وزارة التربية والتعليم

رؤية عُمان 2040 OmanVision

نتقدم بثقة
Moving Forward with Confidence

تقنية المعلومات ٦

الفصل الدراسي الثاني

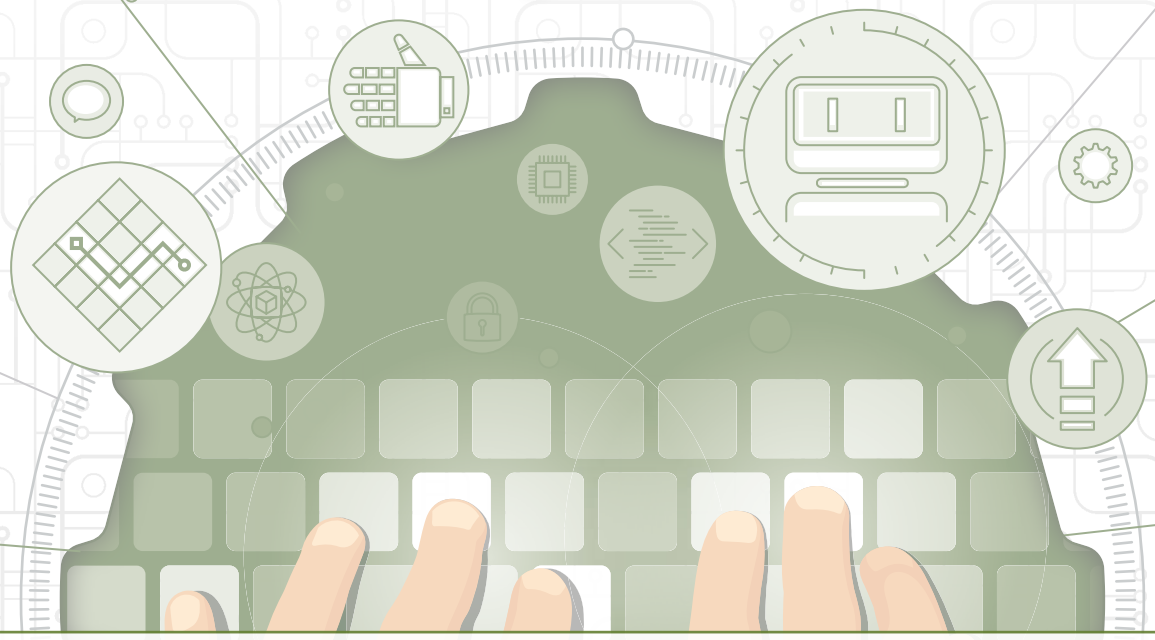
الطبعة التجريبية ١٤٤٧ هـ - ٢٠٢٥ م

binarylogic

وزارة التربية والتعليم

رؤية عُمان 2040
نتقدم بثقة
Moving Forward with Confidence

تقنية المعلومات ٦



الاسم: .....

الشّعبة: .....

الفصل الدراسي الثاني

الطبعة التجريبية ١٤٤٧ هـ - ٢٠٢٥ م

binarylogic

نُشِرَ هـذا الكتـاب بموجـب اتفاقيـة خاصة بين شركـة Binary Logic SA ووزارة التربية والتعليم في سـلطنة عُمان (عقد رقم ٢/٢٠٢٤) للاستخدام في سلطنة عُمان.

حقوق النشر © Binary Logic SA 2025

كما أنَّ جميـع الحقـوق محفوظـة، ولا يجـوز نسـخ أي جـزء من هذا المنشـور أو تخزينـه في أنظمة استرجاع البيانـات، أو نقلـه بـأي شـكل مـن الأشـكال، أو بـأي وسـيلة إلكترونيـة أو ميكانيكيـة أو بالنسـخ الضـوئي أو التسجيل، أو غير ذلك دون إذن كتابي من الناشرين.

كما يُرجـى ملاحظـة مـا يـأتي: يحتـوي هـذا الكتـاب على روابـط إلى مواقـع ويـب لا تُـدار مـن قبـل شركـة Binary Logic. ورغـم أنَّ شركـة Binary Logic تبـذل قصـارى جهدها لضمان دقة هذه الروابط وحداثتها وملاءمتها، فإنها لا تتحمل المسؤولية عن محتوى أي مواقع ويب خارجية.

إشـعار بالعلامـات التجاريـة: أسماء المنتجات أو الشركات المذكورة هنا قد تكـون علامات تجارية أو علامات تجاريـة مُسـجَّلة، وتُسـتخدم فقط بغرض التعريـف والتوضيح، وليس هناك أي نية لانتهـاك الحقوق، حيث تنفـي شركـة Binary Logic وجـود أي ارتبـاط أو رعايـة أو تأييـد مـن جانـب مـالكي العلامـات التجاريـة المعنيين. تُعد Microsoft و Windows و Bing و Edge و Office 365 علامات تجارية أو علامات تجارية مُسـجَّلة لشركة Microsoft Corporation. ويعد اسـم Scratch وشعار Scratch و Scratch Cat علامات تجاريـة لفريـق Scratch. وتعـد VEX و VEX Robotics و VEXCODE علامـات تجاريـة أو علامـات تجارية مسجلة أو علامات خدمة لشركة Innovation First, Inc.

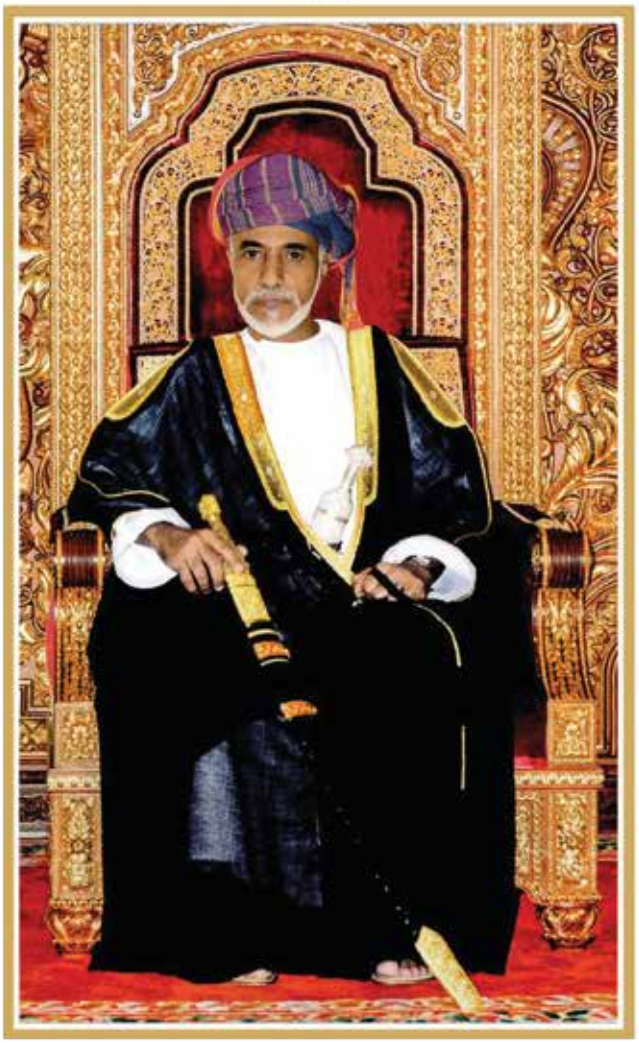
ولا ترعى الشركات أو المنظمات المذكورة أعلاه هذا الكتاب أو تصرح به أو تصادق عليه.

حـاول النـاشر جاهـدًا تتبـع ملاك الحقوق الفكرية كافة، وإذا كان قد سـقط اسـم أيٍّ منهم سـهوًا فسـيكون من دواعي سرور الناشر اتخاذ التدابير اللازمة في أقرب فرصة.

تمت مواءمة الكتاب بموجب القرار الوزاري رقم ٢٠٢٥/١٣٥ واللجان المنبثقة عنه.

جميع حقوق الطبع والنشر والتوزيع في سلطنة عُمان محفوظة لوزارة التربية والتعليم.

ولا يجـوز طبـع الكتـاب أو تصويـره أو إعادة نسـخه كاملًا أو مجزأً أو ترجمته أو تخزينه في نطاق اسـتعادة المعلومـات بـأي شـكل مـن الأشـكال إلا بـإذن كتـابي مسـبق مـن الـوزارة، وفي حـال الاقتباس القـصير يجب ذكر المصدر.



المغفور له
السلطان قابوس بن سعيد
طيّب الله ثراه

حضرة صاحب الجلالة
السلطان هيثم بن طارق المعظم
حفظه الله ورعاه

سلطنـــــة عُمـــــان

(المحافظات والولايات)

النَّشيدُ الْوَطَنِيُّ

يـا رَبَّنـا احْفَـظْ لنـا جَـلالَـةَ الـسُّـلْـطـان
وَالشَّعْـبَ فـي الأَوْطـان بِـالْـعِـزِّ والأَمـان
وَلْـيَـدُمْ مـؤَيَّـدًا عـاهِـلاً مُـمَـجَّـدًا
بِالنُّفـوسِ يُفْتَـدى

يا عُمانُ نَحْنُ مِنْ عَهْدِ النَّبي أَوْفِياءُ مِنْ كِرامِ الْعَرَبِ
فـارْتَقـي هـامَ السَّمـاء وَامْلَئي الْكَـوْنَ ضيـاء
وَاسْعَدي وَانْعَمي بِالرَّخـاء

بسم الله الرحمن الرحيم

تقديم

الحمد لله رب العالمين، والصلاة والسلام على أشرف المرسلين، سيدنا محمد وعلى آله وصحبه أجمعين...

سعت وزارة التربية والتعليم إلى تطوير المنظومة التعليمية في جوانبها المختلفة؛ لمواكبة التطورات المتسارعة في مجالي المعرفة والتقانة، وتلبية متطلبات مؤسسات التعليم العالي، واحتياجات المجتمع العُماني وسوق العمل، وهي بذلك تتوافق مع أهداف رؤية عُمان ٢٠٤٠ وركائزها التي أكدت أهمية رفع جودة التعليم وتطوير المناهج الدراسية والبرامج التعليمية؛ لإعداد متعلم معتز بهُويته، مبدع ومبتكر، ومنافس عالميًّا في جميع المجالات.

كما جاءت المناهج الدراسية منسجمة مع فلسفة التعليم في سلطنة عُمان، والإستراتيجية الوطنية للتعليم 2040، وقانون التعليم المدرسي في تنظيم مجالات العمل التربوي، وتهيئة الفرص المناسبة لبناء الشخصية المتكاملة للمتعلمين، والحرص على امتلاكهم مهارات المستقبل؛ كريادة الأعمال والابتكار، وأخلاقيات العمل، والتعامل مع معطيات التكنولوجيا الحديثة وإنتاج المعرفة، وتعزيز مهارات التفكير والبحث العلمي، ورفع مستوى وعيهم بالقضايا الإنسانية، وقيم السلام والحوار، والتسامح والتقارب بين الثقافات.

ويمثل هذا الكتاب المدرسي ترجمة للمحتوى المعرفي والمهاري للمنهاج الدراسي، الذي وضع ليستقي منه الطالب معلومات شاملة ومتنوعة، وليكتسب منه مهارات تعليمية مختلفة؛ لتحقيق ما تصبو إليه الوزارة من أهداف تربوية، وغايات سامية تسهم في تقدم هذا الوطن العزيز تحت ظل القيادة الحكيمة لمولانا حضرة صاحب الجلالة السلطان هيثم بن طارق المعظم، حفظه الله ورعاه.

واللّه ولي التوفيق.

د. مديحة بنت أحمد الشيبانية

وزيرة التربية والتعليم

بسم الله الرحمن الرحيم

المقدمة

عزيزي الطالب/ عزيزتي الطالبة:

يشهد العالم اليوم تَغَيُّرات جوهرية نتيجة للطفرة التكنولوجية التي أحدثتها الثورة الصناعية الرابعة في مختلف القطاعات؛ لذا أصبحت مهارات تقنية المعلومات من المهارات الأساسية التي تسعى جميع النُّظُم التعليمية الحديثة إلى إكسابها للمتعلمين؛ لجعل المتعلم نشطًا، وفعّالًا، ومنتجًا، ومدركًا لأهمية التعلم، ولِتُكْسِبَه مهارات حل المشكلات، والتفكير المنطقي، والإبداع. ومن هذا المنطلق، يسرُّنا أن نضع بين يديك كتاب تقنية المعلومات للصف السادس (الفصل الدراسي الثاني)، الذي تمت مواءمته وفق إطار سلسلة منهاج تقنية المعلومات من Binary Logic SA. وتُقدَّم هذه السلسلة بطريقة جاذبة وسلسة تحفزك على تطبيق المهارات الأساسية التي تحتاجها في هذه المرحلة، التي ستسهم في تطوير مهاراتك في مجال تقنية المعلومات.

يتناول هذا الكتاب وحدتين دراسيتين. في الوحدة الأولى، ستواصل استكشاف تطبيق Scratch، وستتعلم كيفية تحريك الكائنات باستخدام إحداثيات x و y، واستخدام المتغيرات لحفظ الوقت والنتيجة وعرضهما، وإنشاء مظاهر جديدة، وإرسال الرسائل واستقبالها بين الكائنات، بالإضافة إلى إجراء العمليات الحسابية. أما في الوحدة الثانية، فستستكشف الروبوتات في العالم الحقيقي من خلال التعرف إلى أنواعها واستخداماتها في مختلف المجالات، بالإضافة إلى مزايا وعيوب استخدامها. وستتعرف أيضًا إلى الروبوتات الافتراضية وكيفية برمجتها باستخدام تطبيق VEXcode VR، وستستخدم المستشعرات لزيادة دقة حركة الروبوت وتمكينه من التفاعل مع البيئة المحيطة.

ستلاحظ عزيزي الطالب/عزيزتي الطالبة أن أنشطة الكتاب مرتبطة بأحداث الحياة اليومية وبالمواد الدراسية المختلفة؛ لتوجيهك إلى ضرورة تفعيل استخدام التقنية في كافة مجالات التعلُّم.

وفي الختام، نوجّه عنايتك إلى ضرورة المحافظة على كتابك المدرسي، باعتباره دليلَك ومرجعَك في أثناء تعلّم المادة وتنفيذ أنشطتها.

وفقك الله في مسيرتك التعليمية.

المحتويات

الوحدة ١: البرمجة المرئية ١٢

هل تذكر؟ ١٤
الدرس ١-١: الإحداثيات ١٦
الدرس ١-٢: التحكم في اللعبة ٣٧
الدرس ١-٣: اتخاذ القرارات ٥١
الدرس ١-٤: الآلة الحاسبة ٦٦
الدرس ١-٥: المشروع ٨٢
برامج أخرى ٨٨

الوحدة ٢: الروبوتات ٩٠

الدرس ٢-١: الروبوتات في حياتنا ٩٢
الدرس ٢-٢: الروبوتات الافتراضية ٩٩
الدرس ٢-٣: حركة الروبوت ١٠٦
الدرس ٢-٤: التَّكرارات ورسم الأشكال ١١٦
الدرس ٢-٥: المستشعرات ١٢٤
الدرس ٢-٦: اتخاذ القرارات ١٣٦
الدرس ٢-٧: المتغيِّرات ١٤٢
الدرس ٢-٨: المشروع ١٥٠
برامج أخرى ١٥١

الوحدة ١:

البرمجة المرئية

المقدمة

ستتعرف في هذه الوحدة إلى كيفية استخدام نظام الإحداثيات لتحريك الكائنات في تطبيق Scratch. وستتعلم أيضًا كيفية إنشاء لعبة تعليمية تستخدم المتغيرات لحساب الوقت والنتيجة، وبث الرسائل لجعل الكائنات تتفاعل معًا. كما ستستخدم العمليات الحسابية لبناء مشروع آلة حاسبة.

أهداف التعلّم

ستتعلم في هذه الوحدة:

- مفهوم نظام الإحداثيات.
- خطوات تصميم لعبة.
- التمييز بين لبنات الانتظار في تطبيق Scratch.
- كيفية استخدام المتغيرات لتتبع الوقت والنتيجة.
- التمييز بين المعاملات المنطقية.
- البث في تطبيق Scratch.
- المعاملات الرياضية في البرمجة.

المهارات

ستتمكن بعد هذه الوحدة من:

- تحريك الكائنات باستخدام إحداثيات X (س) و Y (ص).
- استخدام المتغيرات لتخزين وعرض الوقت والنتيجة.
- تخصيص الكائنات من خلال إنشاء المظاهر.
- استخدام اللبنات الشرطية والمعاملات المنطقية.
- إرسال واستقبال الرسائل بين الكائنات.
- استخدام لبنات المُراسِل.
- إجراء عمليات حسابية.

الأدوات

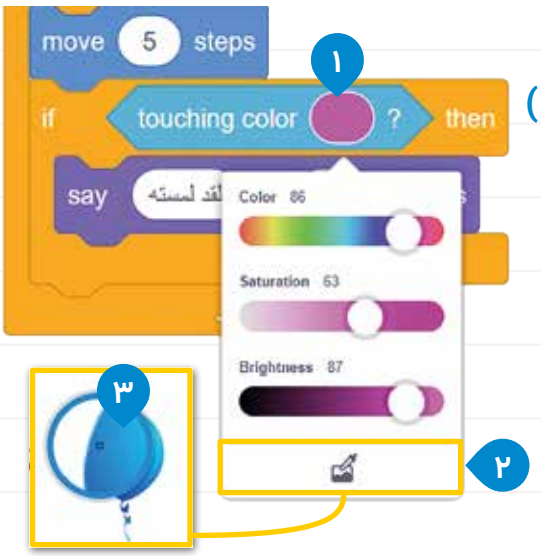
- Scratch

هل تذكر؟

لإنشاء خلفية

١ من قائمة **Choose a Backdrop** (اختيار خلفية)، انقر خيار **Paint** (رسام).

استخدام الفأرة لتحديد لون لبنة touching color ( ) ? (ملامس لـ ( ) ؟)



١ انقر على مربع الألوان للبنة **touching color ( ) ?** (ملامس لـ ( ) ؟).

٢ انقر على أداة **Color picker** (منتقي الألوان) لاختيار لون.

