا من المسافة.
	يعيد ضبط المؤقت إلى الصفر لبدء حساب جديد.
	يحدد زاوية دوران الروبوت حول محوره المحور(Z). مثال: برمجة الروبوت ليتوقف بعد أن يدور 90 درجة.

Control BlocKs

تستخدم هذه الكتل للتحكم في تكرار الأوامر أو تنفيذها عند تحقق شروط معينة، مما يمنح البرامج مرونة وذكاء أكبر.

الأمر	الوظيفة
	(إذا – افعل): ينفذ الأوامر داخل الكتلة إذا تحقق الشرط فقط.
	(إذا – افعل – وإلا): ينفذ أوامر معينة إذا تحقق الشرط، وأوامر أخرى إذا لم يتحقق.
	(كرر للأبد): يعيد تنفيذ الأوامر إلى ما لا نهاية، حتى يوقف البرنامج يدويًا
	(كرر 10 مرات): يعيد تنفيذ الأوامر عددًا معينًا من المرات (يمكن تغيير الرقم).
	(انتظر 500 ملي ثانية): يوقف تنفيذ البرنامج مؤقتًا لمدة نصف ثانية
	(انتظر حتى): يتوقف البرنامج حتى يتحقق شرط معين

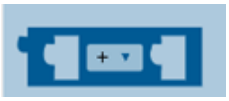
Logic BlocKs

تستخدم هذه الكتل لمقارنة القيم أو دمج أكثر من شرط معًا داخل البرنامج.

الأمر	الوظيفة
	كتلة المساواة (=): تستخدم لمقارنة قيمتين إذا كانت القيمتان متساويتين فالنتيجة True إذا لم تكونا متساويتين فالنتيجة False.
	كتلة المنطق (AND – OR): AND (و): يتحقق الشرط فقط إذا كان الشرطان معًا صحيحين. OR (أو): يتحقق الشرط إذا كان أحد الشرطين صحيحًا.
	القيم المنطقية (Boolean Values): true : تعني أن الشرط صحيح دائمًا. false: تعني أن الشرط خاطئ.

Math BlocKs

تُستخدم لإجراء العمليات الحسابية، وهي مفيدة خصوصًا عند التعامل مع المتغيرات أو المستشعرات.

الأمر	الوظيفة
	القيمة العددية (Number): تمثل رقمًا ثابتًا (يمكن تغييره إلى أي قيمة أخرى)
	العمليات الحسابية (Operators): هذه الكتلة لها مدخلان (يمين ويسار)، وتقوم بإجراء العملية الحسابية (جمع – طرح – ضرب – قسمة) عليهما.

Text BlocKs

تُستخدم لإنشاء نصوص أو التعامل معها (عرضها).

الأمر	الوظيفة
	النصوص: (Text) يمكن كتابة أي نص داخلها تُستخدم غالبًا مع أوامر العرض على الشاشة أو مع كتل المقارنة (Logic).
	دمج النصوص: (Join Texts) كتلة لدمج نصين معًا في نص واحد يمكن استخدامها مع متغيرات لعرض قيم مخصصة.

Colours BlocKs

تُستخدم هذه الكتل لتمثيل الألوان المختلفة داخل البرنامج، سواءً للتعرف عليها عبر المستشعر أو لعرضها على شاشة الروبوت أو في الرسومات.

Variables BlocKs

إنشاء المتغير في كتلة Start هو الخطوة الأولى قبل استخدامه ، إذا لم يُعلن المتغير، لن يتعرف عليه البرنامج.

ما هي المتغيرات؟

المتغير هو "علبة تخزين" يمكن وضع قيمة بداخلها (مثل رقم أو نص) واستخدامها لاحقًا في البرنامج، يمكن تغيير قيمة المتغير أثناء تشغيل البرنامج.

مثال:

تخيل أن لديك علبة مكتوب عليها "السرعة" في البداية وضعت داخلها الرقم 30، لاحقًا تستطيع أن تفتح العلبة وتغيّر الرقم إلى 50، والروبوت سيستخدم القيمة الجديدة.

كيفية إنشاء متغير في Open Roberta Lab

عند الحاجة إلى متغير، يجب أولًا إنشاؤه من خلال :

1. الضغط على رمز + الموجود بجانب كتلة Start.
2. يتم إنشاء متغير باسم item من النوع Number (عدد)، وتمت تهيئته بالقيمة 0.

ملاحظة: يمكن تغيير اسم ونوع المتغير والقيمة المبدئية له حسب الحاجة.

أنواع المتغيرات

- Number (رقم): لتخزين قيم عددية موجبة أو سالبة مثل 10 أو 25.-
- Boolean (قيمة منطقية): لتخزين (True / False)
- String (نص): لتخزين نصوص مثل "مرحباً."
- List (قائمة): لتخزين مجموعة من القيم.

كتل المتغيرات

الأمر	الوظيفة
	تعيين قيمة للمتغير: تُستخدم لإعطاء المتغير قيمة جديدة.
	استدعاء المتغير: تُستخدم للحصول على القيمة المخزنة في المتغير (استرجاع القيمة الموجودة حاليا داخل المتغير).

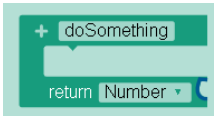
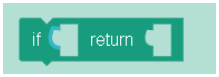
Functions BlocKs

الدوال هذه الكتل متقدمة قليلًا، وتُستخدم لجعل البرنامج منظمًا وقابلًا لإعادة الاستخدام.

ما هي الدالة؟

الدالة هي جزء من البرنامج يُنفّذ مهمة محددة ويمكن استدعاؤه عدة مرات بدلًا من تكرار نفس الأوامر، تساعد على جعل البرامج أكثر تنظيمًا وسهولة في الفهم والتعديل عند الحاجة.

الكتل الأساسية للدوال

الأمر	الوظيفة
	(Function without return): دالة تنفذ أوامر معينة ولا تعيد قيمة. مثال: دالة تجعل الروبوت يصدر صوتًا ويعرض كلمة"Hello" .
	(Function with return) : دالة تنفذ أوامر ثم تعيد قيمة (عدد، نص، منطق). مثال: دالة تحسب المسافة وتقارنها، ثم تعيد رقمًا
	(If return): تُستخدم داخل دالة لتحديد أي قيمة ستُعاد بناءً على شرط. مثال: إذا كانت المسافة <20 أرجع "True"وإلا أرجع"False" .

الدوال تساعد المتعلمين على التفكير المنطقي وتقسيم المشكلة الكبيرة إلى مهام صغيرة، تجعل برامج الروبوت أكثر تنظيمًا وكفاءة، وتعتبر جسرًا نحو مفاهيم البرمجة المتقدمة.

استخدام المحاكي في Open Roberta Lab

تشغيل الروبوت الافتراضي

عند الانتهاء من بناء البرنامج، يمكن الضغط على زر SIM من القائمة الجانبية لتشغيل المحاكي الافتراضي.

- يظهر روبوت على الشاشة (يشبه الروبوت الحقيقي) يمكنه تنفيذ الأوامر البرمجية.
- هذه الخطوة تتيح للمستخدمين رؤية نتيجة الكود مباشرة دون الحاجة إلى وجود روبوت حقيقي في المختبر.

التحكم في البيئة

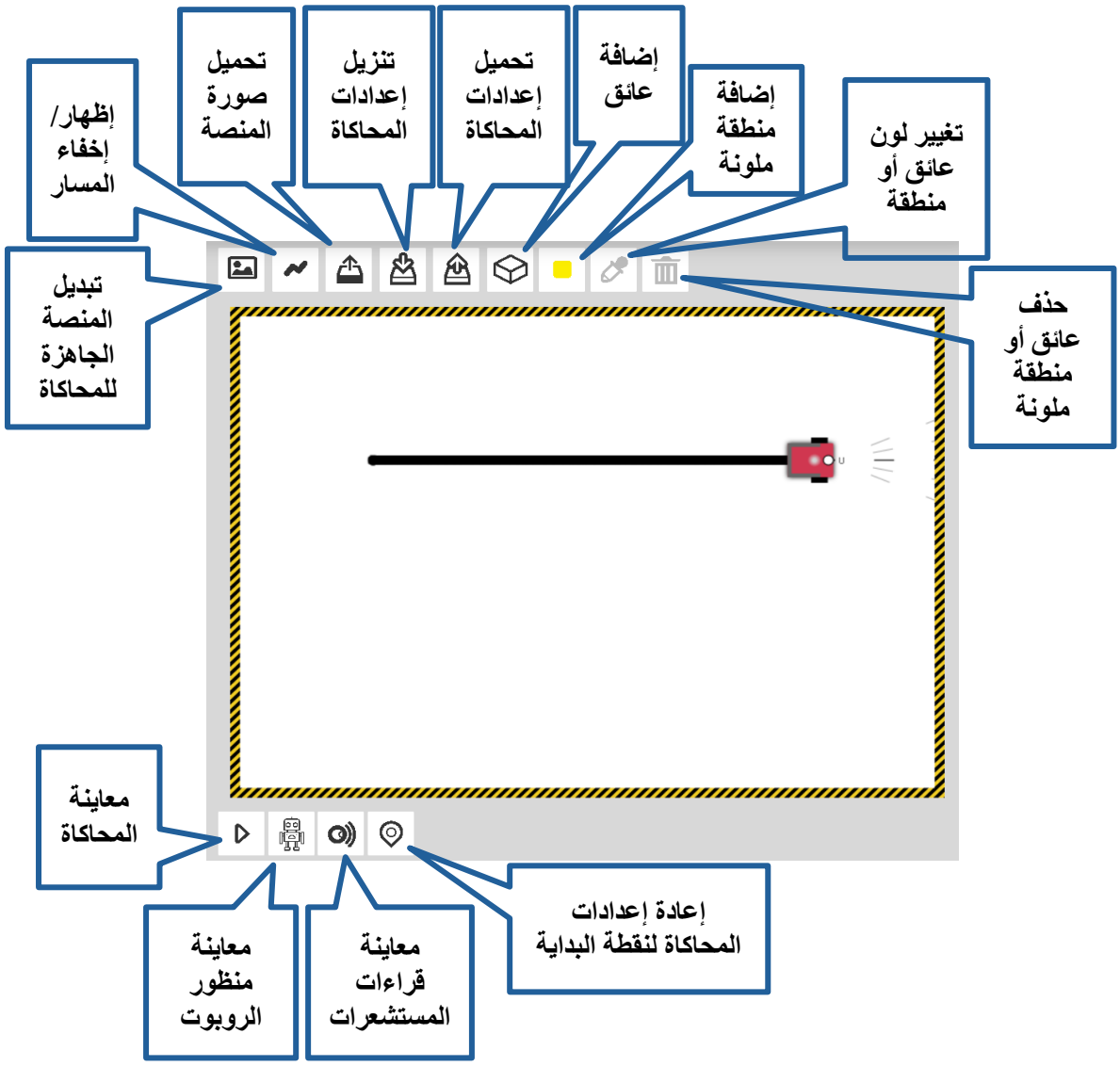
المحاكي يتيح خيارات متنوعة لتغيير بيئة الروبوت وجعل التجارب أكثر واقعية:

1. المنصة: (Arena) يمكن اختيار خلفيات مختلفة (مسار مستقيم – ساحة دائرية – حلبة سومو – شبكة مربعات).
2. إضافة عوائق (Obstacles): يمكن وضع أجسام أمام الروبوت لاختبار عمل مستشعر المسافة.
3. إضافة منطقة لونية (Color Area): يمكن إضافتها داخل الحلبة الافتراضية في المحاكي، للتفاعل مع الألوان باستخدام مستشعر اللون.
4. المسار (Path): يمكن رسم مسارات على الشاشة (خط أسود) لتدريب الروبوت على تتبعها.

إعادة تعيين الروبوت ومراقبة النتائج

- زر إعادة التعيين (Reset) يعيد الروبوت إلى موقعه الأصلي في المحاكي.
- يمكن تشغيل البرنامج أكثر من مرة بعد إجراء تعديلات على الكود لمقارنة النتائج.
- من المهم تدريب المتعلمين على المحاولة والخطأ ومراجعة سبب نجاح أو فشل الحل.

أدوات المحاكاة(Simulation toolbar)



06

دروس الصف السادس

عنوان الوحدة	برمجة الروبوت في العالم الافتراضي		
الدروس	العالم الافتراضي للروبوت ORL	أسرار الحركة الذكية	من الحركة إلى التعبير
عدد الحصص	2	2	2
المشروع	مشروع روبوت 1 ص 150	مشروع روبوت 2 ص 162	مشروع روبوت 3 ص 174

فكرة الوحدة

تركز هذه الوحدة على تنمية مهارات المتعلمين في برمجة الروبوتات والتحكم بها باستخدام بيئة المحاكاة Open Roberta Lab ينتقل المتعلمون من الأساسيات إلى المهارات المتقدمة بشكل تدريجي، بدءًا من تحريك الروبوت في مسارات بسيطة، مرورًا بتصميم حركة ذكية ومتكررة، وصولًا إلى برمجة الروبوت للتفاعل مع الأحداث عبر عرض النصوص وإصدار الأصوات.

في نهاية هذه الوحدة سيكون المتعلم قادرًا على:

1. التعرف على مكونات روبوت LEGO SPIKE Prime وواجهة منصة ORL .
2. إنشاء برامج لتحريك الروبوت في مسارات مستقيمة وأشكال هندسية بسيطة.
3. استخدام كتلة Steer والتحكم في الحركة المنحنية والدائرية.
4. توظيف كتل التكرار لتبسيط البرمجة وإنشاء مسارات أكثر دقة.
5. برمجة الروبوت لعرض النصوص على الشاشة وإصدار الأصوات في أوقات محددة.
6. الربط بين المدخلات (الأحداث) والمخرجات لزيادة تفاعل الروبوت.
7. حفظ واستدعاء المشاريع من منصة البرمجة.
8. إنشاء حساب خاص بالمتعلم على المنصة.

ملخص محتوى الدروس

الدرس الأول: الحركة البسيطة

يتعلم المتعلمون أساسيات المحاكاة وبرمجة الروبوت للتحرك للأمام والخلف، ورسم مسارات هندسية بسيطة، مع التعرف على طريقة حفظ واسترجاع البرامج.

الدرس الثاني: الحركة الذكية

يكتسب المتعلمون مهارات التحكم المتقدم في مسار الروبوت باستخدام كتلة Steer، وإنشاء حركات دائرية وبيضاوية، مع إدخال مفهوم التكرار في البرمجة.

الدرس الثالث: من الحركة إلى التعبير؟

يبرمج المتعلمون الروبوت ليعرض نصوصًا على شاشته ويصدر أصواتًا استجابة لأحداث معينة، مما يضيف بعدًا تفاعليًا إلى مشاريعهم.

استراتيجية التعلم

- ✓ التعلم القائم على المشروع (Project-Based Learning).
- ✓ التعلم بالاكتشاف والتجريب في بيئة آمنة.
- ✓ التدرج من السهل إلى الصعب في تقديم المفاهيم البرمجية.
- ✓ الدمج بين الشرح النظري والتطبيق العملي على المحاكي.

قيم ومهارات مدمجة

- ✓ التفكير الحاسوبي وحل المشكلات.
- ✓ الإبداع في تصميم المشاريع البرمجية.
- ✓ الدقة والانتباه للتفاصيل.
- ✓ تنظيم العمل وتقليل الأخطاء عبر كتابة أكواد منظمة.

التقويم المستمر

تم إعداد مقياس Rubric لتقويم أداء متعلمي الصف السادس في دروس الروبوت، بما يتناسب مع محتوى الدروس المقررة التي تركز على الحركة البسيطة، والحركة الذكية باستخدام التكرار، وانتقال الروبوت من الحركة إلى التعبير عبر النصوص والأصوات. لمساعدة المعلمين في متابعة مستوى إتقان المتعلمين للمهارات البرمجية والعملية بدقة، مع تعزيز التفكير المنطقي والإبداعي لديهم.

الدرس الأول

العالم الافتراضي للروبوت باستخدام منصة ORL

المقدمة

في هذا الدرس سيتعرف المتعلم على كيفية برمجة الروبوت للتحرك في خط مستقيم أو تنفيذ دوران بسيطة باستخدام المحاكي في منصة Open Roberta Lab الدرس يمثل المدخل لفهم أوامر الحركة، حيث سيتعلم المتعلم الفرق بين الحركة المستقيمة والدوران، وكيفية ضبط المسافة أو المدة الزمنية، وكذلك كيفية استخدام وحدة القياس الصحيحة.

خطة الدرس

الحصة	الموضوع	النشاط
الأولى	التعرف على أوامر الحركة	تجربة أوامر التحرك للأمام والخلف وتغيير المسافة
الثانية	الدوران والتحكم بالزوايا	تنفيذ دورانات بزوايا محددة لرسم أشكال هندسية بسيطة
المشروع		مشروع روبوت 1 صـ 150 ـمن الكتاب المدرسي رحلة إلى معالم دولة الكويت
QR Code لفيديوهات شرح وأنشطة تعلمية		ملفات أوراق العمل ملفات مصادر التعلم

نتاجات التعلم

1. شرح مفهوم الواقع الافتراضي، وتعداد مزاياه في التعلم والمحاكاة.
2. التعرف على بيئة برمجة روبوت LEGO SPIKE ومحاكاته الروبوتية الأساسية.
3. استكشاف واجهة ORL محاكي الروبوت، والتعرف على أدوات التحكم البرمجي.
4. إنشاء حركة بسيطة للروبوت باستخدام الكتل البرمجية داخل ORL.
5. معاينة محاكاة حركة الروبوت داخل البيئة البرمجية ORL وملاحظة الاستجابة.
6. حفظ المشروع البرمجي، وإعادة استدعائه لاحقاً بطريقة منظمة.

المصطلحات التقنية

- Move BlocK: كتلة تحريك الروبوت للأمام أو الخلف.
- Turn BlocK: كتلة دوران الروبوت يمينًا أو يسارًا.
- cm: سنتيمتر – وحدة قياس المسافة.
- degrees: درجة – وحدة قياس الزوايا.
- Simulation Mode: وضع المحاكاة لتجربة البرنامج.

ملاحظات للمعلم

- ✓ تأكد من اختيار الروبوت الصحيح وفي حال اختيار المتعلم لروبوت مختلف يتم تغيير الروبوت من قائمة (Robot) العودة لاختيار الروبوت/النظام في المنصة.
- ✓ إذا لم يظهر المحاكي أو تعطل، أعد تحميل الصفحة أو جرّب متصفح Chrome.
- ✓ عند تنفيذ الدوران، وضح للمتعلمين أن القيم العددية الموجبة والقيم العددية السالبة قد تؤثر على اتجاه الدوران.
- ✓ إذا حدث خطأ عند تشغيل الكتل، جرّب استبدال كتلة أو إعادة ترتيبها، ثم شغل البرنامج من جديد.
- ✓ يمكن تحريك الروبوت للحصول على الاتجاه المطلوب عند البداية من خلال أسهم لوحة المفاتيح.
- ✓ يمكن تحريك الروبوت من موقعه الحالي الى موقع جديد من خلال السحب والافلات باستخدام الفأرة.

الأنشطة النظرية والعملية

ملاحظة: تم اقتراح أنشطة إضافية في الدليل لإتاحة خيارات متعددة للمعلم

نشاط - نظري

1. شرح كيف يمكن حساب زاوية دوران الروبوت لرسم مربع أو مثلث.
2. ناقش مع المتعلمين العلاقة بين المسافة والمدة في أوامر الحركة.

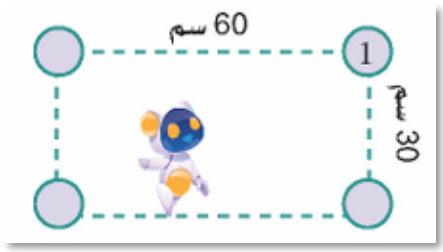
نشاط – عملي

1. برمج الروبوت للتحرك للأمام 50 سم، ثم الدوران 90° ، وكرر ذلك 4 مرات لرسم مربع.
2. حاول تغيير الزاوية لرسم مثلث أو شكل آخر.
3. يمكنك تغيير اتجاه الحركة مرة للأمام ومرة للخلف.

أنشطة الكتاب

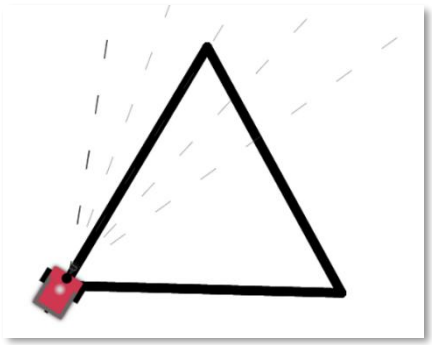
نشاط صـ 147

- صل بين النقاط لاستكمال الشكل.
- استنتج الكتل اللازمة ليرسم الروبوت مستطيل أثناء الحركة.



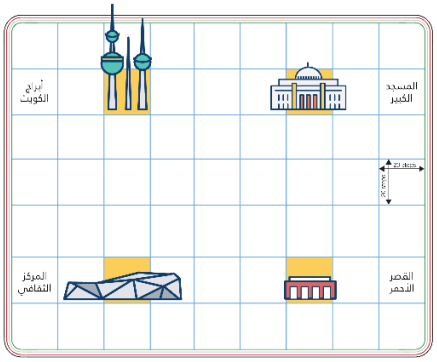
نشاط صـ 148

هل يمكنكم الان استنتاج الكتل اللازمة لجعل الروبوت يرسم مثلث اذا علمت أن طول الضلع 100 سم وزاوية الدوران 120.

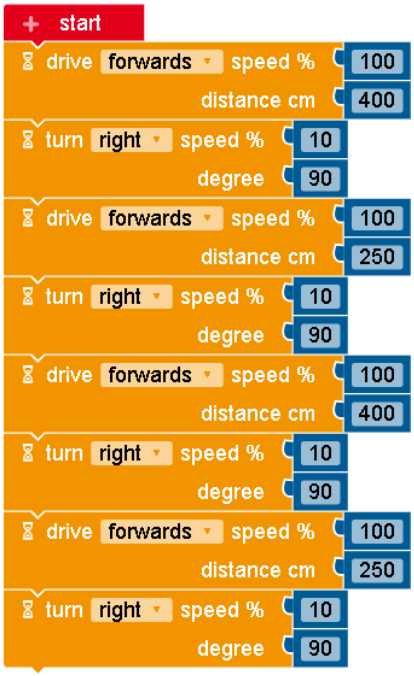


مشروع روبوت 1صـ 150 ـ من الكتاب المدرسي

رحلة إلى معالم دولة الكويت



هذه الأوامر مبرمجة بحيث يتحرك الروبوت في مسار مستطيل ويكرر ذلك مرتين تقريباً، كالآتي:



1. التحرك للأمام بسرعة 100% لمسافة 400 سم.
2. الدوران لليمين ببطء (سرعة 10%) بزاوية 90 درجة.
3. التحرك للأمام بسرعة 100% لمسافة 250 سم.
4. الدوران لليمين بزاوية 90°.

النتيجة:

- الروبوت يرسم مستطيلاً أبعاده 400 سم × 250 سم.
- بما أن الأوامر مكررة مرتين، فهو يلف حول نفس المستطيل مرتين متتاليتين.

التحدي:

يمكنك جعل الروبوت يبدأ الحركة عند معلم مختلف من خلال تحريكه إلى موقع معلم مختلف في كل مرة قبل بدء تشغيل البرنامج ويستكمل طريقة إما للأمام أو للخلف.

تقويم مقترح :

1. ما هي الكتلة البرمجية المسؤولة عن دوران الروبوت؟
2. إذا أردت أن يتحرك الروبوت للخلف 30 سم، أي إعداد ستغير في كتلة الحركة؟
3. ما الفرق بين استخدام زاوية 90° وزاوية 45° في الدوران؟

تقويم مستمر مقترح لنتاجات التعلم وفق مقياس Rubric

المعيار	ممتاز(4)	جيد(3)	مقبول(2)	يحتاج لتحسين(1)
صحة الكود	الكود سليم تمامًا بلا أخطاء.	أخطاء طفيفة لا تؤثر على التنفيذ.	بعض الأخطاء أثرت جزئيًا على التنفيذ.	الكود غير صالح للتنفيذ.
استخدام كتل الحركة(Drive/Turn)	استخدم الكتل بشكل صحيح وفعال.	استخدم الكتل بشكل صحيح مع بعض التكرار غير الضروري.	استخدم الكتل بشكل محدود أو غير مرتب.	لم يستخدم الكتل بشكل صحيح.
نجاح المهمة (خط مستقيم/مستطيل /رحلة إلى معالم دولة الكويت)	الروبوت أنجز المطلوب بدقة كاملة.	الروبوت أنجز المطلوب مع بعض الانحرافات.	الروبوت أنجز جزءًا من المطلوب.	الروبوت لم ينجز المطلوب.
العرض والتفسير	شرح واضح للخطوات والكتل.	شرح مقبول مع بعض النواقص.	شرح جزئي.	لم يشرح أو كان الشرح غير واضح.

الدرس الثاني

أسرار الحركة الذكية التكرار والمسارات المنحنية

المقدمة

في هذا الدرس سيتعرف المتعلم على كيفية التحكم بمسار الروبوت باستخدام كتلة Steer لتنفيذ حركات منحنية أو دائرية، ودمجها مع لبنات التكرار لتنفيذ حركات متكررة بعدد مرات محدد أو بشكل مستمر.

الدرس يركز على كيفية ضبط قيم سرعة العجلات اليمنى واليسرى للحصول على مسارات مختلفة، بالإضافة إلى استخدام التكرار لتسهيل تنفيذ المهام البرمجية دون الحاجة لإعادة كتابة الأوامر.

خطة الدرس

الحصة	الموضوع	النشاط
الأولى	التعرف على كتلة Steer	تجربة قيم مختلفة للعجلات للحصول على مسارات منحنية ودائرية
الثانية	التكرار مع المسارات	دمج لبنة التكرار مع أوامر Steer لرسم مسارات أو أشكال هندسية
المشروع		مشروع روبوت 2 صـ 162 ـمن الكتاب المدرسي الوصول إلى الموقف الآمن
QR Code لفيديوهات شرح وأنشطة تعلمية		

نتاجات التعلم

1. إنشاء مشروع جديد باستخدام بيئة البرمجة ORL للتحكم في الروبوت.
2. شرح مفهوم التوجيهSteering ، وبيان تأثيره على حركة ومسار الروبوت.
3. دمج كتلة التكرار Repeat ضمن المقطع البرمجي لتحقيق حركة متكررة.
4. التحكم في مسار الروبوت عبر تعديل قيم التوجيه والزمن والتكرار.

المصطلحات التقنية

- Steer BlocK: كتلة لتوجيه الروبوت عبر ضبط فرق السرعة بين المحركين.
- Repeat BlocK: كتلة لتكرار الأوامر عددًا محددًا من المرات أو بشكل مستمر.
- Distance (cm): معامل يحدد طول المسار الذي سيقطعه الروبوت.
- Speed (%): سرعة المحرك كنسبة مئوية.
- Simulation Mode: وضع المحاكاة لتجربة البرنامج.

ملاحظات للمعلم

- ذكّر المتعلمين بضرورة تسجيل الدخول قبل البدء لضمان حفظ المشروع.
- لتوضيح فكرة الانعطاف، اربطها بتجربة قيادة السيارة عند الدوار وتغيير اتجاه المقود.
- إذا لم يتحرك الروبوت كما هو متوقع، اطلب من المتعلمين التحقق من اتجاه الحركة وقيم السرعات.
- يمكن تغيير قيم المسافة والسرعة للحصول على مسار أدق.
- قبل تشغيل البرنامج، أعد ضبط الروبوت من أداة الإعدادات لضمان انطلاقه من نقطة البداية.

الأنشطة النظرية والعملية

نشاط – نظري

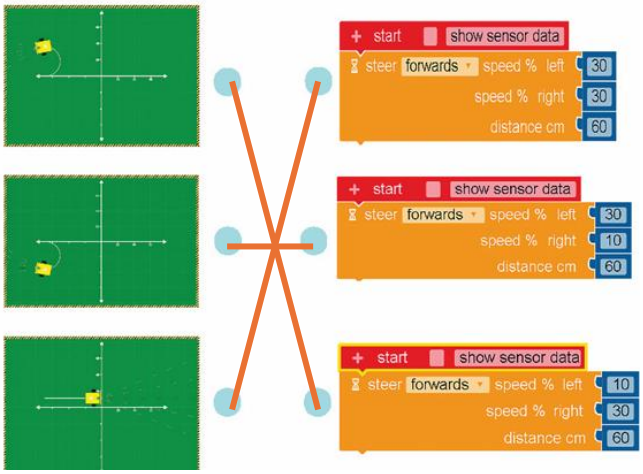
1. وضّح الفرق بين كتلة Drive ، Steer و Turn من حيث وضع المحركات واتجاه الروبوت.
2. ناقش مع المتعلمين كيف يؤثر الفرق في سرعة العجلات على شكل المسار.

نشاط – عملي

1. برمج الروبوت لرسم دائرة باستخدام قيم سرعة متقاربة بين المحركين.
2. غيّر القيم للحصول على مسار بيضاوي أو منحنى أضيق.

أنشطة الكتاب

نشاط صـ 156ـ



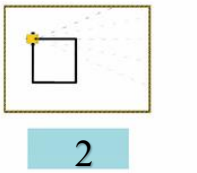
نفذ المقاطع البرمجية التالية، ثم صل كل صورة بالمقطع البرمجي المناسب لها.

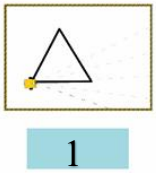
نشاط صـ 159ـ

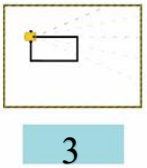
الكتلة البرمجية	Drive BlocK	Steer BlocK	Turn BlocK
وضع المحرك	المحركات تعمل بنفس السرعة	يعمل كل محرك بسرعة مختلفة لكن مرتبطتين معًا	يعمل محرك واحد فقط والأخر ثابت يدور حول محوره دون تقدم
الاتجاه	للأمام وللخلف بخط مستقيم	مسار منحني	دوران
رسم المسار			

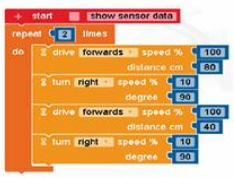
نشاط الكتاب صـ 161ـ

استنتج شكل مسار الروبوت عند تنفيذ المقاطع البرمجية التالية:

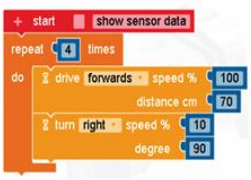

2


1

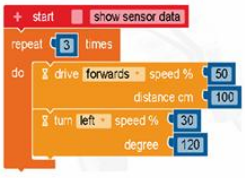

3



3



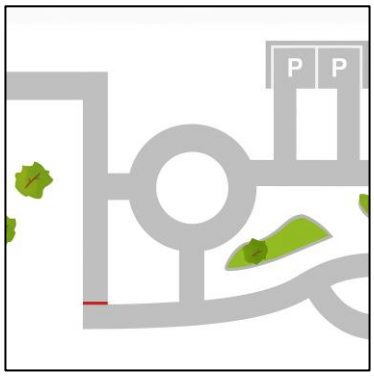
2



1

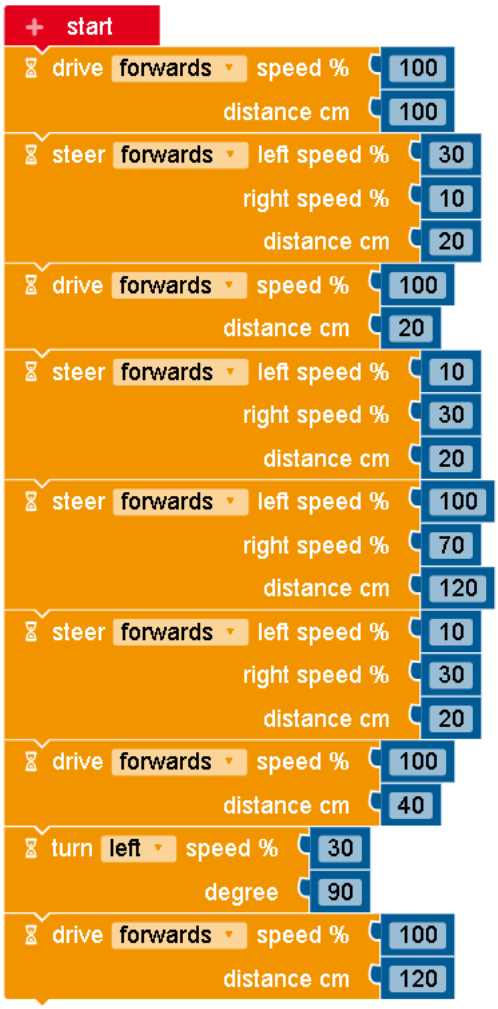
مشروع روبوت 2 صـ 162 من الكتاب المدرسي

الوصول إلى الموقف الآمن



هذه الأوامر تجعل الروبوت يتحرك في مسار معقد يتضمن مقاطع مستقيمة وانحناءات، بالتفصيل:

1. التحرك للأمام بسرعة 100% لمسافة 100سم.
2. الانعطاف بانحناء خفيف لليمين (العجلة اليسرى 30%، اليمنى 10%) لمسافة 20 سم.
3. التحرك للأمام بسرعة 100% لمسافة 60 سم.
4. الانعطاف بانحناء خفيف لليسار (العجلة اليسرى 10%، اليمنى 30%) لمسافة 20 سم.
5. الانعطاف بانحناء لليسار أكبر (العجلة اليسرى 100%، اليمنى 70%) لمسافة 60 سم.
6. الانعطاف بانحناء خفيف لليمين (العجلة اليسرى 30%، اليمنى 10%) لمسافة 20 سم.
7. التحرك للأمام بسرعة 100% لمسافة 40 سم.
8. الدوران لليسار 90 درجة بسرعة 30%.
9. التحرك للأمام لمسافة 120 سم.



النتيجة:

الروبوت يسير بخطوط مستقيمة يتخللها انحناءات يمين ويسار بدرجات مختلفة، ثم يقوم بدوران كامل لليسار وينهي المسار بخط مستقيم طويل.

التحدي:

جعل الروبوت يذهب للموقف الثاني دون لمس الحواف.

تقويم مقترح

1. ما الفرق بين كتلة Steer وكتلة Turn ؟
2. إذا أردت أن يجعل الروبوت دائرة أكبر، هل تقلل الفرق بين سرعة العجلات أم تزيده؟
3. ما فائدة استخدام كتلة التكرار في المشاريع؟

تقويم مستمر مقترح لنتاجات التعلم وفق مقياس Rubric

المعيار	ممتاز(4)	جيد(3)	مقبول(2)	يحتاج لتحسين(1)
توظيف كتلةSteer	توظيف دقيق وبما يناسب المسار المطلوب.	توظيف صحيح مع بعض الأخطاء في القيم.	توظيف جزئي لم يحقق النتيجة.	لم يوظف اللبنة بشكل صحيح.
استخدام كتلة التكرار(Repeat)	استعمل التكرار بكفاءة لتقليل الأوامر.	استعمل التكرار بشكل جزئي.	التكرار غير مناسب أو زائد عن الحاجة.	لم يستخدم التكرار.
دقة التحكم بالمسار (دوار/موقف السيارات)	الروبوت تحرك بدقة وسلاسة.	الروبوت تحرك مع بعض الانحرافات.	الروبوت تحرك بصعوبة أو فقد الاتجاه.	الروبوت لم يحقق المطلوب.
التعاون والإبداع	مشاركة فعالة وحلول مبتكرة.	مشاركة مقبولة وحل تقليدي.	مشاركة محدودة.	لا يوجد تعاون أو ابتكار.

الدرس الثالث

من الحركة إلى التعبير

المقدمة

في هذا الدرس سيتعرف المتعلم على كيفية جعل الروبوت يتفاعل مع الأحداث من خلال عرض نصوص على شاشة الروبوت وإصدار أصوات في أوقات محددة .سيتعلم المتعلم كيفية برمجة الروبوت لإظهار رسائل على الشاشة الافتراضية، ضبط مدة عرض النصوص، مسح الشاشة، وإضافة مؤثرات صوتية مرتبطة بمهام محددة، مما يجعل مشاريع الروبوت أكثر تفاعلية.

خطة الدرس

الحصة	الموضوع	النشاط
الأولى	عرض النصوص على شاشة الروبوت	كتابة رسائل مخصصة تظهر في نقاط محددة من المهمة
الثانية	إضافة الأصوات والمؤثرات	برمجة الروبوت لإصدار نغمات أو أصوات عند أحداث معينة ودمجها مع عرض النصوص
المشروع		مشروع روبوت 3 ص 174 من الكتاب المدرسي رحلة الروبوت خريطة الكنز
QR Code لفيديوهات شرح وأنشطة تعلمية		ملفات أوراق العمل ملفات مصادر التعلم

نتاجات التعلم

1. برمجة الروبوت لإصدار صوت عند حدوث حدث معين.
2. إظهار نصوص على شاشة الروبوت في نقاط محددة أثناء المهمة.
3. الربط بين الحدث والأثر باستخدام الصوت والنصوص.
4. اختيار التسلسل المنطقي الأفضل لتنفيذ المهمة.
5. تحسين مسار الروبوت بناءً على التغذية الراجعة.

المصطلحات التقنية

- Show Text Block: كتلة لإظهار نصوص على شاشة الروبوت.
- Wait Block: كتلة لتحديد مدة الانتظار قبل تنفيذ أمر جديد أو أثناء عرض النص.
- Clear Display Block: كتلة لمسح النصوص من شاشة الروبوت.
- Play Frequency Block: كتلة لإصدار صوت بتردد معيّن (Hz) .
- Play Note Block: كتلة لعزف نغمة موسيقية محددة.

ملاحظات للمعلم

- شجّع المتعلمين على استخدام رسائل واضحة وذات معنى عند عرض النصوص (يمكن أن تكون باللغة العربية).
- وضّح أن وحدة قياس زمن الانتظار في الكتل البرمجية هي الميلي ثانية (1 ثانية = 1000 ميللي ثانية).
- راقب توافق النصوص والأصوات مع سياق المهمة لتجنب التشويش على المشروع.
- يمكن دمج كتل العرض والصوت مع كتل الحركة لرسم أشكال هندسية تفاعلية.
- ذكّر المتعلمين باستخدام أداة إعادة ضبط الروبوت قبل إعادة تشغيل البرنامج

الأنشطة النظرية والعملية

نشاط – نظري

1. ناقش الفرق بين عرض النصوص على الشاشة الافتراضية وإصدار الأصوات من حيث التأثير التفاعلي على المشروع.
2. بيّن دور التغذية الراجعة الصوتية والبصرية في تحسين تجربة المستخدم مع الروبوت.

نشاط – عملي

برمج الروبوت لرسم شكل هندسي (مثل مربع أو مثلث) مع عرض رسالة عند اكتمال كل ضلع، وإصدار صوت مختلف عند الانتقال إلى الضلع التالي.