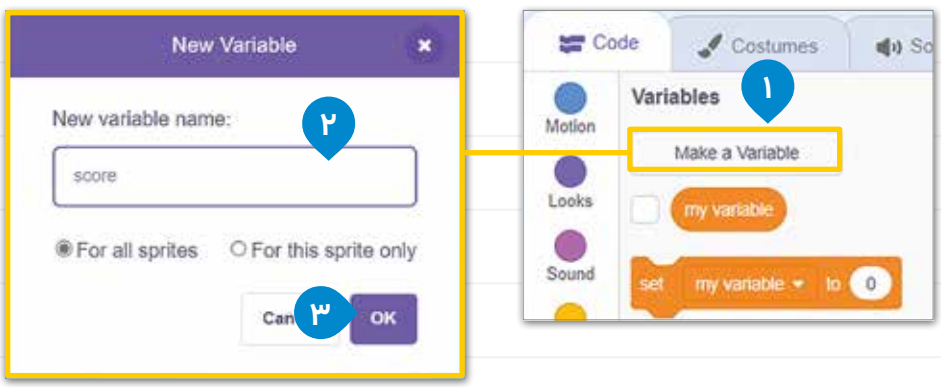
٣ حرّك مؤشر الفأرة ليلامس كائنًا، وانقر عليه لاختيار لونه.

لإنشاء متغير

١ من فئة لبنات **Variables** (المتغيرات)، انقر **Make a Variable** (إنشاء متغير).

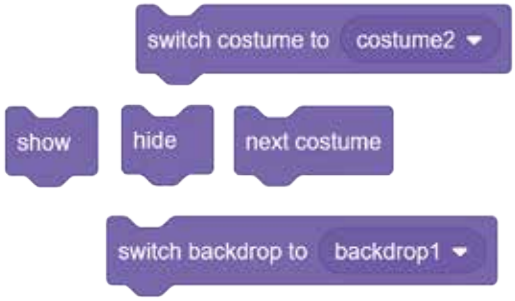
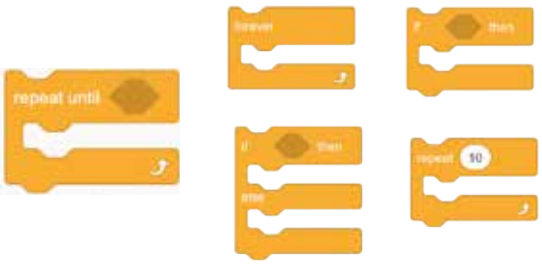
٢ من نافذة **New Variable** (متغير جديد)، اكتب اسمًا للمتغير.

٣ انقر **OK** (موافق).



اللَّبِنات التي استخدمتها

هنـاك 9 فئـات لِلَّبِنـات في لوحة لبنـات تطبيق Scratch. لكل منها لون محدّد وتسـتخدم لأداء مهام محدّدة في البرنامج؛ لإنشاء مقطع برمجي.

فئة اللبنة		اللبنات
Looks الهيئة	تتحكم في مظهر الكائن، أو خلفية المنصة.	
Control التحكم	تتحكم في مسار المقطع البرمجي.	
Sensing الاستشعار	تكشف عن تفاعلات الكائنات مع بعضها أو استشعار الضغط على مفتاح من لوحة المفاتيح أو حركة مؤشر الفأرة.	
Operators العمليات	إجراء العمليات الحسابية والمقارنة بين الأرقام والنصوص أو ربطها ببعضها البعض.	
Variables المتغيرات	تخزين المعلومات مثل النتيجة في مشروع ما، واستخدامها في البرمجة وغيرها من الأغراض المفيدة.	

الدرس ١-١:

الإحداثيات

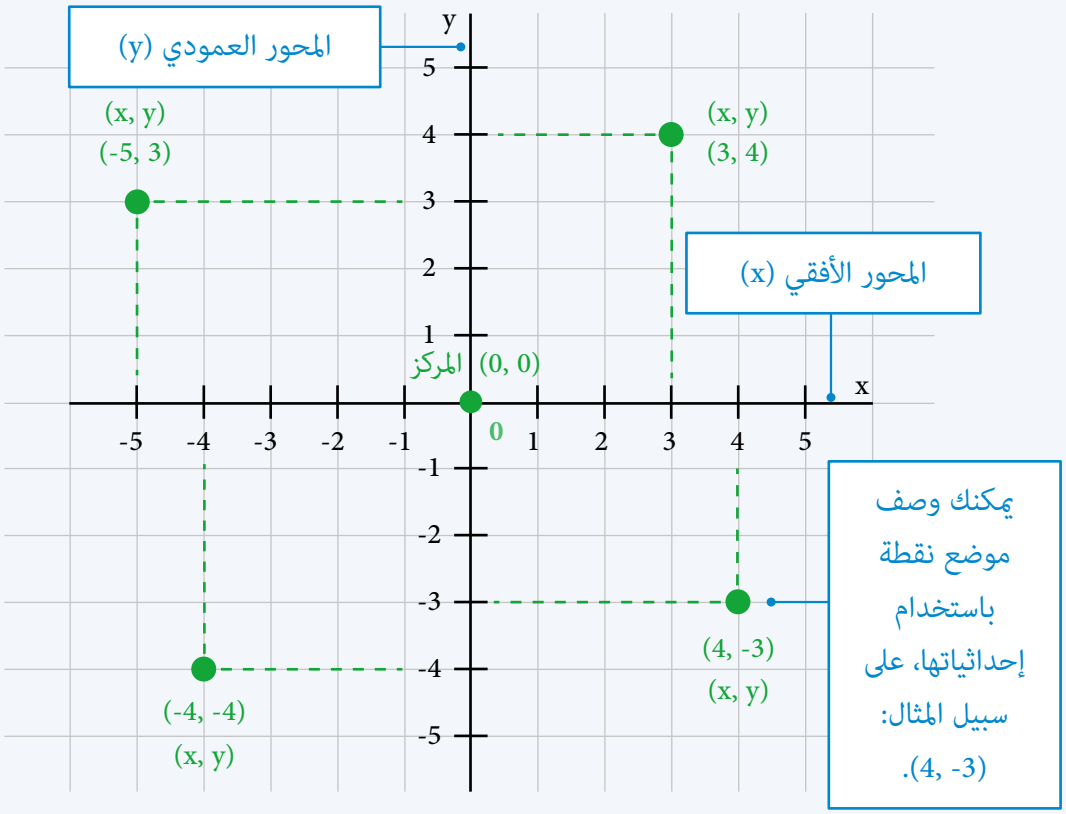
نظام الإحداثيات

نظام الإحداثيات عبارة عن طريقة يستخدمها الحاسوب لتحديد موضع الكائنات على الشاشة والتحكم بها، وهو النظام نفسه الذي نستخدمه في الرياضيات عندما نحدد موضع نقطة ما على ورقة باستخدام الأرقام.

يتكون نظام الإحداثيات من تقاطع خطين، هما:

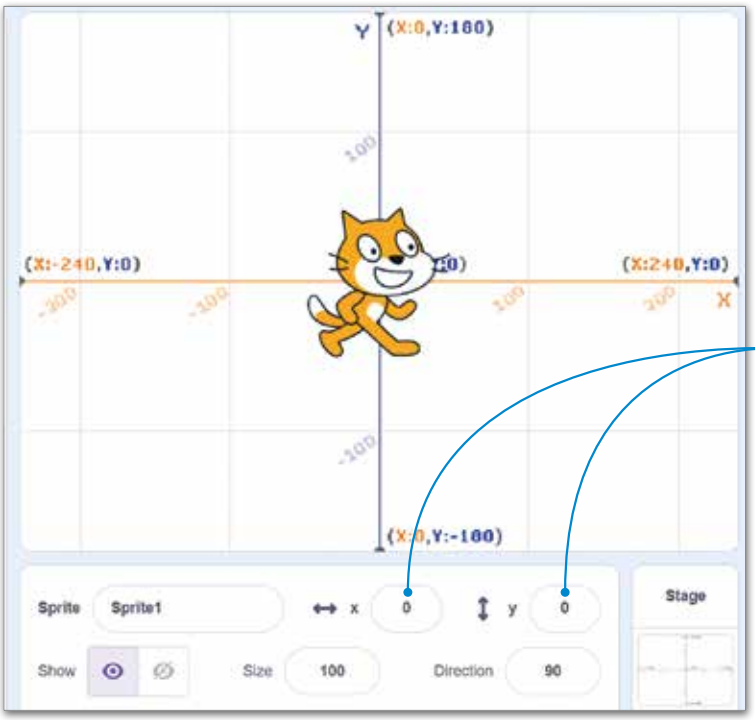
- **المحور الأفقي س (x):** يحدد الاتجاه يمينًا أو يسارًا.
- **المحور العمودي ص (y):** يحدد الاتجاه للأعلى أو للأسفل.

وتسمى نقطة التقاء الخطين بنقطة الأصل أو المركز، وتكون قيمتها (0, 0).



الإحداثيات في تطبيق Scratch

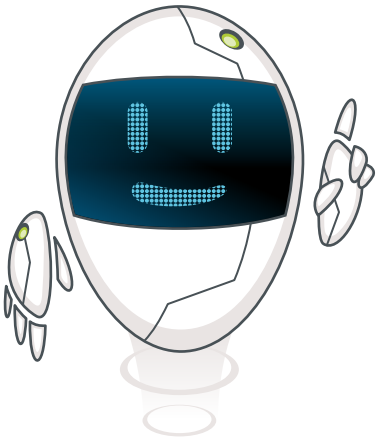
تتكون المنصة في تطبيق Scratch من مجموعة نقاط تُسمى البكسلات، وتشبه جدولًا يحتوي على صفوف وأعمدة. يُشار إلى الموقع في الصف بالرمز x، وفي العمود بالرمز y، حيث إنَّ لكل كائن موقعه الخاص على الشاشة والذي يُحدد بالإحداثيين x و y.



تُظهر هذه القيم موقع الكائن. يقع الكائن (Cat) في مركز المنصة، وهو أيضًا مركز نظام الإحداثيات، وقيمته (0, 0).

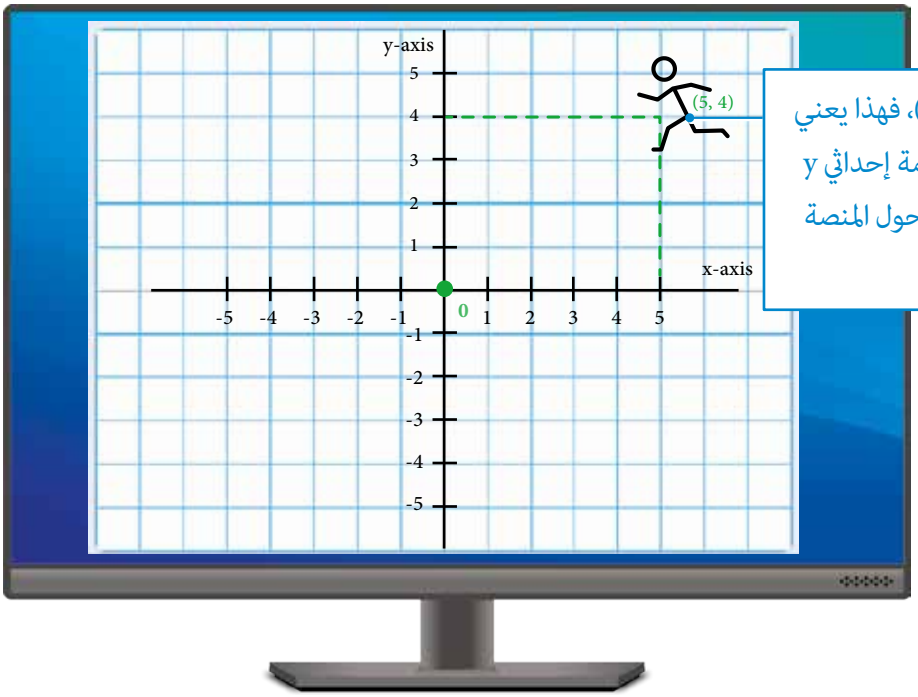
x تشير قيمة x إلى موضع الكائن أفقيًّا على المحور x. عند زيادة هذه القيمة أو تقليلها، يتحرك الكائن يمينًا أو يسارًا، وبالمثل تتغير قيمة x عند تحريك الكائن أفقيًّا بواسطة الفأرة.

y تشير قيمة y إلى موضع الكائن عموديًّا على المحور y. عند زيادة هذه القيمة أو تقليلها، يتحرك الكائن إلى أعلى أو إلى أسفل. وبالمثل تتغير قيمة y عند تحريك الكائن عموديًّا بواسطة الفأرة.

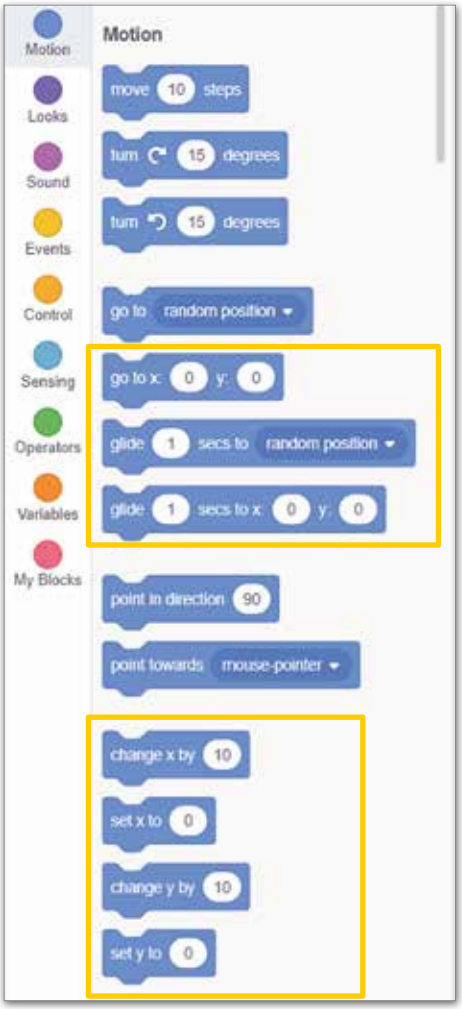


انتقل إلى تطبيق Scratch، وأضف خلفية Xy-grid حرّك الكائن (Cat) باستخدام الفأرة. ماذا تلاحظ على قيم الإحداثيين x وy أثناء الحركة؟

يمكن تعيين إحداثيات x و y الخاصة بموقع الكائن على المنصة وتغييرها من خلال اللبنات الموجودة في فئة لبنات **Motion** (**الحركة**).



إذا كان الكائن في الموضع (5, 4)، فهذا يعني أن قيمة إحداثي **x** هي **5** وقيمة إحداثي y هي **4**. وعندما يتحرك الكائن حول المنصة تتغير إحداثياته.



في تطبيق Scratch يمكن تحريك الكائنات على المنصة باستخدام لبنات محددة تتحكم في موقعها. تتيح لك هذه اللبنات تحديد إحداثيات دقيقة أو تحريك الكائنات في اتجاهات مختلفة من خلال تغيير قيمتي x وy. يُعد فهم كيفية التحكم في موقع الكائنات أمرًا أساسيًّا لإنشاء الرسوم المتحركة، والألعاب، والقصص التفاعلية.

انقر لتعيين إحداثيات **x** و **y** للكائن.

لبنة ( ) **y:** ( ) **go to x:** (**اذهب إلى الموضع س:** ( ) **ص:** ( )): تنقل الكائن على المنصة إلى موضع الإحداثيات المحددة.

لبنة ( ) **change x by (غيّر الموضع س بمقدار ( ))**: تغير موقع الكائن أفقيًا في المحور x وفقًا للقيمة الموجودة في خانة الإدخال. إذا كانت قيمة إحداثيات x موجبة يتحرك الكائن إلى اليمين، وإذا كانت سالبة يتحرك إلى اليسار.

لبنة ( ) **change y by (غيّر الموضع ص بمقدار ( ))**: تغير موقع الكائن عموديًا في المحور y وفقًا للقيمة الموجودة في خانة الإدخال. إذا كانت قيمة إحداثيات y موجبة يتحرك الكائن إلى أعلى، وإذا كانت سالبة يتحرك إلى أسفل.

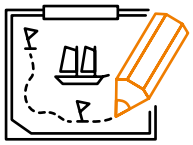
لبنة ( ) **y: ( ) secs to x: ( ) glide (انزلق خلال ( ) ثانية إلى الموضع س: ( ) ص: ( ))**: تحرك الكائن إلى إحداثيات x و y المحددة، وفقًا لعدد الثواني المحدد.

خطوات تصميم لعبة

لا توجد طريقة صحيحة أو خاطئة لتصميم لعبة، ولكن اتباع عملية منظمة يمكن أن يقلل المشكلات ويوفر الوقت أثناء عملية تصميم وتطوير ألعاب تعليمية.

١ **الفكرة**

الخطوة الأولى هي إيجاد فكرة للعبتك.

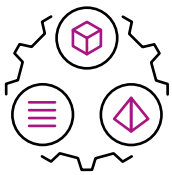


٢ **التخطيط**

يتضمن التخطيط إنشاء خطة للعبة، بما في ذلك هدف اللعبة والكائنات الرئيسة وعناصر التحكم.

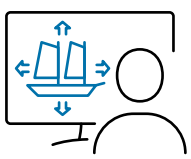
٣ **تصميم النموذج الأولي**

يساعدك تصميم النموذج الأولي على تصميم فكرة اللعبة من خلال إنشاء نموذج بسيط قبل إنشاء برمجتها.



٤ **التنفيذ**

تتضمن هذه الخطوة إنشاء الرسومات والمؤثرات الصوتية وبرمجة قواعد اللعبة.



٥ **الاختبار**

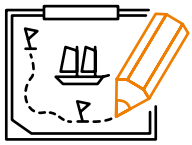
يُعدّ الاختبار خطوةً مهمةً في تطوير الألعاب. عليك اختبار اللعبة للتأكد من أنها تعمل بشكل صحيح، في هذه الخطوة ستُصلح أي أخطاء تجدها.

لعبة البحار العُماني

يمكنك استخدام تطبيق Scratch لإنشاء لعبة تعليمية تحتوي على قارب شراعي يُبحر بين أربع محطات. من المهم تحديد المكونات الرئيسة للعبة أثناء عملية تصميمها، لعمل ذلك عليك اتباع خطوات تصميم اللعبة.

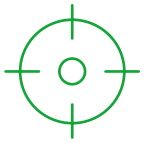
الفكرة

الفكرة الرئيسة للعبة التعليمية هي أن يتحرك القارب الشراعي في البحر، ويتم التحكم به عن طريق لوحة المفاتيح لتجنب الاصطدام باليابسة، مع تقدّم اللاعب في اللعبة من خلال لمس الأعلام والإجابة عن أربعة أسئلة بشكل صحيح في وقت محدد لكسب النقاط.



التخطيط

أهداف اللعبة



الهدف هو أن يجمع القارب الشراعي أكبر عدد ممكن من النقاط قبل انتهاء الوقت. تنتهي اللعبة بانتهاء الوقت المحدد أو عند عدم إجابة اللاعب عن جميع الأسئلة.

الكائنات الرئيسة

تشمل الكائنات الرئيسة القارب الشراعي الذي يتحكم به اللاعب، وأربعة أعلام يتفاعل معها اللاعب لرؤية الأسئلة والإجابة عنها.

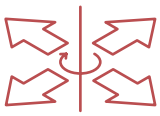
خلفية اللعبة

خلفية اللعبة عبارة عن بحر ويابسة.

قواعد اللعبة

يبدأ اللاعب برصيد (0) من النقاط ومدة (60) ثانية. يكسب النقاط من خلال الإجابة الصحيحة عن الأسئلة، ويجب عليه تجنّب ملامسة القارب الشراعي لليابسة، حيث يعود القارب إلى مركز المنصة عند لمسها. يخسر اللاعب نقطة إذا لم يتمكّن من الإجابة عن السؤال بعد ثلاث محاولات.

عناصر التحكم



يتم التحكم في القارب الشراعي باستخدام مفاتيح الأسهم على لوحة المفاتيح.

ضَع الأسئلة

فكّر في الأسئلة التي من الممكن أن تطرحها على اللاعب، مثل:

- متى تحتفل سلطنة عُمان باليوم الوطني؟
- ما عملة سلطنة عُمان؟
- ما اسم المضيق الذي تطل عليه سلطنة عُمان؟
- في أي قارة تقع سلطنة عُمان؟

تصميم النموذج الأولي

قبل بناء اللعبة في تطبيق Scratch، يُنصح بإنشاء نموذج أولي باستخدام أداة رسم، أو برنامج تصميم، أو حتى ورقة وقلم. يتيح لك هذا النموذج تصوّر شكل ومواقع عناصر اللعبة، مما يجعل عملية التصميم والبرمجة في تطبيق Scratch أسهل وأكثر تنظيمًا.

التنفيذ

بعد الانتهاء من تصميم النموذج الأولي، ستبدأ بإنشاء اللعبة باستخدام تطبيق Scratch.

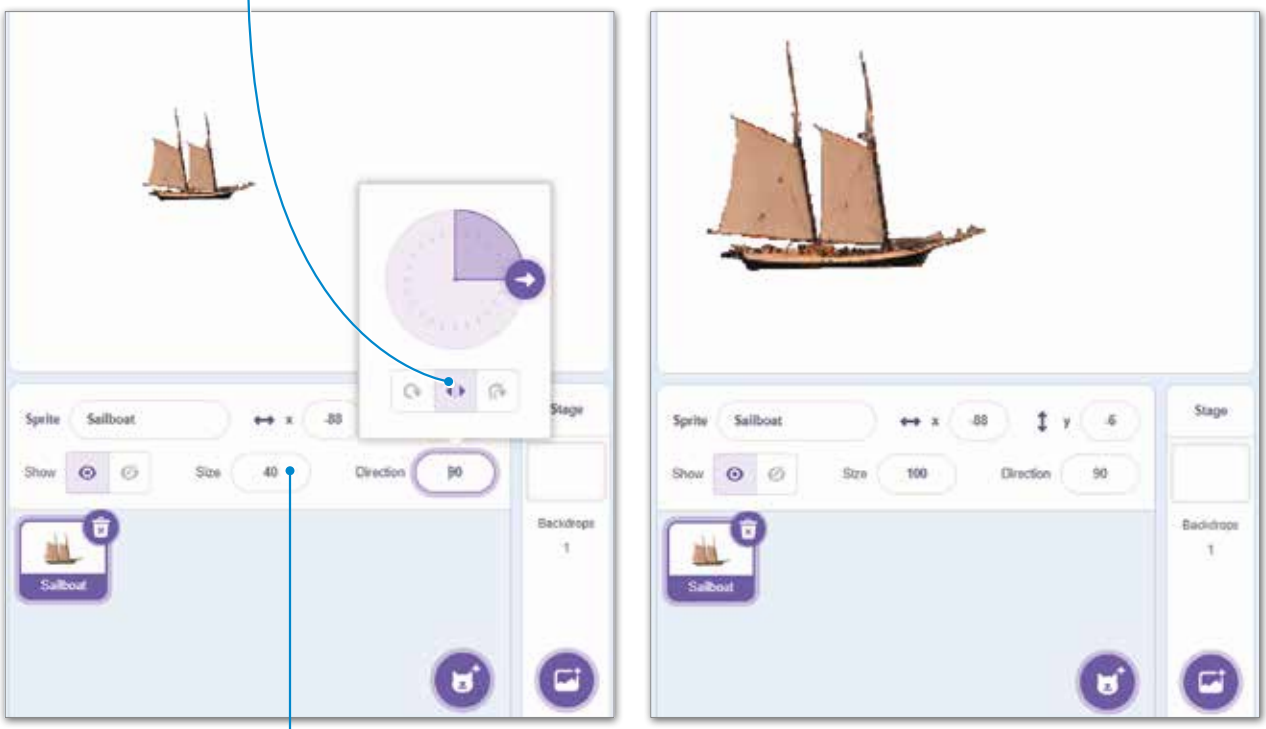
لإنشاء مشروع جديد:

١ افتح تطبيق Scratch.

٢ احذف الكائن (Cat).

٣ أضف الكائن (Sailboat).

غيِّر الاتجاه إلى **Left / Right** (**يمين/ يسار**) حتى لا ينقلب القارب الشراعي.



غيِّر **Size** (**حجم**) الكائن (Sailboat) إلى "**40**".

لرسم الخلفية:

١ انتقل إلى تبويب **Stage** (المنصة)، واختر **Choose a Backdrop** (اختيار خلفية)، ثم **Paint** (رسام).

٢ انقر على **Convert to Bitmap** (تحويل إلى رسم نقطي).

٣ اختر أداة **Fill** (الملء) ولوّن الخلفية باللون الأزرق.

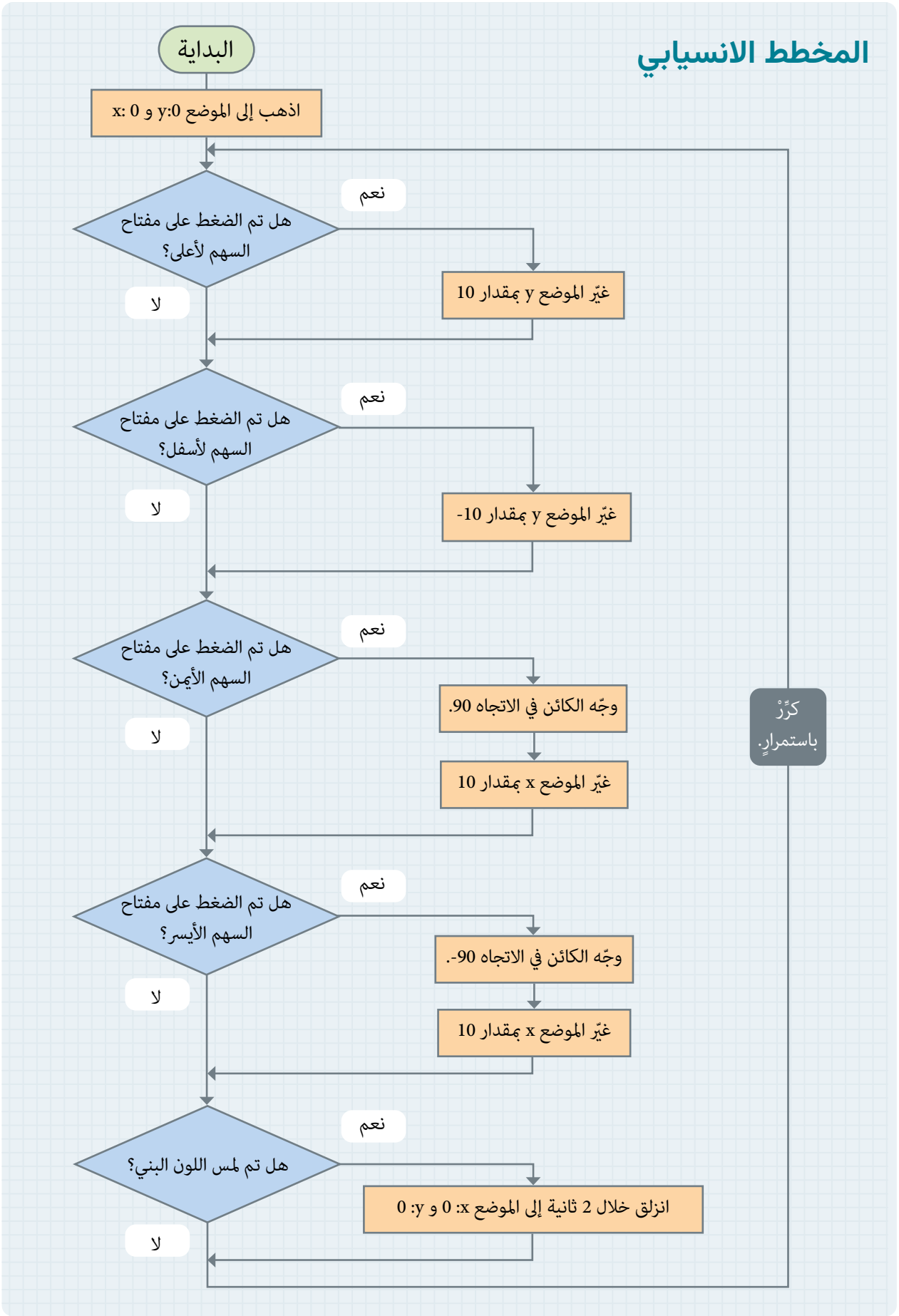
٤ استخدم أداة **Brush** (الفرشاة) لرسم مساحات اليابسة باللون البني.

برمجة القارب الشراعي

في الألعاب، تُستخدم لوحة المفاتيح غالبًا للتحكم في الكائنات. لتحريك الكائن إلى اليمين أو إلى اليسار، عليك تغيير قيمة الإحداثي x، ولتحريك الكائن إلى أعلى أو إلى أسفل، عليك تغيير قيمة الإحداثي y.

ستتحكم في كائن القارب الشراعي عن طريق لوحة المفاتيح باستخدام مفاتيح الأسهم، حيث سينطلق القارب الشراعي من مركز المنصة ويتحرك في اتجاهات مختلفة.

المخطط الانسيابي



للتحكم في حركة القارب الشراعي:

> ١ من فئة لبنات **Events** (الأحداث)، أضف لبنة **when flag clicked** (**عند نقر العلم**).

> ٢ من فئة لبنات **Motion** (الحركة)، أضف لبنة **go to x: ( ) y: ( )** (**اذهب إلى الموضع س: ( ) ص: ( )**) واكتب (0) في الموضع x، و(0) في الموضع y.

> ٣ من فئة لبنات **Control** (التحكم)، أضف لبنتَي **forever** (**كرّر باستمرار**) و **if ( ) then** (**إذا ( )**).

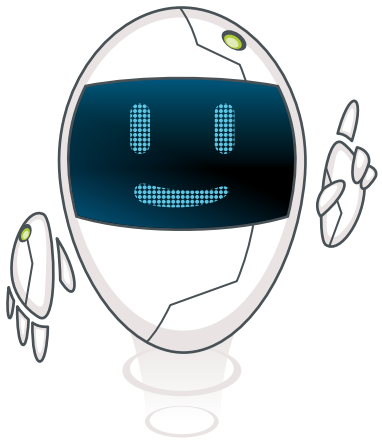
> ٤ من فئة لبنات **Sensing** (الاستشعار)، أضف لبنة **key ( ) pressed?** (**مفتاح ( ) مضغوط؟**)،

٥ ثم اختر مفتاح **up arrow** (**السهم العلوي**).

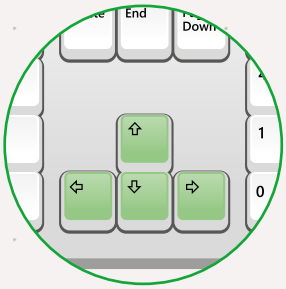
> ٦ من فئة لبنات **Motion** (الحركة)، أضف لبنة **change y by ( )** (**غيّر الموضع ص بمقدار ( )**).

> كرّر الخطوات نفسها مع مفاتيح الأسهم لأسفل، ولليمين، ولليسار لإكمال المقطع البرمجي.

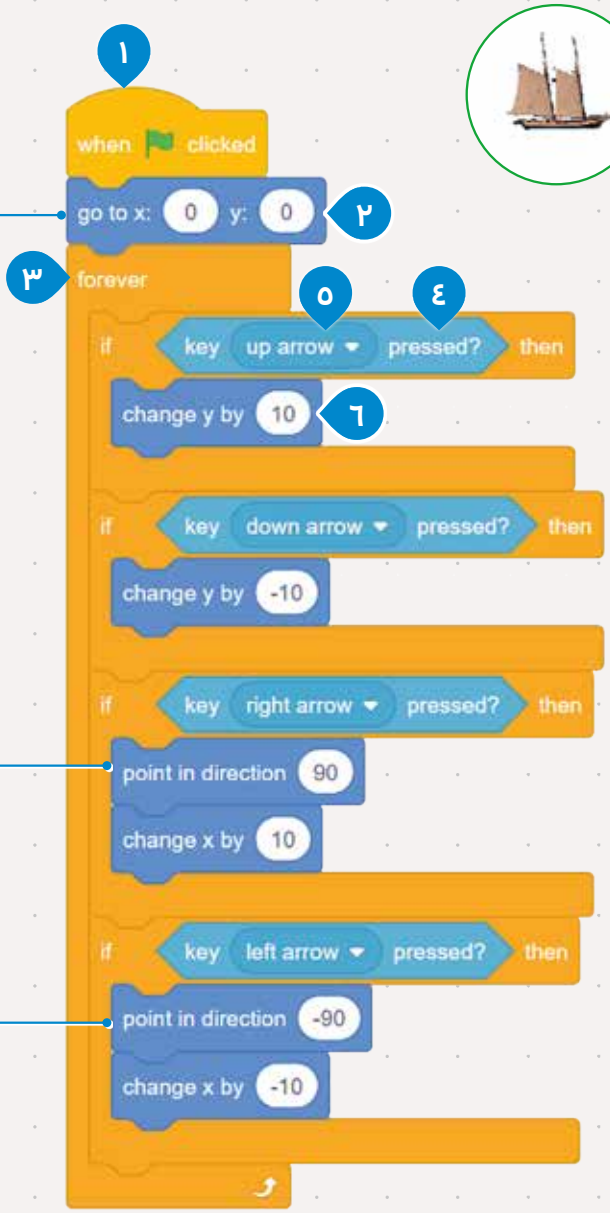
تذكر أن الكائنات يجب أن تتحرك على المحور x باستخدام السهمين الأيمن والأيسر، لذا يجب عليك استخدام لبنة **change x by ( )** (**غيّر الموضع س بمقدار ( )**).

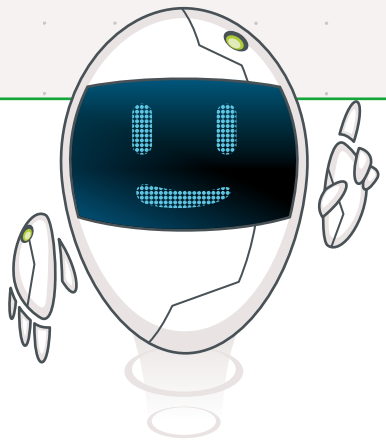


يتحرك الكائن إلى الإحداثيات (0,0) داخل المنصة في كل مرة تنقر فيها على زر Go (انطلق) لبدء اللعبة.



لبنة point in direction ( ) (اتجه نحو الاتجاه ( )) تغير اتجاه الكائن أثناء الحركة إلى اليسار واليمين فقط، بحيث يجعل مقدمة الكائن (Sailboat) موجهة دائمًا نحو اتجاه الحركة.



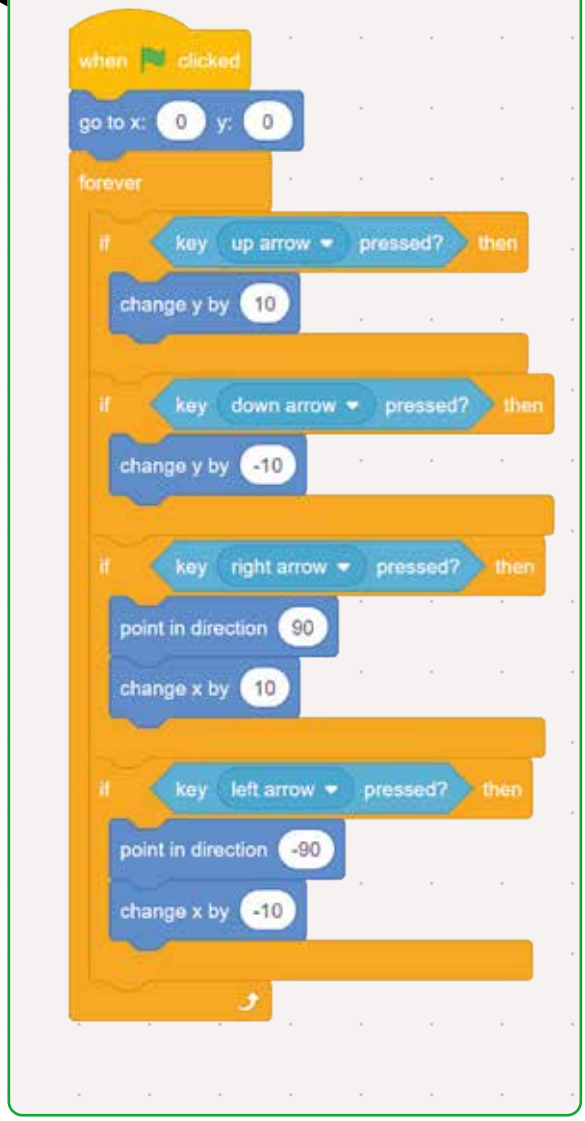


بالنسبة لمشروعك، ربما تكون الإحداثيات مختلفة لأنها تعتمد على الخلفية التي ستنشئها والمكان الذي تريد وضع القارب الشراعي فيه.