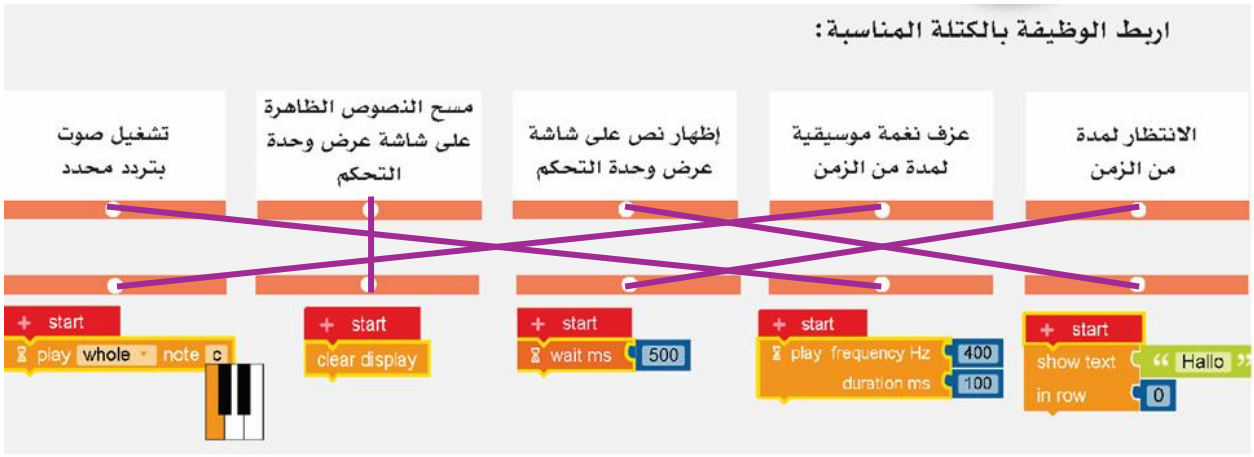
أنشطة الكتاب

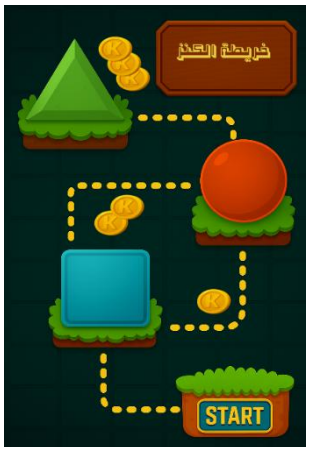
نشاط صـ 170ـ

قارن بين المقطعين الظاهرين أمامك وسجل الاختلاف بينهما.

- في المقطع (1) يظهر النص مرة واحدة فقط في بداية التنفيذ.
- في المقطع :(2) يظهر النص 3مرات لأنه داخل حلقة التكرار.

نشاط ص 173ـ





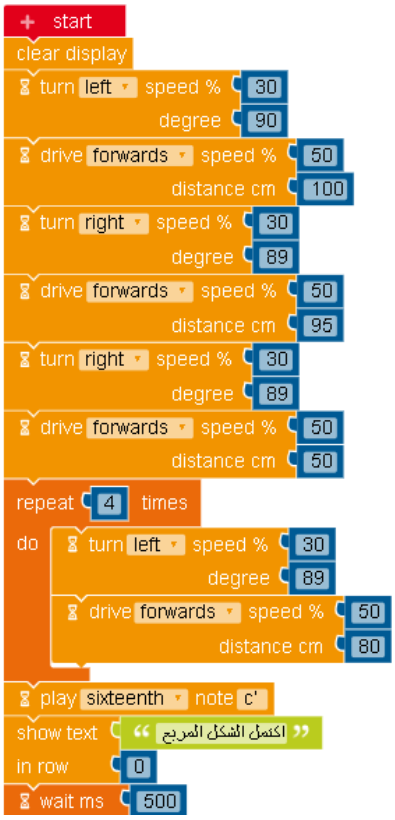
مشروع روبوت 3 ص 174 من الكتاب المدرسي

خريطة الكنز

البرنامج يوجه الروبوت على خريطة الكنز بحيث يزور الأشكال الثلاثة بالترتيب: المربع، ثم الدائرة، ثم المثلث، مع تنفيذ مسارات خاصة بكل شكل وإظهار رسالة عند اكتمال رسم كل واحد منها

شرح المسار بالتفصيل:

1. تهيئة البداية :مسح الشاشة.
2. من نقطة البداية حتى المربع الأزرق



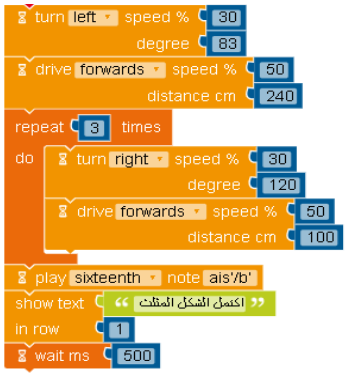
- الروبوت يدور لليسار، يتقدم عدة مسافات مستقيمة، وينعطف يميناً ويساراً حتى يصل إلى موقع المربع.
- عند المربع ينفذ حلقة تكرار 4 مرات لتشكيل المربع (كل مرة دوران يسار 90° + تقدم 80 سم).
- يعرض على الشاشة" :اكتمل الشكل المربع "ويشغل نغمة.

3. من المربع إلى الدائرة الحمراء:

- بعد المربع، يتحرك الروبوت للأمام، يدور لليسار واليمين وفق مسافات دقيقة.
- ينفذ مناورة Steer لعمل انحناء طويل للوصول إلى الدائرة.
- عند الدائرة يعرض" اكتمل الشكل الدائري "مع نغمة.

4. من الدائرة إلى المثلث الأخضر:

- يتحرك للأمام، ثم يدور يسار، يسير لمسافة طويلة.
- ينفذ حلقة تكرار 3 مرات لتشكيل المثلث (كل مرة دوران يمين 120° + تقدم 100 سم).
- يعرض" اكتمل الشكل المثلث "ويشغل نغمة.



النتيجة:

الروبوت يتبع المسار الأصفر على الخريطة بدقة، يتوقف عند كل شكل لإتمامه ثم يعلن اكتماله.

التحدي:

يمكنك جعل الروبوت يرسم طريق العودة إلى نقطة البداية ويظهر نص " استكملت الطريق وحصلت على الكنز "من خلال تكرار الكتل البرمجية مع عكس ترتيب الكتل وجعل الروبوت يتحرك للخلف بدلا من الامام.

تقويم مقترح

1. ما الكتلة البرمجية التي تستخدم لمسح الشاشة؟
2. كيف يمكن ضبط مدة عرض النصوص على الشاشة؟
3. أيهما يستخدم لتحديد النغمة الموسيقية Play Frequency أم Play Note ؟

تقويم مستمر مقترح لنتاجات التعلم وفق مقياس Rubric

المعيار	ممتاز(4)	جيد(3)	مقبول(2)	يحتاج لتحسين(1)
برمجة النصوص والأصوات	الروبوت يعرض النصوص ويصدر الأصوات في الوقت المناسب تمامًا.	الروبوت يعرض النصوص/الأصوات مع تأخير بسيط.	الروبوت يعرض/يصدر جزءًا فقط من المطلوب.	لم ينجح في عرض النصوص أو إصدار الأصوات.
الربط بين الأحداث والاستجابات	الربط منطقي وسليم تمامًا.	الربط موجود مع بعض الثغرات.	الربط ضعيف وغير متماسك.	لا يوجد ربط واضح.
اكتمال المشروع (خريطة الكنز)	المشروع مكتمل وينفذ المطلوب بدقة.	المشروع مكتمل مع بعض النواقص.	المشروع ناقص أو غير واضح.	المشروع غير منجز.
العرض والتفسير	شرح ممتاز للخطوات والكتل.	شرح مقبول مع بعض النواقص.	شرح جزئي أو سطحي.	لم يشرح أو الشرح غير مفهوم.

07

دروس الصف السابع

عنوان الوحدة	التفاعل الذكي بين الروبوت وبيئته		
الدروس	تفاعل الروبوت مع العوائق مستشعر Touch في الروبوت	قياس المسافات بذكاء مستشعر Ultrasonic في الروبوت	خريطة الطريق بعيون الاستشعار
عدد الحصص	2	2	2
المشروع	مشروع روبوت1 ص 166	مشروع روبوت2ص 176	ا مشروع روبوت3 ص 190

فكرة الوحدة

تركز هذه الوحدة على تمكين المتعلمين من برمجة الروبوتات للتفاعل مع بيئتها المحيطة بذكاء باستخدام المستشعرات، من خلال بيئة المحاكاة Open Roberta Lab يتدرج المتعلمون من استخدام المستشعرات البسيطة لاكتشاف العوائق، إلى قياس المسافات والتحكم في الحركة بناءً على القراءات، وصولًا إلى دمج أكثر من مستشعر مع الحلقات والمتغيرات لاتخاذ قرارات معقدة أثناء الحركة.

في نهاية هذه الوحدة سيكون المتعلم قادرًا على:

1. التعرف على مستشعر اللمس وآلية عمله في الروبوت.
2. برمجة الروبوت للتوقف أو تغيير المسار عند ملامسة عائق.
3. استخدام المستشعر فوق الصوتي لقياس المسافات بدقة.
4. التحكم في حركة الروبوت بناءً على قيم المسافة المقاسة.
5. دمج أكثر من مستشعر في برنامج واحد لتحقيق سلوكيات معقدة.
6. استخدام الحلقات المتداخلة والمتغيرات في البرمجة.
7. حفظ واستدعاء المشاريع من منصة البرمجة.

ملخص محتوى الدروس

الدرس الأول: تفاعل الروبوت مع العوائق- مستشعر Touch في الروبوت

يتعلم المتعلمين كيفية استخدام مستشعر اللمس Touch sensor لاكتشاف العوائق أمام الروبوت، وبرمجته للتوقف أو تغيير الاتجاه عند ملامستها.

الدرس الثاني: قياس المسافات بذكاء مستشعر Ultrasonic في الروبوت

يكتشف المتعلمين آلية عمل المستشعر فوق الصوتي Ultrasonic Sensor، وكيفية برمجة الروبوت لقياس المسافات والتحرك أو التوقف أو تغيير المسار بناءً على القيم المقاسة.

الدرس الثالث: خريطة الطريق بعيون الاستشعار

يدمج المتعلمين بين مستشعر اللمس ومستشعر الفوق الصوتي، مع استخدام الحلقات المتداخلة والمتغيرات، لبرمجة الروبوت لاتخاذ قرارات متعددة أثناء الحركة في بيئة مليئة بالعوائق.

استراتيجية التعلم

- ✓ التعلم القائم على المشروع (Project-Based Learning) .
- ✓ التعلم بالاكتشاف والتجريب في بيئة آمنة.
- ✓ التدرج من السهل إلى الصعب في تقديم المفاهيم البرمجية.
- ✓ الدمج بين الشرح النظري والتطبيق العملي على المحاكي.

قيم ومهارات مدمجة

- ✓ التفكير الحاسوبي وحل المشكلات.
- ✓ العمل التعاوني وتبادل الأدوار.
- ✓ الإبداع في تصميم مسارات الروبوت الذكية.
- ✓ الدقة والانتباه للتفاصيل.

التقويم المستمر

تم إعداد مقياس Rubric لتقويم أداء متعلمي الصف السابع في دروس الروبوت، والتي تركز على توظيف المستشعرات بأنواعها (مستشعر اللمس والمستشعر فوق الصوتي) إضافة إلى استخدام الحلقات المتداخلة لبرمجة سلوكيات أكثر تعقيداً. ويهدف الروبريك إلى تمكين المعلمين من متابعة تقدم المتعلمين في ربط المفاهيم العلمية بالبرمجة العملية، وتعزيز قدرتهم على التفكير المنطقي واتخاذ القرارات وفق معطيات بيئية متعددة.

الدرس الأول

كيف يتفاعل الروبوت مع العوائق- مستشعر Touch في الروبوت

المقدمة

في هذا الدرس سيتعرف المتعلم على كيفية برمجة الروبوت للاستجابة للعوائق الموجودة أمامه باستخدام مستشعر اللمس (Touch Sensor). سيتعلم المتعلم كيفية جعل الروبوت يتحرك إلى الأمام ويتوقف أو يغير اتجاهه عند ملامسة عائق، بالإضافة إلى ضبط الحركة للخلف أو الدوران للخروج من المسار المغلق. هذا الدرس يمثل مدخلًا لفهم كيفية تفاعل الروبوت مع بيئته من خلال المستشعرات واتخاذ القرارات البسيطة.

خطة الدرس

الحصة	الموضوع	النشاط
الأولى	التعرف على مستشعر اللمس	تجربة حركة الروبوت للأمام والتوقف عند الاصطدام بعائق
الثانية	تعديل المسار بعد الاصطدام	برمجة الروبوت للتحرك للخلف أو الالتفاف عند ملامسة العائق
المشروع		مشروع روبوت 1 صـ 166 ـمن الكتاب المدرسي رحلة الروبوت في دولة الكويت
QR Code لفيديوهات شرح وأنشطة تعلمية		ملفات أوراق العمل ملفات مصادر التعلم

نتاجات التعلم

1. التعرف على مفهوم المستشعرات ودورها في استشعار البيئة المحيطة بالروبوت.
2. التعرف على أنواع المستشعرات المستخدمة في الروبوتات التعليمية.
3. استنتاج أهمية المستشعرات في تفاعل الروبوت مع العوائق وتعديل سلوكه تلقائيًا.
4. تمييز الكتل البرمجية الخاصة بمستشعر اللمس.
5. تركيب المستشعرات في محاكي الروبوت بشكل صحيح وآمن.
6. إدراج عائق داخل المحاكي لاختبار استجابة الروبوت باستخدام المستشعر الخلفي.
7. تصميم خريطة اختبار تحاكي سيناريوهات اصطدام وتفادي الروبوت للعوائق.
8. برمجة الروبوت على التفاعل مع العوائق من خلال المستشعر الخلفي لضبط حركته.
9. تحقيق التكامل بين المستشعرات وبرمجة الروبوت لتنفيذ المهام.

المصطلحات التقنية

- Touch Sensor: مستشعر اللمس أو الضغط.
- Wait Until BlocK: كتلة انتظار حتى تحقق شرط معين.
- Bumper Trigger : حدث برمجي ينفذ عند ملامسة العائق بالمستشعر.
- :Reverse Motion: حركة الروبوت للخلف عند تحقق شرط محدد.
- Pivot Turn : دوران الروبوت حول إحدى عجلاته لتغيير الاتجاه.
- Condition BlocK: كتلة تحدد شرطًا برمجيًا لتنفيذ أوامر معينة.

ملاحظات للمعلم

- تأكد من إضافة مستشعر اللمس Touch Sensor في إعدادات الروبوت داخل المنصة.
- وضّح للمتعلمين الفرق بين الانتظار حتى الضغط (Wait Until) والانتظار لزمن محدد (Wait) .
- إذا لم يتوقف الروبوت عند ملامسة العائق، تحقق من توصيل مستشعر اللمس بالمنفذ الصحيح في الإعدادات.
- شجع المتعلمين على تجربة أكثر من حل للتعامل مع العائق (رجوع – دوران – مزيج منهما).

الأنشطة النظرية والعملية

ملاحظة: تم اقتراح أنشطة إضافية في الدليل لإتاحة خيارات متعددة للمعلم.

نشاط – نظري

1. اشرح كيف يساعد مستشعر اللمس الروبوت على تجنب العوائق.
2. ناقش ما الذي يمكن أن يحدث لو لم يتم استخدام مستشعر في المهمة.

نشاط – عملي

برمج الروبوت ليتحرك للخلف، وعند الاصطدام بعائق يتحرك للأمام مسافة 20 سم، ثم ينعطف 90° ويكمل المسار.

أنشطة الكتاب

نشاط ص 164ـ

اذكر ثلاثة أمثلة لاستخدام مستشعر الروبوت في حياتك العملية:

1. في المصاعد الضغط على زر الطابق.
2. في الصراف الآلي أو الهاتف الضغط على الأزرار أو الشاشة.
3. في الآلات زر الطوارئ لإيقاف التشغيل

مشروع روبوت 1 صـ 166 من الكتاب المدرسي

رحلة الروبوت في دولة الكويت

رحلة الروبوت على مسار "رحلة الكويت "، حيث يلعب مستشعر اللمس الخلفي دوراً مهماً في الانطلاق بعد الاصطدام

بداية الرحلة:

يكرر الروبوت أول خطوة مرتين: دوران لليمين 90°، ثم الرجوع للخلف بسرعة 30% . (أو يمكن تغيير اتجاه الروبوت باستخدامك أسهم لوحة المفاتيح والاستغناء عن تكرار كتلة الاستدارة لجهة اليمين).

1. **انتظار الإشارة من مستشعر اللمس:**

- الروبوت يتوقف حتى يلمس العائق من الخلف (يكون المستشعر في الجهة الخلفية للروبوت).
- عند الضغط على المستشعر، يعرف الروبوت أنه وصل إلى "نقطة البداية الحقيقية" أو اصطدم بالحائط المخصص للانطلاق.

2. **بدء المسار الأمامي:**

- يتحرك للأمام، يقوم بانحناءات (Steer) يمين ويسار وزوايا مختلفة ليتبع المسار المتعرج.

3. **متابعة الطريق حتى خط النهاية:**

- يتنقل عبر المسار الأسود في الخريطة، متجنباً الانحرافات الخاطئة، حتى يصل إلى علامة FINISH .

دور مستشعر اللمس الخلفي هنا

1. يمنع الروبوت من البدء بالمسار قبل الاصطدام بالعائق الخلفي (مثلاً جدار أو حاجز).
2. يضمن أن الروبوت ينطلق من موضع محدد بدقة قبل تنفيذ المسار.

التحدي:

أضف عائقًا آخر في منتصف الطريق عندما يصطدم به الروبوت، يصدر صوتا مختلفا ويقوم بحركة خاصة (مثلاً يدور دورة كاملة ثم يكمل).

تقويم مقترح

1. ما الكتلة البرمجية التي تجعل الروبوت ينتظر حتى يضغط على مستشعر اللمس؟
2. ماذا يحدث إذا حذفت كتلة Wait Until من البرنامج؟
3. كيف يمكن جعل الروبوت يلتف يسارًا بدلًا من اليمين بعد الاصطدام؟

تقويم مستمر مقترح لنتاجات التعلم وفق مقياس Rubric

المعيار	ممتاز(4)	جيد(3)	مقبول(2)	يحتاج لتحسين(1)
فهم المستشعرات	يوضح مفهوم مستشعر اللمس وأهميته بدقة.	يوضح المستشعر بشكل عام مع بعض النواقص.	يذكر المستشعر دون توضيح كافٍ.	لا يوضح المستشعر.
برمجة الروبوت للتفاعل مع العائق	البرنامج يعمل بسلاسة ويستجيب فورًا للعائق.	البرنامج يعمل مع تأخير أو خطأ بسيط.	البرنامج يعمل جزئيًا أو يستجيب أحيانًا.	البرنامج لا يعمل.
رسم الخريطة/المسار	يرسم خريطة دقيقة لمسار الروبوت.	يرسم خريطة مع بعض الأخطاء.	يرسم خريطة غير مكتملة.	لم يرسم خريطة.
إخراج إضافي (صوت/نص)	يوظف الصوت/النص بشكل صحيح ومناسب.	يوظف الصوت/النص مع بعض الأخطاء.	إخراج محدود أو غير واضح.	لا يوجد إخراج.

الدرس الثاني

قياس المسافات بذكاء مستشعر Ultrasonic في الروبوت

المقدمة

في هذا الدرس سيتعرف المتعلم على مستشعر الموجات فوق الصوتية (Ultrasonic Sensor) وآلية عمله في قياس المسافة بين الروبوت والعوائق دون لمسها. سيتعلم المتعلم العلاقة بين الصوت والزمن وكيفية الاستفادة منها في تحديد المسافة، كما سيقوم ببرمجة الروبوت لاتخاذ قرارات عند الاقتراب من عائق، مثل التوقف أو الالتفاف أو إصدار صوت تنبيهي.

خطة الدرس

الحصة	الموضوع	النشاط
الأولى	التعرف على مستشعر الموجات فوق الصوتية	تجربة قراءة المستشعر للمسافة أمام الروبوت في بيئة المحاكاة
الثانية	اتخاذ قرارات عند الاقتراب من عائق	برمجة الروبوت للتوقف أو الالتفاف أو إصدار صوت بناءً على قراءة المستشعر
المشروع		مشروع روبوت 2 صـ 176 من الكتاب المدرسي رحلة في معالم دولة الكويت
QR Code لفيديوهات شرح وأنشطة تعلمية		ملفات أوراق العمل ملفات مصادر التعلم

نتاجات التعلم

1. التعرف على مستشعر الموجات فوق الصوتية وآلية عمله في قياس المسافات.
2. تفسير العلاقة بين سرعة الصوت والزمن والمسافة لفهم مبدأ عمل المستشعر.
3. متابعة قراءات المستشعرات في بيئة المحاكاة لتقييم تفاعل الروبوت مع محيطه.
4. برمجة الروبوت لاتخاذ قرارات عند الاقتراب من الأجسام باستخدام المستشعرات.
5. ربط المفاهيم العلمية حول الصوت والحركة بتجارب عملية تحاكي الواقع.

المصطلحات التقنية

- Ultrasonic Sensor: مستشعر يقيس المسافة باستخدام الموجات فوق الصوتية.
- Ultrasonic Ping: إشارة صوتية يرسلها المستشعر لقياس المسافة.
- Echo: الصدى – الصوت المرتد عن سطح صلب.
- Echo Time: الزمن المستغرق لعودة الموجة الصوتية بعد ارتدادها عن العائق.
- Proximity CheckK: عملية فحص المسافة بين الروبوت والعائق.
- Alert Tone: نغمة صوتية تصدر عند حدوث إنذار أو اقتراب من العائق.
- Safe Distance Parameter: قيمة تحدد المسافة الآمنة التي يجب الحفاظ عليها.

ملاحظات للمعلم

- وضّح للمتعلمين أن الموجات فوق الصوتية أسرع من الصوت العادي ولكنها غير مسموعة للأذن البشرية.
- استخدم أمثلة من الطبيعة (مثل الخفافيش والدلافين) لشرح آلية تحديد الموقع بالصدى.
- إذا لم يقرأ المستشعر القيم الصحيحة، تحقق من تفعيله في إعدادات الروبوت.
- شجع المتعلمين على تعديل المسافات والسرعات لاختبار دقة الاستجابة.

الأنشطة النظرية والعملية

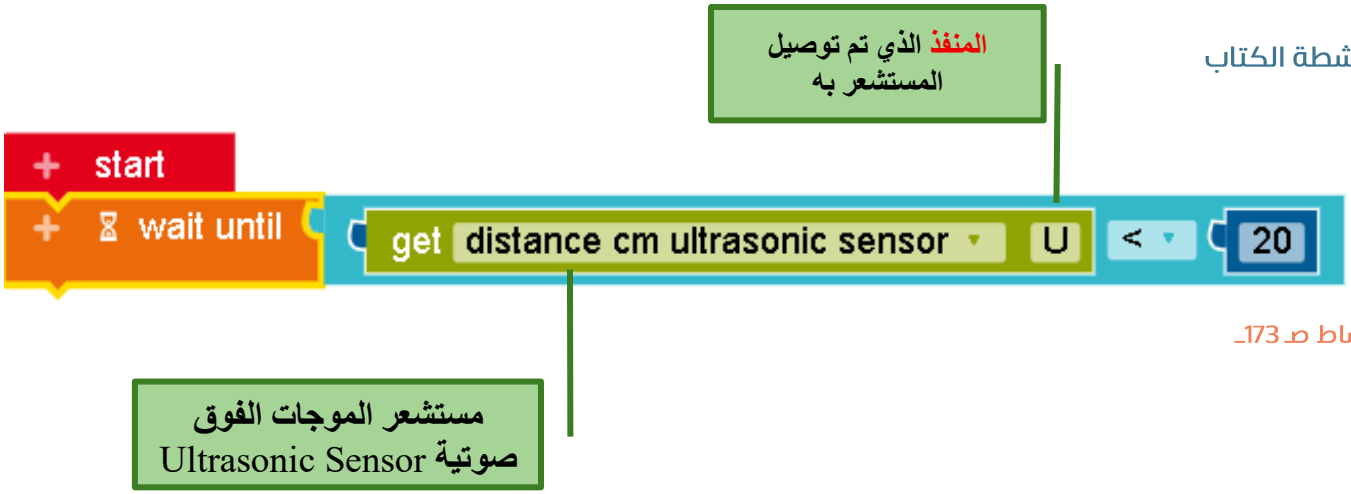
نشاط – نظري

1. ناقش كيف يمكن حساب المسافة باستخدام الزمن وسرعة الصوت.
2. قارن بين مستشعر الموجات فوق الصوتية ومستشعر اللمس من حيث طريقة العمل والاستخدامات.

نشاط – عملي

برمج الروبوت ليتحرك للأمام بسرعة مناسبة، وعند الاقتراب لمسافة أقل من 15 سم من العائق، يتوقف ويصدر صوتًا، ثم ينعطف 90° ويتابع المسار.

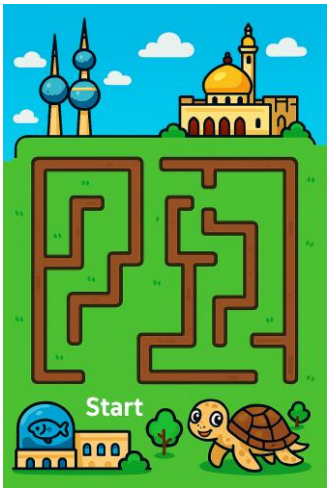
أنشطة الكتاب



نشاط ص 173_

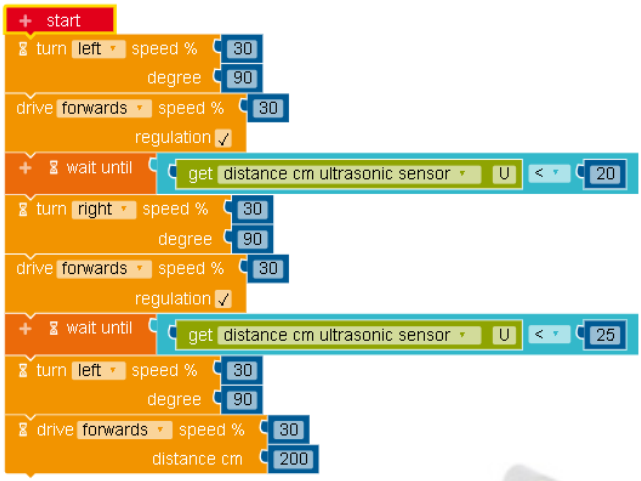
مشروع روبوت 2 صـ 176 من الكتاب المدرسي

رحلة في معالم دولة الكويت



برمجة الروبوت لعبور المتاهة باستخدام مستشعر الموجات فوق الصوتية (Ultrasonic Sensor) في اكتشاف العوائق.

1. الدوران لليسار 90°، ثم التقدم للأمام ببطء (30% سرعة).
2. انتظار حتى يكتشف المستشعر عائقًا على بُعد أقل من 20 سم.
3. عند اكتشاف العائق، يدور الروبوت لليمين 90° ويتحرك للأمام.
4. انتظار مرة أخرى حتى يكتشف عائقًا على بُعد أقل من 25 سم.
5. عند العائق الثاني، يدور لليسار 90° ويتحرك للأمام 200 سم للوصول إلى نهاية المسار.



دور مستشعر الموجات فوق الصوتية هنا:

1. يعمل مثل "عيون" الروبوت، حيث يقيس المسافة للأجسام أمامه.
2. يمنع الاصطدام بالجدران داخل المتاهة عبر التوقف أو تغيير الاتجاه عند الاقتراب من العائق.

التحدي

يمكن جعل الروبوت يعرض عدد العوائق التي مر بها من خلال عرض نص على الشاشة في نهاية الرحلة.

تقويم مقترح

1. ما هي وظيفة Ultrasonic Sensor ؟
2. ماذا يحدث إذا جعلت المسافة في الشرط أكبر من 50 سم؟
3. كيف يمكن جعل الروبوت يتجاهل العوائق البعيدة ويتفاعل فقط مع القريبة جدًا؟

تقويم مستمر مقترح لنتاجات التعلم وفق مقياس Rubric

المعيار	ممتاز(4)	جيد(3)	مقبول(2)	يحتاج لتحسين(1)
فهم المستشعر فوق الصوتي	يشرح آلية عمله وربطه بالواقع بوضوح.	يشرح المستشعر بشكل مقبول مع نقص.	يذكر وظيفته فقط دون تفاصيل.	لا يشرح المستشعر.
برمجة قياس المسافة	البرنامج يقيس المسافة ويتخذ القرار بدقة.	البرنامج يقيس المسافة مع بعض الأخطاء.	البرنامج يعمل جزئيًا أو بشكل غير مستقر.	البرنامج لا يعمل.
التفاعل مع العوائق	الروبوت يتجنب العوائق بكفاءة وسلاسة.	الروبوت يتجنب العوائق مع بعض الانحرافات.	الروبوت يتجنب بعض العوائق فقط.	الروبوت لا يتجنب العوائق.
ربط المفهوم العلمي بالواقع	يربط المفاهيم (الصوت/الزمن/المسافة) بوضوح مع أمثلة.	يربطها بشكل عام دون تفاصيل كافية.	يذكر العلاقة جزئيًا.	لا يربط المفاهيم.

الدرس الثالث

خريطة الطريق بعيون الاستشعار

المقدمة

في هذا الدرس سيتعرف المتعلم على كيفية برمجة الروبوت لاستخدام مستشعر المسافة للتنقل في طريق محدد وتجنب العوائق. سيتعلم المتعلم كيفية قراءة بيانات المستشعر وتخزينها في متغيرات، ثم استخدام هذه البيانات لاتخاذ قرارات مثل التوقف، أو تغيير الاتجاه، أو مواصلة الحركة، وذلك وفقًا لخريطة طريق افتراضية في بيئة المحاكاة. هذا الدرس يركز على اتخاذ القرارات المتعددة باستخدام الشروط المتداخلة والخوارزميات البسيطة.

خطة الدرس

الحصة	الموضوع	النشاط
الأولى	قراءة بيانات مستشعر المسافة	عرض القيم الحية للمسافة على شاشة الروبوت أثناء الحركة
الثانية	برمجة قرارات متعددة حسب المسافة	تصميم برنامج يتحكم في حركة الروبوت لتجنب العوائق وفق خريطة طريق في المحاكاة
المشروع		مشروع روبوت 3 صـ 190 من الكتاب المدرسي التجوال في بلادي بعيون الاستشعار
QR Code لفيديوهات شرح وأنشطة تعلمية		ملفات أوراق العمل ملفات مصادر التعلم

نتاجات التعلم

1. استخدام أكثر من مستشعر داخل مشروع برمجي لتحقيق تفاعل ديناميكي مع البيئة.
2. تمييز وظيفة الحلقات المتداخلة في تنفيذ مهام متكررة تعتمد على قراءات المستشعرات.
3. إدراج أكثر من روبوت في المحاكي لتصميم بيئة تعاونية أو تفاعلية.
4. برمجة الروبوت لاتخاذ قرارات متداخلة استنادًا إلى مستشعرات متعددة.
5. استخدام الجمل الشرطية في برمجة الروبوت لاتخاذ قرارات بناءً على المدخلات.
6. توظيف التداخل Nesting لتضمين قرارات متعددة داخل هيكل برمجي متكامل للتحكم في سلوك الروبوت.

المصطلحات التقنية

- Nested Condition: شرط متداخل داخل شرط آخر لاتخاذ قرارات متعددة.
- Distance Reading: القيمة المقاسة للمسافة بين الروبوت والعائق.
- Multiple Robot Simulation: وضع المحاكاة الذي يسمح بإضافة وتشغيل أكثر من روبوت في نفس المشروع.
- Distance Variables: متغيرات تخزن قياسات المسافات من المستشعر.
- Decision Algorithm: خوارزمية لتحديد الإجراء المناسب بناءً على بيانات المسافة.

ملاحظات للمعلم

- وضّح للمتعلمين أهمية تحديد مسافة آمنة لتجنب الاصطدام.
- شجع المتعلمين على استخدام المتغيرات بدلا من الاعتماد على القيم المباشرة، مما يسهل تعديل البرنامج لاحقًا.
- يمكن تصميم خريطة الطريق في المحاكاة باستخدام مسارات وأشكال متنوعة لاختبار مرونة البرنامج.
- اشرح الفرق بين الشرط البسيط والشرط المتداخل وأهمية كل منهما في اتخاذ القرارات.

الأنشطة النظرية والعملية

نشاط – نظري

1. ناقش كيف تساعد الشروط المتداخلة في تحسين ذكاء الروبوت.
2. اشرح سبب الحاجة إلى مسافة أمان عند برمجة الروبوت.

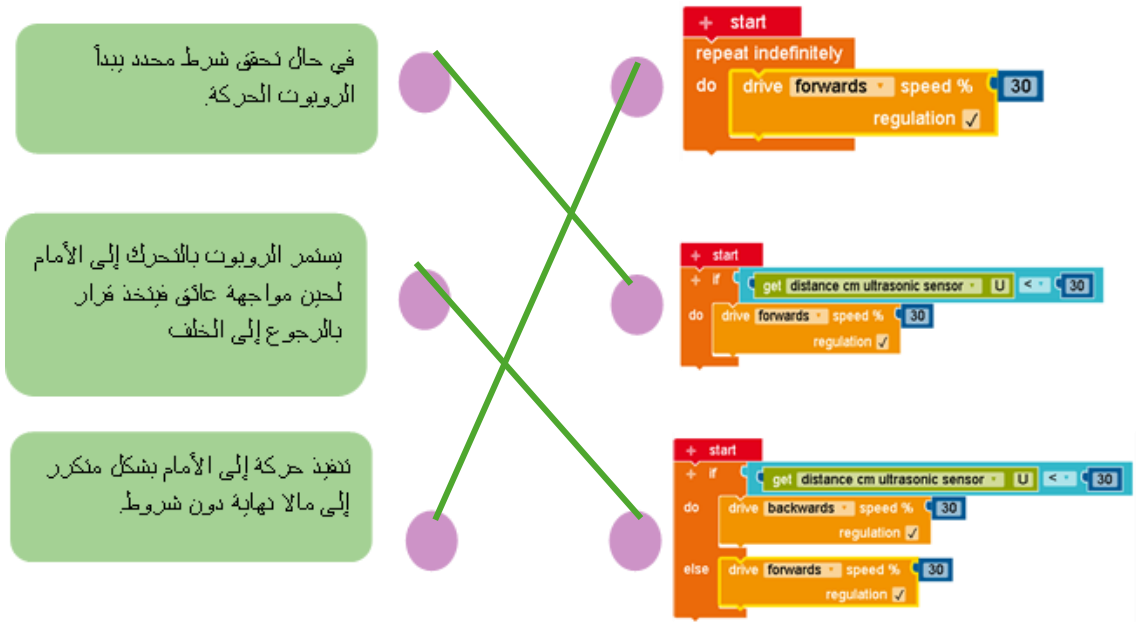
نشاط – عملي

برمج الروبوت للتحرك للأمام، وإذا كانت المسافة أقل من 20 سم يبطئ سرعته، وإذا كانت أقل من 10 سم يتوقف ويدور 90° جهة اليمين.

أنشطة الكتاب

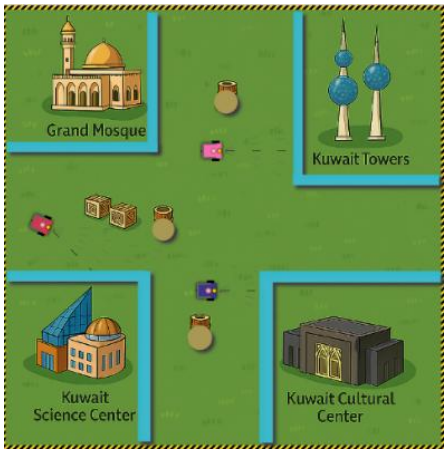
نشاط صـ 184ـ

صل كل مقطع برمجي بوظيفته الصحيحة



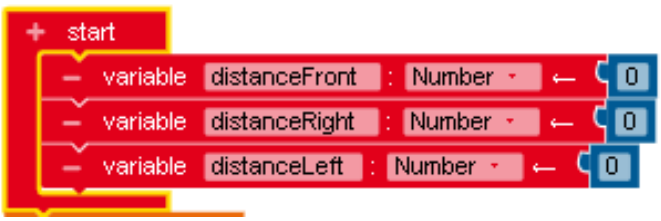
مشروع روبوت 3 صـ 190 من الكتاب المدرسي

التجوال في بلادي بعيون الاستشعار



الروبوت يتصرف بطريقة الملاحة الذاتية باستخدام مستشعر الموجات فوق الصوتية لاختيار الاتجاه الأقل عائقًا، ويتحرك لزيارة المعالم.

إنشاء المتغيرات



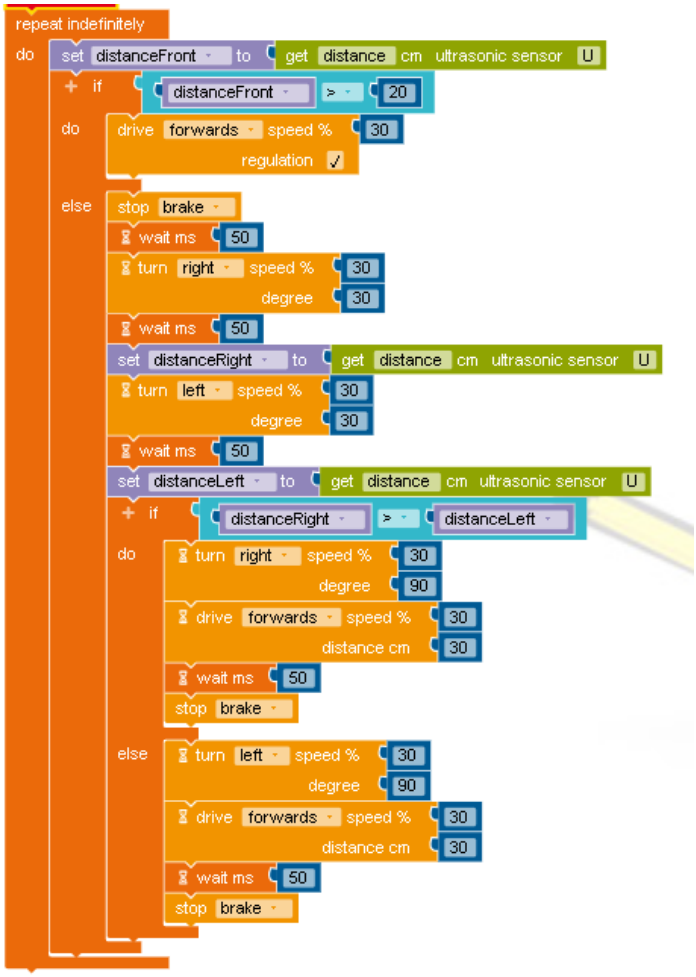
distanceFront، distanceRight،

distanceLeft

لتخزين قراءات المسافات من المستشعر

وتكون من نوع Number وقيمتها الابتدائية 0

.



1. التكرار المستمر(repeat indefinitely)
2. قراءة المسافة الأمامية .(distanceFront)
   - إذا كانت المسافة > من 20 سم يتحرك الروبوت للأمام.
   - إذا كانتٍ ≥ 20 سم يتوقف.
3. فحص اليمين يدور لليمين 30° ويقرأ distanceRight.
4. فحص اليسار يدور لليسار 60° (ليعود إلى وضع اليسار من الأصل) ويقرأ distanceLeft.
5. المقارنة بين المسافتين
   - إذا كانت distanceRight > distanceLeft يدور يمين 90° ويتقدم للأمام.
   - إذا كانت distanceLeft > يدور يسار 90° ويتقدم.

التحدي:

طوّر البرنامج بعرض رسالة "ما أجمل معالم بلدي الكويت!"على شاشة عرض وحدة التحكم عند زيارة كل معلم.

عند تحقق الشرط المسافة الامامية اقل من 20 بعد ان يتم قياس المسافة جهة اليمين/اليسار اضف كتلة الكتابة لإظهار النص المطلوب.