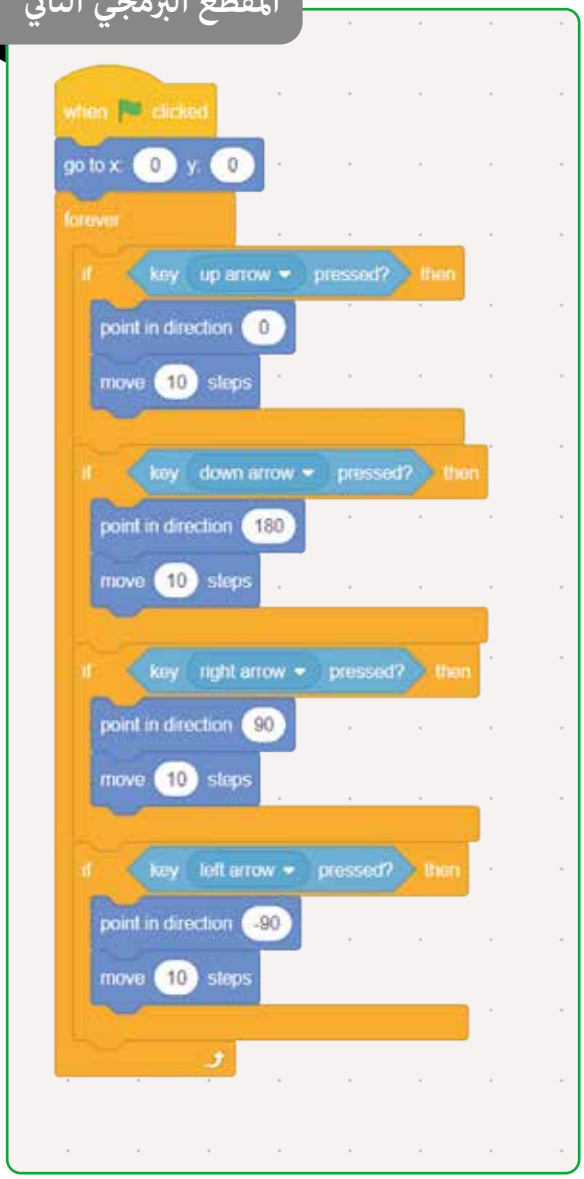
جرب المقطع البرمجي نفسه دون استخدام لبنة point in direction ( ) (اتجه نحو الاتجاه ( )) ولاحظ الفرق في الحركة.

لنرَ الفـرق بين هذيـن المقطـعين البرمجـيين: المقطـع البرمجـي الأول حـرّك الكائـن مبـاشرةً مـن خلال تغـيير إحداثياتـه x و y عنـد الضغـط على مفاتيـح الأسـهم، وأمـا المقطـع البرمجـي الثـاني فيسـتخدم لبنـة **move ( ) steps (تحـرك ( ) خطـوة)** التـي تُحـرّك الكائـن إلى اليـمين افتراضيًّا؛ لـذا يجـب إضافـة لبنة **point in direction ( ) (اتجه نحو الاتجاه ( ))** في كل مرة.

المقطع البرمجي الأول



المقطع البرمجي الثاني



افتـح مشروع Scratch جديـدًا وجـرّب كلا المقطـعين البرمجيين مع ضبط اتجاه الكائـن (Sailboat) على All Around (جميع الاتجاهات)، ثم Left/Right (يمين / يسار) ولاحظ الفرق.

عودة القارب الشراعي إلى نقطة البداية

إذا لامس الكائن (Sailboat) اليابسة ذات اللون البنيّ، فإنه يعود إلى مركز المنصة عند النقطة (0, 0) وهي نقطة البداية التي انطلق منها.

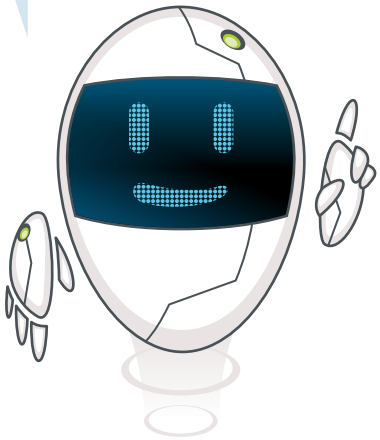
لتحريك الكائن (Sailboat) إلى نقطة البداية:

> ١ من فئة لبنات **Control** (التحكم)، أضف لبنة **if ( ) then** (إذا ( )) **داخل لبنة forever** (كرّر باستمرار).

> ٢ من فئة لبنات **Sensing** (الاستشعار)، أضف لبنة **touching colour ( )?** (ملامس للون ( )؟)،
٣ ثم استخدم أداة **Color picker** (منتقي الألوان) لاختيار لون اليابسة.

> ٤ من فئة لبنات **Motion** (الحركة)، أضف **لبنة glide ( ) secs to x: ( ) y: ( )** (انزلق خلال ( ) ثانية إلى الموضع س: ( ) ص: ( )) ٥ واضبط الثواني إلى (2).

ناقش مع معلمك الفرق بين لبنات ( ) y: ( ) secs to x: ( ) glide (انزلق خلال ( ) ثانية إلى الموضع س: ( ) ص: ( ))، و ( ) change x by (غيّر الموضع س بمقدار ( ))، و ( ) change y by (غيّر الموضع ص بمقدار ( )).



رسم المسار

لرسم أثر لمسار الكائن (Sailboat) أثناء تحركه، استخدم فئة Pen (القلم).

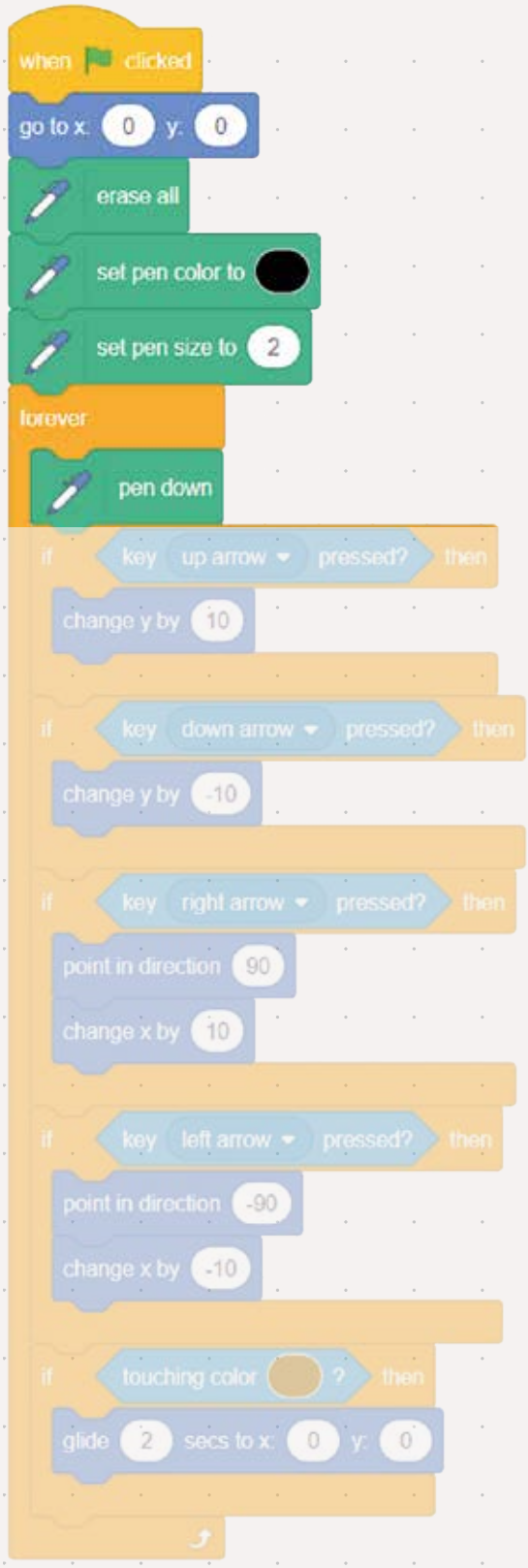
لرسم المسار:

١ انتقل إلى زر **Add Extension** (إدراج إضافة) واختر **Pen** (القلم).

٢ أضف لبنات **erase all** (**مسح الكل**)، و ( ) **set pen color to** (**اجعل لون القلم مساويًا** ( ))، و ( ) **set pen size to** (**اجعل حجم القلم مساويًا** ( )).

٣ اضبط لبنة ( ) **set pen color to** (**اجعل لون القلم مساويًا** ( )) بلون من اختيارك، ولبنة ( ) **set pen size to** (**اجعل حجم القلم مساويًا** ( )) إلى (2).

٤ أضف لبنة **pen down** (**أنزل القلم**) داخل لبنة **forever** (**كرّر باستمرار**).



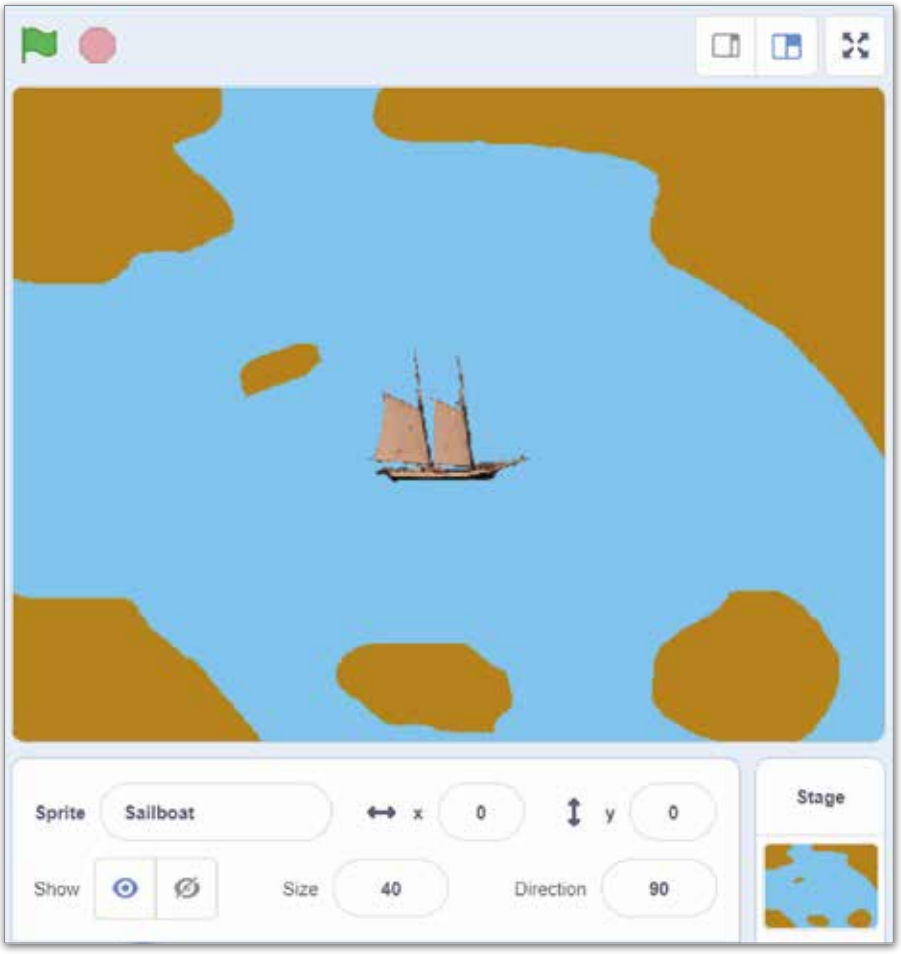
تذكّر أن تحفظ عملك.

الاختبار

جـرّب تشـغيل اللعبـة وتأكـد مـن تنفيـذ المقاطـع البرمجيـة بشـكل صحيـح، ثـم أجب عن الأسئلة الآتية:

- هل لدى القارب الشراعي مساحة كافية للتحرك ورسم مساره؟
- هل أعجبك لون وحجم مسار الرسم؟

بنـاءً على إجاباتـك، أجرِ التغـييرات اللازمة لتحسين اللعبة، مثل تغيير حجم القارب الشراعي أو إعادة تصميم اليابسة وخط الرسم.



لنطبق معًا

تدريب ١

تحقَّق من العبارات الآتية لمعرفة ما إذا كانت صحيحة أم خاطئة، ثم صحِّح العبارات الخاطئة.

	صحيحة	خاطئة
١. تُحدد قيمة y موضع الكائن على المحور الأفقي.	○	○
٢. إذا كانت إحداثيات x وy للكائن تساوي صفرًا، فسيكون الكائن في مركز المنصة.	○	○
٣. تُحدِّد قيمة x موضع الكائن على المحور الأفقي.	○	○
٤. إذا كان الكائن في الموضع (10, 150)، فهذا يعني أن قيمة y = 150 وقيمة x = 10.	○	○
٥. يستخدم تطبيق Scratch نظام إحداثيات محور واحد لوصف موضع الكائن.	○	○
٦. إذا تحرك الكائن على المحور y ، فإنه يتحرك إلى اليسار أو اليمين.	○	○
٧. إذا تحرك الكائن إلى اليسار، فإن إحداثياته على المحور x تزداد.	○	○
٨. لكل كائن موقعه الفريد على منصة تطبيق Scratch، والذي يُحدد بإحداثيين هما x و y، ولا يمكن لأي كائن آخر أن يكون له نفس الموقع.	○	○

تدريب ٢

طابِق اللبنات بوظائفها.

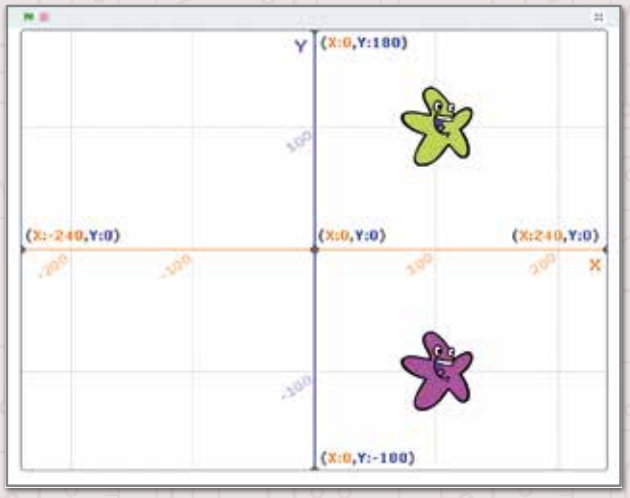
- نقل الكائن إلى إحداثيات x و y المحددة، وفقًا لعدد الثواني المحدد.
- تحريك الكائن إلى موضع معيّن على المنصة وفقًا للإحداثيات المحددة.
- تغيير قيمة x لموضع الكائن بمقدار الرقم المحدد.
- تغيير قيمة y لموضع الكائن بمقدار الرقم المحدد.

تدريب ٣

أكمل الجدول بقيم إحداثيات x و y.

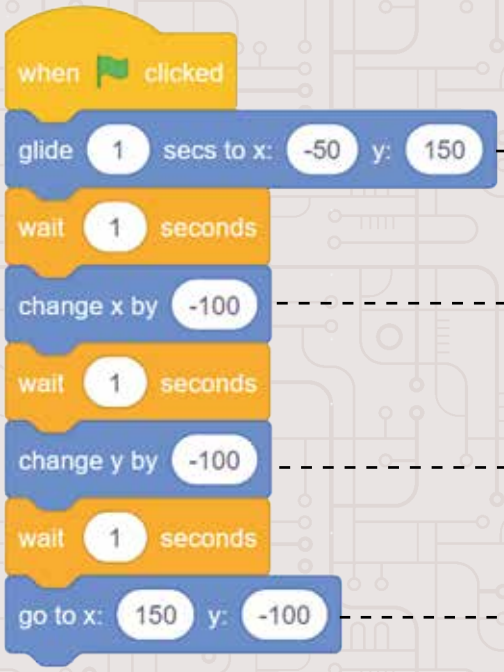
- سجّل إحداثيات موضع النجمتين الخضراء والبنفسجية.
- استخدم المحور y كمرآة، وارسم انعكاس النجمتين على الجانب الآخر من المنصة. على سبيل المثال: إذا كانت النجمة الخضراء في الموقع y، x، فإن انعكاس صورتها على المحور y تكون في الموقع y، x-.
- دوّن إحداثيات موضع الكائنين الجديدين.

الكائن	الإحداثيات	
	y	x



تدريب ٤

اقرأ المقطع البرمجي واكتب قيم إحداثيات x و y. انتقل إلى تطبيق Scratch للتحقق من إجاباتك.



الإحداثيات	
y	x

الدرس ٢-١:

التحكم في اللعبة

في هذا الدرس، ستبرمج نظام تسجيل النقاط ومؤقّت اللعبة باتباع الآتي:

- يبدأ اللاعب برصيد صفر من النقاط.
- لدى اللاعب 60 ثانية للإجابة عن أربعة أسئلة.
- يحصل اللاعب على نقطة واحدة مقابل كل إجابة صحيحة، أما إذا لم يتمكّن من الإجابة عن السؤال ثلاث مرات فسيخسر نقطة.
- تنتهي اللعبة بانتهاء الوقت المحدد أو إذا لم يتمكّن من حل السؤال بعد ثلاث محاولات.

ستحتاج إلى إيقاف تنفيذ المقطع البرمجي مؤقتًا، أو الإنتظار حتى يتحقق شرط محدد في اللعبة، ولتنفيذ ذلك يمكنك استخدام لبنات الإنتظار من فئة لبنات **Control (التحكم)**.

لبنة **seconds ( ) wait (انتظر ( ) ثانية)**: تنتظر عددًا محددًا من الثواني ثم تنتقل إلى اللبنة التالية.

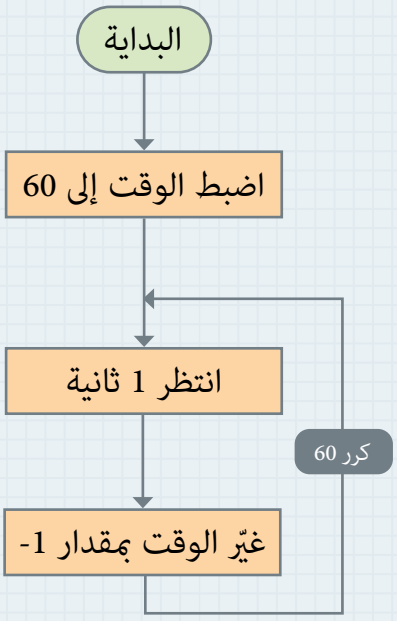
لبنة **( ) wait until (انتظر حتى ( ))**: تُوقف المقطع البرمجي مؤقتًا حتى يتحقق شرط محدد.

لاحظ الفرق بين لبنة **seconds ( ) wait (انتظر ( ) ثانية)** ولبنة **( ) wait until (انتظر حتى ( ))**، حيث إن خانة الإدخال في لبنة **seconds ( ) wait (انتظر ( ) ثانية)** بيضوي الشكل؛ لأن مدخلاته لا يمكن أن تكون إلا قيمًا، أما خانة الإدخال في لبنة **( ) wait until (انتظر حتى ( ))** فهو مضلع الشكل لأن مدخلاته لا يمكن أن تكون إلا شروطًا.

المؤقّت

بعد فتح اللعبة، سترى المقطع البرمجي الخاص بالكائن (Sailboat). اتبع الخطوات الآتية لإنشاء المقطع البرمجي الخاص بالمؤقت. سيكون الوقت متغيراً، حيث يبدأ من 60 ثانية ويتناقص بمقدار 1 كل ثانية حتى يصل إلى صفر.

المخطط الانسيابي



لإنشاء متغير الوقت:

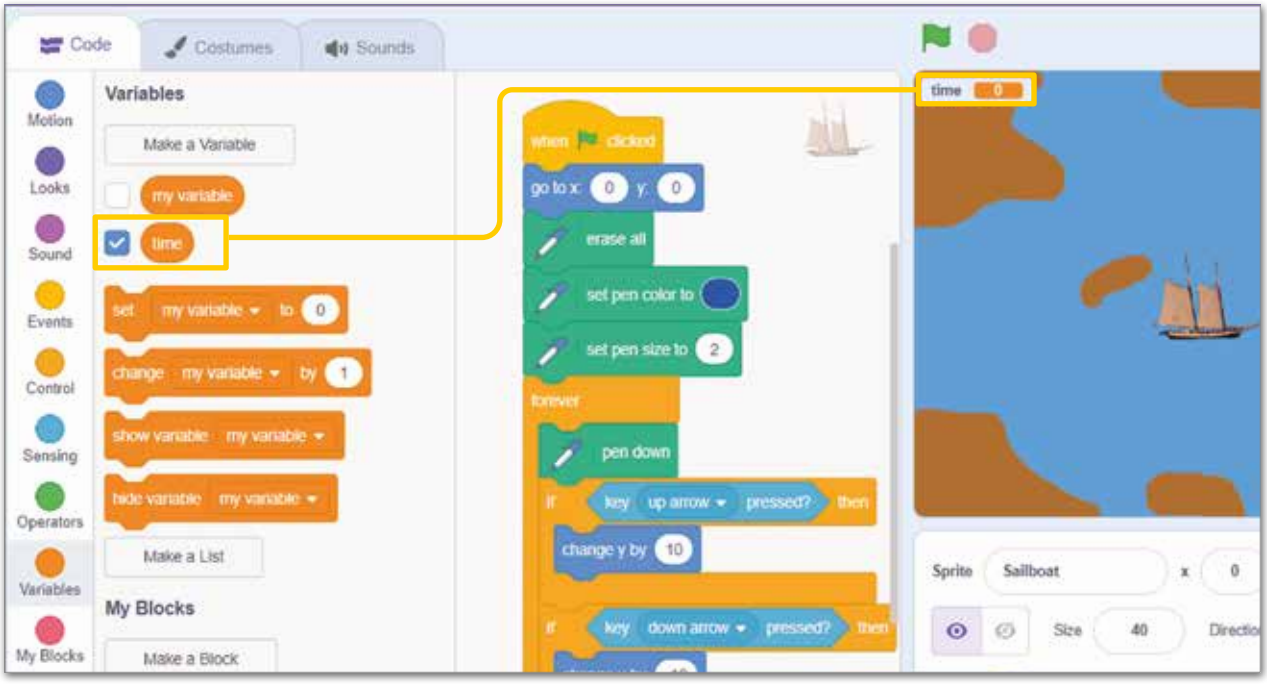
١ بعد فتح لعبة البحّار العُماني، انقر فئة لبنات **Variables** (المتغيرات).

٢ انقر **Make a Variable** (إنشاء متغير).

٣ من نافذة **New Variable** (المتغير الجديد)، اكتب **time** (الوقت) كاسم للمتغير.

٤ انقر **OK** (موافق).

٥ انقر على خانة الإختبار بجوار المتغير **time** (الوقت) لعرضه على المنصة.



لإنشاء المؤقت:

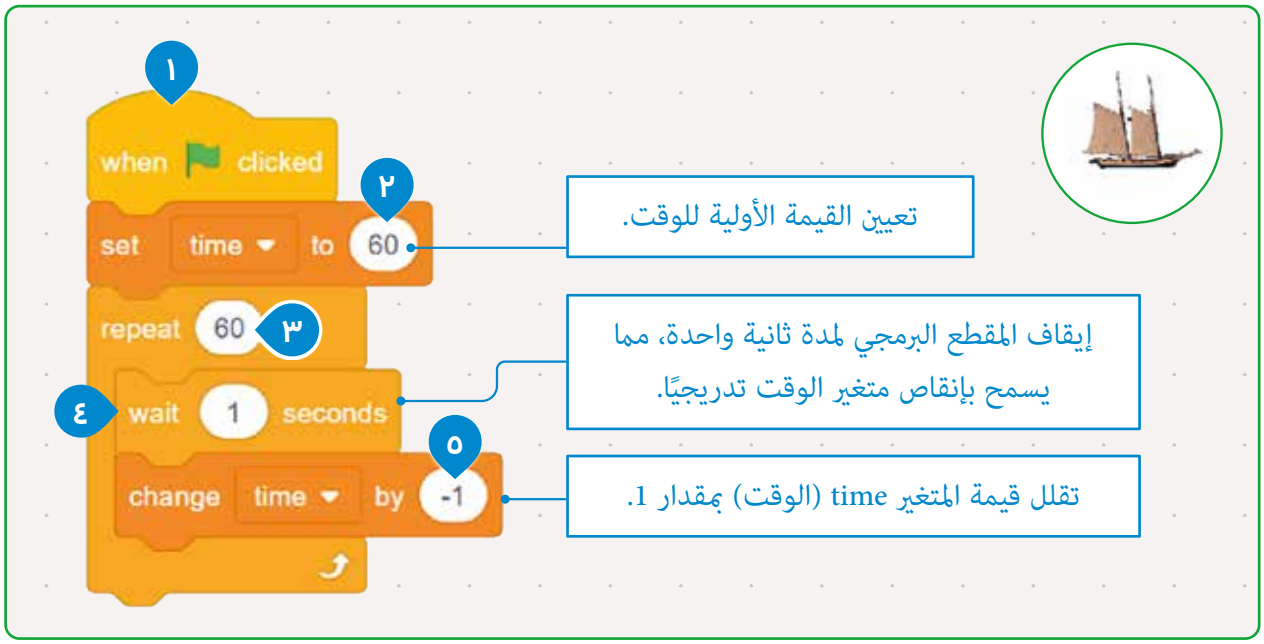
١ من فئة لبنات **Events** (الأحداث)، أضف لبنة **when flag clicked** (**عند نقر العلم**).

٢ من فئة لبنات **Variables** (المتغيرات)، أضف لبنة **set time to ( )** (**اجعل الوقت مساويًا ( )**) واضبط الوقت إلى (**60**).

٣ من فئة لبنات **Control** (التحكم) أضف لبنة **repeat ( )** (**كرّر ( ) مرة**) واضبطها إلى (**60**).

٤ بعد ذلك أدرج لبنة **wait ( ) seconds** (**انتظر ( ) ثانية**) داخل لبنة **repeat ( )** (**كرّر ( ) مرة**)

٥ ومن فئة لبنات **Variables** (المتغيرات)، أضف لبنة **change time by ( )** (**غيّر الوقت بمقدار ( )**) واضبطها إلى (**-1**).



باستخدام لبنة ( ) repeat (كرّر ( ) مرة)، يتوقف تناقص متغير الوقت بعد 60 ثانية، وباستخدام لبنة forever (كرّر باستمرار)، تصبح قيمة المؤقت سالبة بعد مرور 60 ثانية. في هذه الحالة، ما اللبنات التي يجب إضافتها لإيقاف هذه الدائرة عند وصول قيمة المؤقت إلى 0؟ أنشئ المقطع البرمجي.

يجب أن يكون اسم المتغير فريدًا وبسيطًا قدر الإمكان.

احتساب النقاط

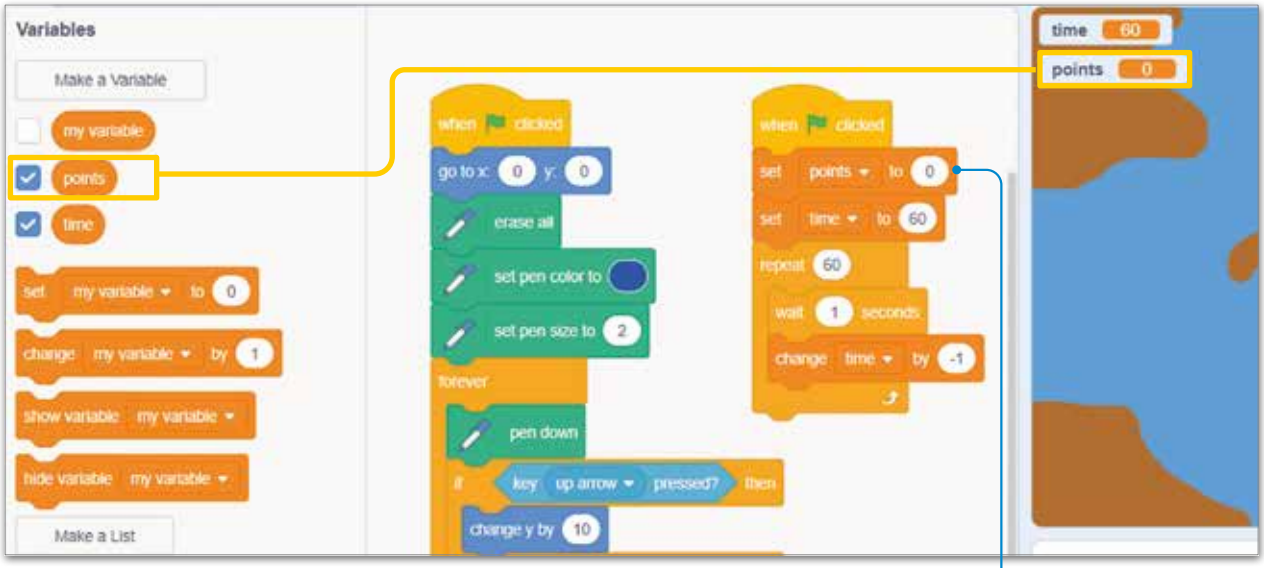
بعد الانتهاء من إنشاء المؤقت، ستُعيّن المتغير الثاني للعبة الذي يُمثّل نقاط اللاعب وسيكون متغير النقاط لجميع الكائنات.

لإنشاء متغير النقاط:

١ أنشئ متغيرًا جديدًا باسم **points** (النقاط).

٢ انقر على خانة الاختيار بجوار المتغير **points** (النقاط) لعرضه على المنصة.

٣ أضف لبنة ( ) **set points to** (**اجعل النقاط مساويًا** ( )) إلى الكائن (Sailboat) واضبطها إلى "0".



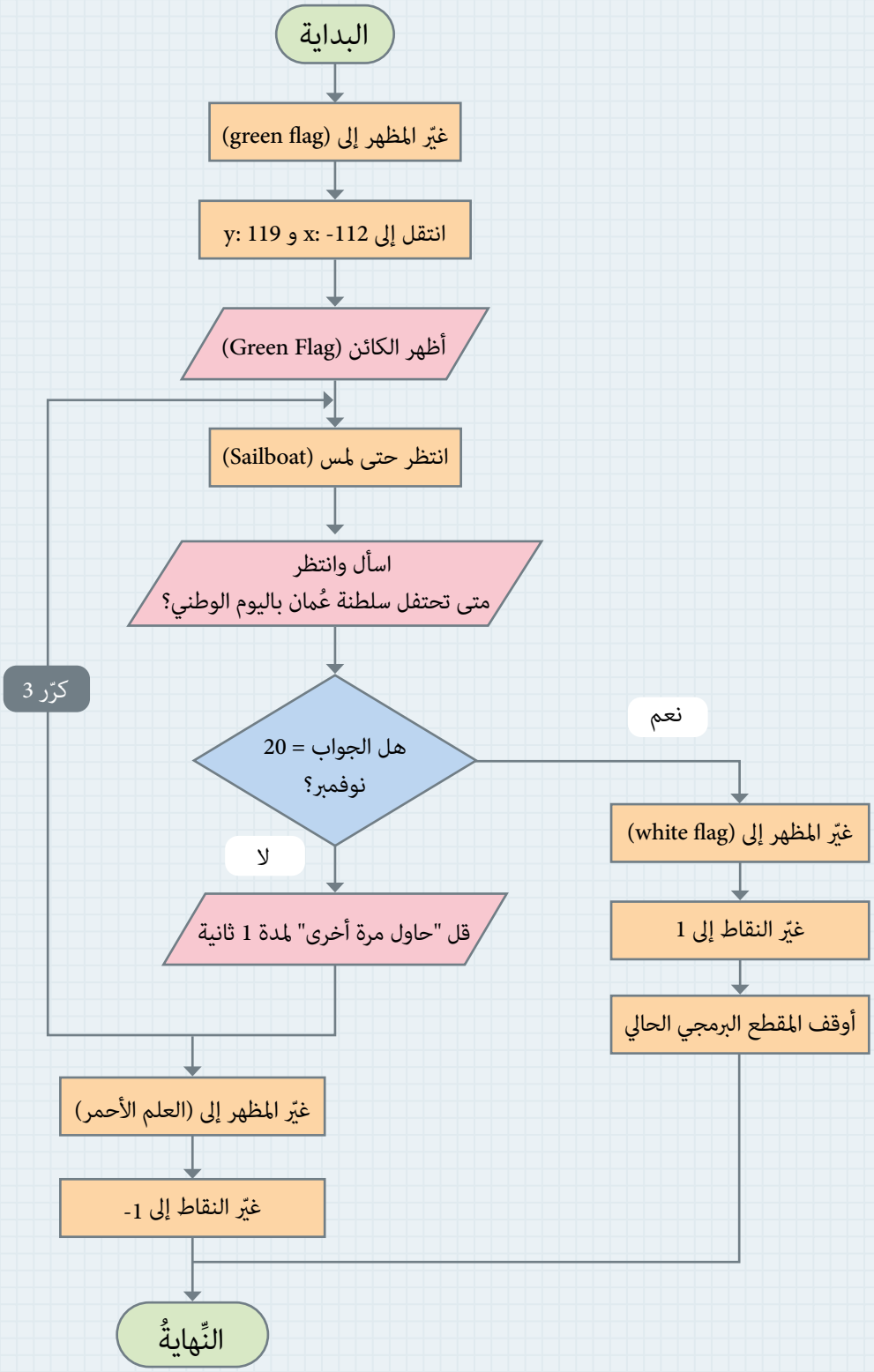
يجب ضبط القيمة التي يبدأ بها المتغير في بداية اللعبة؛ لذلك نقوم بإدراجها بعد لبنة **when flag clicked** (**عند نقر العلم**).

إنشاء مظاهر جديدة لكائن

ستنشئ أربعة كائنات (Green Flag)، بحيث يكون لكل كائن ثلاثة مظاهر، كما سيحمل كل كائن سؤالًا مختلفًا.

- عندما يلمس الكائن (Sailboat) الكائن (Green Flag) يظهر سؤال على الشاشة.
- إذا كانت الإجابة صحيحة، يكسب اللاعب نقطة ويصبح العلم أبيض ويختفي السؤال.
- إذا كانت الإجابة خاطئة، تظهر عبارة حاول مرة أخرى، حيث يتوفر للاعب ثلاث محاولات للإجابة. بعد هذه المحاولات الثلاث يخسر اللاعب نقطة واحدة ويصبح العلم أحمر، ويختفي السؤال.

المخطط الانسيابي



لإنشاء مظاهر جديدة لكائن:

> ١ أضف الكائن (**Green Flag**) إلى اللعبة.

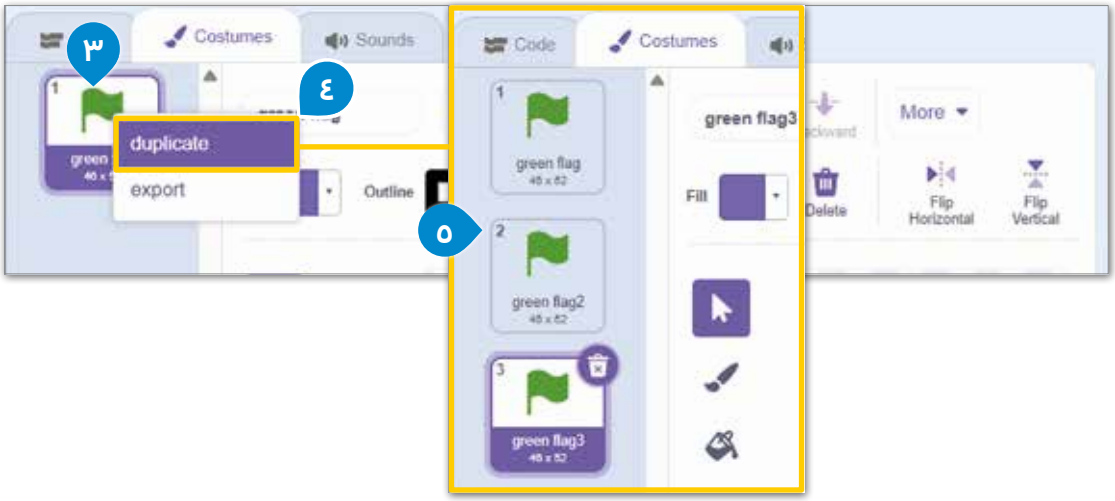
> ٢ انقر تبويب **Costumes** (المظاهر).