تقويم مقترح :

1. ما الفرق بين الشرط البسيط والشرط المتداخل؟
2. كيف يمكن استخدام متغيرات لتخزين بيانات المسافة؟
3. ما الهدف من تحديد مسافة أمان للروبوت؟

تقويم مستمر مقترح لنتاجات التعلم وفق مقياس Rubric

المعيار	ممتاز(4)	جيد(3)	مقبول(2)	يحتاج لتحسين(1)
استخدام أكثر من مستشعر	يوظف مستشعرين أو أكثر بشكل تكاملي.	يوظف أكثر من مستشعر لكن دون تنسيق تام.	يوظف مستشعرًا واحدًا فقط.	لا يوظف المستشعرات.
تطبيق الحلقات المتداخلة (Nested Loops)	يوظفها بكفاءة لحل مشكلة معقدة.	يوظفها بشكل صحيح مع بعض القصور.	يوظفها بشكل جزئي أو غير فعال.	لم يستخدم الحلقات المتداخلة.
اتخاذ القرار المنطقي	الروبوت يتخذ قرارات دقيقة بناءً على المدخلات.	الروبوت يتخذ قرارات صحيحة غالبًا مع بعض الأخطاء.	الروبوت يتخذ قرارات جزئية أو عشوائية.	الروبوت لا يتخذ قرارات صحيحة.
مشروع الخريطة/المتاهة	المشروع مكتمل ويظهر الروبوت طريقًا صحيحًا.	المشروع مكتمل مع بعض الانحرافات.	المشروع ناقص أو الروبوت ضل الطريق.	المشروع غير منجز.

08

دروس الصف الثـامن

عنوان الوحدة	البرمجة الذكية باستخدام المستشعرات والدوال		
الدروس	تفاعل الروبوت مع الالوان	انعطاف ذكي بعيون Gyroscope	البرمجة الذكية تبدأ بالدوال Functions
عدد الحصص	2	2	2
المشروع	مشروع روبوت 1 ص 198	مشروع روبوت 2 ص 214	مشروع روبوت 2 ص 228

فكرة الوحدة

تركز هذه الوحدة على تطوير مهارات المتعلمين في برمجة الروبوتات للتحكم الدقيق والتفاعل الذكي مع بيئتها، باستخدام مستشعرات متقدمة ومفاهيم برمجية احترافية في منصة Open Roberta Lab يبدأ المتعلمين بالتعرف على مستشعر الألوان واستخدامه لتمييز الألوان وشدة الضوء، ثم يتعلمون كيفية الاستفادة من مستشعر Gyroscope sensor لضبط اتجاهات الروبوت بدقة أثناء الحركة، وصولًا إلى توظيف الدوال في تنظيم الكود وتنفيذ مهام متكررة بكفاءة.

في نهاية هذه الوحدة سيكون المتعلم قادرًا على:

1. التعرف على مستشعر الألوان ووظيفته في تمييز الألوان وقياس شدة الضوء.
2. برمجة الروبوت للتفاعل مع ألوان محددة والتحكم في الحركة بناءً على قراءة المستشعر.
3. التعرف على مستشعر Gyroscope sensor واستخدامه في قياس الزوايا وضبط الاتجاهات.
4. تنفيذ مسارات دقيقة وانعطافات محددة باستخدام بيانات Gyroscope sensor.
5. إنشاء دوال برمجية خاصة لتبسيط الكود وتقليل التكرار.
6. دمج المستشعرات مع الدوال لتنفيذ مهام معقدة في بيئة المحاكاة.
7. حفظ واستدعاء المشاريع من منصة البرمجة.

ملخص محتوى الدروس

الدرس الأول: تفاعل الروبوت مع الالوان

يتعرف المتعلمين على مستشعر الألوان، وكيفية قراءة بياناته واستخدامها للتحكم في حركة الروبوت عند التعرف على ألوان أو مستويات إضاءة محددة.

الدرس الثاني: انعطاف ذكي بعيون Gyroscope

يكتشف المتعلمين كيفية برمجة الروبوت باستخدام مستشعر Gyroscope sensor لتنفيذ انعطافات دقيقة وضبط الاتجاهات أثناء الحركة، مع تصحيح الانحرافات في المسار.

الدرس الثالث: البرمجة الذكية تبدأ بالدوالFunctions

يطبق المتعلمين مفهوم الدوال في تنظيم البرامج، وإنشاء تعليمات قابلة لإعادة الاستخدام، ودمجها مع المستشعرات لتنفيذ مهام متعددة بكفاءة ودقة.

استراتيجية التدريس

- ✓ التعلم القائم على المشروع (Project-Based Learning) .
- ✓ التعلم بالاكتشاف والتجريب في بيئة آمنة.
- ✓ التدرج من السهل إلى الصعب في تقديم المفاهيم البرمجية.
- ✓ الدمج بين الشرح النظري والتطبيق العملي على المحاكي.

قيم ومهارات مدمجة

- ✓ التفكير الحاسوبي وحل المشكلات.
- ✓ الإبداع في تصميم المشاريع البرمجية.
- ✓ الدقة والانتباه للتفاصيل.
- ✓ تنظيم العمل وتقليل الأخطاء عبر كتابة أكواد منظمة.

التقويم المستمر

تم إعداد مقياس Rubric لتقويم أداء متعلمي الصف الثامن في دروس الروبوت، والتي تركز على التعمق في توظيف المستشعرات (الألوان وGyroscope sensor) وتطبيق مبادئ التحكم التناسبي، إضافة إلى إدخال مفهوم الدوال لتنظيم الكود وتحقيق كفاءة أعلى في البرمجة. ويهدف إلى تمكين المعلمين من متابعة إتقان المتعلمين لمهارات برمجية متقدمة، وتوظيفها في مشروعات عملية تعكس قدرتهم على التفكير المنظم وحل المشكلات.

الدرس الأول

تفاعل الروبوت مع الألوان

المقدمة

في هذا الدرس يتعرف المتعلم على كيفية استخدام مستشعر الألوان لتمييز الألوان المختلفة والتحكم في سلوك الروبوت بناءً عليها. سيتعلم المتعلم كيفية برمجة الروبوت للاستجابة للألوان المحددة في بيئة المحاكاة، مثل التوقف أو تغيير الاتجاه أو إصدار إشارة، مما يتيح له تنفيذ مهام تتطلب التفاعل مع إشارات أو علامات لونية. يركز الدرس على ربط بيانات المستشعر بالقرارات المنطقية، واستخدام البرمجة الشرطية لتنظيم سلوك الروبوت بشكل ديناميكي.

خطة الدرس

الحصة	الموضوع	النشاط
الأولى	قراءة بيانات مستشعر الألوان	عرض اللون المقروء على شاشة الروبوت أثناء الحركة
الثانية	برمجة الروبوت للاستجابة للألوان	تصميم برنامج يغير سلوك الروبوت حسب اللون المكتشف في المحاكي
المشروع		مشروع روبوت1 صـ 198 من الكتاب المدرسي مدينة الكويت
QR Code لفيديوهات شرح لفيديوهات شرح وأنشطة تعلمية		ملفات أوراق العمل ملفات مصادر التعلم

نتاجات التعلم

1. التعرف على مستشعر الألوان، ووظيفته في التفاعل مع بيئة الروبوت.
2. التعرف على كتلة مستشعر الألوان وأنماطه.
3. برمجة الروبوت للتحرك بشكل تكراري حتى يكتشف لوناً معيناً.
4. استخدام الكتلة البرمجية «wait until» للتحكم في سلوك الروبوت.
5. توظيف عبارة التحكم الشرطي (if - else) لتحديد استجابات مختلفة حسب اللون المكتشف.
6. التحقق من دقة استجابة الروبوت عند اكتشاف اللون المستهدف في بيئة المحاكاة.
7. تطبيق البرمجة المعلّقة لإيقاف حركة الروبوت تلقائياً عند تحقق شرط لوني محدد.
8. تحسين استجابة الروبوت بناءً على التغذية الراجعة أثناء الاختبار والتعديل.

المصطلحات التقنية

- Color Sensor: مستشعر يلتقط اللون السطحي ويعيد قيمته إلى البرنامج.
- Color ID: القيمة الرقمية التي تمثل لونًا محددًا في النظام.
- Color-based Control: أسلوب برمجة يعتمد على اللون لاتخاذ القرار.
- Signal Color: لون يُستخدم كإشارة أو علامة لتنفيذ إجراء معين.
- Color Mapping: مطابقة بين القيم اللونية والأوامر البرمجية.

ملاحظات للمعلم

- وضّح للمتعلمين أن الألوان في المحاكاة قد تختلف قليلًا عن الألوان الحقيقية، لذا يجب الاعتماد على القيم المحددة مسبقًا.
- شجعهم على تجربة ألوان إضافية غير الموجودة في المهمة لمعرفة استجابة الروبوت.
- أظهر لهم كيف يمكن لمستشعر الألوان أن يحاكي إشارات المرور أو العلامات الأرضية.
- اقترح عليهم إضافة أكثر من إجراء للون الواحد لتوسيع فكرة التفاعل.

الأنشطة النظرية والعملية

نشاط – نظري

1. ناقش مع المتعلمين كيف يمكن للألوان أن تستخدم كإشارات تحكم في أنظمة مختلفة.
2. اشرح أهمية تحديد ألوان واضحة ومميزة لضمان دقة الاستشعار.

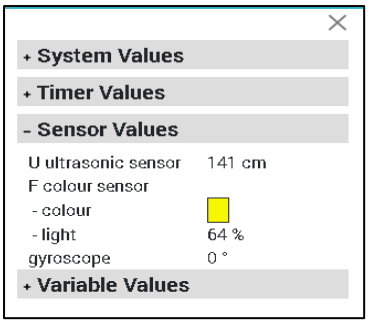
نشاط – عملي

برمج الروبوت بحيث يتوقف عند اللون الأحمر، ويدور عند الأزرق، ويزيد السرعة عند الأخضر

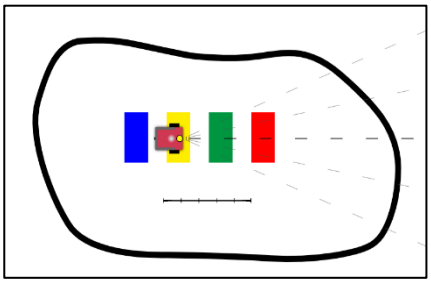
أنشطة الكتاب

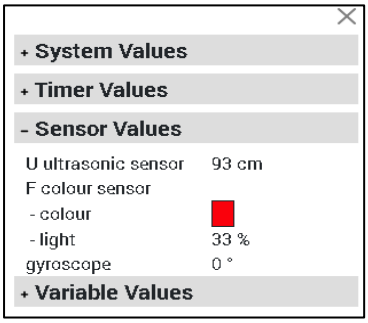
نشاط صـ 196 ـ

سجل اللون المستكشف وشدة الإضاءة الظاهرة أمامك في نافذه عرض بيانات المستشعر

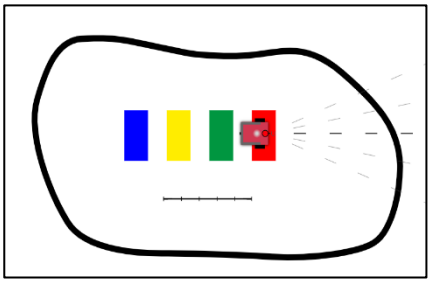


اللون: الأصفر
شدّة الإضاءة: 64%



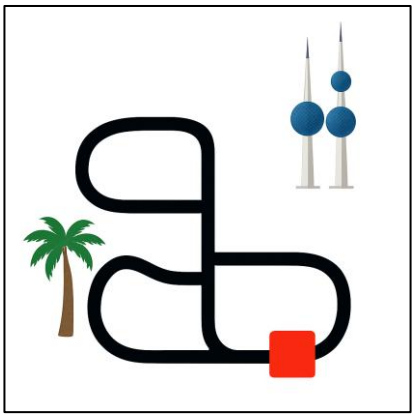


اللون: الأحمر
شدّة الإضاءة: 33%



مشروع روبوت1 صـ 198 من الكتاب المدرسي

مدينة الكويت



التحكم في حركة الروبوت بناءً على قراءات **حساس الألوان** (Colour Sensor) المثبت في المنفذ **F**، وآلية عمله كالآتي:

1. **التكرار المستمر(repeat indefinitely)**

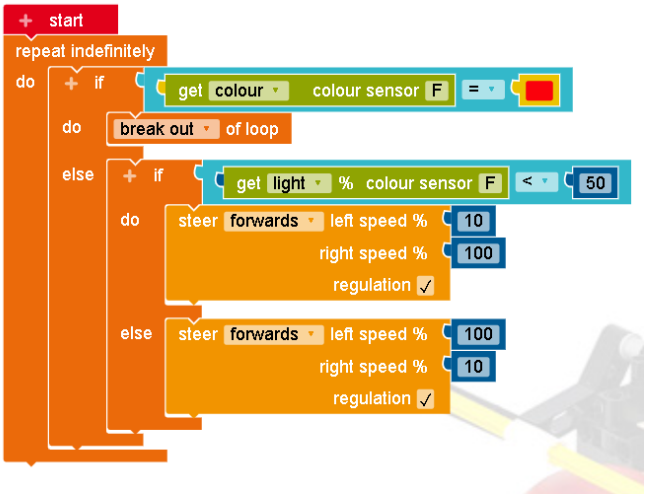
البرنامج يعمل في حلقة لا نهائية حتى يتحقق شرط التوقف.

2. **التحقق من اللون**

إذا اكتشف حساس الألوان اللون **الأحمر** (Red)

يتم **الخروج من الحلقة** (breaK out of loop) أي إيقاف البرنامج أو الانتقال لما بعد الحلقة.

3. **التحقق من شدة الإضاءة**



- إذا لم يكن اللون أحمر، يتحقق البرنامج من شدة الضوء التي يقيسها حساس الألوان.
- إذا كانت نسبة انعكاس الضوء (أقل من 50 %) الروبوت يتجه للأمام مع:
  - السرعة اليسرى: 10%
  - السرعة اليمنى: 100% (أي أنه سيدور قليلاً نحو اليسار).

4. **في الحالة الأخرى انعكاس** (الضوء ≥ 50%) الروبوت يتجه للأمام مع:

-السرعة اليسرى: 100%

-السرعة اليمنى: 10%

(أي أنه سيدور قليلاً نحو اليمين).

النتيجة النهائية

- الروبوت يتحرك للأمام مع تعديل اتجاهه بناءً على شدة الإضاءة التي يكتشفها.
- إذا اكتشف اللون الأحمر، سيتوقف فورًا عن هذه الحركة ويخرج من الحلقة.

التحدي

إضافة كتلة steer لمسافة 360 قبل كتلة breaK out of loop.

تقويم مقترح

1. ما وظيفة مستشعر الألوان في الروبوت؟
2. كيف يمكن استخدام اللون كإشارة تحكم في البرنامج؟
3. ما الفائدة من إضافة أكثر من إجراء للون الواحد؟

تقويم مستمر مقترح لنتاجات التعلم وفق مقياس Rubric

المعيار	ممتاز(4)	جيد(3)	مقبول(2)	يحتاج لتحسين(1)
فهم مستشعر الألوان	يشرح آلية عمل المستشعر وأنماط استخدامه بدقة.	يشرح المستشعر بشكل عام مع بعض النقص.	يذكر وظيفة المستشعر فقط.	لا يوضح المستشعر.
برمجة الروبوت للتفاعل مع الألوان	يوظف كتل البرمجة بفاعلية لتمييز الألوان والاستجابة لها.	يوظف الكتل مع بعض الأخطاء.	يوظفها بشكل جزئي وغير دقيق.	لم يوظفها.
تطبيق التكرار والشروط (if–else)	يدمجها بكفاءة لضبط سلوك الروبوت.	يستخدمها مع بعض الأخطاء.	استخدام جزئي وغير مؤثر.	لم يستخدمها.
نجاح المشاريع (تتبع الخط/خريطة الكويت)	الروبوت ينفذ المهمة كاملة وبنجاح.	الروبوت ينفذ المهمة مع بعض الانحرافات.	الروبوت ينفذ جزءًا من المهمة.	الروبوت لم ينجز المهمة.

الدرس الثاني

انعطاف ذكي بعيون Gyroscope

المقدمة

في هذا الدرس يتعرف المتعلم على كيفية استخدام مستشعر Gyroscope sensor لقياس زاوية دوران الروبوت بدقة، مما يتيح تنفيذ حركات انعطاف متحكم بها بدقة عالية في بيئة المحاكاة. Gyroscope sensor هو جهاز استشعار يقيس سرعة واتجاه الدوران، ويتيح للروبوت معرفة الزاوية التي استدارها حتى يتمكن من تعديل حركته أو التوقف عند زاوية محددة. سيتعلم المتعلم برمجة الروبوت للاستدارة بزوايا محددة أو الحفاظ على اتجاه ثابت، مما يعزز دقة المسارات في المهام المعقدة.

خطة الدرس

الحصة	الموضوع	النشاط
الأولى	قراءة بيانات	عرض زاوية الانعطاف على شاشة الروبوت أثناء الدوران
الثانية	Gyroscope sensor	عرض زاوية الانعطاف على شاشة الروبوت أثناء الدوران
المشروع		مشروع روبوت2 صـ 214 من الكتاب المدرسي زوايا دقيقة بعيون Gyroscope
QR Code لفيديوهات شرح وأنشطة تعلمية		ملفات أوراق العمل ملفات مصادر التعلم

نتاجات التعلم

1. التعرف على Gyroscope sensor ، واكتشاف وظيفته في قياس الدوران والاتجاه.
2. تحديد زاوية الدوران واتجاهه باستخدام Gyroscope sensor داخل بيئة المحاكاة.
3. اكتشاف حركة الدوران في الروبوت من خلال قراءات Gyroscope sensor .
4. حساب مدى دوران الروبوت بدقة بالاعتماد على قراءات المستشعر Gyro .
5. برمجة الروبوت للدوران بزاوية محددة بناءً على بيانات Gyroscope sensor .
6. تحسين دقة الدوران في حركة الروبوت من خلال التجريب والتعديل.
7. تقدير أهمية Gyroscope sensor في تحسين التحكم بالحركة الدقيقة للروبوت.

المصطلحات التقنية

- Gyroscope Sensor: مستشعر يقيس سرعة وزاوية الدوران.
- Yaw Angle: الزاوية حول المحور الرأسي التي يكتشفها الجيروسكوب.
- Angle Calibration: إعادة ضبط الزاوية لضمان قراءة دقيقة.
- Rotational Control: التحكم في حركة الدوران باستخدام بيانات الاستشعار.
- Turn Precision: دقة الانعطاف المحققة في الحركة البرمجية.

ملاحظات للمعلم

- بيّن للمتعلمين أن Gyroscope sensor يقيس الزاوية التراكمية للدوران وليس المسافة.
- وضّح الفرق بين الدوران بزمن محدد والدوران بقياس زاوية محددة.
- شجع المتعلمين على معايرة المستشعر قبل بدء التجربة للحصول على نتائج دقيقة.
- اقترح اختبار البرنامج بزوايا مختلفة للتأكد من دقة الأداء في المحاكاة.

الأنشطة النظرية والعملية

نشاط – نظري

1. ناقش كيف يمكن للجايروسكوب تحسين دقة تحركات الروبوت مقارنة بالدوران المعتمد على الوقت.
2. اشرح أهمية معايرة المستشعر قبل تنفيذ المهمة.

نشاط – عملي

برمج الروبوت لينعطف 90° يمينًا، ثم يتقدم، ثم ينعطف 45° يسارًا قبل إكمال المسار.

أنشطة الكتاب

صـ 210 ـ

قارن بين قراءات مستشعر Gyro و الزاوية المثالية (0) للتأكد من سير الروبوت بخط مستقيم احسب الخطأ ثم استخدم قيمة Kp = 0.3لحساب مقدار التصحيح المطلوب.

الزاوية الحالية من المستشعر	الخطأ = 0 – الزاوية الحالية	التصحيح = الخطأ * Kp	النتيجة
-20	0 – -20 = 20	20*0.3= 6	يجب الانعطاف إلى اليمين لتصحيح المسار
-10	0 – -10 = 10	10*0.3= 3	يجب الانعطاف إلى اليمين لتصحيح المسار
5	0 – 5 = -5	-5 * 0.3 = -1.5	يجب الانعطاف إلى اليسار لتصحيح المسار
15	0 – 15 = -15	-15 * 0.3 = -4.5	يجب الانعطاف إلى اليسار لتصحيح المسار

مشروعروبوت2 صـ 214 من الكتاب المدرسي

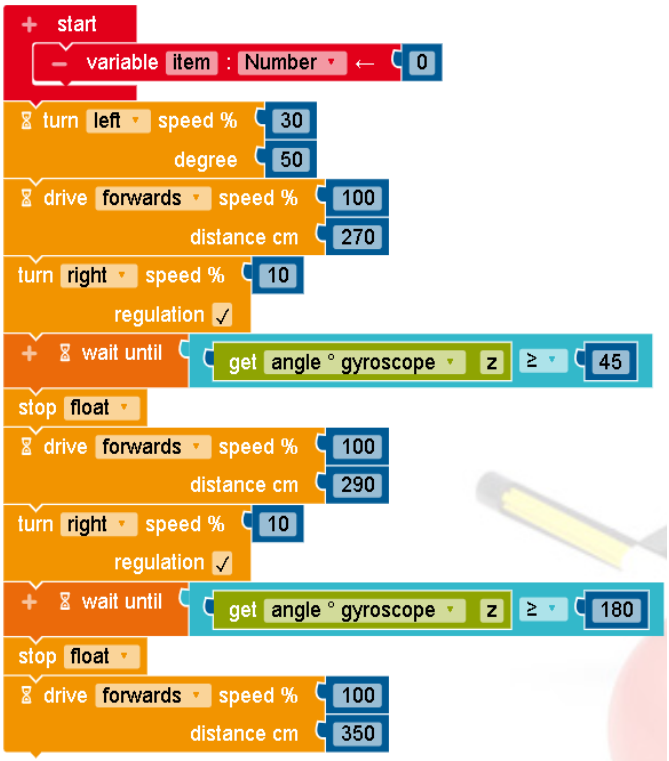
زوايا دقيقة بعيون Gyroscope

يتحرك الروبوت في مسار مثلث باستخدام حساس الجيروسكوب لضبط الزوايا بدقة .

إنشاء متغير

المتغير item من نوع Number يبدأ بقيمة 0.

1. **الدوران جهة اليسار**
   - الروبوت يلتف لليسار بسرعة 30% بمقدار 50 °.
2. **التحرك للأمام (الضلع الأول)**
   - سرعة 100% لمسافة 270 سم.
3. **الانعطاف لليمين ببطء**
   - سرعة 10%، حتى يبدأ الجيروسكوب بقياس زاوية الدوران.
4. **انتظار حتى تصل الزاوية Z إلى 45°**
   - باستخدام الجيروسكوب يتم التأكد من اكتمال الدوران المطلوب بدقة.
5. **التوقف**
6. **التحرك للأمام (الضلع الثاني)**
   - سرعة 100% لمسافة 290 سم.
7. **الانعطاف لليمين ببطء مرة أخرى**
8. **انتظار حتى تصل الزاوية Z إلى 180°**



9. **التوقف**
10. **التحرك للأمام (الضلع الثالث)**
    - سرعة 100% لمسافة 350 سم، ليكمل المثلث ويعود لنقطة البداية تقريبًا.

النتيجة النهائية:

البرنامج يجعل الروبوت يسير في شكل **مثلث** مع ضبط الزوايا بدقة بواسطة الجيروسكوب، لضمان أن المسار منتظم حتى إذا كان هناك انزلاق أو خطأ في العجلات.

التحدي

أضف صوتًا أو رسالة نصية تظهر عند زيارة كل معلم باستخدام شاشة الروبوت.

ج: أضف كتلة إضافة نص بعد كل دوران.

تقويم مقترح

1. ما وظيفة الجيروسكوب في الروبوت؟
2. كيف يختلف الدوران بزمن عن الدوران حتى زاوية معينة؟
3. لماذا تعتبر معايرة المستشعر خطوة مهمة؟

تقويم مستمر مقترح لنتاجات التعلم وفق مقياس Rubric

المعيار	ممتاز(4)	جيد(3)	مقبول(2)	يحتاج لتحسين(1)
فهم مستشعر الجيروسكوب	يوضح دوره في قياس الزوايا والاتجاهات بدقة.	يشرح المستشعر مع بعض النقص.	يذكر وظيفته فقط.	لا يوضح المستشعر.
برمجة الدوران باستخدام Gyroscope sensor	الروبوت يدور بدقة ويحقق الزوايا المطلوبة.	الروبوت يدور مع انحراف بسيط.	الروبوت يدور جزئيًا أو بشكل غير دقيق.	الروبوت لا يدور كما هو مطلوب.
تطبيق التحكم التناسبي (P-Control)	يطبقه بكفاءة لتصحيح المسار.	يطبقه مع بعض الأخطاء البسيطة.	يطبقه بشكل جزئي.	لم يطبقه.
مشاريع التطبيق (المثلث/خريطة الكويت)	الروبوت يكمل المشروع بدقة مع إضافات (صوت/نص).	الروبوت يكمل المشروع مع بعض النواقص.	الروبوت ينفذ جزءًا من المشروع.	المشروع غير منجز.

الدرس الثالث

البرمجة الذكية تبدأ بالدوال Functions

المقدمة

يتعلّم المتعلمون في هذا الدرس كيف يمكن للدوال أن تجعل البرامج أكثر تنظيمًا ووضوحًا. فبدلًا من إعادة كتابة نفس الأوامر مرارًا، يمكن جمعها تحت اسم واحد واستدعاؤها عند الحاجة. هذا المبدأ يساعد على تقليل الأخطاء، وتبسيط التعديل، وربط السلوكيات المختلفة بالروبوت بطريقة أكثر ذكاءً ومرونة.

خطة الدرس

الحصة	الموضوع	النشاط
الأولى	التعرف على مفهوم الدوال	نقاش حول التكرار في الحياة اليومية + إنشاء دالة بسيطة
الثانية	استخدام الدوال في الروبوت	برمجة الروبوت لتفادي العوائق والعثور على الهدف باستخدام دوال
المشروع		مشروع روبوت 3 صـ 228 من الكتاب المدرسي المستكشف الذكي
QR Code لفيديوهات شرح وأنشطة تعلمية		ملفات أوراق العمل ملفات مصادر التعلم

نتاجات التعلم

1. تفسير مفهوم الدالة Function في البرمجة بلغة مبسطة وواضحة.
2. إنشاء دالة بسيطة لأداء سلوك محدد للروبوت.
3. توظيف الدوال مع الشروط والمستشعرات للتحكم في سلوك الروبوت بشكل فعال.
4. المقارنة بين المقاطع البرمجية التي تستخدم الدوال، وتلك التي لا تستخدمها من حيث الكفاءة والتنظيم.
5. تطبيق استخدام الدوال Function في سيناريو واقعي داخل المحاكي لتنفيذ مهمة موجهة للروبوت.

المصطلحات التقنية

- الدالة (Function) : وحدة برمجية تحمل مجموعة أوامر تحت اسم واحد.
- Void Function: دالة تنفذ مهمة دون إرجاع قيمة.

- Return Function: دالة تعيد ناتجًا (عددًا أو نصًا أو قيمة منطقية).
- إعادة الاستخدام Reuse: خاصية تسمح باستدعاء نفس الكود عدة مرات لتقليل الأخطاء.

ملاحظات للمعلم

- استعن بأمثلة حياتية (مثل تشغيل المصباح) لتبسيط مفهوم الدوال.
- وجّه الطلاب لمقارنة الكود مع وبدون دوال لإبراز أهمية التنظيم.
- ركّز على التفرقة بين الدوال الراجعة وغير الراجعة عبر أنشطة عملية.
- شجع المتعلمين على تسجيل نتائج التجارب ليتابعوا تطور أدائهم.

الأنشطة النظرية والعملية

نشاط - نظري

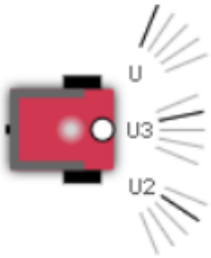
1. ناقش ما الذي يحدث عند تكرار نفس الأوامر في البرنامج؟
2. وضّح كيف تساعد الدوال في تقليل التكرار وتنظيم الكود.

نشاط - عملي

1. أنشئ دالة تجعل الروبوت يتجاوز الزوايا باستخدام المستشعرات.
2. شغّل الروبوت في بيئة المحاكاة ولاحظ كيف تستدعى الدوال أثناء التنفيذ.

أنشطة الكتاب

نشاط صـ 225 ـ حدّد هل يجب أن ينعطف الروبوت يمينًا أم يسارًا؟



U1	25
U2	40
U3	10
الاتجاه المناسب	يتجه نحو اليسار
السبب	المحرك الأيمن (U2) أسرع من الأيسر (U1)

U1	35
U2	20
U3	8
الاتجاه المناسب	يتجه نحو اليمين
السبب	المحرك الأيسر (U1) أسرع من الأيمن(U2)

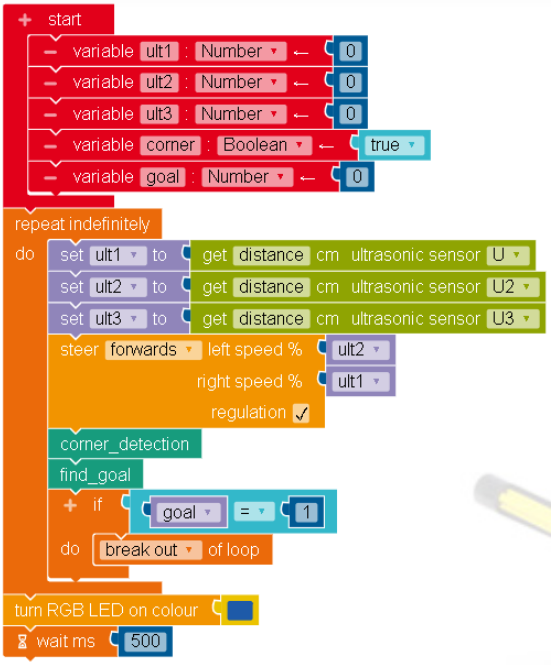
U1	25
U2	25
U3	12
الاتجاه المناسب	إلى الأمام بخط مستقيم
السبب	سرعة المحركين متساويتين

مشروع روبوت 3 صـ 228 ـ من الكتاب المدرسي

المستكشف الذكي

1. **تهيئة المتغيرات(Variables)**

- متغيرات عددية ult1, ult2, ult3 تخزن قيم المسافات المقروءة من المستشعرات U1 و U2 و U3 .
- corner متغير منطقي (Boolean) يستخدم لاكتشاف الزوايا.
- goal متغير عددي يحدد هل وصل الروبوت إلى الهدف أم لا.



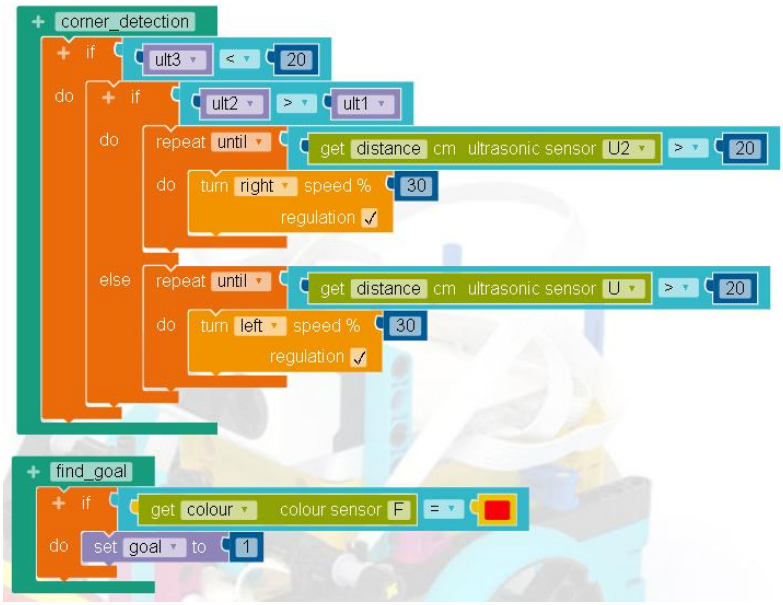
2. **الحلقة التكرارية(repeat indefinitely)**

الروبوت يكرر الأوامر باستمرار:

- يقرأ المسافات من المستشعرات الثلاثة ويخزنها في ult1 ult2 ult3 .
- يتحرك للأمام (steer forwards) مع ضبط سرعة المحركات:
  - المحرك الأيسر = ult2
  - المحرك الأيمن = ult1

اتجاه الروبوت يتحدد حسب الفرق بين المسافتين يمين/يسار.

- يستدعي دالتين فرعيتين:
  - (**corner_detection**) اكتشاف الزوايا.
  - (**find_goal**) البحث عن الهدف.
- إذا كان المتغير goal (goal = 1) هذا يعني أن الروبوت وجد الهدف النتيجة يخرج الروبوت من الحلقة (يتوقف البرنامج).
- يشغل إضاءة RGB بلون أزرق كل 500 مللي ثانية.



3. **دالة (corner_detection) اكتشاف الزوايا**
   - إذا كان المستشعر الأمامي U3> 20 سم (أمامه حاجز قريب):
     - يقارن بين U2 (يمين) و U1 (يسار):
     - إذا كان اليمين أوسع(U2 > U1)
     - يستدير الروبوت يمينًا حتى تصبح المسافة جهة اليمين > 20 سم.
     - وإلا:
     - يستدير الروبوت يسارًا حتى تصبح المسافة جهة اليسار > 20 سم.

(عندما يواجه الروبوت جدار أمامه، يختار الطريق الأوسع (يمين أو يسار)).

4. **دالة find_goal (البحث عن الهدف)**
   - إذا قرأ مستشعر اللون (Red) اللون الأحمر.
     - يغير قيمة goal إلى 1 إشارة إلى أنه وصل للهدف.

النتيجة

الروبوت يمشي للأمام مع تعديل اتجاهه حسب المسافات من الجانبين.

- ✓ إذا واجه عائق أمامه (U3 قريب) يقرر الروبوت الاستدارة جهة اليمين أو اليسار حسب المسافة الأكبر.
- ✓ يظل يبحث حتى يجد الهدف (لون أحمر) عند العثور على الهدف يوقف البرنامج.

تقويم مقترح

1. ما المقصود بالدالة (Function) في البرمجة؟
2. ما الفرق بين الدالة الراجعة (Return Function) والدالة غير الراجعة (Void Function)؟
3. لماذا يساعد استخدام الدوال في جعل الكود أكثر تنظيمًا وأقل عرضة للأخطاء؟

تقويم مستمر مقترح لنتاجات التعلم وفق مقياس Rubric

المعيار	ممتاز(4)	جيد(3)	مقبول(2)	يحتاج لتحسين(1)
فهم مفهوم الدوال (Functions)	يشرحها ويميزها بوضوح عن التكرار.	يشرحها بشكل عام مع بعض النواقص.	يذكرها دون توضيح كافٍ.	لا يوضحها.
تصميم واستخدام الدوال	يصمم دوال مناسبة ويوظفها بكفاءة في الكود.	يصمم دوال مع بعض الأخطاء.	يصمم دوال جزئية أو غير فعّالة.	لم يستخدم الدوال.
دمج الدوال مع المستشعرات والشروط	يوظفها بفاعلية لحلول متقدمة.	يوظفها مع بعض الأخطاء.	يوظفها بشكل محدود وغير دقيق.	لم يوظفها.
مشاريع التطبيق (المتاهة/رسم الأشكال)	المشروع مكتمل وفعّال (تجنب عوائق – الوصول للهدف).	المشروع مكتمل مع بعض الأخطاء.	المشروع جزئي.	المشروع غير منجز.

09

دروس الصف التاسع

وصف الوحدة التعليمية

عنوان الوحدة	إستراتيجيات التحكم الذكي بالروبوتات		
الدروس	الروبوت يعرف طريقه دائما	الروبوت المصارع	الروبوت الحارس الذكي
عدد الحصص	2	2	2
المشروع	مشروع روبوت 1 ص 198	مشروع روبوت 2 ص 213	مشروع روبوت 3 ص 232

فكرة الوحدة

تركز هذه الوحدة على تنمية مهارات المتعلمين في تصميم استراتيجيات تحكم ذكية للروبوتات باستخدام بيئة المحاكاة Open Roberta Lab ينتقل المتعلمون من برمجة الروبوت لتتبع المسارات بدقة، مرورًا بتطوير سلوكيات هجومية ودفاعية في بيئة تنافسية، وصولًا إلى بناء برامج معقدة تستخدم القوائم والمنطق لاتخاذ قرارات متعددة وفقًا للظروف المحيطة.

في نهاية هذه الوحدة سيكون المتعلم قادرًا على:

1. برمجة الروبوت لتتبع المسارات باستخدام المستشعرات والتعامل مع الانحرافات.
2. تطبيق استراتيجيات هجومية ودفاعية في محاكاة مسابقة سومو روبوتيك.
3. التعرف على مفهوم القوائم واستخدامها لتخزين البيانات ومعالجتها.
4. توظيف العبارات المنطقية (AND – OR) لاتخاذ قرارات مركبة.
5. استخدام المؤقتات لتنظيم زمن الاستجابة في البرنامج.
6. دمج أكثر من مستشعر في سيناريو واحد لتحقيق سلوكيات ذكية.
7. حفظ واستدعاء المشاريع من منصة البرمجة.

ملخص محتوى الدروس

الدرس الأول: الروبوت يعرف طريقه دائما

يتعلم المتعلمين كيفية برمجة الروبوت لتتبع المسارات بخطوط واضحة، مع تعديل سلوكه للتعامل مع الانحرافات أو العوائق الطارئة.

الدرس الثاني: الروبوت المصارع

يبرمج المتعلمين الروبوت لاكتشاف الخصوم ووضع استراتيجيات للبقاء داخل الحلبة ودفع الخصم خارجها، عبر التحكم في الدفع والتوازن وتجنب الخروج.

الدرس الثالث: الروبوت الحارس الذكي

يتعرف المتعلمين على كيفية استخدام القوائم والمنطق (AND – OR) لتنظيم قرارات الروبوت، مع دمج المؤقتات لزيادة دقة الاستجابة في مواقف متعددة.

استراتيجية التدريس

- ✓ التعلم القائم على المشروع .(Project-Based Learning)
- ✓ التعلم بالاكتشاف والتجريب في بيئة آمنة.
- ✓ التدرج من السهل إلى الصعب في تقديم المفاهيم البرمجية.
- ✓ الدمج بين الشرح النظري والتطبيق العملي على المحاكي.

قيم ومهارات مدمجة

- ✓ التفكير الحاسوبي وحل المشكلات.
- ✓ العمل التعاوني ووضع الاستراتيجيات.
- ✓ الإبداع في تصميم الحلول البرمجية.
- ✓ الدقة والانتباه للتفاصيل.

التقويم المستمر

تم إعداد مقياس Rubric لتقويم أداء متعلمي الصف التاسع في دروس الروبوت، والتي تركز على مفاهيم متقدمة تشمل التحكم التناسبي–التكاملي–التفاضلي (PID) لتتبع الخط، وتطوير استراتيجيات ذكية للهجوم والدفاع في روبوت السومو، إضافة إلى توظيف القوائم والمنطق والمؤقت في برمجة الروبوت الحارس. ويهدف مقياس Rubric إلى تمكين المعلمين من متابعة قدرة المتعلمين على دمج المفاهيم النظرية بالحلول العملية، وتعزيز مهاراتهم في التحليل واتخاذ القرار البرمجي.

الدرس الأول

الروبوت يعرف طريقه دائما

المقدمة

في هذا الدرس يتعرف المتعلم على كيفية جعل الروبوت يتبع خطًا مرسومًا على الأرض بدقة باستخدام مستشعرات اللون وتقنية التحكم PID تتبع الخط ليس مجرد حركة بسيطة، بل هو مثال على كيفية تعامل الروبوتات مع الانحرافات عن المسار الصحيح. من خلال خوارزمية PID يمكن تحقيق توازن بين السرعة والدقة، حيث يتم تصحيح حركة الروبوت باستمرار بناءً على الفارق بين موقعه الحالي والمسار المطلوب.سيتعلم المتعلم كيفية ضبط معاملات P و I و D واكتشاف تأثيرها على أداء الروبوت في بيئة المحاكاة.