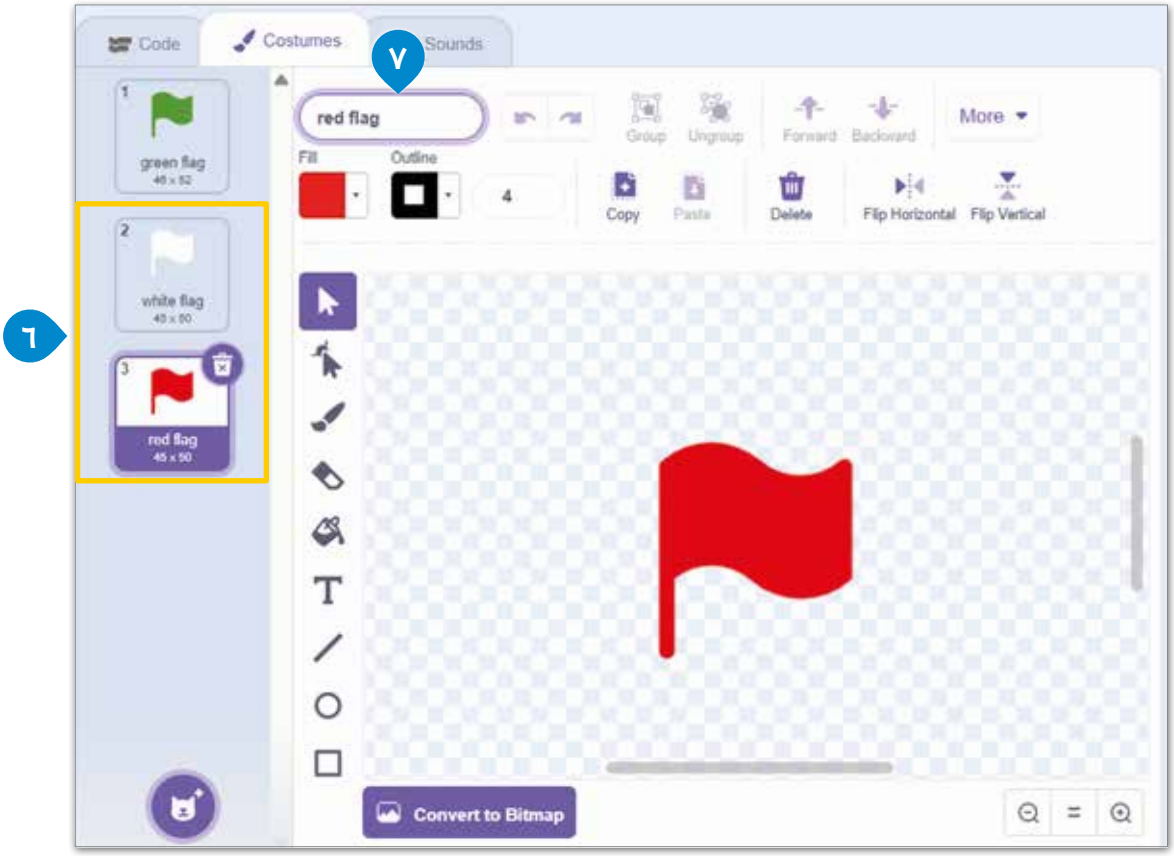
> ٣ انقر بزر الفأرة الأيمن على المظهر الأول المسمى (**green flag**)، ٤ ثم انقر **duplicate** (مضاعفة).

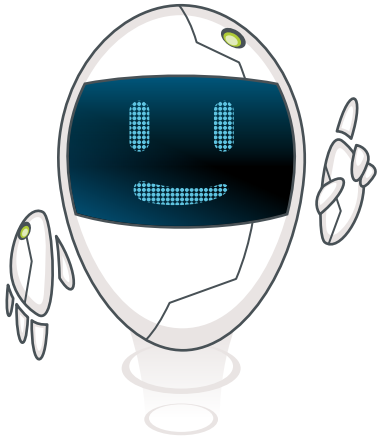
> ٥ كرّر الخطوات السابقة مرة أخرى لإنشاء مظهر آخر.

> ٦ لوّن المظهر (**green flag2**) باللون الأبيض والمظهر (**green flag3**) باللون الأحمر.

> ٧ غيّر اسمَي المظهرين الجديدين.



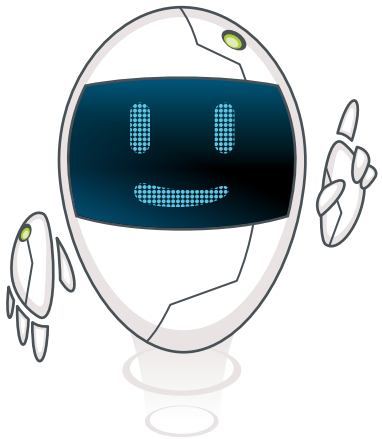




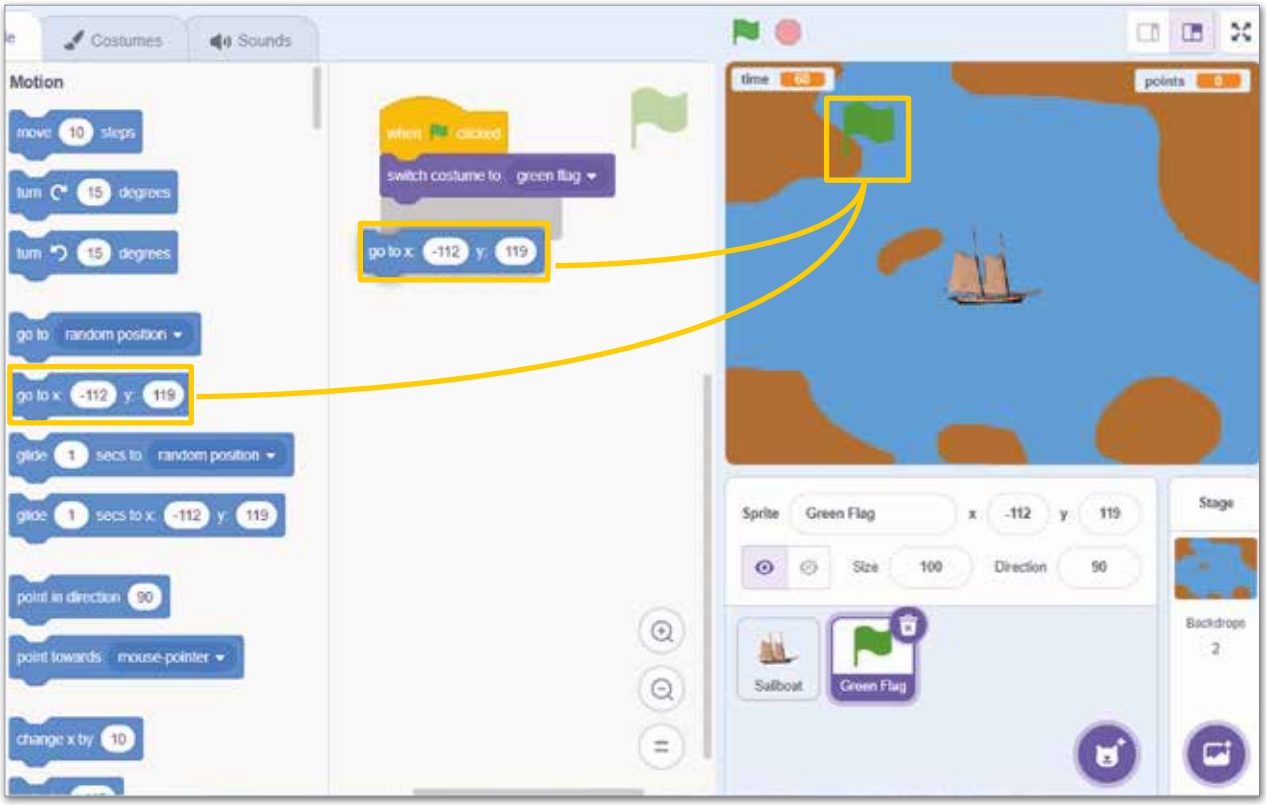
الآن، مظاهـر الكائـن الأول أصبحـت جاهزة ويمكنـك برمجته. يمكنك تـوفير الوقـت ببرمجـة الكائن الأول ثـم تكراره ثلاث مـرات، ولكن في هذه الحالة تذكّر تغيير الأسئلة وإحداثيات x و y.

برمجة العلَم

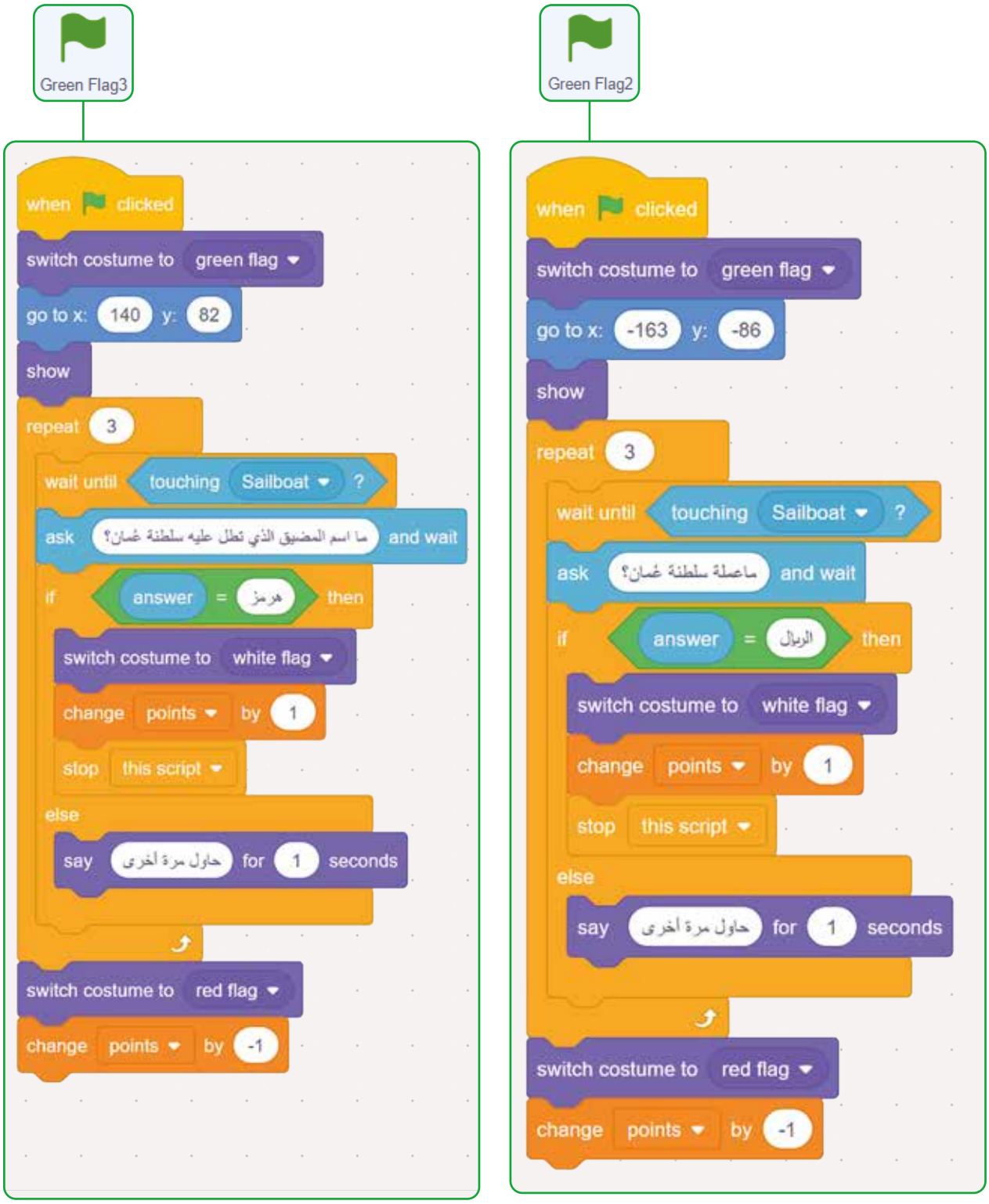
بعد إنشاء المظاهر الجديدة للكائن (Green Flag)، ستنشئ المقطع البرمجي الخاص به. تؤدي لبنة ( ) **wait until** (**انتظر حتى** ( )) إلى إيقاف المقطع البرمجي مؤقتًا والانتظار حتى يلمس الكائنُ (Sailboat) الكائنَ (Green Flag).

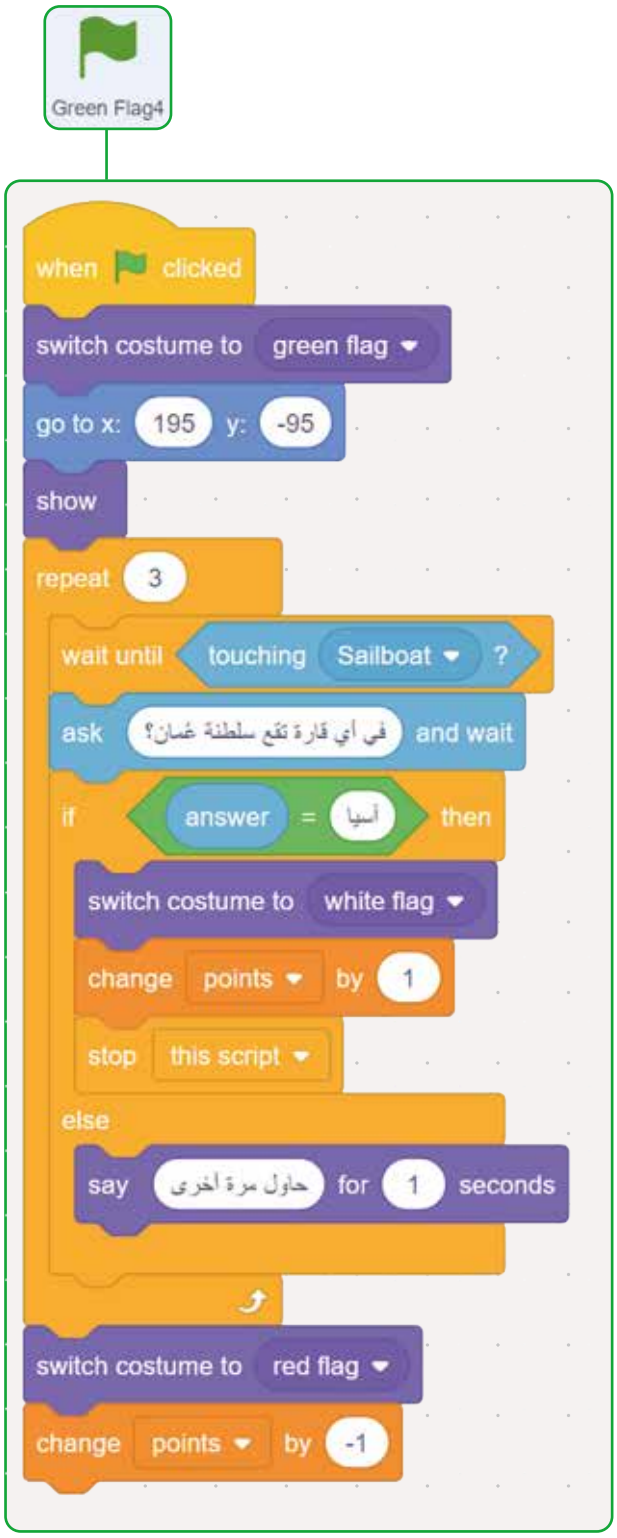


قبل إنشاء المقطع البرمجي لكل عَلَم، اسحبه وأفلته إلى النقطة المطلوبة. بهذه الطريقة سيتم ضبط إحداثيات x و y تلقائيًّا في لبنة **go to x: ( ) y: ( ) (اذهب إلى الموضع س: ( ) ص: ( ))**.

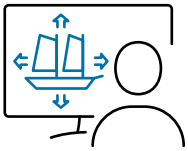


أكمل برمجة الكائن (Green Flag2)، والكائن (Green Flag3)، والكائن (Green Flag4) كالآتي:




Green Flag4
when clicked
switch costume to green flag
go to x: 195 y: -95
show
repeat 3
wait until touching Sailboat ?
ask في أي قارة تقع سلطنة عُمان؟ and wait
if answer = آسيا then
switch costume to white flag
change points by 1
stop this script
else
say حاول مرة أخرى for 1 seconds
switch costume to red flag
change points by -1

ستحتوي المنصة النهائية على أربعة أعلام في أربعة مواضع مختلفة. من الجيد وجود مساحة كافية بين الأعلام ليتمكن القارب الشراعي من التحرك بينها.



الاختبار

بعد الانتهاء من إنشاء المؤقت، ونظام تسجيل النقاط، والمقاطع البرمجية الخاصة بالأعلام الأربعة، اختبر اللعبة وأجب عن الأسئلة الآتية:

- هل هناك مساحة كافية بين الأعلام ليتمكن الكائن (Sailboat) من السير بينها؟
- هل لدى اللاعب وقت كافٍ، أو يحتاج إلى وقت أطول للإجابة عن الأسئلة؟
- هل الأسئلة مناسبة لزملائك؟

بناءً على إجاباتك، قم بإجراء بعض التغييرات لتحسين اللعبة، مثل تغيير متغيّر الوقت، أو تعديل المسافة بين الأعلام، أو تغيير الأسئلة.

تذكّر أن تحفظ عملك.

لنطبق معًا

تدريب ١

اشرح أهمية المتغيرات في اللعبة.

تدريب ٢

أنشئ لعبة بالونات باستخدام تطبيق Scratch. أضف ثلاثة كائنات (Balloon)، واجعلها تظهر الواحد تلو الآخر باستخدام لبنة wait ( ) seconds (انتظر ( ) ثانية).

تدريب ٣

أنشئ لعبة سرعة باستخدام تطبيق Scratch، حيث يتحرك كائن على المنصة يمينًا ويسارًا بشكل مستمر. أمام اللاعب 10 ثوانٍ قبل انتهاء اللعبة، وعليه النقر على الكائن لزيادة نقاطه.

ستحتوي اللعبة على كائن واحد.

ستحتوي اللعبة على متغيرين: أحدهما للنقاط (النتيجة)، والآخر لحساب الوقت.

تدريب ٤

أنشئ لعبة تفاعلية باستخدام تطبيق Scratch بحيث تحتوي على كائنين وهما (Hare) و (Jar).

١. في هذه اللعبة، سيسأل الكائن (Hare) اللاعبَ السؤال الآتي: "هل تفضل العسل أو المربى؟".

٢. بناءً على إجابة اللاعب، سيُغيّر الكائن (Jar) مظهره ليُظهر إما العسل أو المربى.

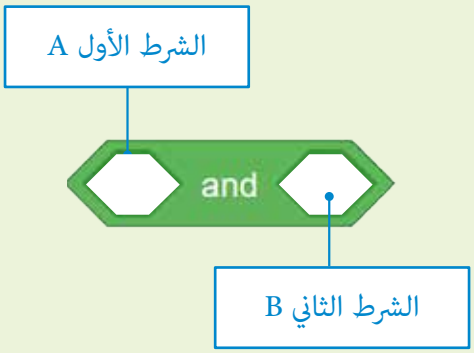
٣. استخدم لبنات if (إذا) الشرطية.

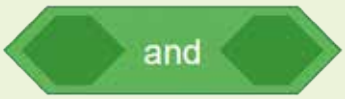
الدرس ٣-١:

اتخاذ القرارات

المعاملات المنطقية في تطبيق Scratch

لاتخاذ قرارات في اللعبة تُستخدم المعاملات المنطقية للتحقق من الشروط. يمكن أن تتضمن المضلعات السداسية الأضلاع لبنات أخرى سداسية الأضلاع، مثل لبنة **key ( ) pressed?** (**مفتاح ( ) مضغوط؟**) و **touching ( ) ?** (**ملامس لـ ( ) ؟**).

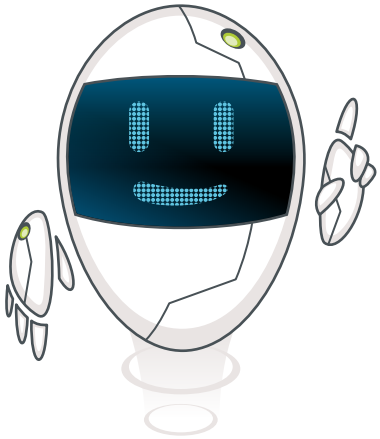




لبنة **and** ( ) (( ) **و** ( )): تتحقق من شرطين. إذا كان كلا الشرطين صحيحين، فإن اللبنة تعطي نتيجة صحيحة.

لبنة **or** ( ) (( ) **أو** ( )): تتحقق من شرطين. إذا كان أحد الشرطين صحيحًا، فإن اللبنة تعطي نتيجة صحيحة.

لبنة ( ) **not** (**ليس** ( )): تتحقق من شرط واحد وتعكس نتيجته. إذا كان الشرط صحيحًا، تصبح نتيجته خاطئة. وإذا كان الشرط خاطئًا، تصبح نتيجته صحيحة. على سبيل المثال: إذا كان الشرط "أنا كتبت الواجب"، فإن لبنة ( ) **not** (**ليس** ( )) تجعل نتيجته "أنا لم أكتب الواجب".

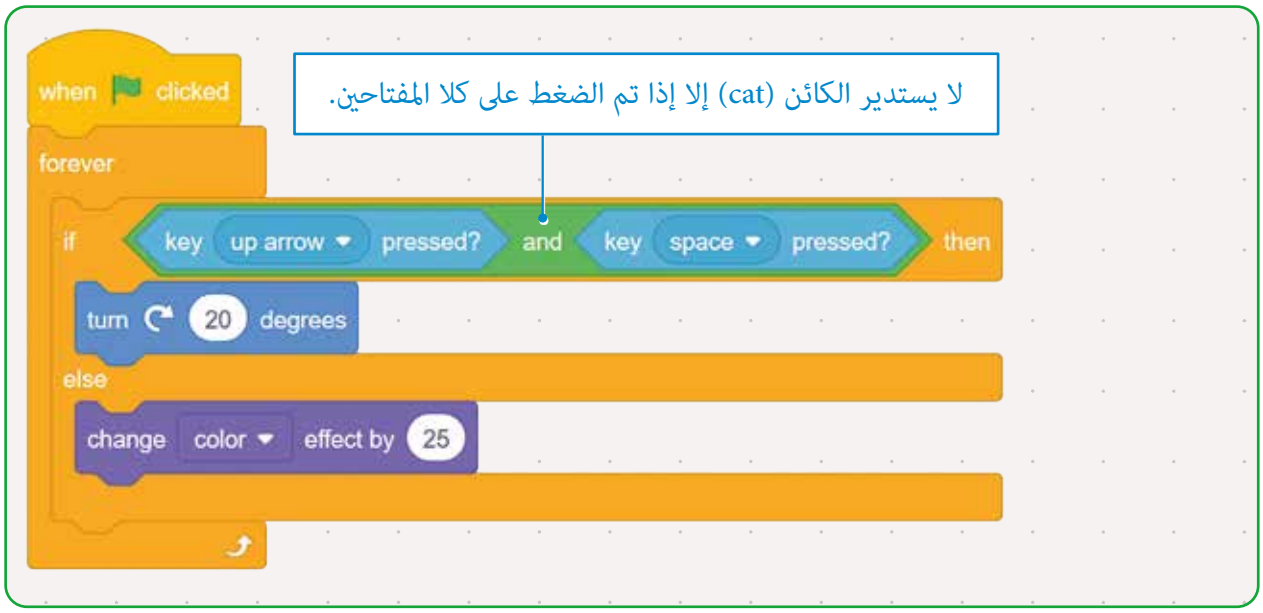


انتقل إلى تطبيق Scratch وبرمج الكائن (Cat) بالمقاطع البرمجية الآتية للمعاملات المنطقية الثلاثة. ماذا تلاحظ؟ ناقش ذلك مع معلمك.

المعامل المنطقي ( ) and ( ) (( ) و ( ))

في بعض الأحيان، تحتاج إلى تحقق شرطين في الوقت نفسه حتى يحدث شيء معين.

في المثال التالي يغيّر الكائن (Cat) لونه باستمرار، لكنه يتوقف عن تغيير لونه ويبدأ في الدوران فقط عندما يتم الضغط على مفتاح السهم لأعلى ومفتاح المسافة معًا في الوقت نفسه.



المعامل المنطقي ( ) or ( ) (( ) أو ( ))

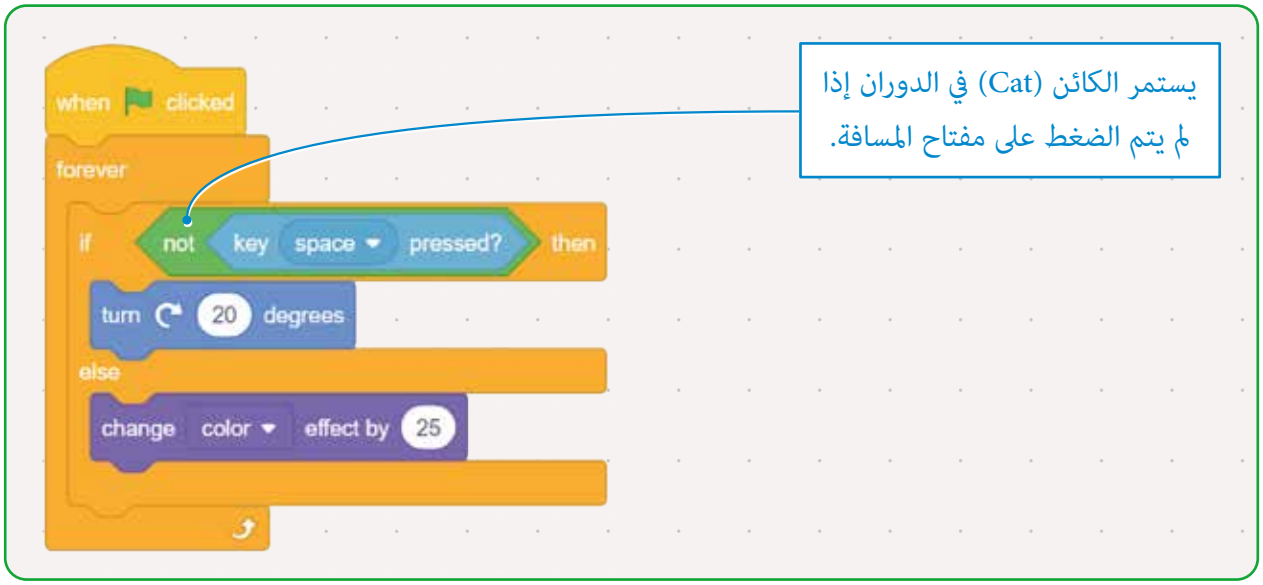
في بعض الأحيان، يتطلب أن يتحقق شرط واحد فقط حتى يحدث شيء معين.

في هـذا المثـال، يـغيّر الكائـن (Cat) لونـه، ولكنه يتوقـف عن تغيير لونـه ويبدأ بالدوران فقـط إذا ضغطت على مفتاح السهم لأعلى أو مفتاح المسافة من لوحة المفاتيح.

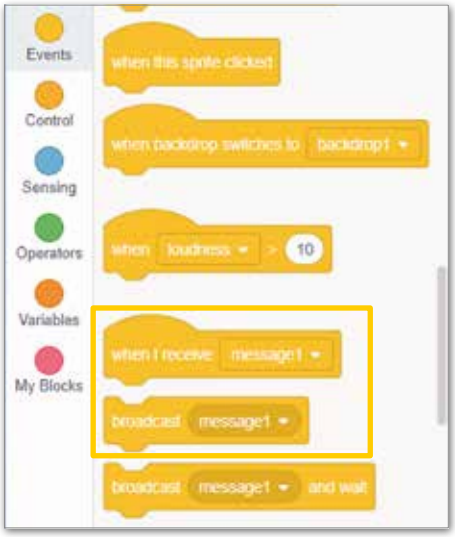
المعامل المنطقي ( ) not (ليس ( ))

في بعض الأحيان، يمكن أن يحدث شيء ما عندما لا يتحقق شرط معين.

في هـذا المثـال، يسـتمر الكائـن (Cat) في الـدوران طـالما أنـك لا تضغـط على مفتـاح المسـافة. ولكـن عنـد الضغط على مفتاح المسافة، يتوقف عن الدوران ويبدأ في تغيير لونه.



البث



البـث يشـبه إرسـال رسـالة إلى كائنـات أخرى أو أجـزاء أخرى من المقطـع البرمجـي. عندمـا يقوم أحـد الكائنات ببث رسـالة، يمكن برمجـة الكائنـات الأخرى لتلقي تلك الرسـالة وتنفيذ إجراء معين. يُعـدّ البـث مفيـدًا في لعبتـك؛ لأنـه يـؤدي إلى تفعيل إجـراءات في أجـزاء مختلفـة مـن اللعبـة. توجد لبنـات البث ضمـن فئة لبنات Events (الأحداث).

لبنـة ( ) **broadcast (بـث ( )):** تحمـل اسمًا محـددًا، وتقـوم بإرسـال رسـالة بـث إلى جميـع أجـزاء المشروع. حيـث سـيتم تفعيـل أي مقاطـع برمجيـة في الكائنـات التـي تسـتخدم لبنـة ( ) **when I receive (عندمـا أتلقـى ( ))** والمُعـدّة لاسـتقبال رسالة البث المحددة.

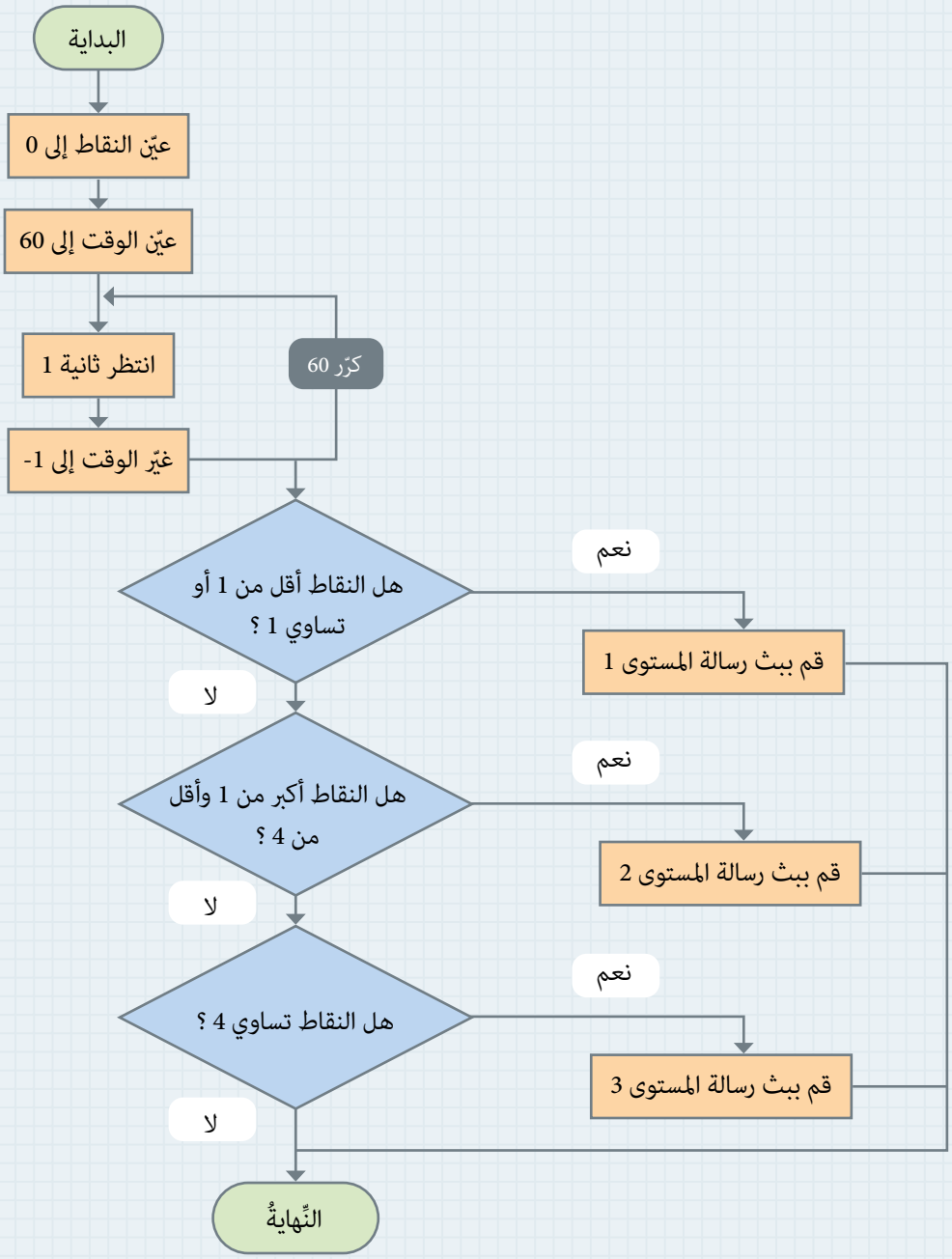
لبنـة ( ) **when I receive (عندمـا أتلقـى ( ))**: تُفعِّـل المقطع البرمجي الخاص بها عند إرسـال رسـالة البـث المحددة من قِبل مقطـع برمجـي آخر. عند إرسـال رسـالة بـث إلى جميـع أجزاء المشروع، يتم تفعيل جميع اللبنات المُعدّة لاستقبالها.

نهاية لعبة البحّار العُماني

عندما يصل متغير time (الوقت) إلى صفر، تنتهي اللعبة، وتظهر نقاط اللاعب من خلال الآتي:

- يتحقق المقطع البرمجي من عدد النقاط باستخدام العوامل المنطقية.
- يؤدي كل مستوى من مستويات النقاط إلى تفعيل مقطع برمجي مختلف لكل كائن.

المخطط الانسيابي



لإرسال رسالة:

> ١ أضف لبنة **if** ( ) **then** (**إذا** ( )) أسفل لبنة **repeat** ( ) (**كرّر** ( ) **مرة**) في المقطع البرمجي الخاص بالكائن (**Sailboat**).

> ٢ من فئة لبنات **Operators** (العمليات) أدرج المعامل ( ) **or** ( ) (( ) **أو** ( )).

> ٣ في المعامل المنطقي، أدخل الشرط.

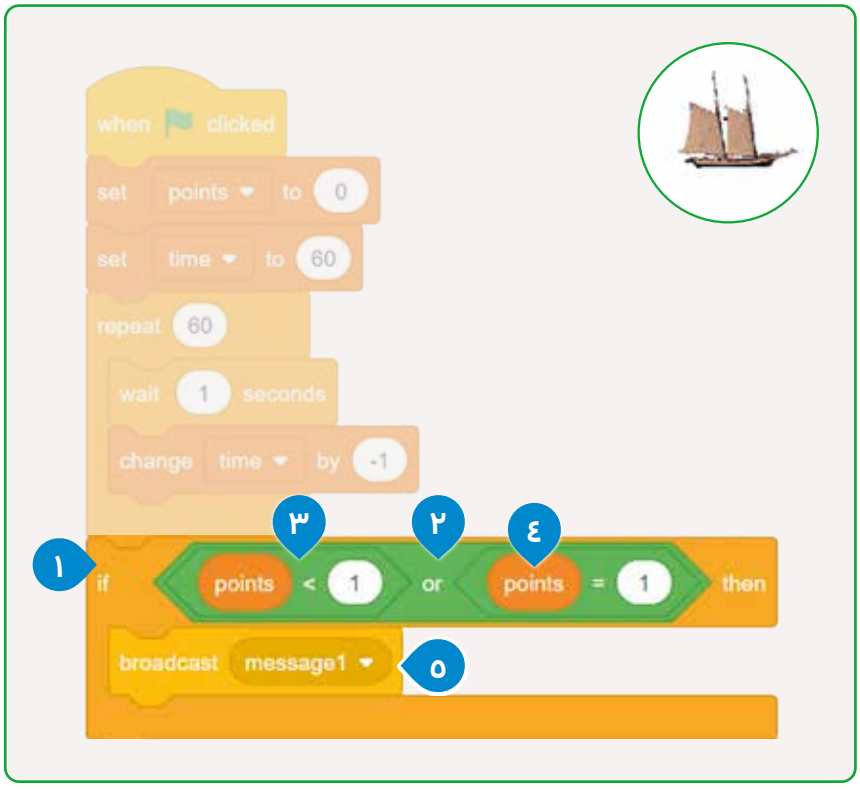
> ٤ ثم أضف المتغير **points** (النقاط) والأرقام المستخدمة للمقارنة.