خطة الدرس

الحصة	الموضوع	النشاط
الأولى	التعرف على مفهوم الدوال	نقاش حول التكرار في الحياة اليومية + إنشاء دالة بسيطة
الثانية	استخدام الدوال في الروبوت	برمجة الروبوت لتفادي العوائق والعثور على الهدف باستخدام دوال
المشروع		مشروع روبوت 1 صـ 198 من الكتاب المدرسي الروبوت لا يضيع الطريق
QR Code لفيديوهات شرح وأنشطة تعلمية		ملفات أوراق العمل ملفات مصادر التعلم

نتاجات التعلم

1. برمجة روبوت قادر على تتبع مسار محدد.
2. تحليل سلوك الروبوت عند الخروج عن الخط، واقتراح التعديلات اللازمة لتحسين الأداء.
3. التعرف على نظام التحكم PID.
4. تقييم فعالية البرمجة في تمكين الروبوت من متابعة المسار وتحقيق الهدف المطلوب.

المصطلحات التقنية

- **PID Control**: خوارزمية تحكم تعتمد على ثلاثة معاملات (P, I, D) لضبط الحركة بدقة.
- **Error(الخطأ)**: الفرق بين قيمة الهدف والمسار الذي يقرأه المستشعر.
- **Correction(التصحيح)**: قيمة التعديل المضافة أو المطروحة من حركة المحركات لتقليل الخطأ.
- **Proportional (P)**: معامل يحدد سرعة الاستجابة للتغيرات الفورية.
- **Integral (I)**: معامل يعالج تراكم الأخطاء الصغيرة على المدى الطويل.
- **Derivative (D)**: معامل يحد من الزيادة المفاجئة ويحقق استقرارًا أفضل.

ملاحظات للمعلم

- وضّح للمتعلمين أن التحكم بالوقت وحده لا يكفي لتتبع خط بدقة.
- نبّه أن تغيير معاملات PID بشكل عشوائي قد يؤدي إلى اهتزاز أو بطء زائد.
- شجع المتعلمين على تسجيل ملاحظاتهم عند تغيير كل معامل ومقارنة النتائج.
- اقترح استخدام السرعات المنخفضة في البداية لمراقبة التصحيح بوضوح.

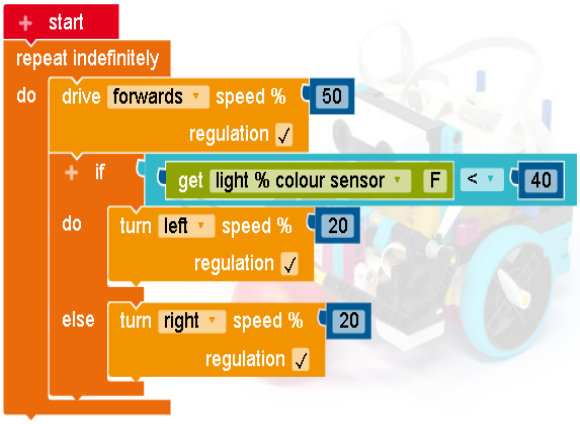
الأنشطة النظرية والعملية

نشاط – نظري

- ناقش مع المتعلمين ما الذي يحدث إذا انحرف الروبوت قليلاً عن الخط؟
- اشرح كيف تساعد خوارزمية PID في إعادة الروبوت إلى مساره.

نشاط – عملي

- برمج الروبوت ليتحرك على خط أسود بخلفية بيضاء، مع تطبيق التحكم التناسبي (P) فقط.
- أضف معاملات I و D تدريجيًا ولاحظ الفرق في الأداء.
- أنشطة الكتاب



نشاط ص 185 ـ

1. كيف تقيم أداء الروبوت؟

   الأداء مقبول لكنه ليس دقيقًا جدًا، لأن القرار يعتمد فقط على قيمة الضوء وليس على معايرة أو دمج مستشعرات أخرى.

2. ما المشكلة في المقطع البرمجي؟

   الشرط يعتمد على قراءة مستشعر اللون فقط دون تحديد واضح للون أو البيئة، قد يخطئ الروبوت إذا تغيرت الإضاءة أو الأرضية.

3. كيف يصحح الروبوت حركته بدقة؟

   عبر حساب الفرق بين القراءات (قيمة الخطأ) وتطبيق تصحيح نسبي (مثل التصحيح = الخطأ *Kp)

نشاط ص 188 ـ

قارن بين قراءات مستشعر اللون (حالة Light mood) وخط المنتصف (القيمة المرجعية = 50).
احسب الخطأ ثم استخدم قيمة Kp = 0.3 لحساب مقدار التصحيح المطلوب.

قراءة المستشعر	الخطأ = القراءة - 50	التصحيح = الخطأ * Kp	النتيجة
30	20 = 30 – 50	6 = 0.3 * 20-	يجب الانعطاف إلى اليمين لتصحيح المسار
75	-25 = 75 – 50	-7.5 = 0.3*-25	يجب الانعطاف إلى اليسار لتصحيح المسار
40	10 =40-50	3 = 0.3*10	يجب الانعطاف إلى اليمين لتصحيح المسار

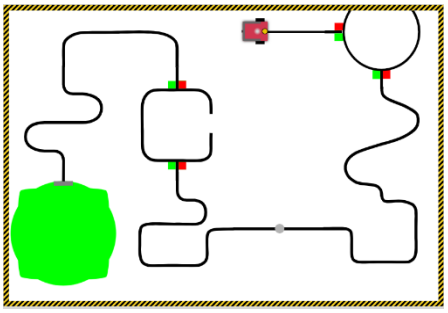
نشاط ص 192 ـ

تخيل أن الخطأ في كل لحظة = 2 (خطأ صغير) وأن معامل التحكم التكاملي KI = 0.1 استكمل الجدول التالي:

الزمن	الخطأ(error)	مجموع الأخطاء(sumErrors)	احسب التحكم التكاملي(I)
1	2	2	0.2 = 0.1*2
2	2	4	0.4
3	2	6	0.6
4	2	8	0.8
5	2	10	1

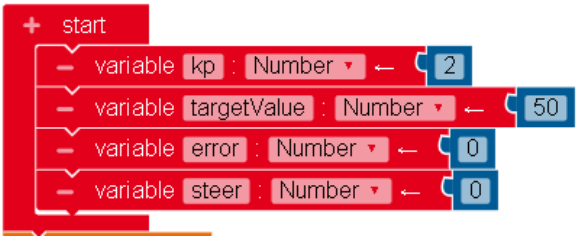
مشروع روبوت 1 صـ 198 ـ من الكتاب المدرسي

الروبوت لا يضيع الطريق



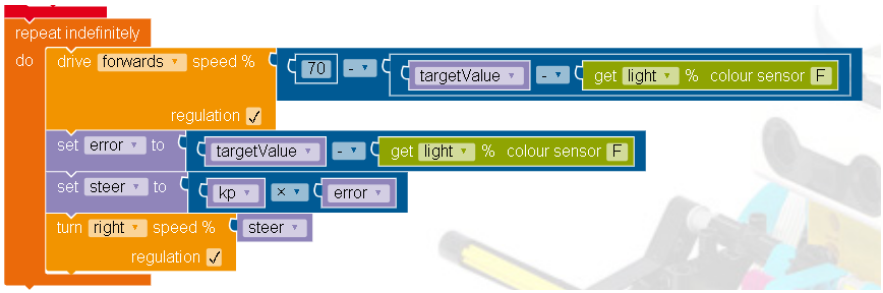
1- تعريف المتغيرات

- **Kp** : معامل التناسب (Proportional Gain) يتحكم بمدى قوة التصحيح عند الانحراف.
- **targetValue**: القيمة المرجعية (المستوى المطلوب من الضوء أي النقطة المتوسطة بين الأسود والأبيض).
- **Error**: متغير لتخزين مقدار الخطأ (الفرق بين القراءة الحالية والقيمة المرجعية).
- **steer**: متغير لتخزين قيمة التوجيه (اتجاه الانعطاف).



ملاحظة: جميع المتغيرات نوعها عددي ولها قيم مبدئية مختلفة يمكن تغييرها.

الحلقة التكرارية (repeat indefinitely)



الروبوت ينفذ ما يلي باستمرار:

التحرك للأمام بسرعة محسوبة

- السرعة الأساسية = 70 – (targetValue – قراءة المستشعر).
- الهدف: جعل السرعة مرتبطة بمدى انحراف الروبوت.

حساب الخطأ

- error = (targetValue - قراءة مستشعر اللون).
- إذا كانت القراءة أقل من (targetValue) إذا الروبوت ابتعد عن اللون الأسود.
- إذا كانت أعلى من (targetValue) ابتعد نحو الأبيض.

حساب قيمة التوجيه

- steer = Kp × error.
- كلما زاد الانحراف (الخطأ) يزداد مقدار الانعطاف.

تصحيح اتجاه الحركة

- الروبوت ينعطف يمينًا بسرعة مقدارها = قيمة steer .
- هذا التصحيح يعيده تدريجيًا إلى الخط.

النتيجة

الروبوت يتحرك على طول الخط الأسود/الأبيض.

- إذا انحرف يمينًا أو يسارًا، فإن **الخطأ error** يزداد.
- البرنامج يحسب **steer** بناءً على هذا الخطأ.
- الروبوت يعدل اتجاهه فورًا بحيث يعود إلى الخط.

التحدي

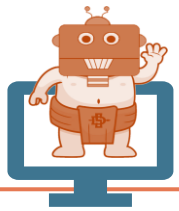
عند الوصول إلى منطقة الهدف (الأخضر) يتوقف الروبوت، يلتف 180°، ثم يعيد استخدام تتبّع الخط نفسه ليعود عكسيًا حتى يرى لون بداية محدّد (مثل الاصفر عند خط البداية).

تقويم مقترح

1. ما وظيفة كل من P و I و D في خوارزمية PID ؟
2. كيف يحدد الروبوت مقدار الانحراف عن الخط؟
3. ما الفرق بين التحكم البسيط بالزمن والتحكم باستخدام PID ؟
4. كيف يمكن تحسين أداء الروبوت إذا كان يهتز كثيرًا عند تتبع الخط؟

تقويم مستمر مقترح لنتاجات التعلم وفق مقياس Rubric

المعيار	ممتاز(4)	جيد(3)	مقبول(2)	يحتاج لتحسين(1)
فهم مفهوم PID	يشرح بوضوح مكونات التحكم (P,I,D) ودورها.	يشرح المكونات مع بعض النقص.	يذكر مكونًا واحدًا فقط دون تفسير كافٍ.	لا يوضح PID.
توظيف المعادلات (Kp, Ki, Kd)	يطبق القيم بدقة ويضبطها للتجربة.	يطبق القيم مع بعض الأخطاء.	يوظف قيمًا جزئية أو عشوائية.	لم يوظفها.
برمجة الروبوت لتتبع الخط	الروبوت يتبع الخط بسلاسة ويتفاعل مع المنعطفات.	الروبوت يتبع الخط مع بعض الانحرافات.	الروبوت يتبع الخط بشكل جزئي.	الروبوت لا يتبع الخط.
التفاعل مع العلامات اللونية (أصفر/أحمر/أخضر)	الروبوت يميز الألوان ويستجيب بدقة.	الروبوت يميز الألوان مع بعض الأخطاء.	الروبوت يميز لونًا واحدًا فقط.	الروبوت لا يميز الألوان.



الدرس الثاني

الروبوت المصارع

استراتيجيات ذكية للبقاء في الحلبة

المقدمة

في هذا الدرس يتعرف المتعلم على مفهوم **روبوت السومو**، حيث يتنافس عدد 2 من الروبوتات داخل حلبة دائرية لمحاولة دفع الخصم خارجها باستخدام البرمجة الذكية والمستشعرات. سيتعلم المتعلم كيفية استخدام مستشعرات مختلفة (المسافة، اللون/الضوء، اللمس) لبناء منطق برمجي يتيح للروبوت اتخاذ قرارات هجومية ودفاعية، مع الحفاظ على وجوده داخل الحلبة. يساعد هذا النشاط المتعلمين على تطوير التفكير المنطقي، وتحليل الاستراتيجيات، وفهم مفاهيم الدفع والاحتكاك والتوازن.

خطة الدرس

الحصة	الموضوع	النشاط
الأولى	**مفهوم روبوت السومو والحساسات المستخدمة**	مناقشة قواعد التحدي، شرح مفهومي الدفع والاحتكاك وأهمية التوازن، والتعرف على دور المستشعرات (المسافة – اللون – اللمس).
الثانية	**برمجة الروبوت في مباراة السومو**	تصميم برنامج يعتمد على المستشعرات للهجوم والدفاع وتجنب الحدود، ثم اختباره في المحاكي ORL وتعديل الاستراتيجيات.
المشروع		مشروع روبوت 2 ص 213 من الكتاب المدرسي البقاء لمن يمتلك استراتيجية
QR Code لفيديوهات شرح وأنشطة تعلمية تعلمية		ملفات أوراق العمل ملفات مصادر التعلم

نتاجات التعلم

1. التعرف على قواعد التحدي داخل حلبة السومو.
2. بناء برنامج روبوت السومو باستخدام منطق تحكم ذكي يشمل (الهجوم، التراجع، التجنب، الالتزام بالحدود).
3. التخطيط لاستراتيجيات هجومية ودفاعية تتناسب مع سيناريوهات تنافسية متنوعة.
4. تحليل المواقف التنافسية داخل الحلبة، واستنتاج السلوك المناسب للروبوت.
5. تعديل الكتل البرمجية وفقًا لملاحظته لسلوك الروبوت أثناء التفاعل مع الخصم وحدود الحلبة.
6. تقييم أداء الروبوت داخل بيئة المحاكاة.
7. تحسين أداء الروبوت بناءً على نتائج التقييم والتجربة.

المصطلحات التقنية

- **Force (الدفع)**: القوة الناتجة عن المحركات لدفع الخصم.
- **Friction (الاحتكاك)**: القوة التي تمنع انزلاق الروبوت وتزيد ثباته.
- **Balance (التوازن)**: توزيع الوزن بشكل متساو للحفاظ على استقرار الروبوت.
- **Sumo Robot Logic**: تسلسل برمجي يجمع بين الهجوم والدفاع وتجنب الخروج.
- **Sumo Strategy**: منطق سلوك تنافسي مبني على أولويات.
- **Arena Boundary (حدود الحلبة)**: الخط الفاصل الذي يجب على الروبوت تجنبه للبقاء في التحدي.
- **State-based Behavior**: ترتيب الحالات لضمان قرار صحيح في كل لحظة.

ملاحظة:

تعريفات الحساسات الأساسية سبق تناولها في صفوف سابقة، لذا نكتفي هنا بدورها داخل الاستراتيجية.

ملاحظات للمعلم

- وضّح للمتعلمين أن الفوز يعتمد على المنطق البرمجي الذكي أكثر من القوة.
- ثبّت أولوية «الحافة أولاً» قبل أي قرار هجومي.
- شجّع المتعلمين على تجربة استراتيجيات هجومية ودفاعية متنوعة.
- ذكّرهم بأهمية اختبار المستشعرات قبل البدء بالمنافسة.
- اقترح عليهم تعديل السرعة أو زاوية الدوران لاكتشاف تأثيرها على الأداء.

الأنشطة النظرية والعملية

نشاط – نظري

- ناقش كيف يؤثر (الدفع والاحتكاك) على أداء الروبوت.
- اشرح دور (التوازن) في بقاء الروبوت داخل الحلبة.
- ناقش ترتيب الأولويات: لماذا «الحافة» قبل «الخصم»؟ أعطِ أمثلة لفشل القرار عند عكس الترتيب.
- صِف حالات خطأ محتملة: خصم سريع جدًا، أو حافة باهتة اللون، وكيف نعالجها بمنطق برمجي

نشاط – عملي

برمجة الروبوت ليقوم بمطاردة خصمه بطريقة أكثر ذكاءً بدلًا من مجرد التقدم للأمام.

مثال:

الروبوت يتحرك في مسار دائري ببطء باحثًا عن الخصم، عند اكتشاف الخصم، يندفع بسرعة كبيرة لمحاولة دفعه.

أنشطة الكتاب

نشاط ص 204 ـ

1- ما وظيفة حساس اللون في هذا التحدي؟

ج: اكتشاف حدود الحلبة

2- كيف يمكن للروبوت أن يعرف مكان خصمه؟

ج: باستخدام مستشعر المسافة Ultrasonic senssor لقياس قرب الأجسام أمامه وتحديد موقع الخصم.

3- إذا اقترب الروبوت من حافة الحلبة، ما السلوك الصحيح الذي يجب أن يقوم به؟

ج: يتراجع للخلف أو يغيّر اتجاهه بسرعة حتى لا يخرج من الحلبة.

4- لماذا يتم الاستعانة بمستشعر اللمس علماً بأنه في محاكي RCJ يتواجد في الجهة الخلفية للروبوت؟

ج: ليحس الروبوت إذا دفعه الخصم من الخلف، فيتفاعل مباشرة إمّا بالتقدم للأمام بسرعة أو بالانعطاف للهجوم المضاد.

نشاط ص 212 ـ

1- افترض أنك تريد استخدام نفس الأوامر في أكثر من شرط (مثل الرجوع للخلف والدوران). ما الفكرة البرمجية التي يمكنك استخدامها لتجنب تكرار المقطع البرمجي؟ وما فائدتها؟

ج: استخدام الدوا (Functions) ، تجعل الكود منظمًا وأكثر وضوحًا، وتمنع تكرار الأوامر، كما تسهّل تعديل السلوك لاحقًا في مكان واحد فقط بدل تعديل كل المقاطع.

2- صمّم شرطًا جديدًا للروبوت باستخدام مستشعر إضافي. ما النموذج؟ وما السلوك المقترح؟

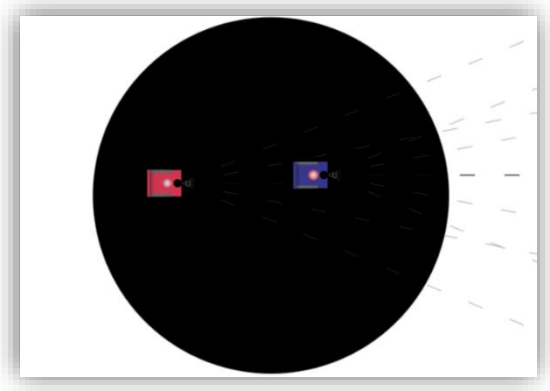
ج: استخدام أكثر من مستشعر الموجات الفوق صوتية وضع واحد في الأمام وآخر على الجانب، إذا اكتشف المستشعر الأمامي أن الخصم قريب جدًا يتحرك الروبوت للأمام بسرعة للهجوم، إذا كان المستشعر الجانبي يقرأ مسافة صغيرة ينحرف الروبوت نحو الخصم من الجانب لمهاجمته.

3- تخيل أن البرنامج سيعمل في مسابقة حقيقية. ما المكون الذي تراه الأكثر أهمية؟ ولماذا؟

ج: الأكثر أهمية هو المستشعرات (مثل الموجات فوق الصوتية أو اللون) لأنها تساعد الروبوت على اكتشاف الخصم أو الحافة بدقة، وبالتالي يتصرف بذكاء ويستطيع الفوز في المباراة

مشروع روبوت 2 ص 213 من الكتاب المدرسي

البقاء لمن يمتلك استراتيجية



الحلقة الأساسية

- البرنامج يبدأ ويعمل بشكل متكرر **بدون توقف**.
- في كل دورة، يتحقق من شروط المستشعرات وينفذ الأوامر المناسبة.

الشرط الأول: مستشعر اللون

إذا اكتشف الروبوت لونًا فاتحًا (عادة الأبيض على حدود الحلبة):

1. يتوقف فورًا (stop braKe).
2. يرجع للخلف بسرعة 100% لمسافة 50 سم.
3. ينعطف لليمين بسرعة 30% بزاوية 90° .

الهدف: تجنّب الخروج من الحلبة

الشرط الثاني: مستشعر المسافة

إذا كان هناك جسم/خصم أمام الروبوت

1. يتقدم للأمام بسرعة 100% مع تنظيم الحركة (regulation)

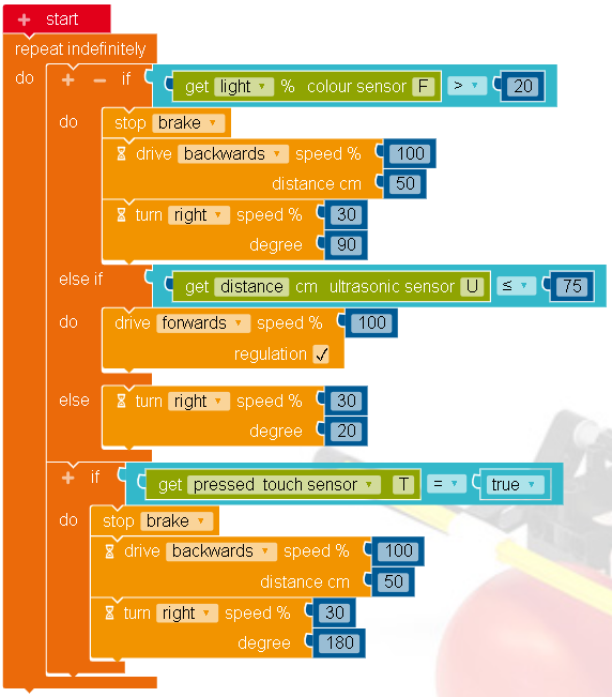
الهدف: الهجوم على الخصم

الشرط الثالث: (إذا لم يتحقق الشرطان السابقان)

إذا لم يرَ الروبوت اللون الأبيض ولم يكتشف خصمًا قريبًا

1. ينعطف لليمين بسرعة 30% بزاوية 20° .

الهدف: البحث عن الخصم داخل الحلبة.



الشرط الرابع: مستشعر اللمس الخلفي

إذا ضغط مستشعر اللمس الخلفي (أي إذا تعرض الروبوت لهجوم من الخلف)

1. يتوقف (stop braKe).
2. يرجع للخلف بسرعة 100% لمسافة 50 سم.
3. ينعطف لليمين بسرعة 30% بزاوية 180° .

الهدف: الدفاع والتخلص من الهجوم من الخصم.

النتيجة

الكود يجعل الروبوت يتصرف مثل مصارع سومو ذكي.

التحدي

برمج الروبوت باستخدام استراتيجيات مختلفة أو بإضافة مستشعرات:

- إضافة أكثر من مستشعر مسافة تمنح الروبوت رؤية أوسع وتسمح له باختيار الاتجاه الأنسب للهجوم أو الدفاع بدقة. يمكن برمجته ليتجه نحو أقرب خصم أو يتفادى الهجوم الجانبي بالانعطاف المناسب.
- إضافة مستشعر Gyroscope sensor تمكّن الروبوت من قياس زاوية الدوران بدقة والحفاظ على اتجاه ثابت أثناء الحركة. يساعد ذلك على تنفيذ الانعطافات الذكية وتجنب الانحراف أثناء الهجوم أو التراجع.
- يمكن برمجة الروبوت ليقوم بمسار بحث دائري بطيء حتى يكتشف الخصم ثم يندفع بسرعة للهجوم. وإذا اقترب من حدود الحلبة يتراجع فورًا ويدور بزاوية عشوائية للعودة بأمان.

تقويم مقترح

1. ما وظيفة مستشعر اللون في الحلبة؟
2. لماذا يُعتبر الاحتكاك عاملاً أساسيًا في أداء الروبوت؟
3. كيف يساعد مستشعر اللمس الخلفي في الدفاع؟
4. ما التحسين البرمجي الذي يمكن أن يرفع من فرص الفوز؟

تقويم مستمر مقترح لنتاجات التعلم وفق مقياس Rubric

المعيار	ممتاز(4)	جيد(3)	مقبول(2)	يحتاج لتحسين(1)
فهم مفهوم السومو وقوانينه	يشرح بدقة الهدف والقوانين (البقاء داخل الحلبة، إخراج الخصم).	يشرح بشكل عام مع بعض النواقص.	يذكر جزءًا من القوانين.	لا يوضح المفهوم.
توظيف المستشعرات (مسافة، لون، لمس)	يوظفها جميعًا بذكاء لتحقيق الهجوم والدفاع.	يوظف معظم المستشعرات مع بعض الأخطاء.	يوظف مستشعرًا واحدًا فقط.	لا يوظف المستشعرات.
بناء استراتيجيات هجومية ودفاعية	يبرمج الروبوت بخطط متنوعة وفعّالة.	يبرمج بخطة واحدة صحيحة.	يبرمج بخطة بسيطة وغير فعّالة.	لا توجد خطة واضحة.
تقييم الأداء وتحسين الكود	يحلل الأداء ويقترح تحسينات واقعية.	يحلل الأداء مع اقتراح محدود.	يقيم الأداء دون اقتراح.	لا يقيم الأداء.

الدرس الثالث

الروبوت الحارس الذكي

المقدمة

في هذا الدرس يتعلم المتعلم كيفية استخدام القوائم البرمجية لتخزين البيانات المتعددة، بالإضافة إلى دمج العبارات المنطقية (AND/OR) في اتخاذ القرارات المركبة. كما يتعرف على دور المؤقت (Timer) في تنظيم الاستجابات البرمجية. من خلال ذلك، سيبرمج المتعلمين روبوتًا يقوم بدور "الحارس" الذي يراقب بيئته، يخزن بيانات المستشعرات، ويصدر إنذارًا عند تحقق شروط معينة.

خطة الدرس

الحصة	الموضوع	النشاط
الأولى	القوائم والعبارات المنطقية	شرح أنواع القوائم (Number, Boolean, String, Colour)، وكيفية استخدام AND/OR لدمج أكثر من شرط.
الثانية	برمجة الروبوت الحارس	تصميم برنامج يخزن بيانات المستشعرات في قائمة، ويطلق إنذارًا عند تحقق الشرط باستخدام المؤقت.
المشروع		مشروع روبوت 2 صـ 232 من الكتاب المدرسي الروبوت الحارس
QR Code لفيديوهات شرح وأنشطة تعلمية		ملفات أوراق العمل ملفات مصادر التعلم

نتاجات التعلم

1. التعرف على القوائم البرمجية، وتحديد دورها في تخزين البيانات.
2. إنشاء قائمة، وإدارة محتواها باستخدام بيئة برمجة بصرية.
3. توظيف العبارات المنطقية (AND — OR) في الشروط البرمجية.
4. التمييز بين وظيفتي AND و OR في اتخاذ القرارات.
5. استخدام المؤقت لتنظيم تسلسل تنفيذ الأوامر.
6. تصميم برنامج لدمج القوائم والمنطق والمؤقت لحل مشكلة محددة.

المصطلحات التقنية

- **List (قائمة)**: بنية بيانات لتخزين مجموعة من العناصر.
- **Boolean Value (قيمة منطقية)**: متغير يأخذ أحد القيمتين (True أو False).
- **Logical AND (و)**: شرط مركب لا يتحقق إلا إذا تحقق الشرطان معًا.
- **Logical OR (أو)**: شرط مركب يتحقق إذا تحقق أحد الشرطين أو كلاهما.
- **Timer (المؤقت)**: أداة لحساب الزمن وضبط الاستجابات البرمجية.

ملاحظات للمعلم

- ركّز على أن القوائم تسمح بتخزين عدة قراءات بدل متغير واحد.
- وضّح أن AND تقلل من احتمال تحقق الشرط، بينما OR تزيد من احتمالاته.
- شجع المتعلمين على استخدام المؤقت لتنظيم الاستجابة بدل الاعتماد على الحلقات فقط.
- اقترح أمثلة من الحياة اليومية (قائمة حضور، شرط دخول، إنذار أمني) لتبسيط المفهوم.

الأنشطة النظرية والعملية

نشاط – نظري

- ناقش مع المتعلمين مثالًا من حياتهم اليومية لقائمة بيانات (مثل قائمة أسماء المتعلمين).
- اطلب منهم التفكير في مواقف تحتاج لاستخدام (AND و OR) في القرارات.
- وضّح كيف يمكن للمؤقت أن يساعد في تنظيم الاستجابات (مثل إضاءة لمدة 5 ثوان).

نشاط – عملي

برمج الروبوت ليقوم بما يلي:

1. يستخدم مستشعر المسافة لقياس المسافة ويضيف القيمة إلى قائمة **Distances**.
2. إذا اكتشف لونًا معينًا (مثلاً الأحمر) باستخدام مستشعر اللون، يخزن القيمة في قائمة **Colours**.
3. إذا تحقق الشرطان معًا (المسافة ≤ 30 سم **AND** اللون = أحمر)، يشغّل إنذارًا صوتيًا ويضيء باللون الأحمر.
4. استخدم **Timer** لضبط زمن الإنذار بحيث يستمر 3 ثوان ثم يتوقف تلقائيًا.

أنشطة الكتاب

نشاط صـ 219 ـ 220ـ

1- ما نوع الهيكل البرمجي الأنسب لتخزين عدة قيم مثل عدد التهديدات التي يكتشفها الروبوت؟

ج: قائمة (List)

2- ما الكتلة المنطقية التي نستخدمها عندما نريد تنفيذ أمر فقط إذا تحقق شرطان معاً؟

ج: كتلة AND

3- ما الهدف من استخدام المؤقت (Timer) في هذا البرنامج؟

ج: تنظيم توقيت تنفيذ الأوامر وتحديد مدة الاستجابة للأحداث.

نشاط صـ 226 ـ

اذا كانت قائمة المسافات هي distance = [80, 60, 50, 40] الحلقة تقوم بالتكرار 4 مرات في كل مرة يتم استخدام كتلة (in list distance get # I) تأمل الجدول التالي وأجب عن الأسئلة:

الدورة(i)	العنصر من القائمة(distance[i])
0	80
1	60
2	50
3	40

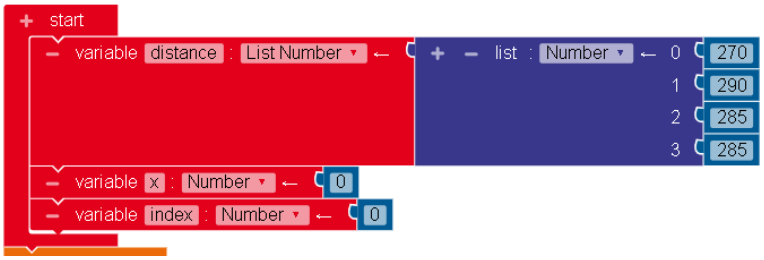
ما الفرق بين كتابة distance[0] و distance[3]؟

distance [0] الفهرس 0 = أول عنصر، distance [3] الفهرس 3 = رابع عنصر، الفرق أن الفهرس [index] يبدأ من 0 وليس من 1.

مشروع روبوت 3 صـ 232 من الكتاب المدرسي

الروبوت الحارس

يعمل مع الروبوت باستخدام **قائمة من القيم** وحساس المسافة.



1) **تعريف المتغيرات**

- **(distance)**: قائمة أعداد تحتوي على أربع قيم 270، 290، 285، 285.
- **(X)**: متغير عددي يبدأ بقيمة (0).
- **(index)**: متغير عددي يبدأ بقيمة (0).

2) **الحلقة (التكرار إلى ما لا نهاية)**

الروبوت يكرر ما يلي باستمرار:

- **تعيين x** : يأخذ الروبوت قيمة من القائمة distance بحسب رقم الموقع index .
- يتحرك الروبوت بسرعة 100% لمسافة مساوية للقيمة المخزنة في x.

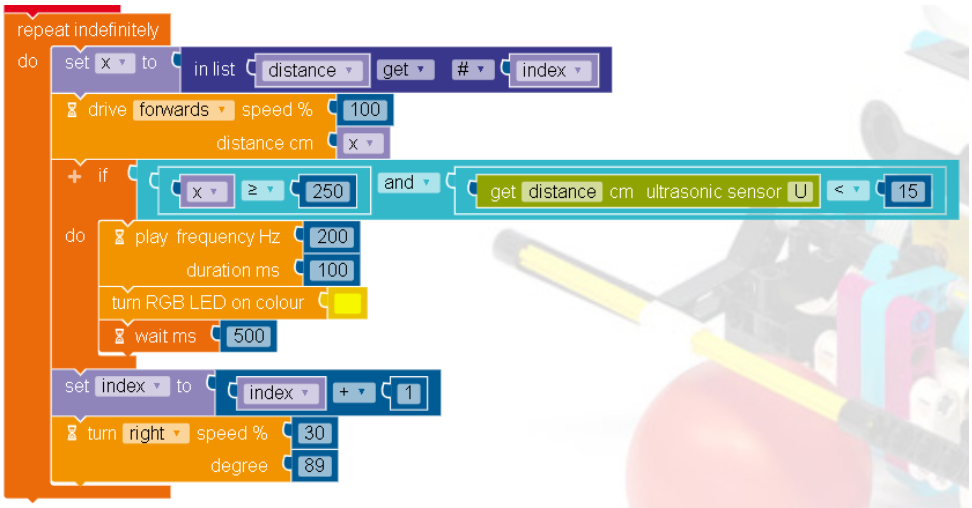
3) **شرط (if):**

إذا كانت قيمة (x ≥ 250) **و** كانت المسافة المقاسة بمستشعر Ultrasonic (أقل من 15سم):

- يشغّل صوت بتردد 200 Hz لمدة 100 ms.
- يضيء الـ LED باللون الأصفر.
- ينتظر 500 ms.

4) **تحديث الفهرس** يزيد قيمة index بمقدار 1.

5) **يستدير الروبوت** يمينًا بسرعة 30% بمقدار 89°.



النتيجة

الكود يجعل الروبوت يسير في مسار مربع أو شبه مربع اعتمادًا على القيم في القائمة، ويعطي إشارة (صوت وضوء أصفر) إذا واجه عائقًا قريبًا وهو يسير لمسافة كبيرة.

التحدي

عدّل برنامج الروبوت الحارس ليعرض ضوءًا بلون مختلف حسب المسافة:

- إذا كانت المسافة **أقل من 20 سم**، يضيء باللون **الأحمر** (خطر قريب).
- إذا كانت المسافة **بين (20 – 50) سم**، يضيء باللون **الأصفر** (تنبيه).
- إذا كانت المسافة **أكثر من 50 سم**، يضيء باللون **الأخضر** (آمن).
- اجعل الإنذار الصوتي يعمل فقط عند تحقق الشرط المركب

(المسافة ≤ 30 AND اللون = أحمر) .

تقويم مقترح

- ما وظيفة **القائمة (List)** في برنامج الروبوت الحارس؟
- ما الفرق بين استخدام **AND** و **OR** في الشرط البرمجي؟ أعطِ مثالًا.
- كيف يساعد المؤقت (Timer) في ضبط زمن الاستجابة للإنذار؟
- ما ميزة دمج القوائم مع المستشعرات مقارنة باستخدام متغير واحد فقط؟

تقويم مستمر مقترح لنتاجات التعلم وفق مقياس Rubric

المعيار	ممتاز(4)	جيد(3)	مقبول(2)	يحتاج لتحسين(1)
فهم القوائم	يعرّف القوائم ويميز أنواعها ويستخدمها عمليًا.	يعرّف القوائم ويستخدمها مع بعض الأخطاء.	يذكر مفهوم القائمة فقط دون تطبيق واضح.	لا يوضح القوائم.
توظيف المنطق (AND – OR)	يوظف العبارات بدقة لحل المشكلات.	يوظفها بشكل صحيح مع بعض النقص.	يوظفها بشكل جزئي أو غير صحيح.	لم يوظفها.
استخدام المؤقت (Timer)	يوظفه بفعالية لتنظيم الأحداث.	يوظفه مع بعض الأخطاء.	يوظفه بشكل جزئي.	لم يستخدمه.
مشروع الروبوت الحارس	المشروع مكتمل ويعكس برمجة ذكية.	المشروع مكتمل مع بعض النواقص.	المشروع ناقص أو غير منظم.	المشروع غير منجز.

10

خطة الدرس

خطة الدرس

أهمية خطة الدرس

يُعدّ تحضير الدروس أحد المكونات الجوهرية التي تعزّز كفاءة المعلم وتسهم في نجاح العملية التعليمية. وتزداد أهمية التحضير عند تدريس مادة الروبوت والبرمجة، خاصة عند تقديم المفاهيم البرمجية باستخدام محاكي منصة Open Roberta Lab، لما تتطلبه هذه المهارات من ترتيب منهجي وتسلسل منطقي. ومن أبرز الجوانب التي يحققها تحضير الدروس ما يلي:

- تنظيم المحتوى وترتيب الأفكار:

يتيح للمعلم تنظيم عناصر الدرس وتحديد تسلسلها، مثل: التقديم النظري للمفاهيم (الحساسات، الحركات، الكتل البرمجية)، ثم الانتقال إلى الأنشطة العملية داخل المحاكي، مما يساعده على توقع المواقف الصفية المختلفة والتعامل معها بثقة ومرونة.

- اختيار الاستراتيجيات والأنشطة المناسبة:

يسهم في تمكين المعلم من اختيار المهام العملية والأنشطة التفاعلية على المحاكي بما يتناسب مع أهداف الدرس ومستوى المتعلمين، مثل التحكم في حركة الروبوت، استخدام الحساسات، أو بناء خوارزميات تعتمد على المنطق (AND، OR).

- توثيق المحتوى وتيسير الرجوع إليه:

يُعدّ مرجعاً مكتوباً يمكن للمعلم الرجوع إليه متى ما احتاج، خاصة عند مراجعة عند مراجعة خطوات بناء برنامج أو إعادة شرح وظيفة كتلة برمجية مثل "التكرار" أو "الشرط".

- تتبّع التقدّم وتوزيع الزمن:

يساعد ا على متابعة ما أنجزه الطلاب من تمارين عملية ومهام محاكية، وتحديد ما تبقى منها، مما يسهل تنظيم الجدول الزمني لإنهاء المنهج في الوقت المحدد.

- أداة للتقييم والمتابعة المهنية:

يُظهر التزام المعلم بالمنهج وتنظيمه، ويُستخدم كمعيار للتقييم الفني والتربوي لأدائه.

- تعزيز الثقة والتمكّن من الأداء:

يمنح التحضير المسبق للإعداد الكتابي لخطة الدرس المعلم قدرة أكبر على التحكم في مجريات الدرس، ويُقلّل من التشتّت أو التردّد، مما ينعكس إيجاباً على أداء المعلم وتفاعل المتعلمين.

- توظيف الموارد التعليمية والتقنية بشكل فعّال:

يتيح التحضير للمعلم أن يخطط لاستخدام الأدوات الرقمية والوسائل التعليمية المناسبة، مثل العروض التقديمية، المشاريع المصممة مسبقًا على Open Roberta Lab، بما يضفي على الدرس طابعًا عمليًا شيقًا ومباشرًا.

- وضع خطط بديلة عند الحاجة:

يمكن الإعداد الكتابي الجيد لخطة الدرس المعلم من الاستعداد بخيارات بديلة عند حدوث ظرف طارئ، مثل ضعف الاتصال بالإنترنت أو تقليص زمن الحصة. في هذه الحالات، يمكن اللجوء إلى أنشطة نظرية أو أوراق عمل مرتبطة بالمفاهيم نفسها لضمان تحقيق أهداف الدرس.

سجل الإعداد الكتابي لخطة الدرس

يجب أن يتضمن سجل الإعداد الكتابي الصفحات التالية:

1. بيانات المعلم الأساسية:
    - الاسم الرباعي.
    - الرقم المدني.
    - الجنسية (اختياري).
    - رقم الملف. اختياري
    - تاريخ التعيين.
    - مركز العمل السابق.
    - المؤهل العلمي.
2. التقويم الدراسي.
3. الجدول الدراسي للمعلم:
    - المواد التي يقوم بتدريسها.
    - الأعمال المسندة إليه (مثل الأنشطة أو المهام الإدارية).
4. خطة توزيع المقرر للفصل الدراسي الحالي.
5. الأهداف التربوية:
    - الهدف الشامل للتربية في دولة الكويت.
    - أهداف المرحلة التعليمية.
    - الأهداف العامة لتدريس مادة الحاسوب.
6. خطط الدروس اليومية بحسب خطة توزيع المنهج المقرر.
7. سجل متابعة المتعلمين اليومي:
    - يمكن أن يُضمَّن في سجل الإعداد الكتابي أو يُرفق كسجل مستقل مع كل حصة.

خطــوات إعداد خطة الــدرس

يمثل إعداد الدرس عملية منظمة تهدف إلى تمكين المعلم من تقديم محتوى علمي متماسك، وتحقيق نواتج التعلم المرجوة بكفاءة وفاعلية. ويُعد الإعداد الجيد ضرورة خاصة في مادة الحاسوب، لما تتطلبه من ترتيب منطقي للمفاهيم، واستعداد تقني مسبق. وتشمل خطوات إعداد الدرس ما يلي:

1. القراءة الدقيقة لموضوع الدرس:

يبدأ الإعداد بمراجعة شاملة للدرس عبر الكتاب المدرسي ودليل المعلم والمواد الإثرائية. يساعد ذلك على بناء فهم عميق للمفاهيم النظرية مثل الحساسات، الحركات، الكتل المنطقية، وربطها بالتطبيق العملي في المحاكي.

2. تحديد أهداف الدرس / نواتج التعلم بدقة ووضوح:

ينبغي تحديد ما يُتوقع من المتعلّمين تحقيقه بنهاية الحصة، سواء من حيث المعرفة (مثل التعرّف على الكتل البرمجية)، أو المهارات (مثل إنشاء مشروع تفاعلي)، أو الاتجاهات (مثل تنمية روح التعاون أو التفكير المنطقي). ويُفضل صياغة الأهداف وفق نموذج سلوكي واضح، قابل للقياس والملاحظة.

3. الإعداد الذهني:

قبل البدء في كتابة خطة الدرس، على المعلم أن يُكوّن تصوراً ذهنياً متكاملاً عن خطوات الشرح، وربطها بالأنشطة التعليمية والوسائل المناسبة. يشمل ذلك التفكير في أسلوب التقديم، وطريقة الربط بين المفاهيم، وكيفية إدارة التفاعل داخل الفصل.

4. الإعداد الكتابي:

بعد تكوّن التصور الكامل لطريقة سير الدرس، يتم **تسجيل الخطة على شكل خطوات واضحة ومحددة**، شاملةً جميع عناصر الإعداد، مع مراعاة عامل الوقت وارتباط كل خطوة بنتاجات التعلم.

5. إعداد المتطلبات والوسائل التعليمية:

يشمل تجهيز الأدوات المطلوبة مسبقاً، مثل:

- جهاز حاسوب لكل متعلم (أو مجموعات).
- الدخول على منصة Open Roberta Lab والتأكد من عمل المحاكي.
- توفير مشاريع جاهزة كنماذج.
- استخدام العروض التقديمية أو السبورة الذكية إن وُجدت.
- كما يُوصى بتجربة الوسائل التقنية مسبقاً لضمان جاهزيتها وتفادي أي أعطال محتملة.

6. التخطيط لأساليب التقويم:

يُعد تحديد أدوات التقويم جزءاً أساسياً من الإعداد، ويمكن أن تشمل:

- أسئلة شفهية أثناء الحصة.
- بطاقات ملاحظة لأداء المتعلمين خلال تنفيذ المهام البرمجية.
- أوراق عمل قصيرة أو تحديات برمجية بسيطة.
- تقويم المشروع النهائي أو المنتج البرمجي.

7. توقّع الفروق الفردية وتهيئة البدائل:

ينبغي على المعلم أن يُراعي اختلاف قدرات المتعلمين، من خلال:

- إعداد أنشطة داعمة للمتعلمين الذين يحتاجون إلى تعزيز.
- توفير مهام إثرائية للمتعلمين المتفوقين (التحدي).
- خطط بديلة عند حدوث خلل تقني أو نقص الوقت (مثل نشاط نظري أو مناقشة جماعية).

8. تضمين جوانب تربوية ومهارية:

يُستحسن أن يتضمن الإعداد عناصر تعزز القيم والمهارات الحياتية، مثل:

- تعزيز العمل الجماعي عند تنفيذ المشاريع.
- تنمية مهارات حل المشكلات.
- تشجيع المتعلمين على التفكير النقدي ومراجعة الأخطاء البرمجية.

أخطاء شائعة يقع فيها المعلم عند إعداد خطة الدرس

1	الإهمال في كتابة التاريخ أو الاكتفاء بكتابة فصل واحد فقط.
2	تجاهل استخدام وسيلة تعليمية أثناء الشرح، رغم الإشارة إليها في بند "مصادر التعلم والوسائل التعليمية".
3	في المقدمة والنشاط الاستهلالي، لا يصح كتابة "مراجعة قصيرة للدرس السابق" بشكل عام، بل يجب تحديد بنود المراجعة بشكل دقيق.
4	عدم ذكر حل التمارين الموجودة في صفحة معينة.
5	عدم ملاءمة أسلوب العرض من حيث عدد النقاط،، إذ يجب ألا تكون قصيرة جداً أو طويلة جداً.
6	عدم تحديد دور المعلم أو دور المتعلم بشكل واضح.
7	اختيار أسئلة تقويمية بصياغة خاطئة، مثل: ○ الأسئلة المركبة التي تطرح أكثر من فكرة في آن واحد. ○ الأسئلة التي تكون إجابتها "نعم / لا" دون تعليل. ○ الأسئلة الغامضة غير محددة الهدف. ○ الأسئلة الموحية بالإجابة.

نمـــوذج خطة درس

عنوان الدرس يصاغ بحسب خطة المقرر الدراسي					
الأسبوع	الوحدة	اليوم	التاريخ	الفصل	الحصة

يتم تسجيلها لكل حصة دراسية

مصادر التعلم والوسائل التعليمية	القيمة التربوية
تُسجّل المصادر التي يستخدمها المعلم أثناء عرض الدرس لإثراء الموقف التعليمي (مواد، أدوات رقمية). لا يوجد عدد معين للمصادر التي يمكن استخدامها في الدرس الواحد، فهذا يتوقف على طبيعة الدروس.	تحديد القيمة التربوية حسب الجدول الزمني المقترح، مع وصف زمني لكيفية تفعيلها خلال الحصة بما يتلاءم مع عرض مفاهيم الدرس.

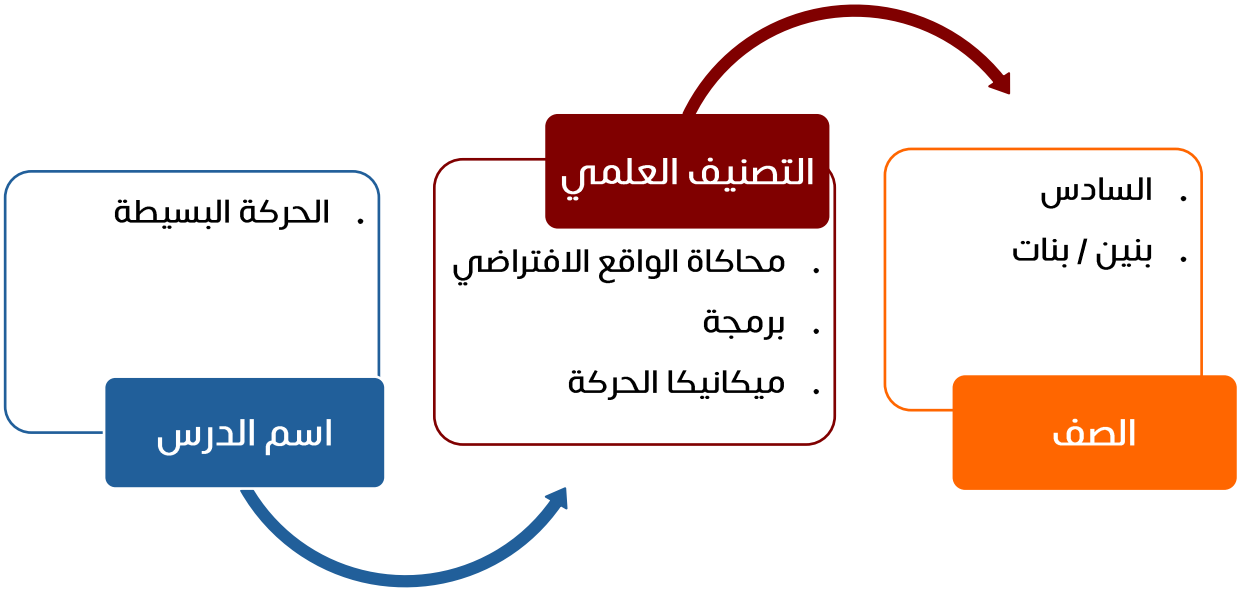
	النشاط الاستهلالي	الزمن
المقدمة والتمهيد	تهيئة عقول المتعلمين للمعلومات الجديدة وتشويقهم لها، وتختلف حسب موضوع الدرس. أمثلة للمقدمة: • عرض الناتج النهائي للوحدة ليكون للمتعلم فكرة عن المفاهيم التي ستُعرض في الدروس القادمة. • مراجعة جزء سبق دراسته له علاقة بالدرس. • عرض فيديو متعلق بموضوع الدرس أو استعراض مواقع وصفحات على الإنترنت. ملاحظة: يراعى في المقدمة وضوحها ومناسبتها للفئة العمرية للمتعلمين وارتباطها بالدرس، وعدم الإطالة في المقدمة حتى لا تؤثر على الزمن المخصص لشرح الدرس.	3-5 دقائق

م	نواتج التعلم	محتوى التعلم (الأنشطة والاستراتيجيات)	التقويم	الزمن
	تتضمن ما الذي يُتوقع من المعلم تحقيقه نتيجة لأنشطة التعلم، ويُعبّر عنها على شكل معارف أو مهارات أو اتجاهات، ويجب أن تكون واضحة تصف ما الذي يُتوقع أن يتعلمه المتعلمون نتيجة اكتمال عملية التعلم.	محتوى التعلم يجب أن يُجيب عن السؤال التالي: ما الذي يجب تعلمه؟ ويصف الأنشطة والاستراتيجيات المستخدمة مع وصف مختصر للدور المناط بكل من المعلم والمتعلم.	صياغة لغوية واضحة للأسئلة والتعليمات التي توجه نشاط المتعلم، منعاً للارتجال وتحسيناً للتواصل اللفظي مع المتعلم باستخدام أدوات ووسائل التقويم، وتعكس هذه الأدوات إنجازات المتعلم وقياسها في مواقف حقيقية، فيبدو كنشاطات تعلم يُمارس فيها المتعلمون مهارات التفكير العليا وحل المشكلات الحياتية الحقيقية.	تقدير الزمن المناسب لتنفيذ النشاط، لاستثمار وقت الحصة بشكل سليم.
	التقويم الختامي	حل ورقة عمل: ..... صفحة: ..... أو ورقة عمل بديلة من تصميم المعلم بحيث تقيس النواتج بشكل شامل		

اليوم والتاريخ	الفصل	الحصة	التأمل الذاتي				
			الرضا عن تحقق الأهداف			تحديات واجهتني	مقترحات لتحسين الأداء
			☺	😑	☹	التعرّف على الأسباب والمشكلات، ووضع خطط حلولها. (من التحديات: توقف الإنترنت — ضعف أحد الأنشطة لتوصيل المهارة المطلوبة).	يقود المعلم إلى تحسين أدائه وإجراء التغييرات المناسبة التي تنعكس إيجابياً على التعلم. (استبدال نشاط بآخر والتي تساعد على تحسين الأداء).

مقترح خطة درس روبوت

بطاقة توضيحية للدرس



وصف موجز للمهارات التي يقدمها الدرس:

1. استيعاب مفهوم الواقع الافتراضي وتطبيقه في بيئة الروبوت.
2. التعرف على مكونات روبوت LEGO SPIKE.
3. التعامل مع واجهة برنامج **ORL** .
4. التعرف على كتل الحركة (Drive, Turn) في بيئة **ORL** .
5. توظيف الأوامر البرمجية للتحكم في الاتجاه والمسافة.
6. بناء مسارات هندسية بسيطة (مستطيل – مثلث).
7. الربط بين القيم المدخلة (مسافة/ زاوية/ سرعة) وسلوك الروبوت على المحاكي.
8. تشغيل المحاكاة وملاحظة النتائج.
9. حفظ المشروع البرمجي واستدعاؤه عند الحاجة.

عنوان الدرس	العالم الافتراضي ORL - الحركة البسيطة				
الأسبوع	الوحدة	اليوم	التاريخ	الفصل	الحصة
التاسع	الأدوات الرقيمة الروبوت التعليمي				
الحصة					
الثانية					

مصادر التعلم والوسائل التعليمية	القيمة التربوية
– فيديو تعليمي عن الروبوت التعليمي – عرض تقديمي لعرض الدرس (خطوات تشغيل بمنصة ORL - مناطق البرنامج - خطوات استدعاء و حفظ) – بطاقات تعليمية تحتوي ..... – منصة ORL و أجهزة الحاسوب في المختبر للتطبيق العملي – السبورة الذكية لعرض الدرس	تقدير العلم و العلماء عرض صورة لعلماء عرب والتعرف على دورهم في مجال التقدم العلمي واختراعاتهم التي أسهمت في وضع خارطة الطريق لعلوم الروبوت

	النشاط الاستهلالي	الزمن
المقدمة والتمهيد	• عرض صورًا (سيارة ذاتية القيادة، غسالة أوتوماتيكية، إنسان آلي صناعي، إنسان، هاتف ذكي) ثم سؤال المتعلمين هل هذه الاختراعات روبوت أم لا؟ • توضيح مفهوم الروبوت وما هي مكونات الروبوت	5 دقائق

ملاحظات تساعدك:

- **تنفيذ مهام تحاكي السلامة المرورية أو إدارة الوقت.**
- **استلهام مسارات مرتبطة ببيئة الكويت (معالم، شوارع، خرائط مبسطة).**

✓ **هذه النماذج لا تمثل خطة جامدة، بل هي أدوات مرنة تساعد المعلم على تصميم حصص تفاعلية تجعل المتعلم محور العملية التعليمية، وتحقق الهدف من إدراج الروبوت في المناهج الدراسية.**

م	نواتج التعلم	محتوى التعلم (الأنشطة والاستراتيجيات)	التقويم	الزمن
1	شرح مفهوم الواقع الافتراضي وبيئات المحاكاة.	باستخدام استراتيجية العصف الذهني: - . باستخدام استراتيجية العصف الذهني والمناقشة عرض فيديو قصير لمحاكاة الطيران ثم مناقشة: - ما هو مفهوم المحاكاة في التعليم؟ - لماذا نستخدم المحاكاة بدلاً من التجربة المباشرة؟ - حل نشاط استكشاف ص137	باستخدام استراتيجية التعلم التعاوني: جعل كل مجموعة تجيب عن الأسئلة: - عرّف مفهوم المحاكاة التعليمية بكلماتك - أذكر فائدة واحدة للمحاكاة في التعليم مقارنة بالتجربة المباشرة.	5 دقائق
2	التعرف على مكونات روبوت LEGO SPIKE ووحداته البرمجية الأساسية.	من خلال استراتيجية الاستكشاف: - باستخدام استراتيجية التعلم التعاوني من خلال توزيع بطاقات تحتوي على صور مكونات الروبوت LEGO SPIKE وأخرى تحتوي على أسماء ووظيفة كل مكون ويطلب من كل مجموعة مطابقة الصور مع أسمائها ووظائفها. -حل نشاط استكشاف ص138	أجب على السؤال التالي: - سؤال شفهي لكل مجموعة :ما وظيفة الحساس/ المحرك/ المتحكم في الروبوت؟	5 دقائق
3	استكشاف واجهة ORL والتعرف على أدوات التحكم البرمجي.	من خلال الشرائح الايضاحية و باستخدام التعلم بالاكتشاف الموجه يقوم المعلم بعمل جولة استكشافية داخل المنصة، يطلب من المتعلمين الضغط على الأيقونات المختلفة (Drive, Turn, Repeat)ومشاركة وظائفها مع زملائهم في الفصل .	نشاط عملي - يوضح كل طالب وظيفة أيقونة جربها.	5 دقائق
4	إنشاء حساب على منصةORL	باستخدام البيان العملي و باستخدام استراتيجية التعلم بالممارسة: توجيه المتعلمين لإنشاء حساب خطوة بخطوة على المنصة وحفظ مشروع فارغ للتأكد من نجاح العملية.	نشاط عملي - يتأكد المعلم من نجاح تسجيل دخول كل طالب.، يطلب من المتعلم فتح المشروع الفارغ الذي تم حفظه..	5 دقائق
5	إنشاء حركة بسيطة للروبوت باستخدام الكتل البرمجية داخل ORL	من خلال المناقشةوباستخدام استراتيجية التعلم التجريبي يطلب من المتعلم أن ينشئ برنامجًا قصيرًا لتحريك الروبوت للأمام لمسافة محددة، ثم يغيّر القيم بنفسه لملاحظة الفرق في الحركة. -عرض الخطوات بعد التأكد من تحقق النشاط للمتعلمين	باستخدام استراتيجية المشاركة و نشاط عملي: - أطلب من المتعلم وصف ما يحدث عند تغيير المسافة من 20 إلى 50.	5 دقائق
6	محاكاة حركة الروبوت داخل البيئة البرمجية ORL وملاحظة الاستجابة.	باستخدام استراتيجية التعلم القائم على التحدي يطلب من المتعلم تغيير المسافة والسرعة للحركة مع تغيير اتجاه الحركة (أمام/خلف) وإنشاء مسارات مختلفة. - المهمة الأولى ص 145 ومتابعة أداء المتعلمين	من خلال استراتيجية الملاحظة: قدم المتعلم مساره الخاص أمام زملائه في المجموعة ثم يتم مناقشة كيف يؤثر تغيير السرعة على زمن الحركة؟	5
7	حفظ المشروع البرمجي وإعادة استدعائه لاحقًا	باستخدام استراتيجية التعلم بالاستكشاف بعد الانتهاء من المهمة السابقة يطلب من المتعلم استكشاف خطوات حفظ المشروع لأول مرة.	من خلال استراتيجية السؤال الموجه: ما أهمية تسمية المشروع بأسماء واضحة ومنظمة؟	5
التقويم الختامي		نشاط ص145		5قيقة

** تذكير المتعلمين بضرورة إغلاق الملف و البرنامج وإعادة تنظيم المختبر .

اليوم والتاريخ	الفصل	الحصة	التأمل الذاتي				
			الرضا عن تحقق الأهداف			تحديات واجهتني	مقترحات لتحسين الأداء
الأحد 2025-11-11	1\6	الأولى	☺	😐	☹	ضيق وقت الحصة، بسبب النشاط في طابور الصباح لم يتم شرح اخر بند للدرس	عدم الاسهاب في الشرح النظري واستخدام استراتيجية البيان العملي مع استراتيجية التعلم المباشر
الثلاثاء 2025-11-13	2\6	الثالثة	☺	😐	☹	لا يوجد	لا يوجد

ملاحظات عامة للمعلم

- ✓ تخطيط الدرس المعروض في هذا الدليل هو نموذج مقترح يمكن الاستفادة منه كإطار عام، مع إمكانية التعديل والتكييف وفق:
  - مستوى المتعلمين وخبراتهم السابقة.
  - الزمن المتاح للحصة.
  - الأدوات والتقنيات المتوفرة في المدرسة.
- ✓ الهدف الرئيسي من إدراج الروبوت في المنهج هو:
  - ربط المفاهيم النظرية بواقع الحياة عبر أنشطة عملية في حصة الحاسوب.
  - تنمية مهارات التفكير الناقد وحل المشكلات باستخدام برمجة الروبوت في مواقف حياتية.
  - تعزيز التكامل بين المواد الدراسية (الرياضيات، العلوم، اللغة العربية، الدراسات الاجتماعية) من خلال بناء المسارات، الحسابات، والمحاكاة.
  - تنمية مهارات القرن الحادي والعشرين (التعاون – الإبداع – الاتصال – التفكير الحاسوبي).
- ✓ يمكن للمعلم إدخال تعديلات على الأنشطة لتصبح أكثر قربًا من حياة المتعلمين اليومية، مثل:
  - برمجة الروبوت لمحاكاة حركة مركبة ذاتية القيادة.

إرشادات للمعلم عند التدريس باستخدام المحاكي

1) التركيز على الفهم المنطقي

وضّح للطلاب أن الهدف هو تعلم التفكير البرمجي والمنطقي وليس فقط تشغيل الروبوت.

شجعهم على تصور ما يحدث لو كان الروبوت حقيقيًا (كيف سيتحرك، ماذا سيواجه).

2) استخدام الرسوم التوضيحية

اعرض صورًا أو مقاطع فيديو قصيرة لروبوت LEGO SPIKE أو EV3 أثناء تنفيذ نفس الأوامر، هذا يعوّض غياب الروبوت الحقيقي ويقرب الصورة في ذهن الطلاب.

3) تشجيع التخيل والمحاكاة الذهنية

اطلب من المتعلمين أن يتخيلوا: ماذا سيحدث لو سار الروبوت للأمام 5 ثوانٍ؟

اجعلهم يرسمون مسار الروبوت المتوقع على ورقة قبل تجربة الكود على المحاكي.

4) التركيز على دقة الكود

بما أن المتعلمين لن يواجهوا مشاكل فيزيائية (مثل انزلاق العجلات أو ضعف البطارية)، ركز على صحة الكود وترتيبه.

قارن بين الكود المنظم (باستخدام الدوال/الشروط) والكود المكرر.

5) تدريب المتعلمين على استكشاف الأخطاء (Debugging) وتصحيحها.

عوّد المتعلمين على قراءة رسالة الخطأ أو ملاحظة نتيجة المحاكاة وتحليل السبب.

شجعهم على تعديل القيم (مثل السرعة، الزاوية) والتجربة أكثر من مرة.

6) التعلم التعاوني

اجعل المتعلمين يعملون بشكل ثنائي: واحد يكتب الكود، والآخر يتابع ويقترح الحلول ثم يتبادلان الأدوار.

7) الواجبات والتطبيقات الإبداعية

اطلب من المتعلمين تصميم تحديات صغيرة على المحاكي مثل:

a. رسم مربع أو مثلث.

b. تفادي عقبة.

c. البحث عن "هدف" ملون.

8) ربط المحاكي بالواقع

ناقش مع المتعلمين: لو كان الروبوت حقيقيًا، ما المشكلات التي يمكن أن تواجهه؟ (سطح مائل، إضاءة ضعيفة، عوائق غير متوقعة) هذا يجعلهم يدركون أن البرمجة وحدها لا تكفي، بل هناك عوامل فيزيائية.

9) التقييم بالمنطق والشرح

لا تكتفِ بالنتيجة في المحاكي، بل اطلب من المتعلم أن يشرح كيف فكّر؟ ولماذا استخدم هذه الكتلة؟

10) استخدام التدرج في الصعوبة

- ابدأ بمهام بسيطة (تحريك للأمام/الدوران).
- ثم أضف الشروط (إذا/وإلا).
- ثم المستشعرات (اللون/الموجات فوق الصوتية).
- وأخيرًا الدوال والتكرار.

11

دليل المعلم لحل مشاكل Open Roberta Lab

1) الحسابات والمجموعات

لإنشاء مجموعات خاصة بفصولك يجب تسجيل البريد الالكتروني عند انشاء الحساب

← لا يمكن إنشاء مجموعة:

تحقق من تفعيل البريد الإلكتروني الخاص بك عبر رابط التحقق (إن لم يصل، أعد التسجيل).

← نسيان كلمة مرور المتعلم:

أعد التعيين من شاشة إدارة المجموعة (تعود لكلمة المرور الافتراضية).

← فشل تسجيل الحساب / البريد غير معروف:

أعد المحاولة باستخدام بريد Gmail واتصال إنترنت مستقر.

← الخروج التلقائي بعد تغيير اللغة/تحديث الصفحة:

أعد تسجيل الدخول يدويًا.

2) الواجهة والمحاكي(SIM/NEPO)

← الروبوت المختار خاطئ أو الكتل غير متوافقة:

من القائمة Robot اختر الامر (back to robot/system selection) ستعود للصفحة الرئيسية اختر الروبوت الصحيح (RCJ rescue).

← رسالة "robot not properly configured":

تحقق من مطابقة منافذ المستشعرات في الإعداد مع الكتل.

← نسخ/لصق الكتل لا يعمل :

صغّر أو أغلق نافذة المحاكاة مؤقتًا ثم أعد المحاولة.

← إضافة حساسات تؤدي لأخطاء/انهيار المحاكي:

استخدام عدد كبير من الحساسات قد يؤدي إلى بطء أو أخطاء في المحاكي أزل الحساسات الزائدة.

← الصور/المحاكاة تظهر غريبة أو فارغة:

امسح كاش المتصفح (Ctrl+F5) واضبط التكبير على 100%.

← دوران الروبوت أكبر/ أصغر من المطلوب:

خفّض سرعة الدوران لتحسين الدقة.

3) أخطاء الخادم

خطأ عند فتح الجولة التفاعلية "TaKe an interactive tour" :

ابدأ بدون الجولة التفاعلية، التلميحات داخل الـ Lab كافية.

خطأ خادم مع كتلة "show text" في البرنامج:

استبدله بكتلة "show on Serial Monitor".

رسالة "لم يمكن ترجمة البرنامج" :

امسح الكاش أو راجع توافق الإعداد (Configuration) مع الكتل.

4) الاتصال ونقل البرامج (USB/Wi-Fi/Bluetooth)

انقطاع الاتصال المتكرر:

الأفضل تشغيل خادم محلي في المدرسة (Local Server).

اتصال USB لا يعمل:

افتح المنافذ 80 و443 في الجدار الناري؛ تأكد من استخدام الكابل الصحيح.

زر Connect يتأخر في Windows :

قد يستغرق دقيقة عند أول تشغيل بعد التثبيت.

المتصفح الموصى به:

استخدم دائمًا Google Chrome أو متصفحات Chromium حديثة.

5) مشاكل خاصة بأجهزة معينة

SpiKe Prime/Robot Inventor

أعد تثبيت أو استعادة PybricKs عبر تطبيق LEGO SPIKE Legacy.

micro:bit/Calliope

إن فشل النسخ اليدوي جرّب أداة MbedUploader.

6) ملاحظات الصف والبيئة التعليمية

- **كثير من المشاكل(رموز غريبة، بطء، فشل compile):**

غالبًا تُحل بمسح الكاش + إعادة تسجيل الدخول + تأكد من شبكة الانترنت.

- **التوصية للمدارس:**
  - وفر إنترنت مستقر.
  - استخدم متصفح Google Chrome.
  - فكر في تشغيل خادم محلي لتفادي مشاكل الشبكات الصارمة.
- **لا استطيع الحفظ:**

استخدم"save as" للحساب أو"export program" لتنزيل الملف

"import program" لإعادة تحميله لاحقًا.

7) الدعم والإبلاغ

- اطلع أولًا على قسم Help / WiKi في المنصة.
- إن استمرت المشكلة، راسل فريق الدعم عبر البريد الالكتروني مع: وصف الخطوات + صورة الشاشة + نسخة من السجل (Logging).
- يمكن الاستفادة من مجموعات Google وGitHub Issues لمتابعة الأعطال الرسمية.

12

دروس الموائمة من الصف السابع للصف الثامن

وزارة التربية
Ministry of Education
State of Kuwait | دولـــة الكويـــت

موائمة منهج الروبوت للصف السابع المتوسط

للعام الدراسي 2025 - 2026م

الروبوت التعليمي ■ الفصل الدراسي الأول

نواتج التـــعلم

1	التعرف على بيئة البرمجة ORL واستخدامها في بناء مشروع جديد.
2	استخدام منصة Open Roberta Labلإنشاء مشروع جديد.
3	برمجة الروبوت للتحرك للأمام والاستدارة باستخدام كتل الحركة.
4	تطبيق لبنة التكرار لتكرار الأوامر بسهولة.
5	إضافة تفاعل للروبوت من خلال عرض نصوص وإصدار أصوات.
6	إنجاز مهمة بسيطة تجمع بين الحركة والتكرار والتفاعل.

الاستكشاف

التعلم

التعرف على بيئة
Open Roberta Lab

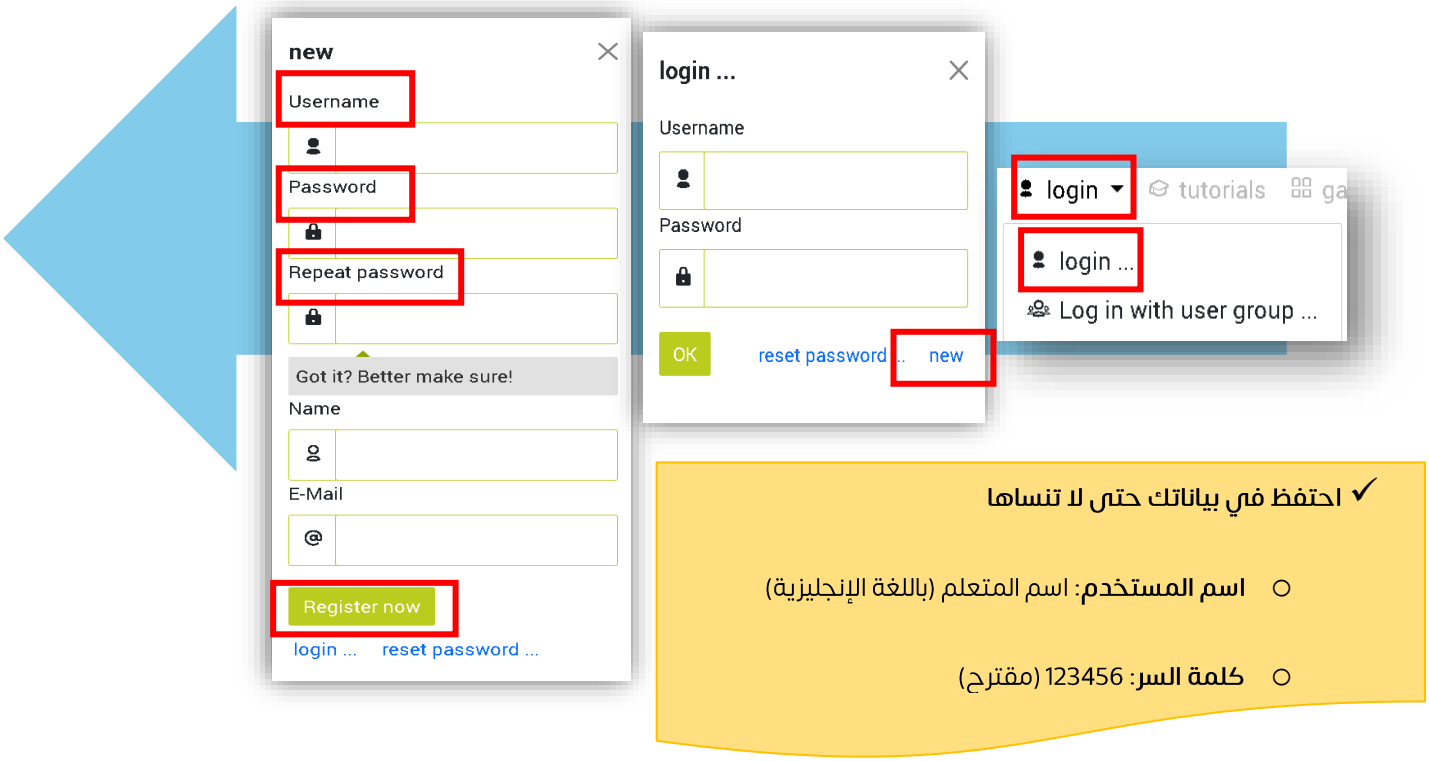
الدخول إلى منصة ORL

لبرمجة الروبوت في بيئة افتراضية، نتوجه إلى موقع المنصة عبر الخطوات التالية:

1. فتح متصفح الإنترنت Chrome.
2. الدخول على الرابط lab.open-roberta.org.
3. الضغط على مفتاح Enter لفتح الصفحة الرئيسية للمنصة .

إنشاء حسابك الخاص في برنامج ORL

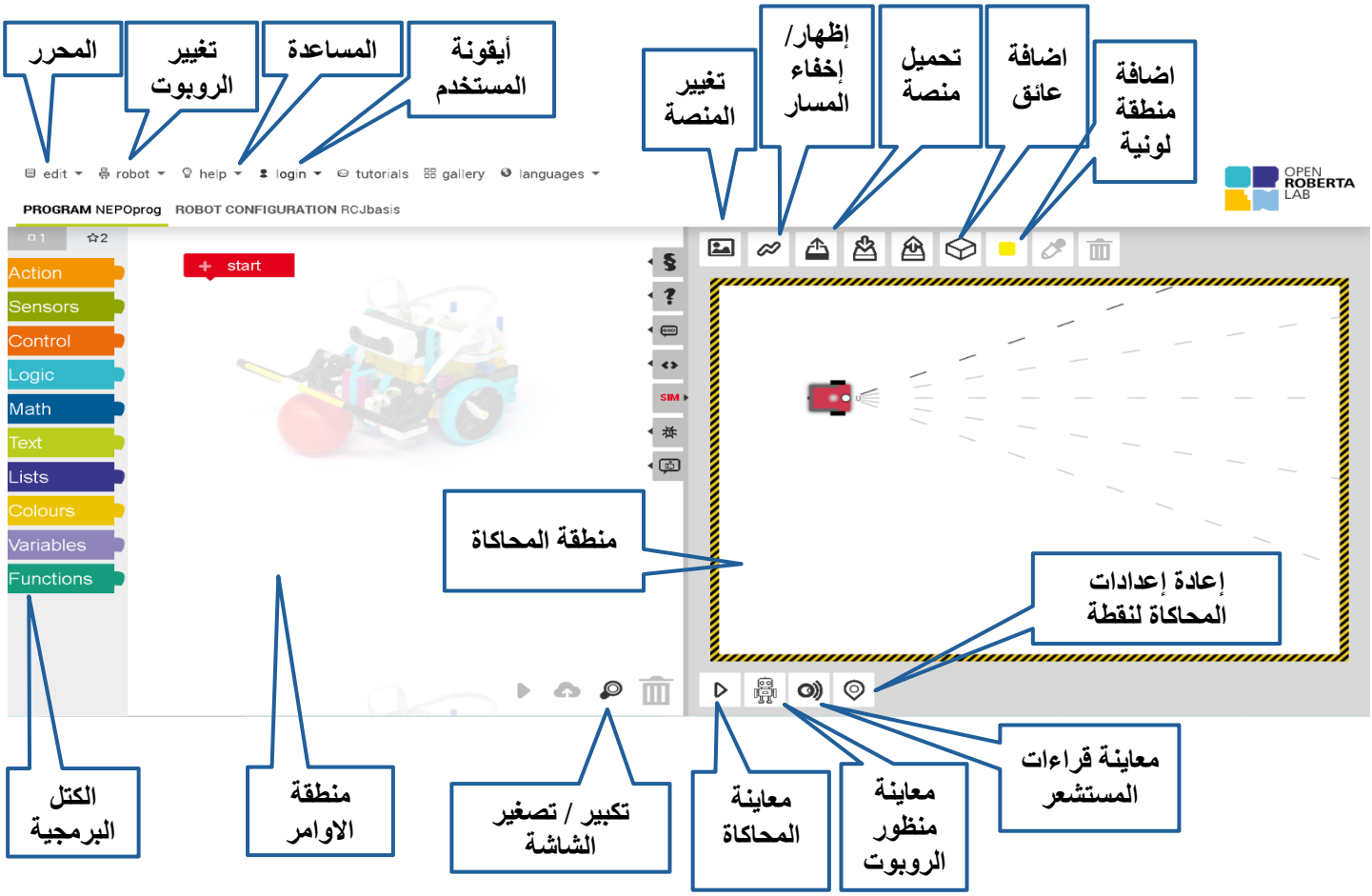
من الصفحة الرئيسية، اضغط على القائمة التي تسمى "Login"، ستظهر خيارات منها:



✓ احتفظ في بياناتك حتى لا تنساها

- **اسم المستخدم**: اسم المتعلم (باللغة الإنجليزية)
- **كلمة السر**: 123456 (مقترح)

التعرف على محاكي RCJ RescueOnlineSim



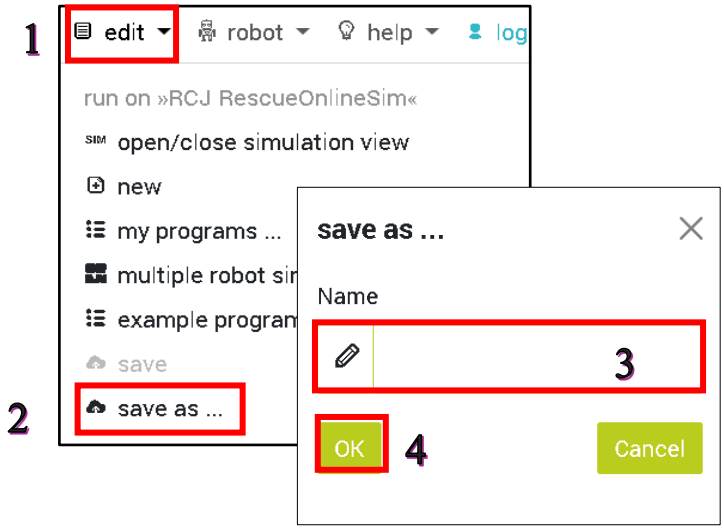
حفظ المشروع

يحفظ المشروع على المنصة إذا قمت بتسجيل الدخول على الحساب الخاص بك قبل بدء المشروع، بعد انتهائك من البرمجة

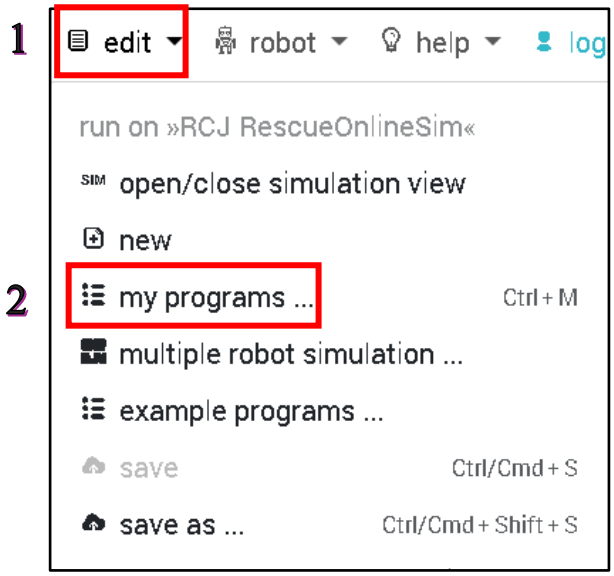
والتجريب اتبع الخطوات التالية للحفظ:

1. اختر قائمة التحرير (edit).
2. اختر من القائمة المنسدلة الامر (save as) احفظ المشروع أول مرة.
3. اكتب اسم البرنامج في مربع الحوار المنبثق في خانة (name).
4. اضغط موافق(ok) لحفظ الملف في حسابك الافتراضي على المنصة.

❖ عند حفظ تعديل على البرنامج سيظهر لك خيار حفظ (save)



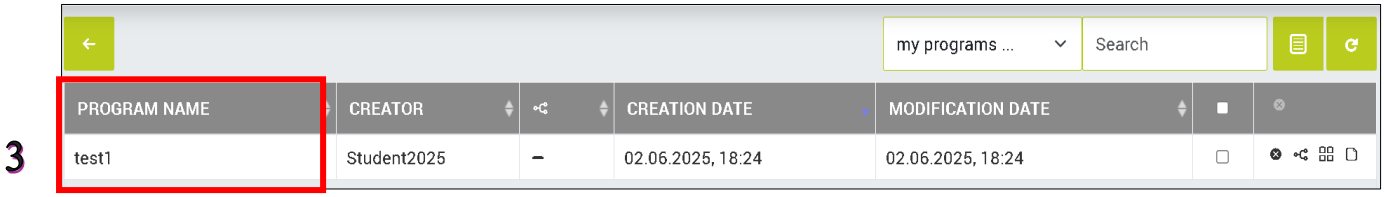
استدعاء المشروع



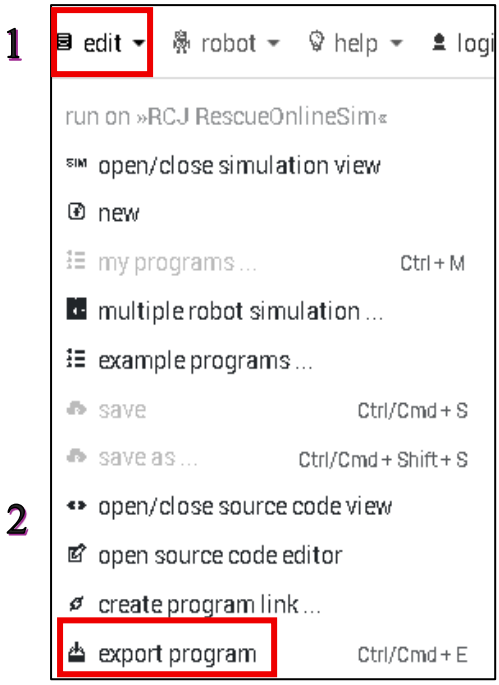
1- ختر قائمة التحرير (edit).

2- اختر من القائمة المنسدلة الامر (My programs) ستنقل الى الصفحة التي يتم تخزين كافة البرامج التي انشأتها على حسابك الافتراضي في المنصة.

3- اضغط على اسم البرنامج المطلوب وسيتم تحميله مباشرة على الصفحة.



تصدير المشروع



1- ختر قائمة التحرير (edit).

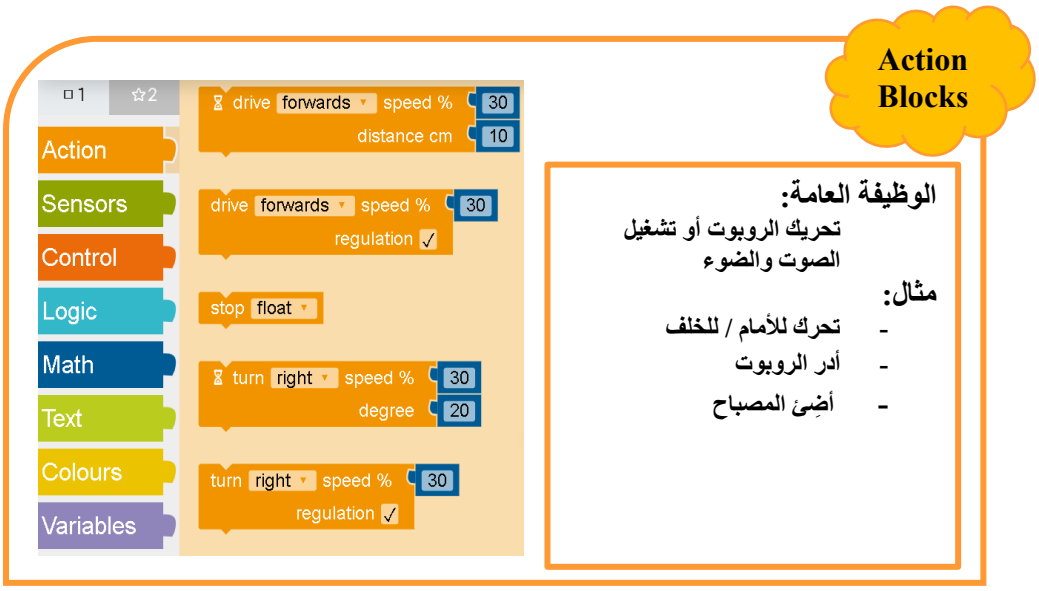
2- اختر من القائمة المنسدلة الامر (export program) سيتم تلقائيًا تحميل ملف المشروع على جهازك بصيغة (.xml).

النشاط

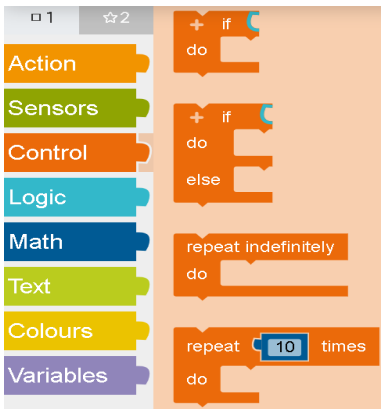
1. افتح منصة Open Roberta Lab وسجّل الدخول إلى حسابك.
2. من خيارات المحاكيات اختر الروبوت **RCJ RescueOnlineSim** .
3. أنشئ مشروعًا جديدًا واحفظه باسم **RescueStart** .

الكتل البرمجية

- في برنامج Open Roberta Lab تنظم الكتل البرمجية في مجموعات بحسب وظيفتها، وتتميز كل مجموعة بلون خاص بها للتعرف عليها بسهولة.
- عند البدء بمشروع جديد ستجد كتلة البدء Start ظاهرة في منطقة الأوامر حيث إن أي برنامج لا يعمل دونها.



Control Blocks



الوظيفة العامة:
تنظيم تسلسل الأحداث، مثل:
التكرار(loops) / الشروط (if/else)
الانتظار(wait)

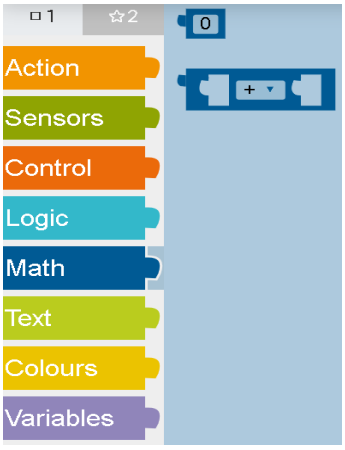
مثال:
- تكرار مجموعة أوامر عددًا معينًا من المرات أو للأبد.
- ينفذ أمرا عند تحقق شرط معين.

Text Blocks

الوظيفة العامة:
التعامل مع النصوص

مثال:
- تخزين نصوص في متغيرات
- عرض نصوص على الشاشة

Math Blocks



الوظيفة العامة:
كتل العمليات الحسابية مثل الجمع- الطرح – الضرب والقسمة.

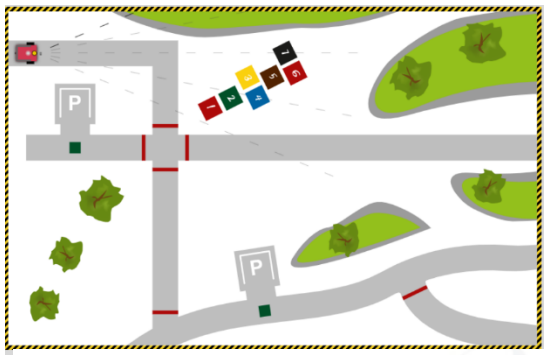
مثال:
- العمل مع المتغيرات (مثل العدادات أو سرعة محسوبة)
- إجراء العمليات الأساسية (جمع/طرح/ ضرب/ قسمة)

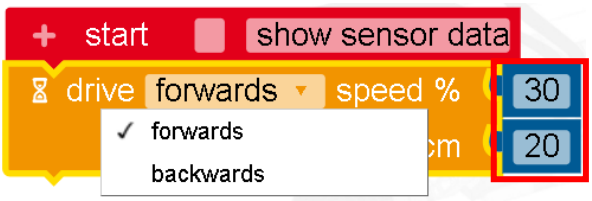
انتبه

- عند استخدامك للكتل لتنفيذ مهمة برمجية سيقوم البرنامج بتنفيذ الأوامر حسب ترتيبك للكتل واحدة تلو الأخرى وهذا ما يعرف بالأوامر المتتالية.
- يوجد مظهرين لمجموعات الكتل يمكنك التنقل بينها عن طريق الضغط على رقم المظهر

المهمة الأولى

أنشئ مشروعًا جديدًا ساعد الروبوت على الوصول الى نقطة توقفه الصحيحة عند الرمز P سنستعين بكتلتي drive و steerg من Action BlocKs لتوجيه الروبوت للتحرك بخط مستقيم والانعطاف عندما يصل الى زوايا الطريق ليصل الى المكان المطلوب، ثم احفظ المشروع باسم RescueTurn



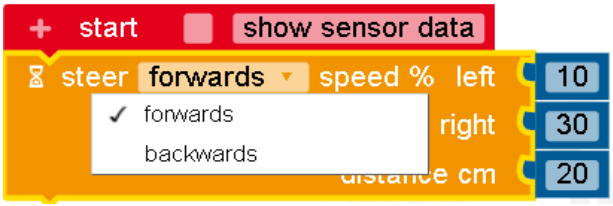


Drive BlocK تحتوي على معاملين:

المسافة (Distance) مشابه لعدد الخطوات المطلوب من الروبوت قطعها.

معامل السرعة (Speed) لتحديد سرعة المحرك.

لاحظ وجود سهم يساعدك لإظهار خيارات اتجاه الحركة للأمام Forward أو للخلف BacKward.



لاحظ وجود سهم يساعدك لإظهار خيارات اتجاه الانعطاف للأمام Forward أو للخلف BacKward.

Speed % left : لضبط سرعة المحرك الأيسر

Speed % right : لضبط سرعة المحرك الأيمن

Distance cm : لضبط مسافة الزاوية المقطوعة

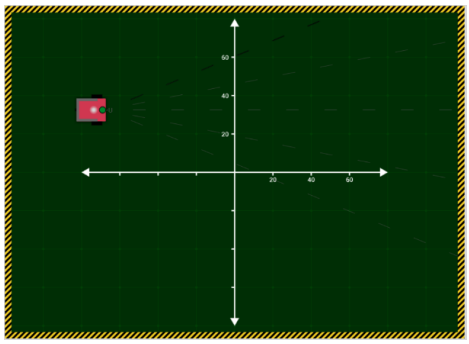
انتبه

- **عند استخدام الكتلة steer كلما اقتربت سرعات العجلتين من بعض، أصبح الانعطاف أوسع والدائرة أكبر، وكلما كان الفرق بين السرعتين كبيرًا، أصبح الانعطاف أشد والدائرة أصغر.**
- في كل مرة تضيف كتلة جديدة للمقطع البرمجي عليك باستعادة الوضع الافتراضي للروبوت بالضغط على زر إعادة الضبط.

التطبيق

أشكالي الهندسية

ورقة عمل



1- الاعداد

- افتح بيئة المحاكاة التعليمية RCJ RescueOnlineSim في منصة Open Roberta Lab.
- شغل المحاكي بالضغط على SIM
- بدل المنصات للوصول للشكل الظاهر أمامك.

2- مهمتك

برمج الروبوت ليتحرك ويرسم مربع طول ضلعة 40 سم، بعد الانتهاء من رسم الشكل اجعل الروبوت يظهر عبارة "تم رسم الشكل المربع" لمدة 3 ثوان ثم تشغيل تردد صوتي.

3- بناء البرنامج

- استخدم كتلة التكرار بعدد لتنفيذ المطلوب.
- تحرك إلى الأمام لمسافات محددة (عدد خطوات الشبكة).
- استدر بزاوية 90° يمين أو يسار حسب الاتجاه المطلوب.

4- تلميحات

- استخدم كتلة التكرار بعدد أضلاع الشكل الهندسي المطلوب تنفيذه.
- يتم حساب زاوية الشكل من خلال المعادلة التالية (360/ عدد أضلاع الشكل الهندسي).
- عند تنفيذ استدارة خفف السرعة ليتم تنفيذ الدوران بدقة.

تابع ورقة عمل

5- **شغل وجرب**

- عدل الزوايا والمسافات إذا خرج الروبوت عن المسار.
- تأكد من الضغط على زر Reset قبل كل مرة تجرب فيها البرنامج.

6- **احفظ البرنامج**

باسم Shapes على حسابك الشخصي.

7- **صدر البرنامج على الجهاز.**

8- **تحد إضافي**

استخدم كتلة steer لرسم دائرة علما بأن المسافة اللازمة لرسمها 90 سم دون الحاجة لتغيير سرعة المحركات.

وزارة التربية
Ministry of Education
State of Kuwait | دولـــة الكويـت

موائمة منهج الروبوت للصف الثامن المتوسط

للعام الدراسي 2025 - 2026م

الروبوت التعليمي ■ الفصل الدراسي الأول

الروبوت يكتشف العالم من المنصة إلى المستشعر

نواتج التـــعلم

1	التعرف على بيئة البرمجة ORL واستخدامها في بناء مشروع جديد.
2	استخدام منصة Open Roberta Labلإنشاء مشروع جديد.
3	التعرف على مفهوم المستشعرات ووظيفتها في الروبوت.
4	برمجة الروبوت للتوقف أو تغيير مساره عند وجود عائق باستخدام شرطية بسيطة.
5	إدراك العلاقة بين الحدث (عائق) والاستجابة (حركة أو صوت أو نص).
6	تنفيذ مهمة صغيرة تدمج بين الحركة والمستشعر والتفاعل.

التعلم

التعرف على بيئة
Open Roberta Lab

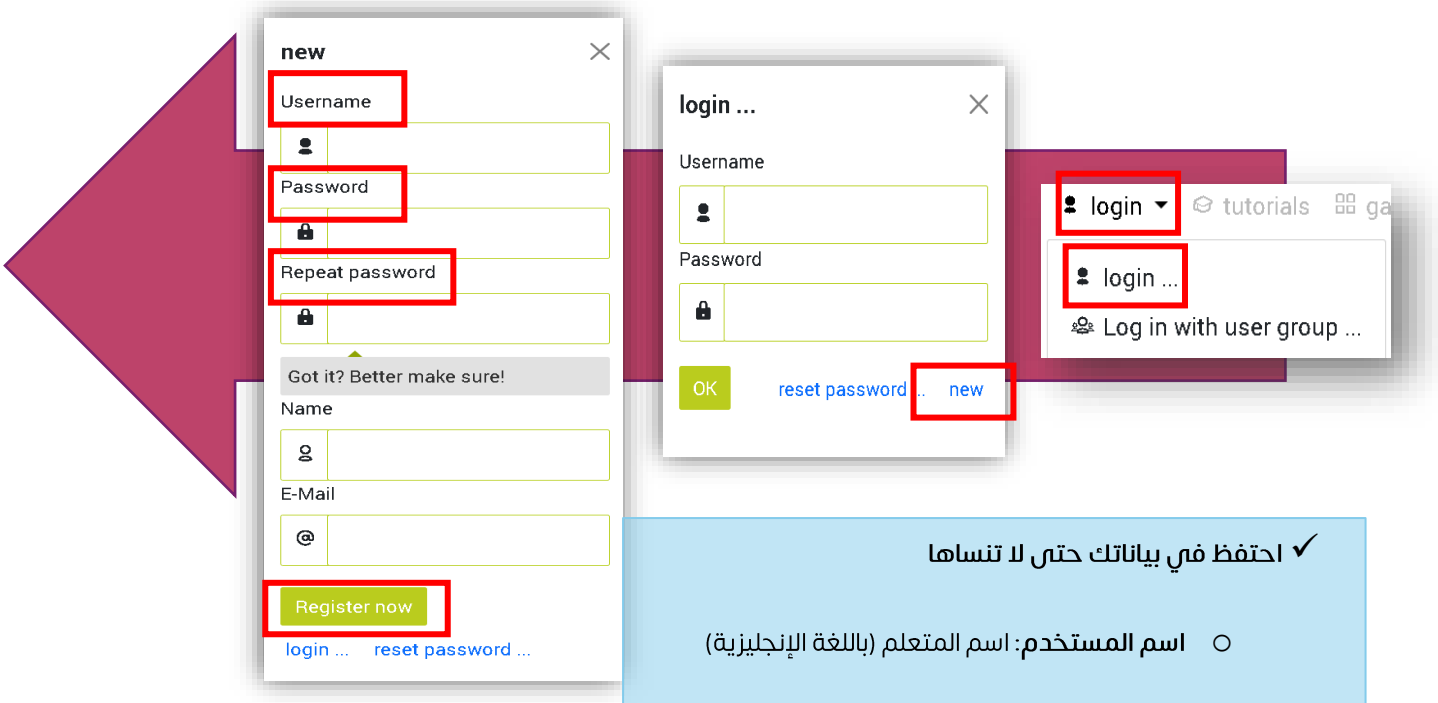
الدخول إلى منصة ORL

لبرمجة الروبوت في بيئة افتراضية، نتوجه إلى موقع المنصة عبر الخطوات التالية:

4. فتح متصفح الإنترنت Chrome.
5. الدخول على الرابط lab.open-roberta.org.
6. الضغط على زر Enter لفتح الصفحة الرئيسية للمنصة .

إنشاء حسابك الخاص في برنامج ORL

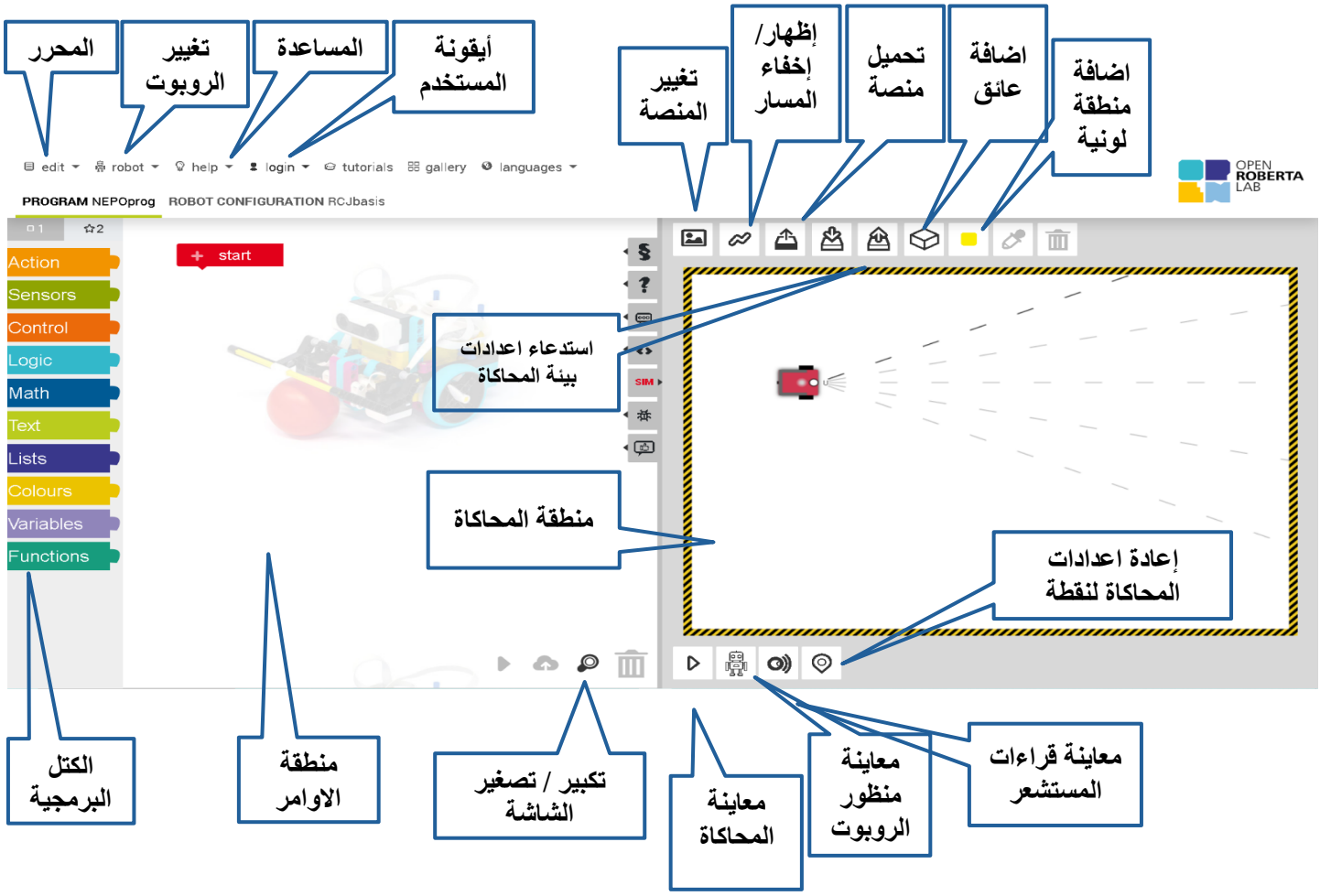
من الصفحة الرئيسية، اضغط على القائمة التي تسمى "Login"، ستظهر خيارات منها:



✓ احتفظ في بياناتك حتى لا تنساها

- **اسم المستخدم**: اسم المتعلم (باللغة الإنجليزية)
- **كلمة السر**: 123456 (مقترح)

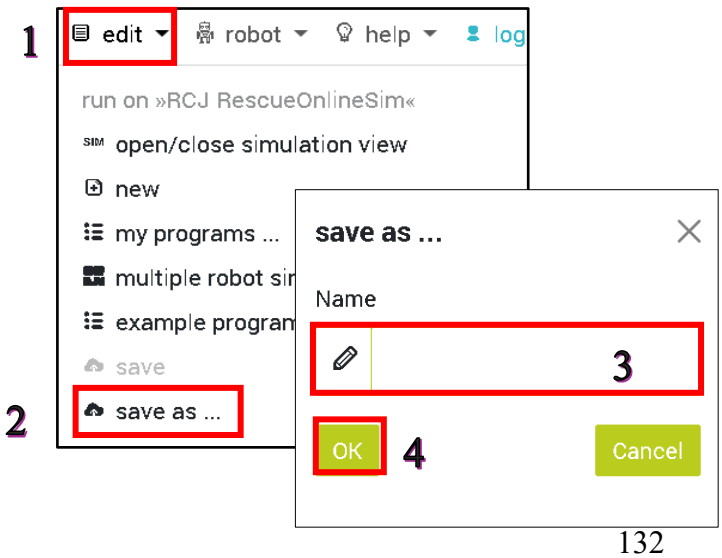
التعرف على محاكي RCJ RescueOnlineSim



حفظ المشروع

يحفظ المشروع على المنصة إذا قمت بتسجيل الدخول على الحساب الخاص بك قبل بدء المشروع، بعد انتهائك من البرمجة

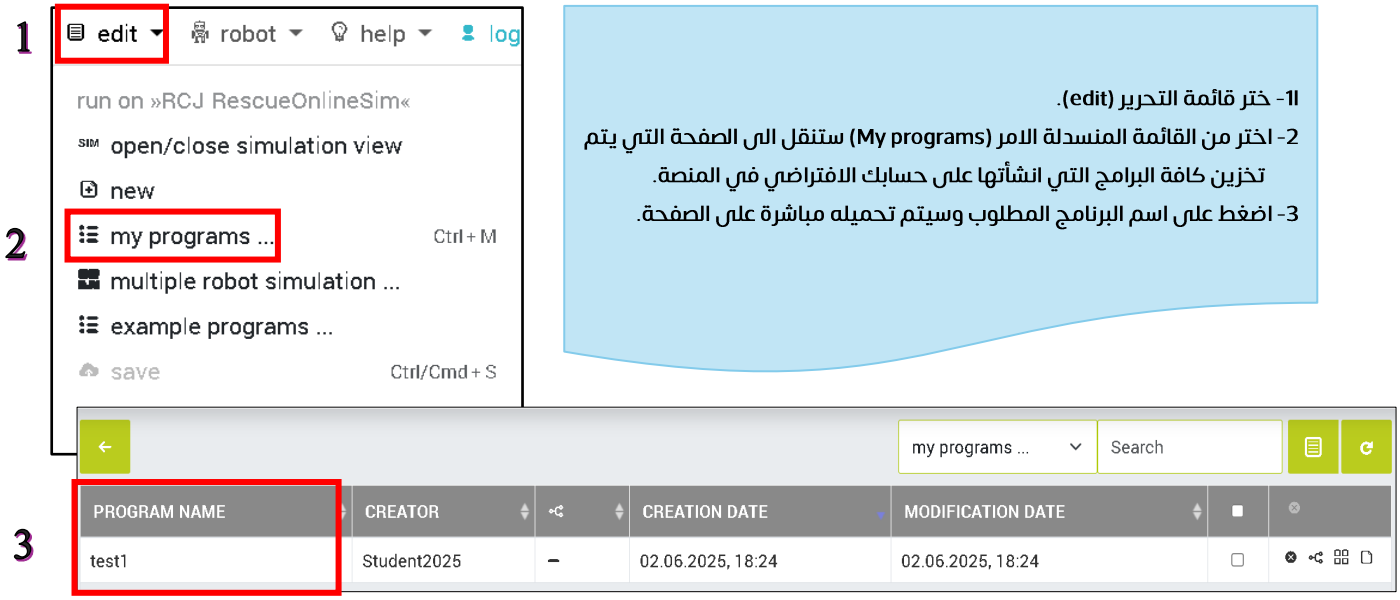
والتجريب اتبع الخطوات التالية للحفظ:



5. اختر قائمة التحرير (edit).
6. اختر من القائمة المنسدلة الامر (save as) احفظ المشروع أول مرة.
7. اكتب اسم البرنامج في مربع الحوار المنبثق في خانة (name).
8. اضغط موافق(ok) لحفظ الملف في حسابك الافتراضي على المنصة.

❖ عند حفظ تعديل على البرنامج سيظهر لك خيار حفظ (save)

استدعاء المشروع

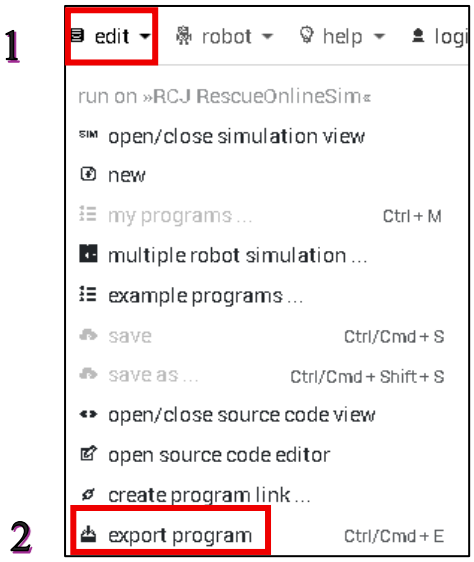


1- ختر قائمة التحرير (edit).

2- اختر من القائمة المنسدلة الامر (My programs) ستنقل الى الصفحة التي يتم تخزين كافة البرامج التي انشأتها على حسابك الافتراضي في المنصة.

3- اضغط على اسم البرنامج المطلوب وسيتم تحميله مباشرة على الصفحة.

تصدير المشروع



1- ختر قائمة التحرير (edit).

2- اختر من القائمة المنسدلة الامر (export program) سيتم تلقائيًا تحميل ملف المشروع على جهازك بصيغة (.xml).

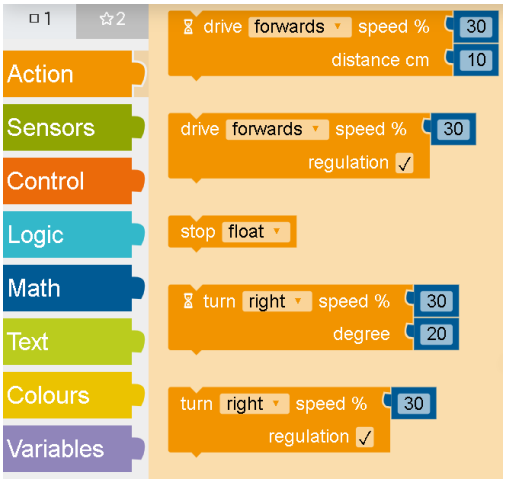
النشاط

4. افتح منصة Open Roberta Lab وسجّل الدخول إلى حسابك.
5. من خيارات المحاكيات اختر الروبوت RCJ RescueOnlineSim .

الكتل البرمجية

- في برنامج Open Roberta Lab تنظم الكتل البرمجية في مجموعات بحسب وظيفتها، وتتميز كل مجموعة بلون خاص بها للتعرف عليها بسهولة.
- عند البدء بمشروع جديد ستجد كتلة البدء Start ظاهرة في منطقة الأوامر حيث إن أي برنامج لا يعمل دونها.

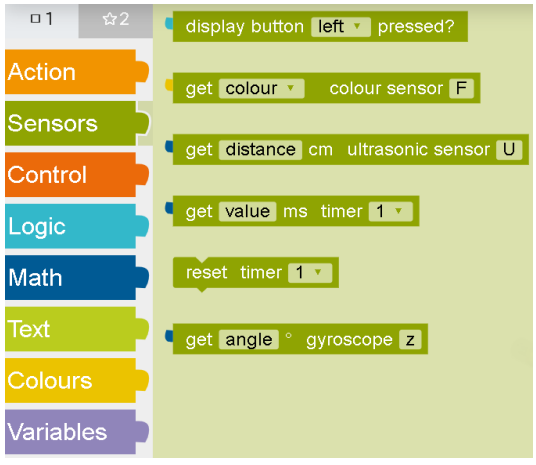
Action Blocks



الوظيفة العامة:
تحريك الروبوت أو تشغيل الصوت والضوء

مثال:
- تحرك للأمام / للخلف
- أدر الروبوت
- أضِئ المصباح

Sensors Blocks



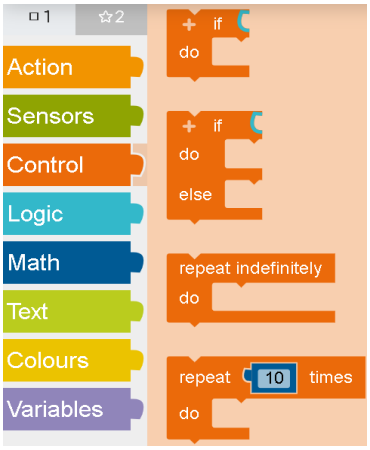
الوظيفة العامة:
قراءة بيانات من المستشعرات (لون- لمس - صوت)

مثال:
- التعرف على لون لاتخاذ قرار
- استشعار وجود عائق أمام الروبوت

انتبه

- عند استخدامك للكتل لتنفيذ مهمة برمجية سيقوم البرنامج بتنفيذ الأوامر حسب ترتيبك للكتل واحدة تلو الأخرى وهذا ما يعرف بالأوامر المتتالية.
- يوجد مظهرين لمجموعات الكتل يمكنك التنقل بينها عن طريق الضغط على رقم المظهر

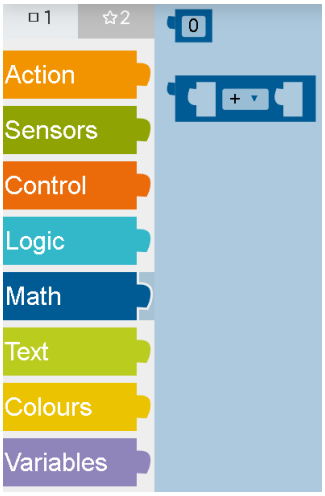
Control Blocks



الوظيفة العامة:
تنظيم تسلسل الأحداث، مثل:
التكرار(loops) / الشروط (if/else)
الانتظار(wait)
مثال:

- تكرار مجموعة أوامر عددًا معينًا من المرات أو للأبد.
- ينفذ أمرا عند تحقق شرط معين.

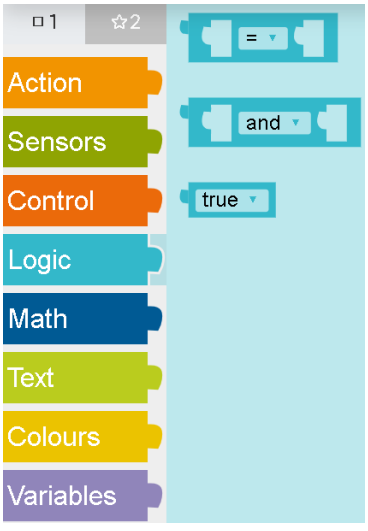
Math Blocks



الوظيفة العامة:
كتل العمليات الحسابية مثل الجمع- الطرح – الضرب والقسمة.
مثال:

- العمل مع المتغيرات (مثل العدادات أو سرعة محسوبة)
- إجراء العمليات الأساسية (جمع/طرح/ ضرب/ قسمة)

Logic Blocks



الوظيفة العامة:
بناء شروط منطقية مثلAND/OR وقيم صحيحة أو خاطئة.
مثال:
تساعد الروبوت في اتخاذ قرارات مركبة

- هل تحقق شرطان معًا (AND)
- هل القيمة تساوي شيئًا معينًا؟(==)

المستشعرات

المستشعرات هي الطريقة التي يتفاعل بها الروبوت مع العالم من حوله، تمامًا كما نفعل نحن باستخدام حواسنا .

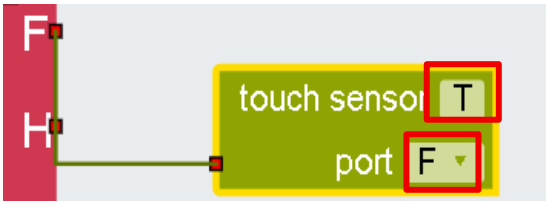
أنواع المستشعرات في محاكي روبوت RCJ Rescue داخل بيئة المحاكاة Open Roberta Lab.

اسم المستشعر	وظيفته	موقعه في الروبوت الافتراضي
مستشعر اللمس Touch sensor	يكشف عن اصطدام الروبوت بعائق خلفي (يستجيب عند الضغط عليه).	الخلف
مستشعر الألوان Color sensor	التعرف على ألوان الأسطح التي يتحرك فوقها الروبوت لتحديد المسار المناسب	في الأسفل باتجاه الأرض
مستشعر فوق صوتي Ultrasonic sensor	يقيس المسافة بين الروبوت والأجسام أمامه باستخدام الموجات فوق الصوتية	الأمام
مستشعر الجيروسكوب Gyroscope sensor	يقيس زاوية دوران الروبوت واتجاهه ويساعد في الحفاظ على التوازن وتحديد ما إذا كان الروبوت قد انحرف أو دار أثناء التنقل	مدمج داخل الروبوت

المستشعرات في منافذ محاكي الروبوت

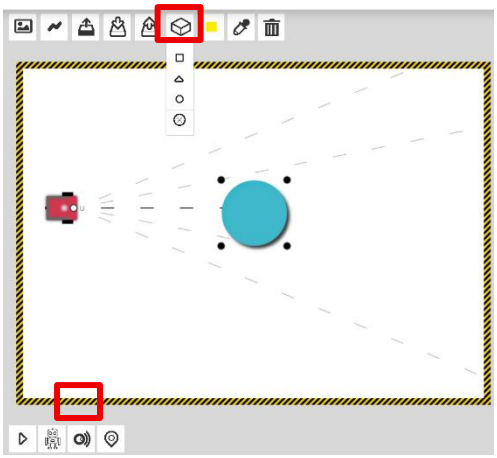
لاحظ وجود جميع المستشعرات موضحة بالمنفذ الموصلة به على معالج الروبوت الافتراضي لتحاكي توصيلها في منافذ الروبوت الحقيقي، ولكن لا يوجد مستشعر لمس ظاهر.

لإضافة أي مستشعر جديد اضغط على بطاقة sensors تظهر لك مجموعة المستشعرات التي يمكنك اضافتها.



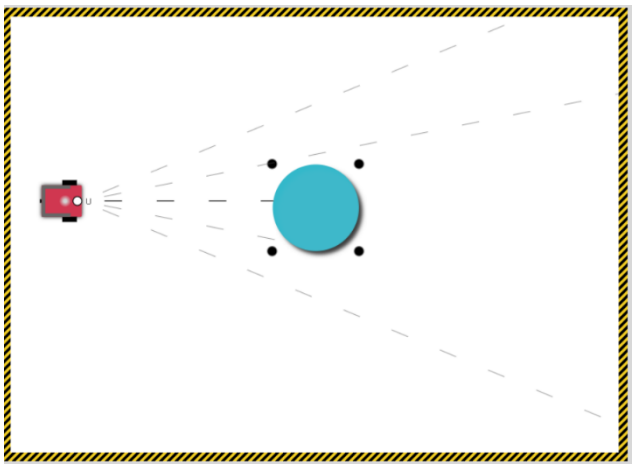
في الصورة الظاهرة أمامك لاحظ وجود حقلين
الأول يظهر حرف اختصار (T) اسم المستشعر الذي تم تحديده
الثاني يظهر لك حرف منفذ المعالج الذي تم توصيل المستشعر عليه
اكتب رمز المنفذ الظاهر port .....

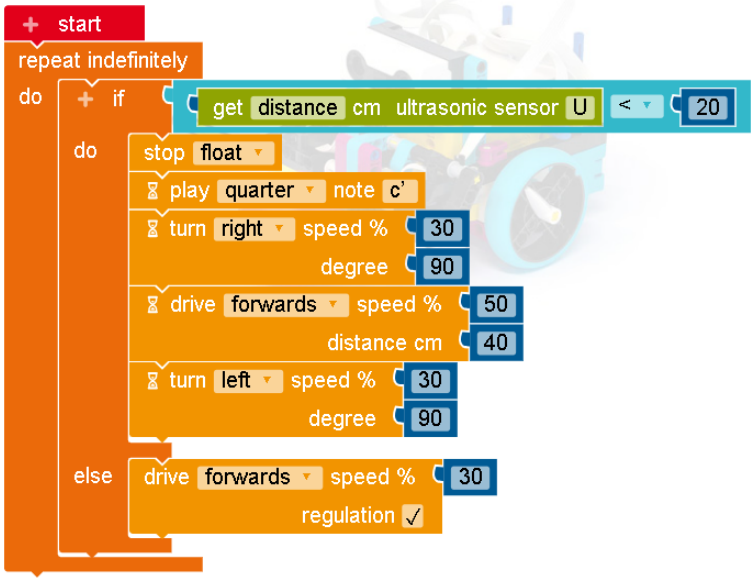
المهمة الأولى

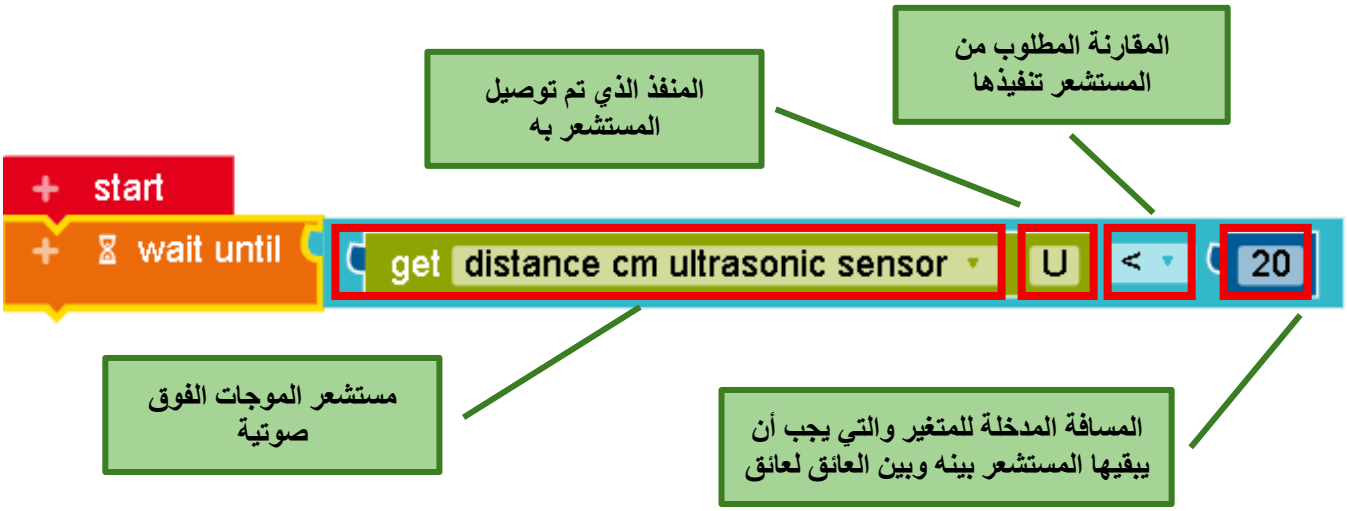


أنشئ مشروع جديد. اضف عائق دائري كما هو موضح في الصورة. نفذ المقطع البرمجي الظاهر في الصورة أمامك ارسم مسار حركة الروبوت الناتجة عن المقطع. شغل شاشة عرض بيانات المستشعر أثناء التشغيل سجل قراءة المسافة لدى مستشعر الموجات الفوق صوتية قبل التشغيل .....

ارسم المسار هنا



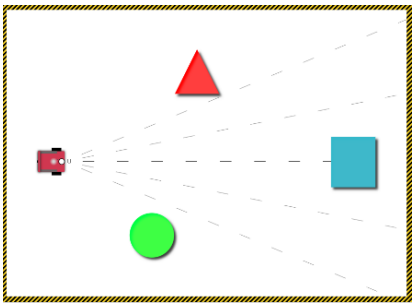




التطبيق

تجنب العوائق

ورقة عمل



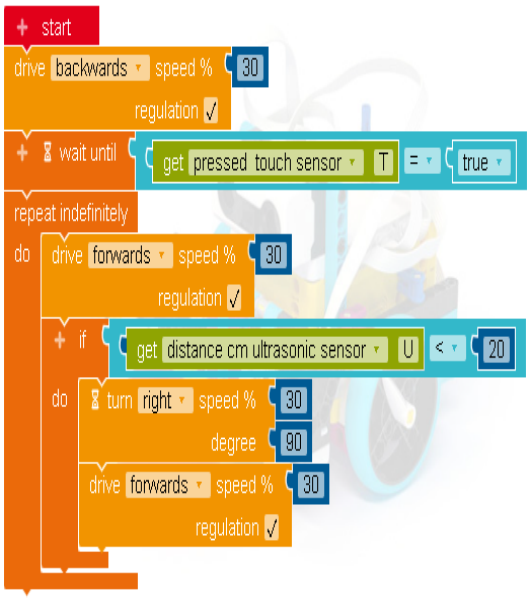
9- الاعداد

- افتح بيئة المحاكاة التعليمية RCJ RescueOnlineSim في منصة Open Roberta Lab.
- شغل المحاكي بالضغط على SIM
- إضافة عوائق إلى المنصة كما هو ظاهر أمامك.
- إضافة مستشعر اللمس Touch Sensor.

10- مهمتك

ساعد الروبوت بالتحرك للخلف إلى أن يصطدم بعائق ليبدأ التحرك إلى الامام بصورة تلقائية، عند استشعار الروبوت لأي عائق أمامه يستدير جهة اليمين بزاوية قائمة ثم يستكمل حركته للأمام.

11- بناء البرنامج



- يتحرك للخلف إلى أن يصطدم بعائق ليبدأ التحرك إلى الامام .
- استخدم كتلة التكرار إلى مالا نهاية.
- يتحرك الروبوت للأمام إذا كان هناك جسم أمام الروبوت على بعد أقل من 20 سم فانه يلتف جهة اليمين بمقدار 90 درجة ويستكمل الحركة للأمام .

7. تلميحات

- البدء بالحركة عند الضغط على Touch sensor.
- التحرك إلى الأمام إذا كان الطريق خالياً.
- تأكد أن منافذ المستشعرات صحيحة أثناء كتابة المقطع البرمجي

تابع ورقة عمل

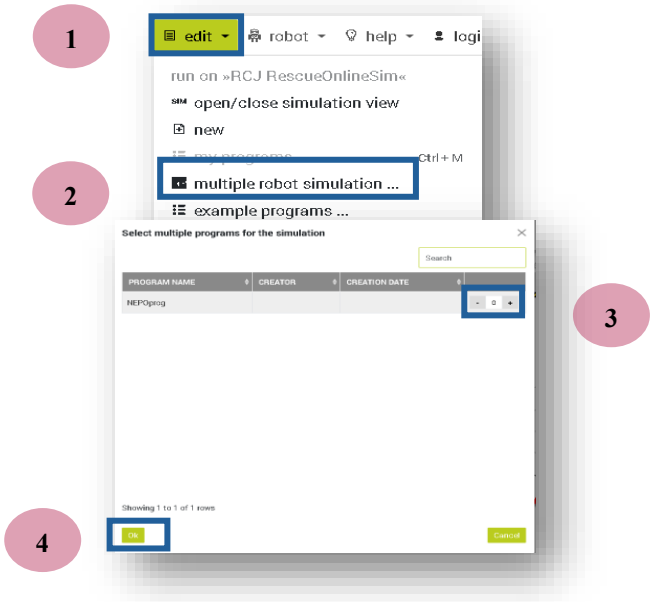
8. **شغل وجرب**
   - راقب الروبوت هل يختار الطريق المناسب عندما يواجه عائقاً.
   - حاول تعديل زوايا الدوران والسرعة إن لزم الأمر.
   - تأكد من الضغط على زر Reset قبل كل مرة تجرب فيها البرنامج.

9. **احفظ البرنامج**

   باسم Avoid على حسابك الشخصي.

10. **صدر البرنامج على الجهاز.**

11. **تحد إضافي**
    1. إضافة أكثر من روبوت للمشروع.



2. تحريك موقع كل نسخه إلى مكان مختلف داخل المحاكي.

وزارة التربية
Ministry of Education
State of Kuwait | دولـــة الكويت

موائمة منهج الروبوت للصف التاسع المتوسط

للعام الدراسي 2025 - 2026م

الروبوت التعليمي ■ الفصل الدراسي الأول

الروبوت الذكي من الألوان إلى الدوال

نواتج التـــعلم

1	التعرف على بيئة البرمجة ORL واستخدامها في بناء مشروع جديد.
2	استخدام منصة Open Roberta Labلإنشاء مشروع جديد.
3	التعرف على مفهوم المستشعرات ووظيفتها في الروبوت.
4	استخدام مستشعر الألوان لإيقاف الروبوت أو تفعيل استجابة عند رؤية لون محدد.
5	فهم فكرة الدالة (Function) كأداة لتقليل التكرار وتنظيم البرنامج.

الاستكشاف

التعلم

التعرف على بيئة
Open Roberta Lab

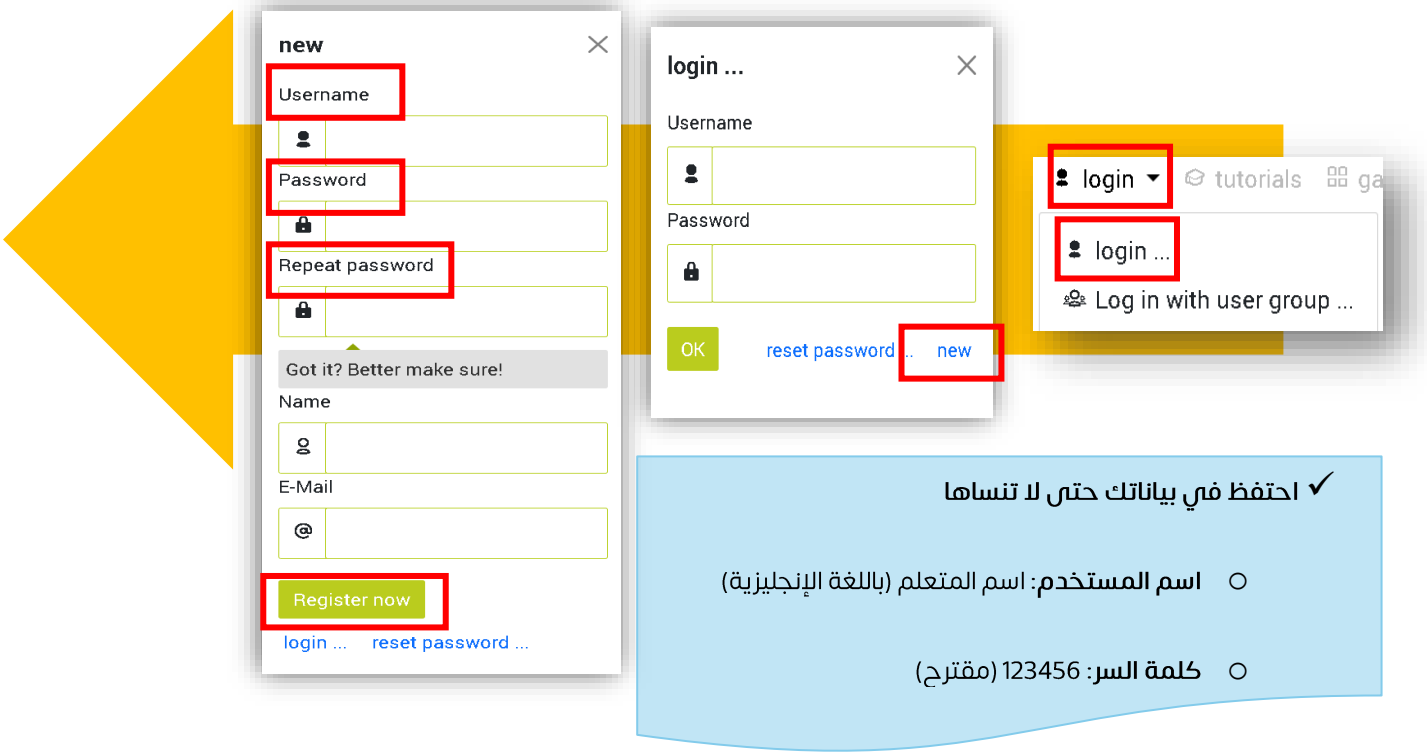
الدخول إلى منصة ORL

لبرمجة الروبوت في بيئة افتراضية، نتوجه إلى موقع المنصة عبر الخطوات التالية:

7. فتح متصفح الإنترنت Chrome.
8. كتابة الرابط lab.open-roberta.org.
9. الضغط على زر Enter لفتح الصفحة الرئيسية للمنصة .

إنشاء حسابك الخاص في برنامج ORL

من الصفحة الرئيسية، اضغط على القائمة التي تسمى "Login"، ستظهر خيارات منها:

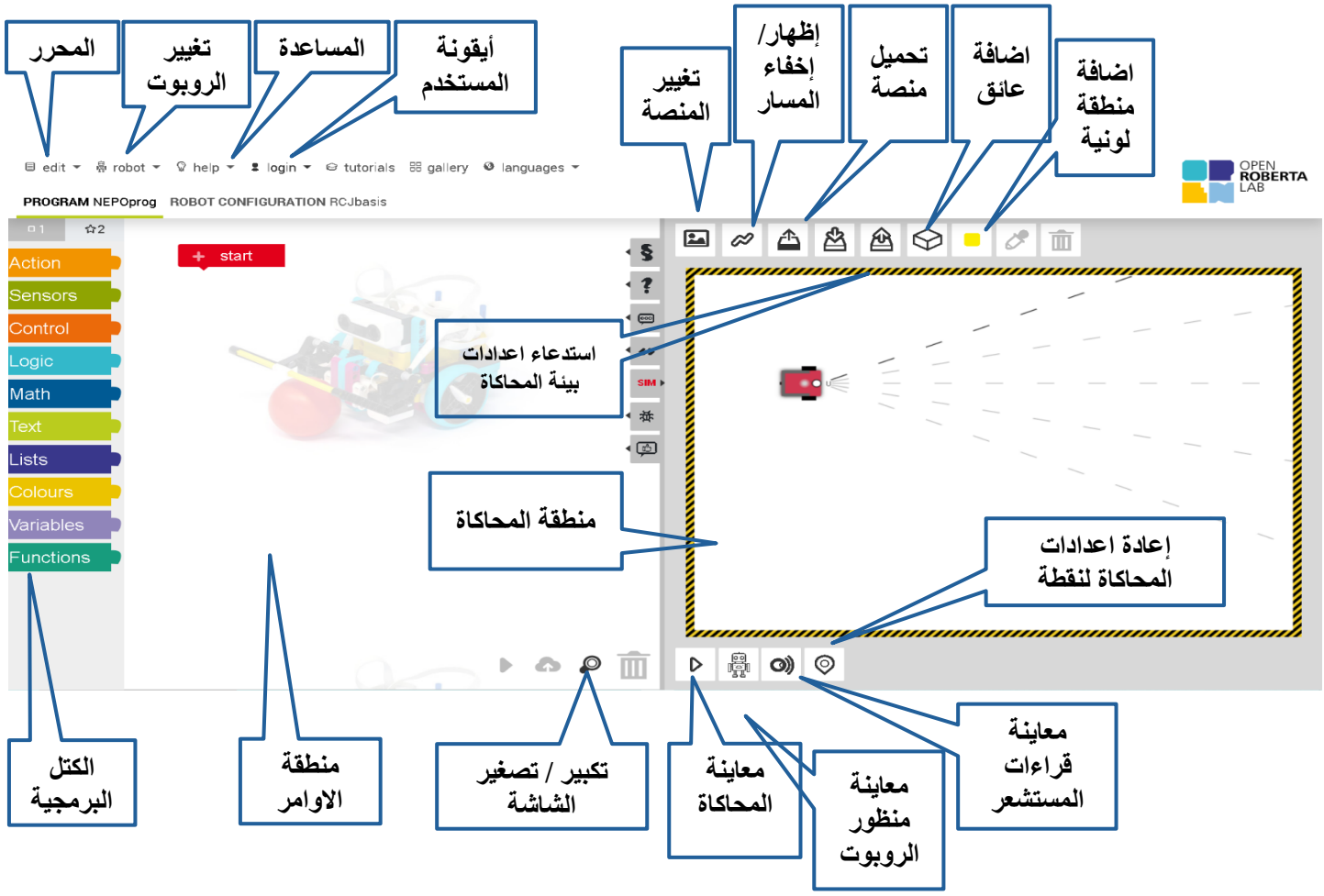


✓ احتفظ في بياناتك حتى لا تنساها

- **اسم المستخدم**: اسم المتعلم (باللغة الإنجليزية)
- **كلمة السر**: 123456 (مقترح)

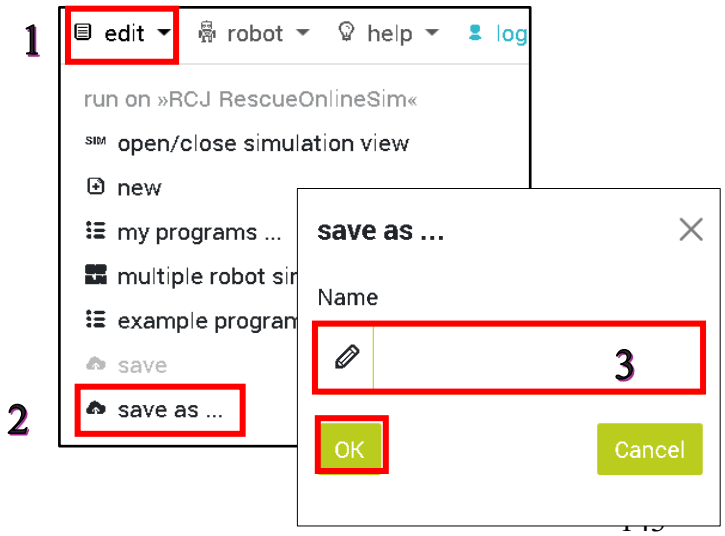
الروبوت الذكي من الألوان إلى الدوال

التعرف على محاكي RCJ RescueOnlineSim



حفظ المشروع

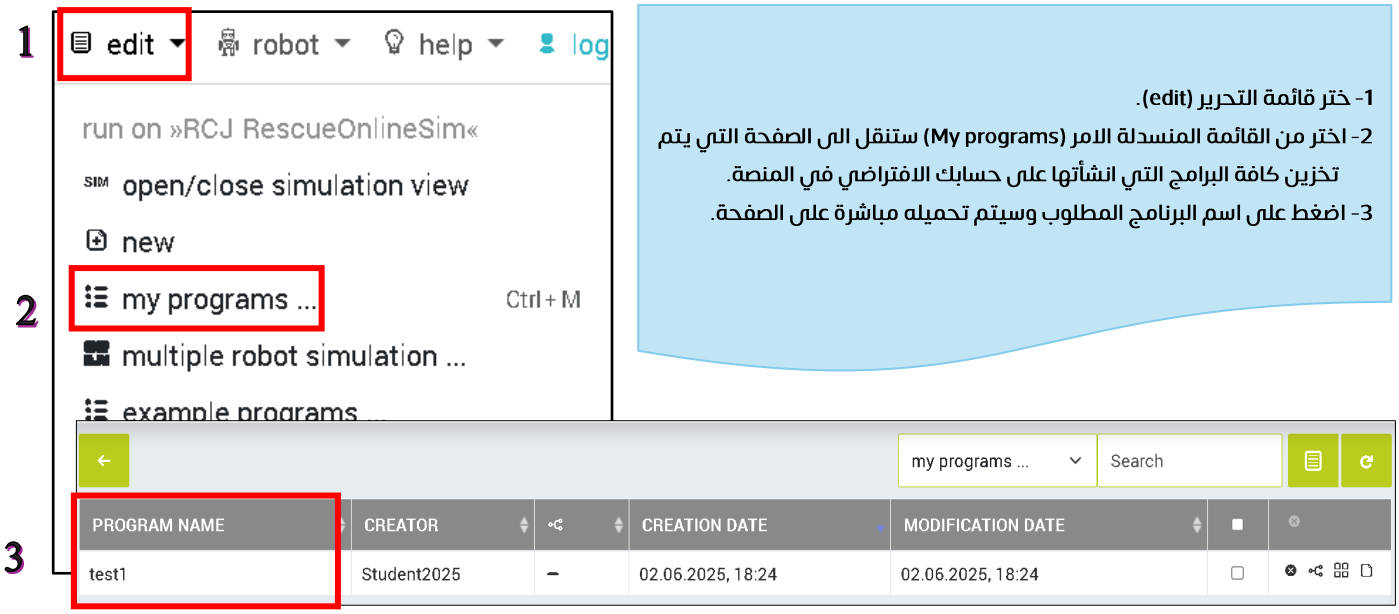
يحفظ المشروع على المنصة إذا قمت بتسجيل الدخول على الحساب الخاص بك قبل بدء المشروع، بعد انتهائك من البرمجة والتجريب اتبع الخطوات التالية للحفظ:



9. اختر قائمة التحرير (edit).
10. اختر من القائمة المنسدلة الامر (save as) احفظ المشروع أول مرة.
11. اكتب اسم البرنامج في مربع الحوار المنبثق في خانة (name).
12. اضغط موافق(ok) لحفظ الملف في حسابك الافتراضي على المنصة.

❖ عند حفظ تعديل على البرنامج سيظهر لك خيار حفظ (save)

استدعاء المشروع

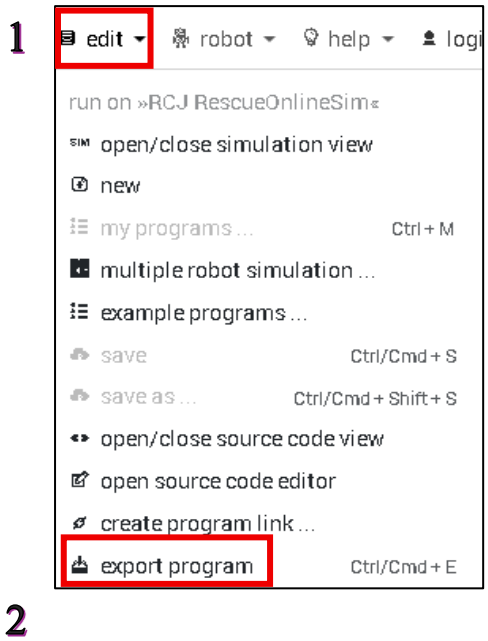


1- ختر قائمة التحرير (edit).

2- اختر من القائمة المنسدلة الامر (My programs) ستنقل الى الصفحة التي يتم تخزين كافة البرامج التي انشأتها على حسابك الافتراضي في المنصة.

3- اضغط على اسم البرنامج المطلوب وسيتم تحميله مباشرة على الصفحة.

تصدير المشروع



1- ختر قائمة التحرير (edit).

2- اختر من القائمة المنسدلة الامر (export program) سيتم تلقائيًا تحميل ملف المشروع على جهازك بصيغة (.xml).

الروبوت الذكي من الألوان إلى الدوال

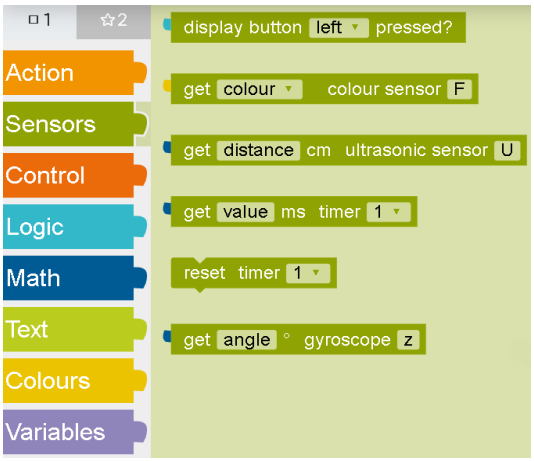
النشاط

1. افتح منصة Open Roberta Lab وسجّل الدخول إلى حسابك.
2. من خيارات المحاكيات اختر الروبوت **RCJ RescueOnlineSim** .
3. أنشئ مشروعًا جديدًا واحفظه باسم **RescueStart** .

الكتل البرمجية

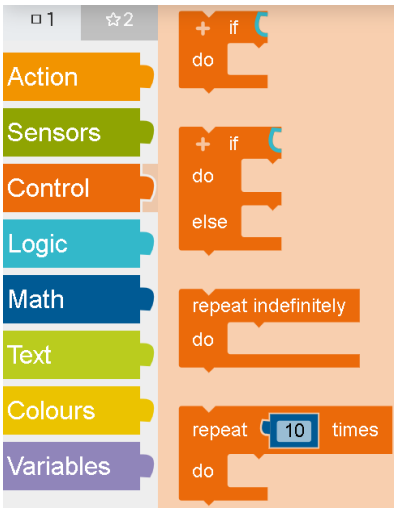
- في برنامج Open Roberta Lab تنظم الكتل البرمجية في مجموعات بحسب وظيفتها، وتتميز كل مجموعة بلون خاص بها للتعرف عليها بسهولة.
- عند البدء بمشروع جديد ستجد كتلة البدء Start ظاهرة في منطقة الأوامر حيث إن أي برنامج لا يعمل دونها.

Sensors Blocks



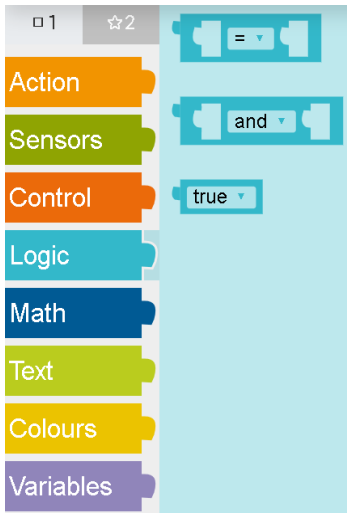
الوظيفة العامة:
قراءة بيانات من المستشعرات (لون- لمس - صوت)
مثال:
- التعرف على لون لاتخاذ قرار
- استشعار وجود عائق أمام الروبوت

Control Blocks



الوظيفة العامة:
تنظيم تسلسل الأحداث، مثل:
التكرار(loops) / الشروط (if/else)
الانتظار(wait)
مثال:
- تكرار مجموعة أوامر عددًا معينًا من المرات أو للأبد.
- ينفذ أمرا عند تحقق شرط معين.

Logic Blocks

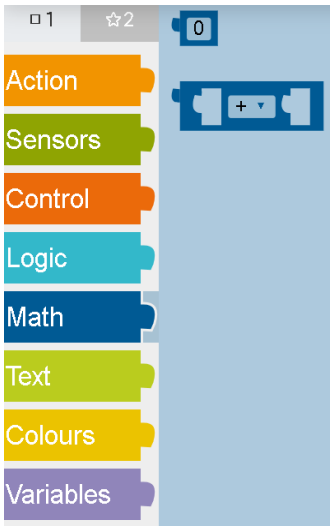


الوظيفة العامة:
بناء شروط منطقية مثلAND/OR وقيم صحيحة أو خاطئة.
مثال:
تساعد الروبوت في اتخاذ قرارات مركبة
- هل تحقق شرطان معًا (AND)
- هل القيمة تساوي شيئًا معينًا؟ (==)

Text Blocks

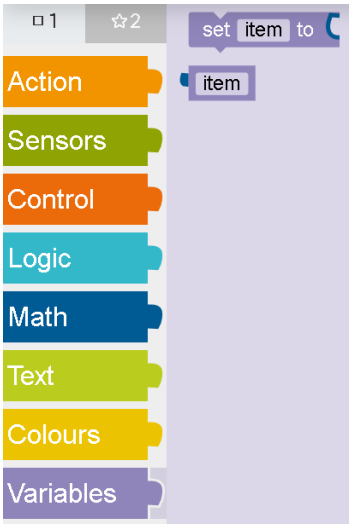
الوظيفة العامة:
التعامل مع النصوص
مثال:
- **تخزين نصوص في متغيرات**
- **عرض نصوص على الشاشة**

Math Blocks



الوظيفة العامة:
كتل العمليات الحسابية مثل الجمع- الطرح – الضرب والقسمة.
مثال:
- **العمل مع المتغيرات (مثل العدادات أو سرعة محسوبة)**
- **إجراء العمليات الأساسية (جمع/طرح/ ضرب/ قسمة)**

Variables Blocks



الوظيفة العامة:
كتل لإنشاء متغيرات وتخزين البيانات فيها.
مثال:
- تخزين بيانات لوقت لاحق
- زيادة قيمة متغير
- استخدام متغير داخل شرط

انتبه

- عند استخدامك للكتل لتنفيذ مهمة برمجية سيقوم البرنامج بتنفيذ الأوامر حسب ترتيبك للكتل واحدة تلو الأخرى وهذا ما يعرف بالأوامر المتتالية.
- يوجد مظهرين لمجموعات الكتل يمكنك التنقل بينها عن طريق الضغط على رقم المظهر

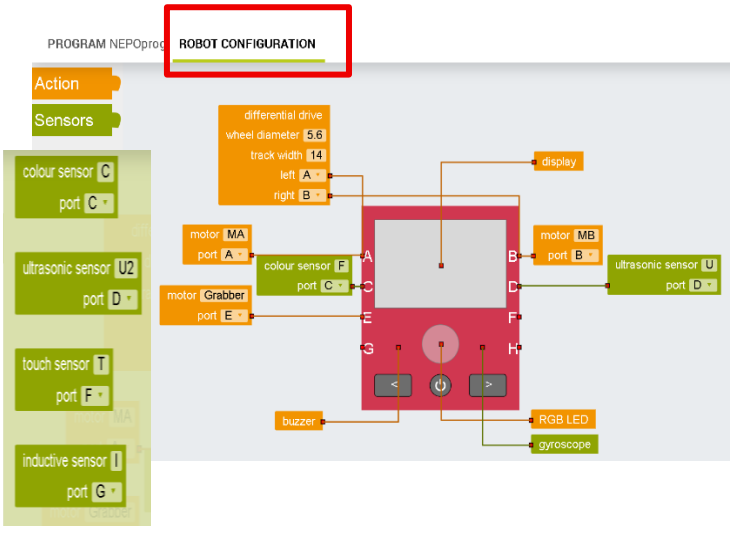
المستشعرات

المستشعرات هي الطريقة التي يتفاعل بها الروبوت مع العالم من حوله، تمامًا كما نفعل نحن باستخدام حواسنا .

أنواع المستشعرات في محاكي روبوت RCJ Rescue داخل بيئة المحاكاة Open Roberta Lab.

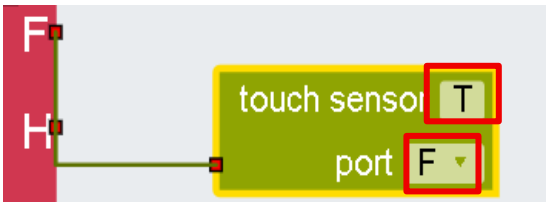
اسم المستشعر	وظيفته	موقعه في الروبوت الافتراضي
مستشعر اللمس Touch sensor	يكشف عن اصطدام الروبوت بعائق خلفي (يستجيب عند الضغط عليه).	الخلف
مستشعر الألوان Color sensor	التعرف على ألوان الأسطح التي يتحرك فوقها الروبوت لتحديد المسار المناسب	في الأسفل باتجاه الأرض
مستشعر فوق صوتي Ultrasonic sensor	يقيس المسافة بين الروبوت والأجسام أمامه باستخدام الموجات فوق الصوتية	الأمام
مستشعر الجيروسكوب Gyroscope sensor	يقيس زاوية دوران الروبوت واتجاهه ويساعد في الحفاظ على التوازن وتحديد ما إذا كان الروبوت قد انحرف أو دار أثناء التنقل	مدمج داخل الروبوت

المستشعرات في منافذ محاكي الروبوت



لاحظ وجود جميع المستشعرات موضحة بالمنفذ الموصلة به على معالج الروبوت الافتراضي لتحاكي توصيلها في منافذ الروبوت الحقيقي، ولكن لا يوجد مستشعر لمس ظاهر.

لإضافة أي مستشعر جديد اضغط على بطاقة sensors تظهر لك مجموعة المستشعرات التي يمكنك اضافتها.

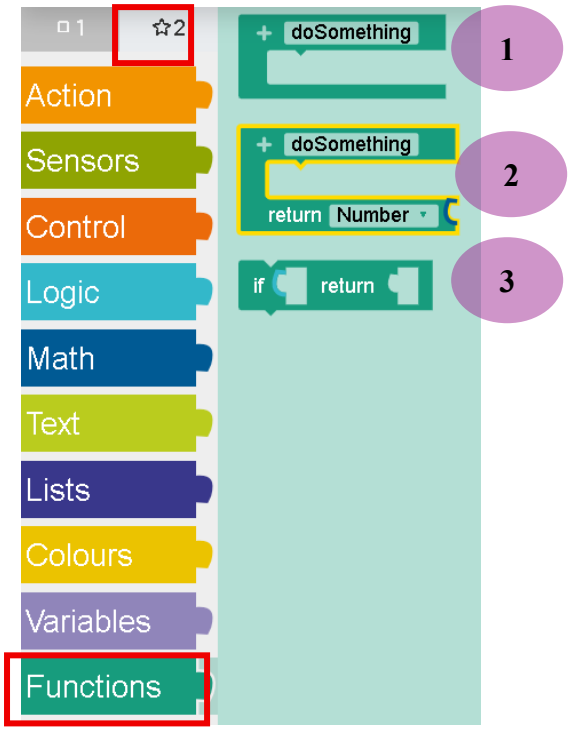


في الصورة الظاهرة أمامك لاحظ وجود حقلين
الأول يظهر حرف اختصار (T) اسم المستشعر الذي تم تحديده
الثاني يظهر لك حرف منفذ المعالج الذي تم توصيل المستشعر عليه
اكتب رمز المنفذ الظاهر ..... port

الدوال (Functions)

الدالة: هي مجموعة من الأوامر (الكتل) نضعها معًا تحت اسم معيّن، ونستطيع استخدام هذا الاسم لاحقًا لتشغيل كل تلك الأوامر مرة واحدة، بدل أن نعيد كتابتها كل مرة.

كيفية إنشاء دالة؟



1. لإنشاء دالة بدون قيمة راجعة (void function)

مثال:

انشاء دالة عند اصطدام الروبوت بعائق تصدر صوت تنبيه وتشغل ضوء.

2. لإنشاء دالة تُعيد قيمة (return function)، عددية أو نصية أو منطقية حسب نوع القيمة المحدد.

مثال:

انشاء دالة تُعيد عدد الدورات (rotations) التي قام بها المحرك ثم استخدام ذلك العدد داخل عملية حسابية.

3. تُستخدم داخل الدوال عندما نريد إعادة قيمة فقط إذا تحقق شرط معيّن.

مثال:

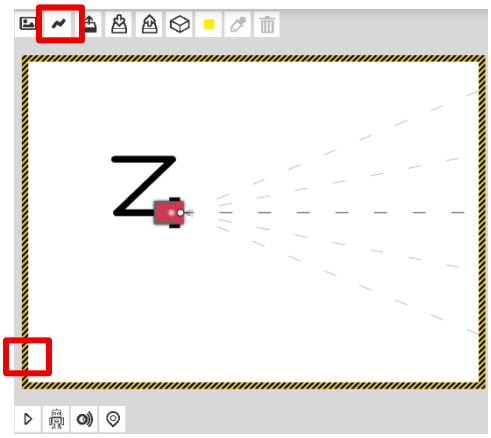
نريد إعادة القيمة فقط إذا كانت المسافة أكبر من 10 سم

if (المسافة > 10) return المسافة

هذا مفيد عندما نريد التحكم في ما إذا كانت الدالة تُعيد قيمة أم لا، حسب شرط معين.

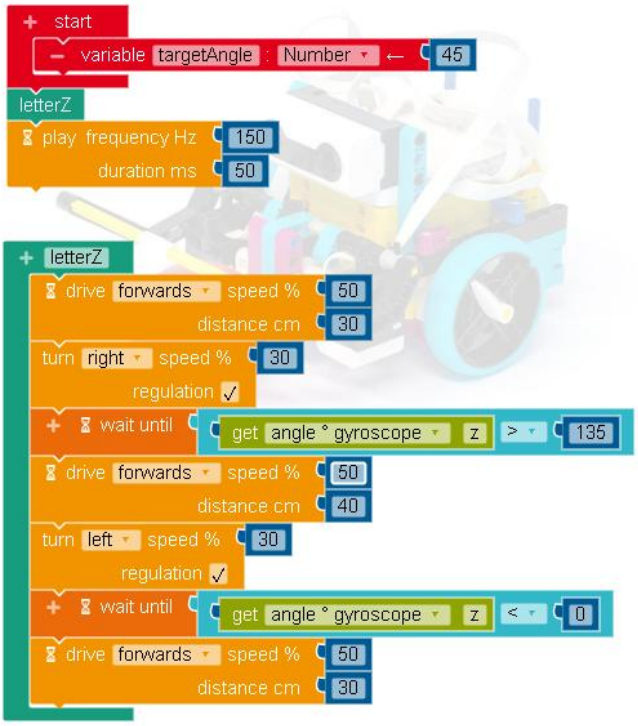
المهمة الأولى

أنشئ مشروع جديد اجعل الروبوت الذكي يرسم مسارًا يشبه حرف Z باستخدام مستشعر Gyroscope، يحسب الزوايا بدقة ويتنقل كما لو كان على طريق جبلي متعرّج، وعند انتهائه يصدر صوت بتردد معين.



✓ التهيئة العامة

- يتم انشاء متغير targetAngle وتعيين قيمة ابتدائية له 45.
- استدعاء
- letterZ
- تفعيل صوت



✓ الدالة letterZ

- تحرك للأمام 30 سم.
- دوران يمين حتى زاوية 135°.
- تحرك للأمام 40 سم.
- دوران يسار حتى زاوية 0° (أي رجوع لاتجاه البداية).
- تحرك للأمام 30 سم.

✓ النتيجة

رسم شكل حرف Z باستخدام زوايا محددة بدقة عبر مستشعر Gyro.

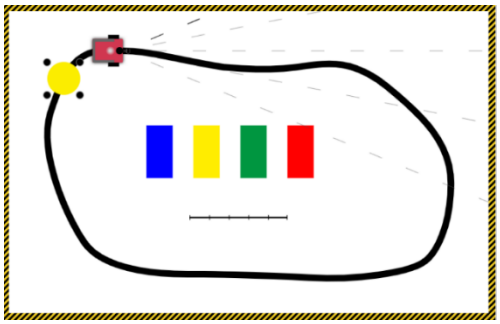
التطبيق

تتبع الخط

ورقة عمل

1- الاعداد

- افتح بيئة المحاكاة التعليمية RCJ RescueOnlineSim في منصة Open Roberta Lab.
- شغل المحاكي بالضغط على SIM
- قم بتغيير خلفية المنصة للوصول للشكل الظاهر أمامك.
- إضافة مساحة لونية إلى المنصة كما هو ظاهر أمامك.

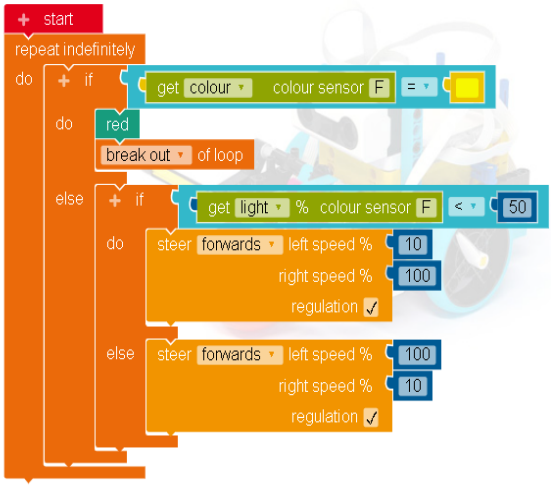


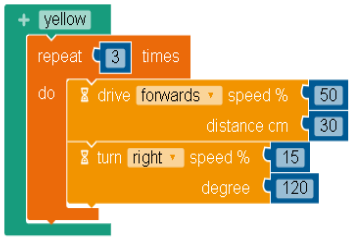
1- مهمتك

ساعد الروبوت على تتبع الخط، الأسود على أرضية بيضاء باستخدام حساس اللون، عند الوصول إلى اللون الأصفر، يتم استدعاء دالة yellow لرسم شكل مثلث علما بان طول ضلعة 30 سم ثم يتوقف الروبوت.

2- بناء البرنامج

- مستشعر اللون في وضع الضوء المنعكس light لتتبع اللون الأسود.
- مستشعر اللون في وضع colour لاستشعار اللون الأصفر وانهاء البرنامج.
- استخدم كتلة التكرار إلى مالا نهاية.
- أضف دالة yellow لجعل الروبوت يرسم الشكل المثلث.





3- **تلميحات**

- استعمل شرط التوقف في أعلى الكود لضمان عدم تجاوزه
- استخدم steer بدلًا من drive لتسهيل التحكم بالاتجاه.

4- **شغل وجرب**

- راقب الروبوت هل يتتبع الطريق بشكل مناسب.
- جرب الكتل خطوة بخطوة وفي كل مره استخدم أداة استعادة اعدادات المنصة للحصول على نتيجة صحيحة.
- عدل السرعة أو قراءة المستشعر إذا خرج الروبوت عن المسار.

5- **احفظ البرنامج**

- باسم FollowLine على حسابك الشخصي.

6- **صدر البرنامج على الجهاز.**

7- **تحد إضافي**

1.إضافة رسالة "أكملت الطريق بنجاح" لمدة 3 ثوان.

تحدي الأبطال

تحدي الأبطال
Champions Challenge

هو مشروع يهدف إلى قياس مدى تمكّن المتعلم من المهارات البرمجية من خلال تطبيقها في بيئات محاكاة متنوعة، بما يتيح له القدرة على توظيف هذه المهارات في أي بيئة برمجية أخرى. ويعزز المشروع التكامل العملي للمفاهيم المكتسبة في إطار تجربة جديدة.

الأهداف الرئيسية لتحدي الأبطال

تطبيق المفاهيم المكتسبة: التأكد من قدرة المتعلم على تطبيق المبادئ الأساسية للبرمجة والروبوتات التي تعلمها في Open Roberta Lab في بيئة محاكاة جديدة ومختلفة .

تعزيز مهارات حل المشكلات: تنمية قدرة المتعلم على تحليل المشكلات وتصميم حلول فعالة وتنفيذها باستخدام الروبوت الافتراضي، وذلك في سياق تحدي جديد يتطلب تفكيراً إبداعياً وحلولاً مختلفة .

التكيف مع بيئات برمجية مختلفة: قياس مرونة المتعلم في التكيف مع واجهات وكتل برمجية مختلفة، مما يعزز فهمه العميق للمفاهيم الأساسية بعيدًا عن الاعتماد على بيئة واحدة.

تطوير التفكير المنطقي والتحليلي: تشجيع المتعلم على التفكير بشكل منطقي لتحديد أفضل المسارات والقرارات للروبوت لتحقيق الأهداف المطلوبة ضمن التحدي باستخدام التسلسل البرمجي المناسب.

بناء الثقة بالنفس: منح المتعلم فرصة لإظهار قدراته بشكل مستقل، مما يعزز ثقته بمهاراته البرمجية وقدرته على مواجهة التحديات التقنية.

التعلم الذاتي: تشجيع المتعلم على استكشاف وتعلم أدوات ومفاهيم جديدة بشكل مستقل للتعامل مع متطلبات بيئات مختلفة من البرمجة.

تقدير الذات والاستقلالية: عبر تقديم المشروع كمهمة غير مُدرسة من قبل المعلم .

حمل برنامج المحاكاة
Simulator Micromelon Robot
لتنفيذ التحدي ومعاينة الروبوت

حمل برنامج
Micromelon Code Editor
لكتابة التعليمات البرمجية

تحدي الأبطال
Free Space

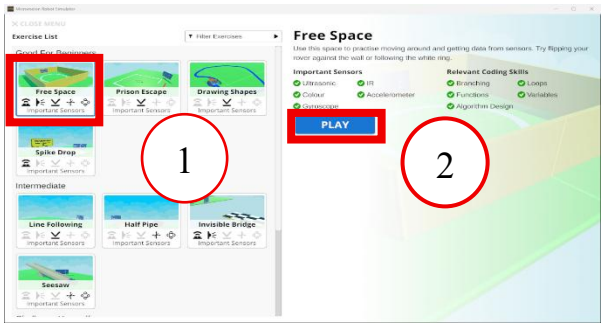
نتاجات التعلم

1	برمجة الروبوت في بيئة Micromelon Code Editor ليُنفذ المهام المطلوبة بدقة وكفاءة.
2	التعامل مع واجهة المستخدم واستخدام الكتل البرمجية في برنامج Micromelon Code Editor دون توجيه مباشر من المعلم .
3	تجربة التعليمات البرمجية واختبار البرنامج في بيئة المحاكاة Micromelon Robot Simulator .
4	تطبيق المفاهيم البرمجية الأساسية (الحركة، الصوت) في برنامج Micromelon Code Editor .
5	برمجة الروبوت ليتحرك إلى الأمام بمسافة معينة وعمل مسار دائري .
6	التعديل على التعليمات البرمجية و تصحيح الأخطاء لتحسين أداء الروبوت .

أهداف تحدي Free Space

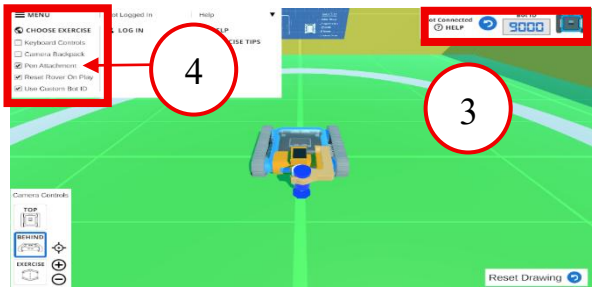
- ربط مهارة استخدام كتل الحركة والصوت وتفعيل القلم في بيئة البرمجة Micromelon Code Editor.
- تحقيق التوظيف العملي لكتل الحركة والصوت وتفعيل القلم لرسم المسار عبر Free Space في المحاكي Micromelon Robot Simulator .

حل مقترح لتحدي Free Space



1- من منصة Micromelon Robot Simulator افتح بيئة المحاكاة التعليمية Free Space.

2- شغّل المحاكي بالضغط على PLAY.



3- من Bot ID عرف المحاكاة بالرقم 9000.

4- من قائمة MENU فعل الخيار Pen Attachment .

5- من منصة Micromelon Code Editor،
أنشئ مشروع جديد .

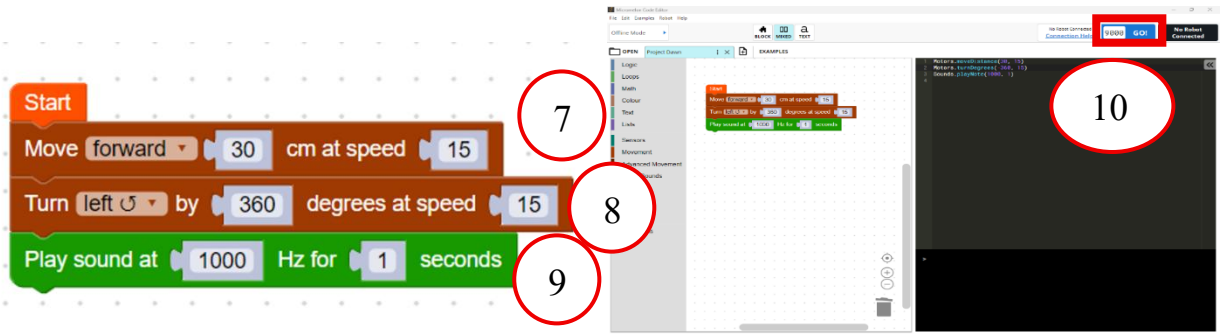
6- في الخانة داخل زر GO أدرج رقم معرف المحاكاة الرقم 9000 .

7- من المجموعة Movement اختر الكتلة البرمجية Move forward cm at speed وحدد لها قيمة المسافة 30 وقيمة السرعة 15 .

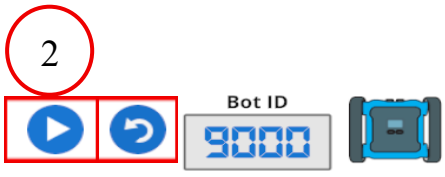
8- من المجموعة Movement اختر الكتلة البرمجية Turn left degrees at speed وحدد لها قيمة زاوية دوران 360 وقيمة السرعة 15 .

9- من المجموعة Lights/Sounds اختر الكتلة البرمجية Play sound at Hz for seconds.

10- أضغط على الزر GO لتشغيل البرنامج .



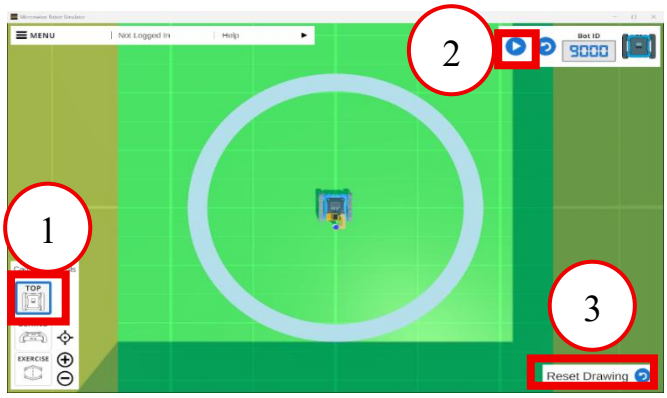
من منصة المحاكاة (Micromelon Robot Simulator)



1- من منصة المحاكاة (Micromelon Robot Simulator)
اختر الخيار المناسب لك وليكن Top

2- شغل البرنامج بالضغط على زر التشغيل .
راقب حركة الروبوت داخل تحدي Free Space.

3- اضغط على زر Reset Drawing.
جرب الحل بأكثر من طريقة :
مثلاً: زيادة أو نقصان المسافة والسرعة وتغير اتجاه الدوران.



يمكنك التعديل على البرمجة بعدة طرق مختلفة منها:
- تغير أتجاه الدوران .
- زيادة أو نقصان قيمة السرعة.
- تغير قيمة المسافة.

نتاجات التعلم

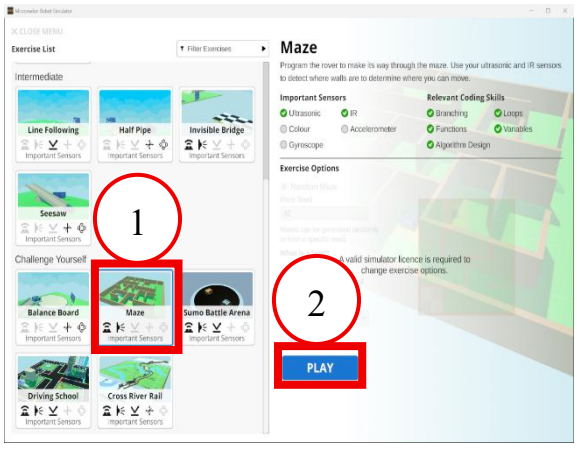
1	برمجة الروبوت في بيئة Micromelon Code Editor لِينفذ المهام المطلوبة بدقة وكفاءة.
2	التعامل مع واجهة المستخدم واستخدام الكتل البرمجية في برنامج Micromelon Code Editor دون توجيه مباشر من المعلم .
3	تجربة التعليمات البرمجية واختبار البرنامج في بيئة المحاكاة Micromelon Robot Simulator.
4	تطبيق المفاهيم البرمجية الأساسية (مثل الحلقات التكرارية، الجمل الشرطية، المتغيرات) في برنامج Micromelon Code Editor .
5	تحليل المشكلة المطروحة في تحدي Maze .
6	برمجة الروبوت لحل المتاهة باستخدام مستشعرUltrasonic .
7	التعديل على التعليمات البرمجية وتصحيح الأخطاء لتحسين أداء الروبوت للوصول إلى نهاية المتاهة بأقل وقت وأكثر كفاءة .
8	اظهار مهارات التفكير المنطقي والاستنتاج لتحديد أفضل الحلول والاستراتيجيات لاجتياز المتاهة.

أهداف تحدي Maze

4- ربط مهارة استخدام مستشعر Ultrasonic و حلقة التكرار ، و الأوامر الشرطية في بيئة البرمجة Micromelon Code Editor .

5- التوظيف العملي لمستشعر Ultrasonic من خلال سياق فهم تأثير السرعة على قراءة المستشعر للمسافة و قرار اتخاذ التفاف دوران اتجاه الزاوية بحيث يحاكي مهمة اجتياز المتاهة (Maze) في المحاكي Micromelon Robot Simulator.

حل مقترح لتحدي Maze

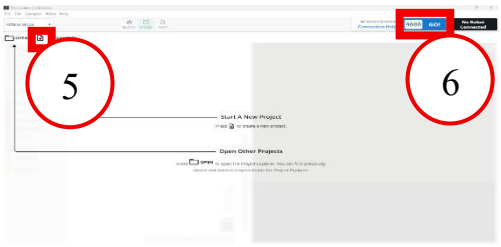


1- من منصة Micromelon Robot Simulator افتح بيئة المحاكاة التعليمية Maze .

2- شغّل المحاكي بالضغط على PLAY.

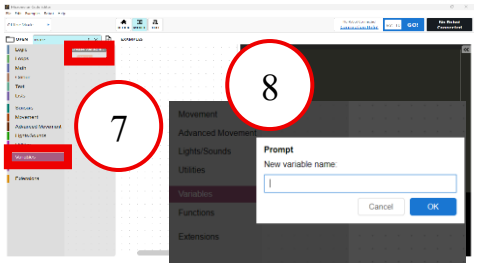
3- من Bot ID عرف المحاكاة بالرقم 9000.

4- من Camera Controls عدل على رؤية المشهد وليكن الخيار Top مع التصغير.



5- من منصة Micromelon Code Editor،
أنشئ مشروع جديد واختر له اسم .

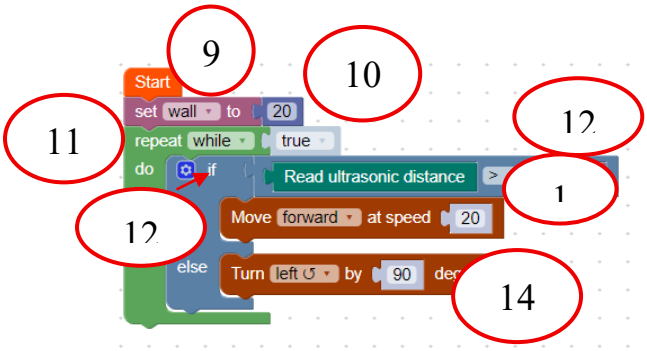
6- في الخانة داخل زر GO أدرج رقم معرف المحاكاة الرقم 9000 .



7- من المجموعة Variables اختر الأمر Create Variables .

8- تظهر شاشة Prompt اختر اسم للمتغير يمثل الحاجز وليكن wall واضغط على زر OK .

9- من المجموعة Variables أدرج الكتلة البرمجية set wall.



10- من المجموعة Math أدرج الكتلة البرمجية الخاصة بالعدد وحدد لها قيمة 20 (المسافة بين الروبوت والحائط) .

11- من المجموعة Loops أدرج الكتلة البرمجية لأمر التكرار while repeat true.

12- في الحلقة التكرارية repeat while true داخل الأمر do أدرج الكتل البرمجية التالية :

a) من المجموعة Logic أدرج كتلة الأوامر الشرطية if -else

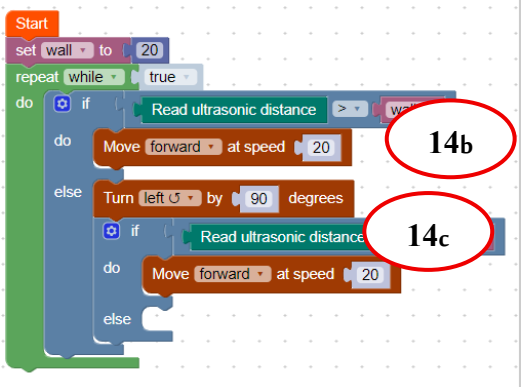
b) حدد الشرط : من الوحدة البرمجية sensors أدرج الكتلة البرمجية Read ultrasonic distance ، حدد رمز أكبر ، ومن الوحدة البرمجية Variables أدرج الكتلة البرمجية للمتغير wall .

13- داخل do في كتلة الأوامر الشرطية if -else :

- من المجموعة Movement أدرج الكتلة البرمجية Move forward at speed وحدد قيمة السرعة 20 .

14- داخل else في كتلة الأوامر الشرطية if -else :

a) من المجموعة Movement أدرج الكتلة البرمجية Turn left by degrees وحدد قيمة الزاوية 90.

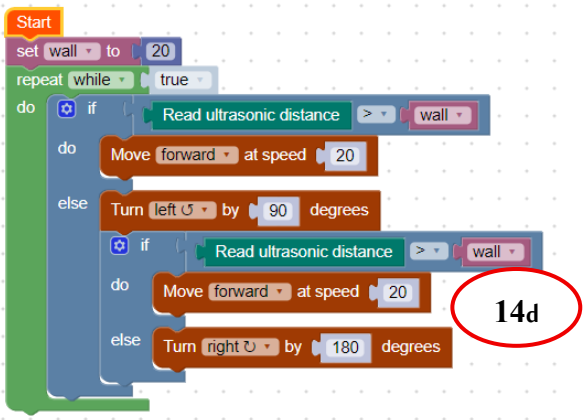


b) من المجموعة Logic أدرج كتلة الأوامر الشرطية if -else

حدد الشرط : من المجموعة sensors أدرج الكتلة البرمجية Read ultrasonic distance ، اختر رمز أكبر من ، ومن المجموعة Variables أدرج الكتلة البرمجية للمتغير wall.

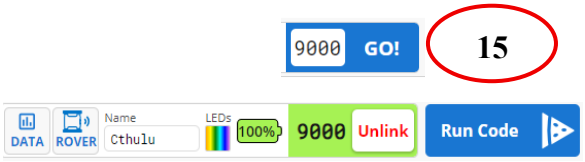
c) داخل do في كتلة الأوامر الشرطية if -else:

من المجموعة Movement أدرج الكتلة البرمجية Move forward at speed وحدد قيمة السرعة 20 .



d) داخل else في كتلة الأوامر الشرطية if -else:
من المجموعة Movement أدرج
الكتلة البرمجية Turn left By degrees
غير الاتجاه إلى right وحدد قيمة الزاوية
180.

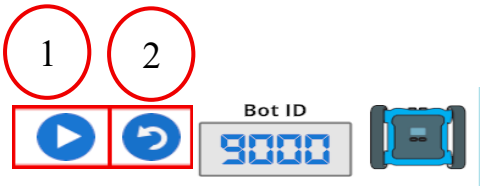
15- أضغط على الزر GO لتشغيل البرنامج.



من منصة المحاكاة (Micromelon Robot Simulator)

1- شغل محاكاة الروبوت بالضغط على زر التشغيل ،
ثم راقب حركة الروبوت داخل تحدي Maze .

2- لإعادة ضبط مكان الروبوت في نقطة البداية اضغط زر إعادة الضبط .



يمكنك التعديل على البرمجة بعدة طرق مختلفة منها:
- التبديل بين أتجاه الالتفاف حيث يبدأ بالالتفاف يميناً ثم يساراً .
- زيادة قيمة السرعة.
- تغير قيمة الحاجز.

تحدي الأبطال
line Following

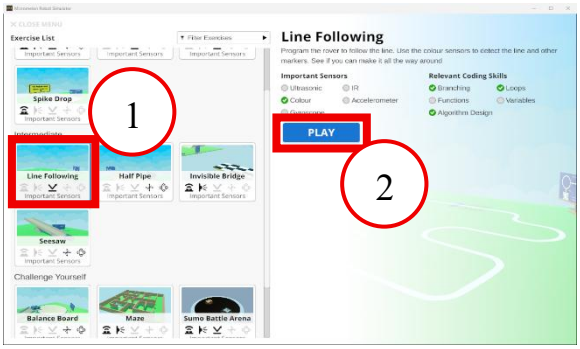
نتاجات التعلم

1	برمجة الروبوت في بيئة Micromelon Code Editor ليُنفذ المهام المطلوبة بدقة وكفاءة.
2	التعامل مع واجهة المستخدم واستخدام الكتل البرمجية في برنامج Micromelon Code Editor دون توجيه مباشر من المعلم .
3	تجربة التعليمات البرمجية واختبار البرنامج في بيئة المحاكاة Micromelon Robot Simulator.
4	تطبيق المفاهيم البرمجية الأساسية (مثل الحلقات التكرارية، الجمل الشرطية المتداخلة) في برنامج Micromelon Code Editor .
5	تحليل مسار تتبع الخط في تحدي line Following.
6	برمجة الروبوت لتحدي line Following باستخدام مستشعر الألوان Color sensor.
7	التعديل على التعليمات البرمجية وتصحيح الأخطاء لتحسين أداء الروبوت للوصول إلى نقطة النهاية.
8	اظهار مهارات التفكير المنطقي والاستنتاج لتحديد أفضل الحلول والاستراتيجيات لاجتياز التحدي.

أهداف تحدي line Following

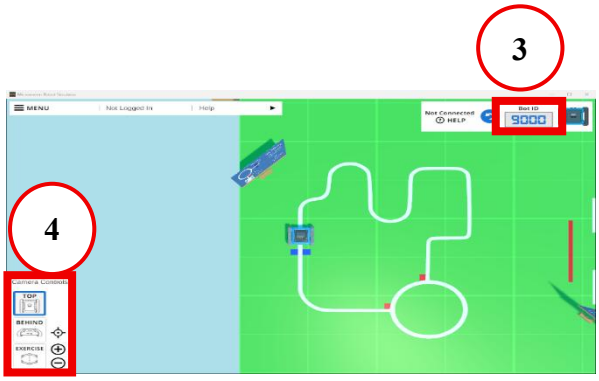
- ربط مهارة استخدام مستشعر الألوان Color sensor ، وحلقة التكرار ، و الأوامر الشرطية في بيئة البرمجة Micromelon Code Editor.
- التوظيف العملي لمستشعر الألوان Color sensor من خلال سياق سردي تحفيزي يظهر العلاقة بين قراءة مستشعر الألوان مع سرعة الروبوت والزوايا والمتعطفات بحيث يحاكي مهمة تحدي line Following في المحاكي Micromelon Robot Simulator .

حل مقترح لتحدي line Following



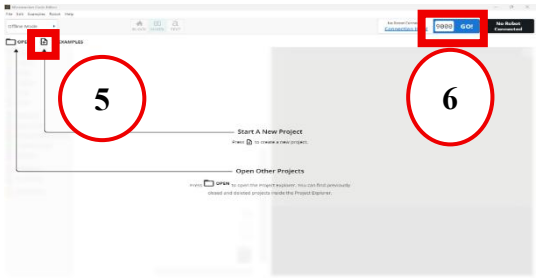
1- من منصة Micromelon Robot Simulator افتح بيئة المحاكاة التعليمية line Following.

2- شغّل المحاكي بالضغط على PLAY.



-3 من Bot ID عرف المحاكاة بالرقم 9000.

-4 من Camera Controls عدل على رؤية المشهد وليكن الخيار Top مع التصغير

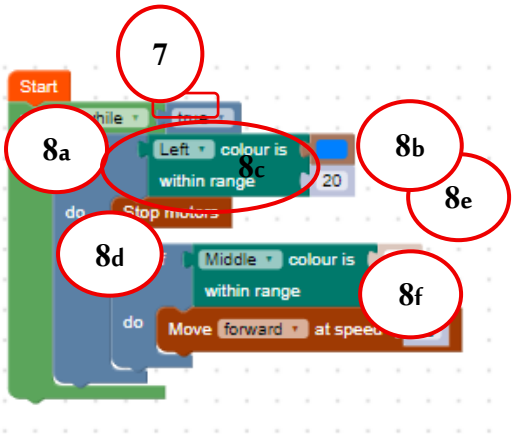


-5 من منصة Micromelon Code Editor ، أنشئ مشروع جديد باسم linefollowing.

-6 في الخانة داخل زر GO أدرج رقم معرف المحاكاة الرقم 9000.

-7 من المجموعة Loops أدرج الكتلة البرمجية لأمر التكرار while true repeat.

-8 في الحلقة التكرارية repeat while true داخل الأمر do أدرج الكتل البرمجية التالية :



a) من المجموعة Logic أدرج كتلة الأوامر الشرطية if -else

b) حدد الشرط : من المجموعة sensors أدرج الكتلة البرمجية مستشعر الألوان Middle colour is within range، اختر خيار القراءة من Left ، اختر قراءة اللون الأزرق ،اجعل قيمة within range 20.

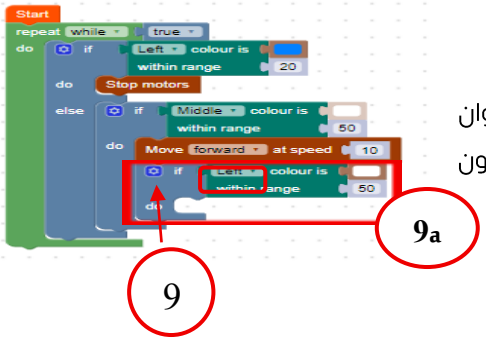
c) داخل do في كتلة الأوامر الشرطية if -else :

من المجموعة Movement أدرج الكتلة البرمجية Stop motors .

d) داخل else في كتلة الأوامر الشرطية if -else : من المجموعة Logic أدرج كتلة الأوامر الشرطية if.

e) حدد الشرط: من المجموعة sensors أدرج الكتلة البرمجية مستشعر الألوان Middle colour is within range ، اختر قراءة اللون الأبيض ، اجعل قيمة within range 50.

f) داخل do في كتلة الأوامر الشرطية if: من المجموعة Movement أدرج الكتلة البرمجية Move forward at speed وحدد قيمة السرعة 10.



-9 من المجموعة Logic أدرج كتلة الأوامر الشرطية if :

a) حدد الشرط : من المجموعة sensors أدرج الكتلة البرمجية مستشعر الألوان Middle colour is within range ، اختر خيار القراءة من Left ، اختر قراءة اللون الأبيض ، اجعل قيمة within range 50 .

b) داخل do في كتلة الأوامر الشرطية if: من المجموعة Movement أدرج الكتلة البرمجية Turn Left at speed, حدد قيمة السرعة 5.

10- من المجموعة Logic أدرج كتلة الأوامر الشرطية if :

a) حدد الشرط : من المجموعة sensors أدرج الكتلة البرمجية مستشعر الألوان Middle colour is within range ، اختر خيار القراءة من Right ، اختر قراءة اللّون الأبيض ، اجعل قيمة 50 within range .

b) داخل do في كتلة الأوامر الشرطية if: من المجموعة Movement أدرج الكتلة البرمجية Turn right at speed, حدد قيمة السرعة 5.

11- من المجموعة Logic أدرج كتلة الأوامر الشرطية if :

حدد الشرط : من المجموعة sensors أدرج الكتلة البرمجية مستشعر الألوان Middle colour is within range ، اختر خيار القراءة من Right ، اختر قراءة اللّون الأحمر ، اجعل قيمة 50 within range .

12- داخل do في كتلة الأوامر الشرطية المتداخلة if : من المجموعة Loops أدرج الكتلة البرمجية لأمر التكرار repeat while true .

a) استبدل true : من المجموعة sensors أدرج الكتلة البرمجية مستشعر الألوان within range is Middle colour، اختر خيار القراءة من Left ، اختر قراءة اللّون الأبيض ، اجعل قيمة 20 within range .

b) داخل do في كتلة التكرار while true repeat: من المجموعة Movement أدرج الكتلة البرمجية Turn Left at speed, حدد قيمة السرعة 5 .

13- من المجموعة Logic أدرج كتلة الأوامر الشرطية if:

a) حدد الشرط : من المجموعة sensors أدرج الكتلة البرمجية مستشعر الألوان Middle colour is within range، اختر خيار القراءة من Left ، اختر قراءة اللّون الأحمر، اجعل قيمة 50. within range

b) من المجموعة Movement أدرج الكتلة البرمجية Turn Left at speed حدد قيمة السرعة 5.

14- أضغط على الزر GO لتشغيل البرنامج .

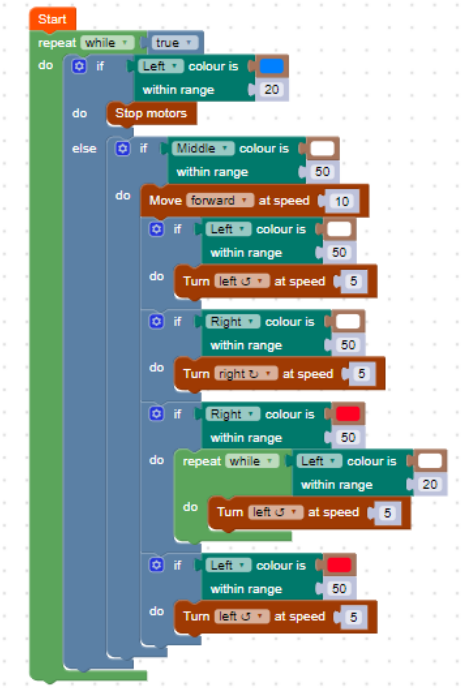
من منصة المحاكاة (Micromelon Robot Simulator)

1 2

1- شغل محاكاة الروبوت بالضغط على زر التشغيل ،

ثم راقب حركة الروبوت داخل تحدي line Following.

2- لإعادة ضبط مكان الروبوت في نقطة البداية اضغط زر إعادة الضبط .



يمكنك التعديل على البرمجة بعدة طرق مختلفة منها:

- التعديل على القيم في البرنامج .
- زيادة قيمة السرعة.
- التعديل على الاستراتيجية المتبعة في قراءة اللّون الأحمر:
  - حذف أمر التكرار .
  - قراءة اللّون الأحمر من الاتجاهين الأيمن والأيسر .

تحدي الأبطال

Sumo Battle Arena

نتاجات التعلم

1	برمجة الروبوت في بيئة Micromelon Code Editor لِينفذ المهام المطلوبة بدقة وكفاءة.
2	التعامل مع واجهة المستخدم واستخدام الكتل البرمجية في برنامج Micromelon Code Editor دون توجيه مباشر من المعلم .
3	تجربة التعليمات البرمجية واختبار البرنامج في بيئة المحاكاة Micromelon .Robot Simulator
4	تطبيق المفاهيم البرمجية الأساسية (مثل الحلقات التكرارية، الجمل الشرطية، المتغيرات) في برنامج Micromelon Code Editor .
5	تحليل المشكلة المطروحة في تحدي Sumo Battle Arena.
6	برمجة الروبوت لتحدي السومو باستخدام المستشعرات Color sensor ، Ultrasonic .
7	التعديل على التعليمات البرمجية وتصحيح الأخطاء لتحسين أداء الروبوت للوصول إلى أفضل أداء في تحدي السومو.
8	اظهار مهارات التفكير المنطقي والاستنتاج لتحديد أفضل الحلول والاستراتيجيات للفوز بالتحدي.

أهداف تحدي Sumo Battle Arena

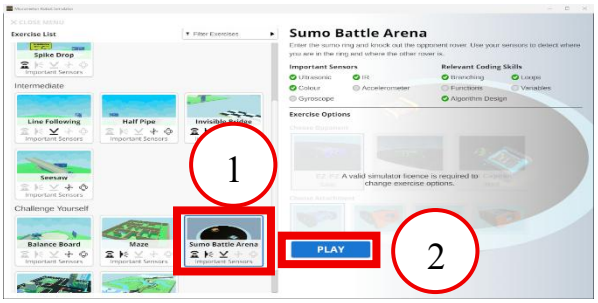
ربط مهارة استخدام مستشعر المسافة Ultrasonic و مستشعر الألوان Color sensor وحلقة التكرار و الأوامر الشرطية في بيئة البرمجة Micromelon Code Editor.

التوظيف العملي لمستشعر المسافة Ultrasonic و مستشعر الألوان Color sensor من خلال سياق سردي تحفيزي يحاكي مهمة تحدي السومو في المحاكي Micromelon Robot Simulator .

حل مقترح لتحدي Sumo Battle Arena

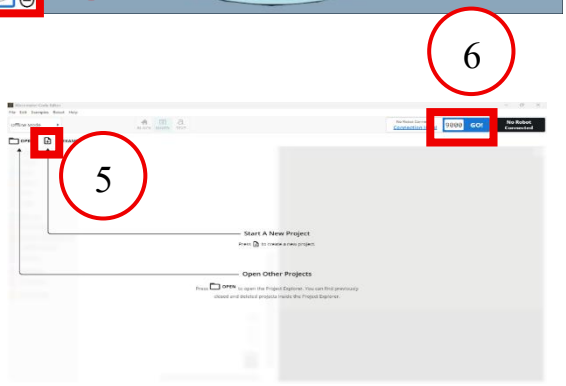
1- من منصة Micromelon Robot Simulator افتح بيئة المحاكاة التعليمية Sumo Battle Arena.

2- شغّل المحاكي بالضغط على PLAY.



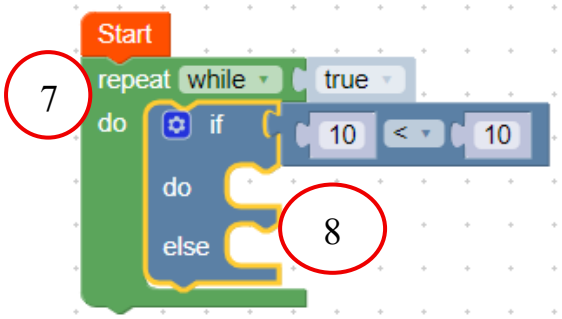
-3 من Bot ID عرف المحاكاة بالرقم 9000.

-4 من Camera Controls عدل على رؤية المشهد وليكن الخيار EXERCISE مع التصغير.



-5 من منصة Micromelon Code Editor أنشئ مشروع جديد باسم sumo .

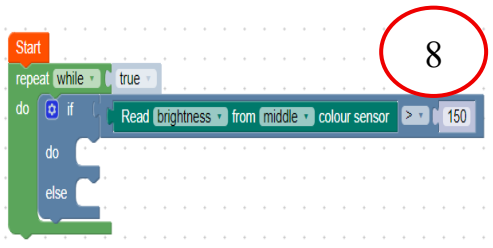
-6 في الخانة داخل زر GO أدرج رقم معرف المحاكاة الرقم 9000



-7 من المجموعة Loops أدرج الكتلة البرمجية لأمر التكرار repeat while true.

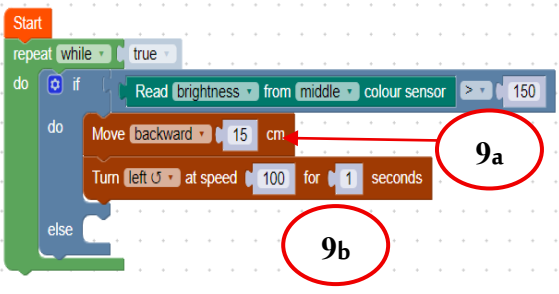
-8 داخل do في الحلقة التكرارية repeat while true أدرج الكتل البرمجية التالية :

a) من المجموعة Logic أدرج كتلة الأوامر الشرطية - if else.



b) حدد الشرط : من المجموعة sensors أدرج الكتلة البرمجية مستشعر الألوان

Read hue from middle colour sensor، اختر خيار قراءة انعكاس الضوء brightness ، حدد رمز أكبر من، واجعل قيمة سطوع الضوء 150.



-9 داخل do في كتلة الأوامر الشرطية if -else :

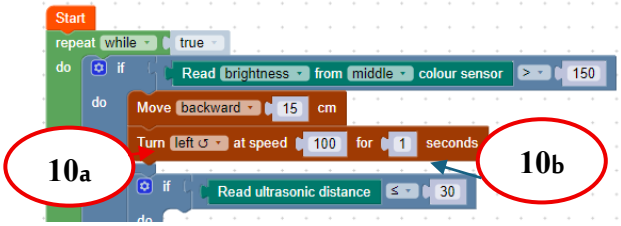
a) من المجموعة Movement : أدرج الكتلة البرمجية Move forward cm, اختر الأمر backward وحدد قيمة المسافة 15.

b) أدرج الكتلة البرمجية Turn left at speed for seconds وحدد قيمة السرعة 100وحدد قيمة الزمن 1.

10- داخل else في كتلة الأوامر الشرطية if-else:

a) من المجموعة Logic أدرج كتلة الأوامر الشرطية if

b) حدد الشرط : من المجموعة sensors أدرج الكتلة البرمجية Read ultrasonic distance، اختر رمز أصغر من أو يساوي ، حدد القيمة 30.

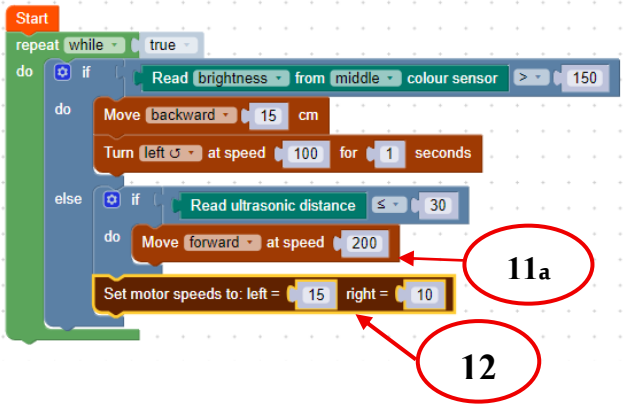


11- داخل do في كتلة الأوامر الشرطية if:

a) من المجموعة Movement أدرج الكتلة البرمجية at Move forward speed

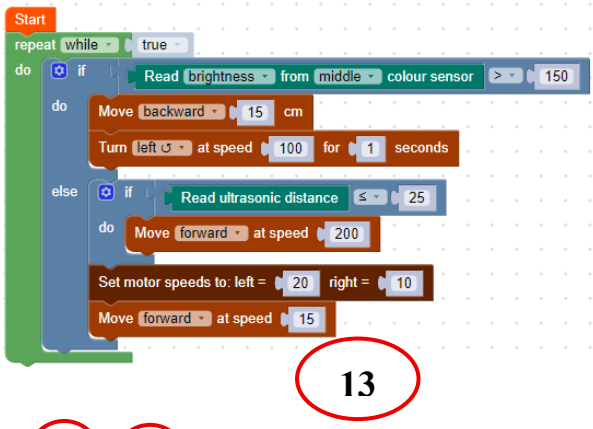
وحدد قيمة السرعة 200 .

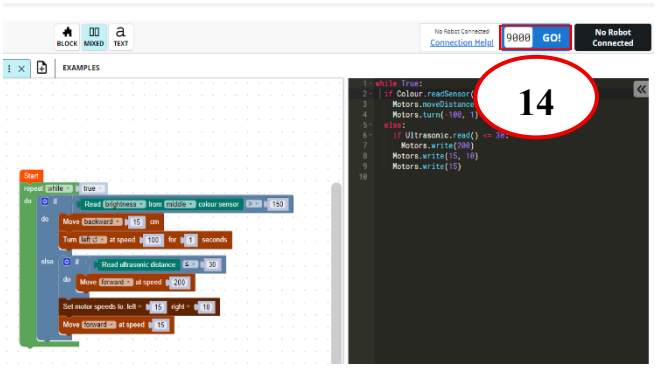
12- من المجموعة Advanced Movement أدرج الكتلة البرمجية Set motor speeds to left= right= حدد قيمة left =15، right =10



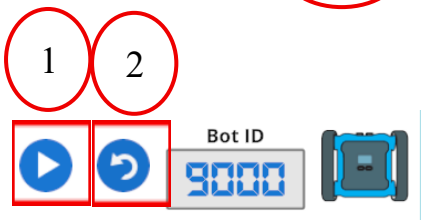
13- أدرج الكتلة البرمجية Move forward cm من المجموعة Movement, وحدد قيمة المسافة 15.

14- أضغط على الزر GO لتشغيل البرنامج.





من منصة المحاكاة (Micromelon Robot Simulator)



1- شغل محاكاة الروبوت بالضغط على زر التشغيل ، ثم راقب حركة الروبوت داخل تحدي Sumo Battle Arena.

2- لإعادة ضبط مكان الروبوت في نقطة البداية اضغط زر إعادة الضبط .

يمكنك التعديل على البرمجة بعدة طرق مختلفة منها:

- التعديل على القيم في البرنامج .
- زيادة قيمة السرعة.
- التعديل على الاستراتيجية المتبعة في تمييز اللون الأبيض والأسود.
- التعديل بقيمة الالتفاف والدوران والزاوية.

المراجع

1. **OpenAI. (2025).** ***Image of LEGO SPIKE Prime [AI-generated]*****. ChatGPT. Retrieved September 3, 2025, from https://chat.openai.com/**
2. **Open Roberta Lab. (n.d.).** ***Open Roberta Lab*****. Retrieved from https://lab.open-roberta.org**
3. **LEGO Education. (n.d.).** ***LEGO® Education SPIKE™ Prime*****. LEGO. Retrieved from https://education.lego.com/**
4. **مركز صباح الأحمد للموهبة والإبداع. (2018). الدليل الإرشادي للروبوت التعليمي.**
5. **Micromelon. (n.d.).** ***Simulator Micromelon Robot*** **[Computer software]. Micromelon. https://micromelon.com**
6. **Micromelon. (n.d.).** ***Micromelon Code Editor*** **[Computer software]. Micromelon. https://micromelon.com**