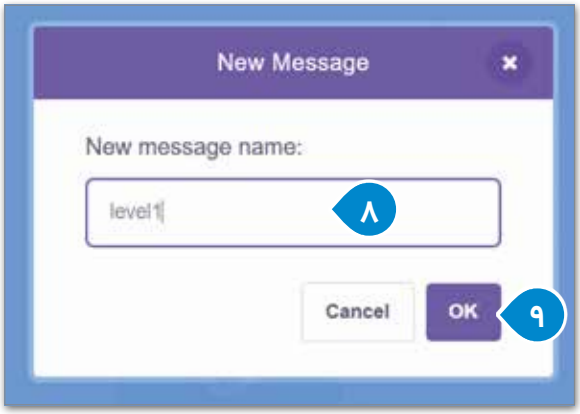
> ٥ أضف لبنة ( ) **broadcast** (**بث** ( )) داخل لبنة ( ) **if** (**إذا** ( )).

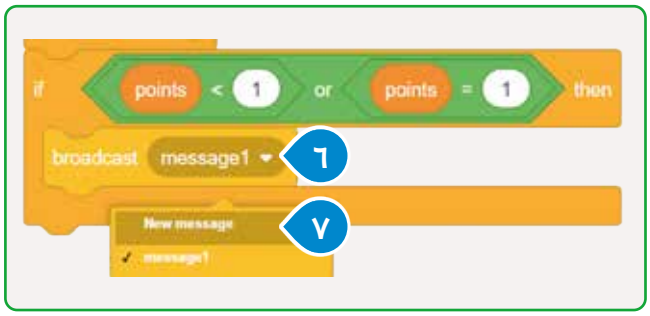
> ٦ انقر لبنة ( ) **broadcast** (**بث** ( ))، ٧ واختر **New message** (رسالة جديدة).

> ٨ في النافذة المنبثقة، اكتب اسم الرسالة، ٩ وانقر **OK** (موافق).

> ١٠ كرّر الخطوات السابقة لإضافة المعاملات ( ) **and** ( ) (( ) **و** ( ))، و ( ) = ( )، وأدخل التعبيرات الشرطية ورسائل البث المناسبة.







يتحقق هذا الجزء من المقطع البرمجي من حصول اللاعب على نقطة واحدة أو أقل. إذا تحقق ذلك، يُرسل المقطع البرمجي الرسالة **level1** **(المستوى 1)**.

يتحقق هذا الجزء من المقطع البرمجي من حصول اللاعب على نقطتين إلى ثلاث نقاط. إذا تحقق ذلك، يُرسل المقطع البرمجي الرسالة **level2** **(المستوى 2)**.

١٠

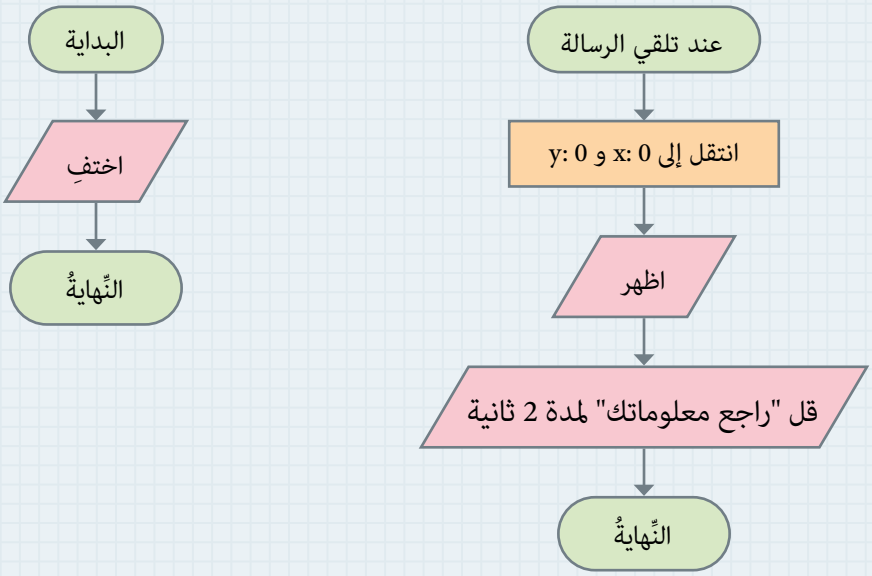
يتحقق هذا الجزء من المقطع البرمجي من حصول اللاعب على أربع نقاط. إذا تحقق ذلك، يُرسل المقطع البرمجي الرسالة **level3** **(المستوى 3)**.

استقبال الرسالة من الكائن المرسل

ستنشـئ الآن ثلاثـة كائنـات (Button4)، و (Button5)، و (Earth) تظهـر في نهايـة اللعبـة بنـاءً على أداء اللاعـب. سـتكون هـذه الكائنـات مخفيـة في بدايـة اللعبـة، وسـيظهر واحـد منهـا فقـط في النهاية، حسـب النقاط التي سيسجلها اللاعب.

سـيظهر الكائن بعد اسـتلام رسـالة محددة، حيث سيرسـل الكائن (Sailboat) رسـالة واحدة فقط في نهاية اللعبـة. يتـم اختيـار هذه الرسـالة باسـتخدام شروط وعمليـات منطقيـة في المقطع البرمجـي، والتي تتحقق بناءً على إجمالي نقاط اللاعب.

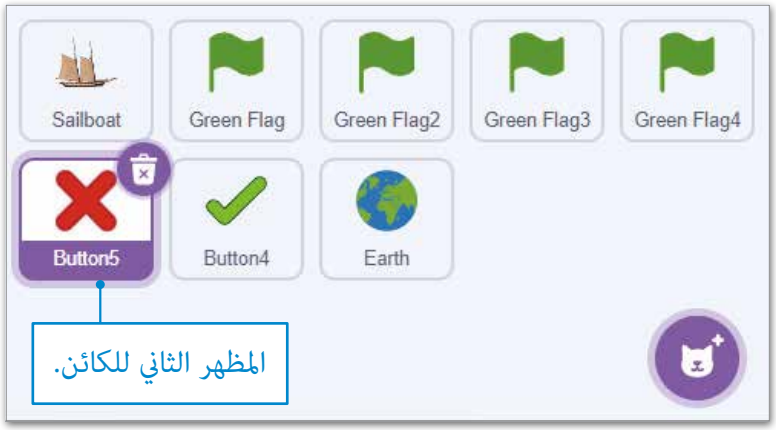
المخطط الانسيابي



لإنشاء المقاطع البرمجية الخاصة بمستوى اللاعب:

١ أضف ثلاثة كائنات جديدة، واحد لكل مستوى من مستويات اللاعب.

٢ اضبط الرسالة الصحيحة لكل لبنة ( ) **when I receive (عندما أتلقى** ( )).

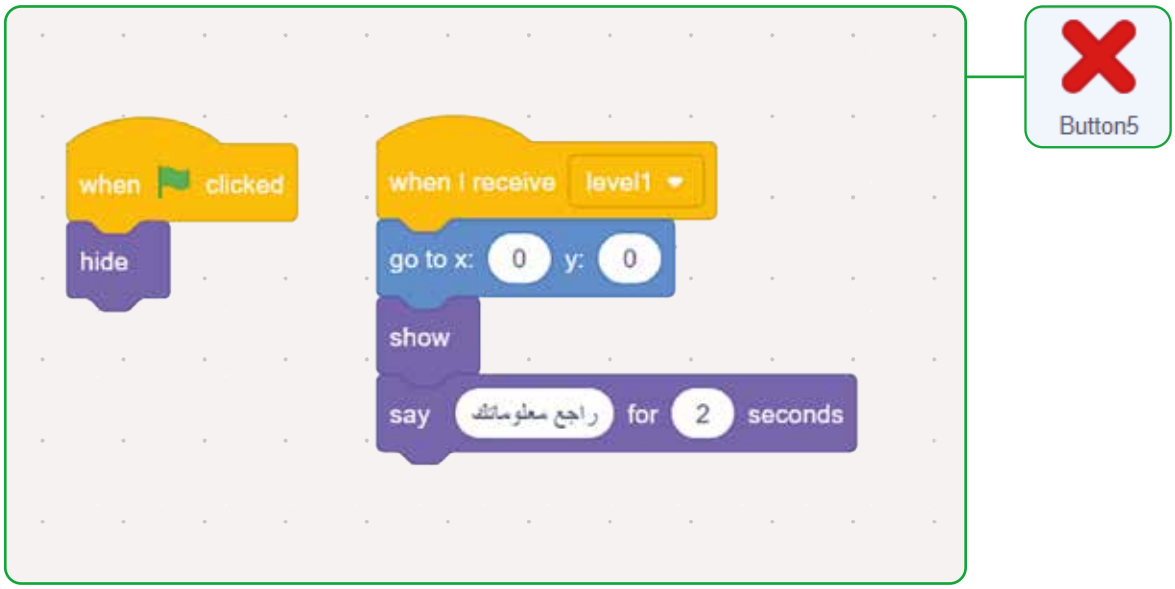


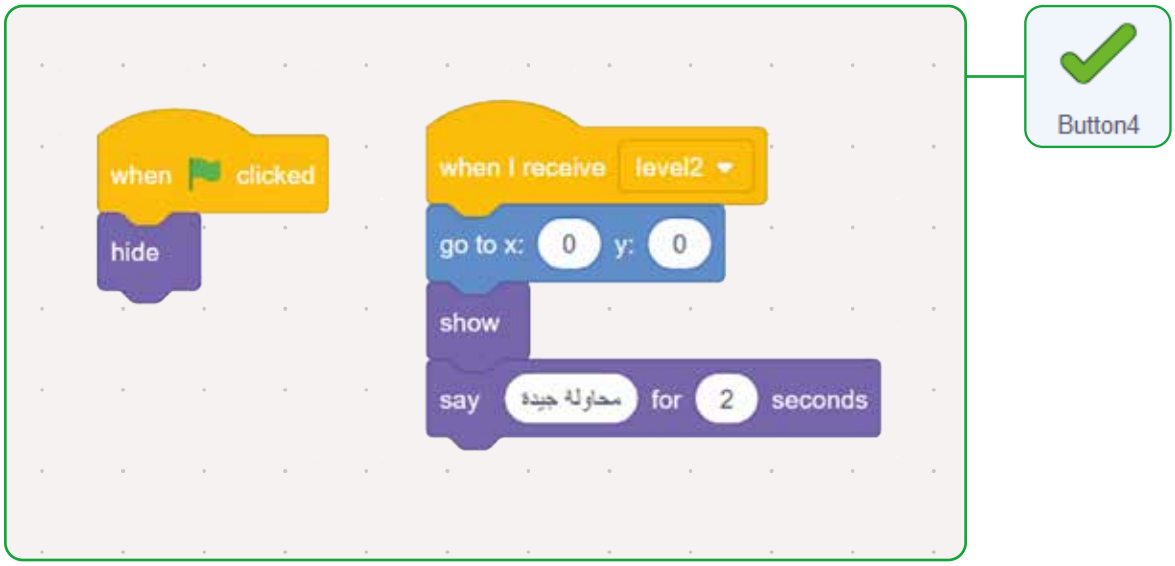
جدول تسجيل نقاط اللعبة يساعدك على تحديد الكائن الذي سيظهر حسب النقاط، مما يُمكّنك من ضبط الرسالة الصحيحة لكل كائن بسهولة.

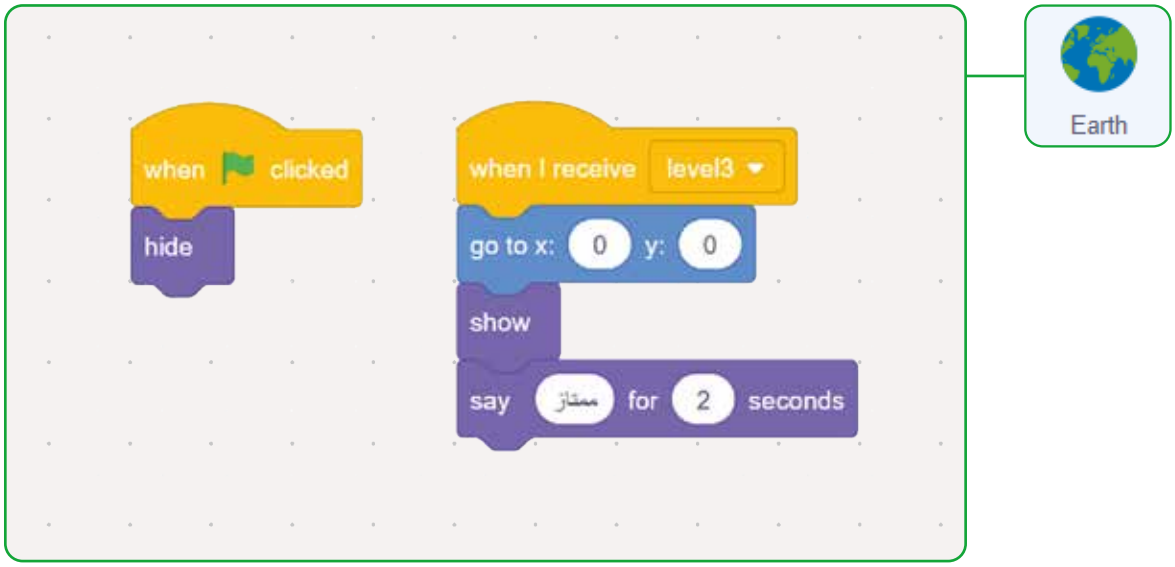
تسجيل النقاط

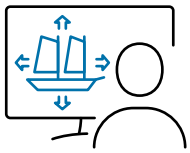
الكائن	المستوى	النقاط
Button5	1	1 أو أقل
Button4	2	2-3
Earth	3	4

المقاطع البرمجية للكائنات الثلاثة.







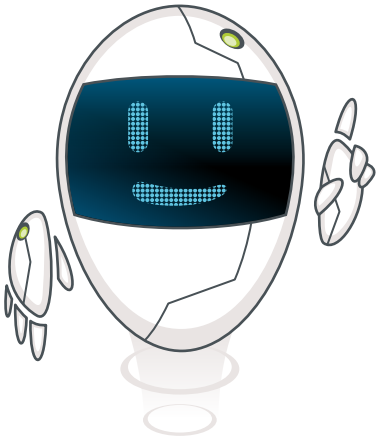


الاختبار

بعد إكمال اللعبة جرب تشغيلها، ثم أجب عن الأسئلة الآتية:

- هل نظام تسجيل النقاط صحيح؟
- هل تعمل لبنات البث بشكل صحيح؟
- هل الكائنات التي تظهر في النهاية مناسبة؟

بناءً على إجاباتك، قم بإجراء تغييرات لتحسين اللعبة، مثل تغيير النقاط والمستويات، أو استبدال الكائنات، أو تغيير الرسالة الأخيرة التي تظهر على الشاشة.



جرب اللعبة واكتب عدد النقاط التي سجلتها، ثم شاركها مع زملائك في الصف.

تذكّر أن تحفظ عملك.

اكتب المعلومات الجديدة التي تعلمتها عن الجغرافيا من هذه اللعبة، ثم اكتب سؤالين مختلفين ترغب في تضمينهما في اللعبة حول الثقافة العُمانية.

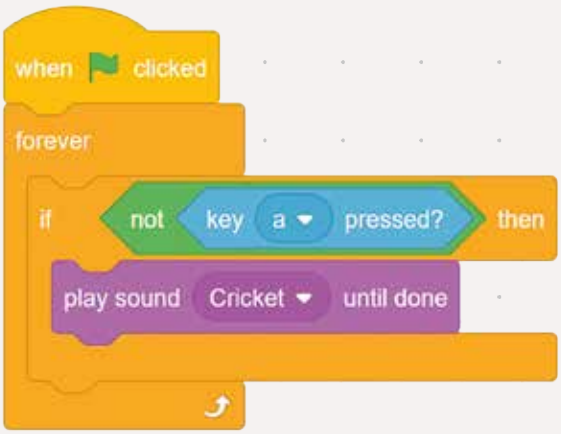
لنطبق معًا

تدريب ١

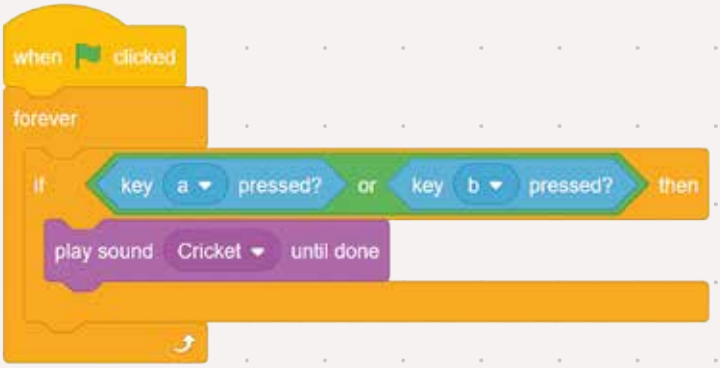
صِفْ وظيفة اللبنات الآتية:

تدريب ٢

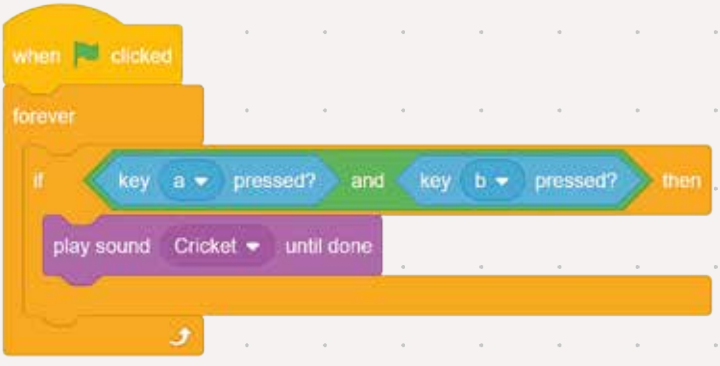
اقرأ المقاطع البرمجية، ثم اختر المفتاح أو المفاتيح التي تحتاج إلى الضغط عليها لتشغيل الصوت.



- ◯ مفتاح a.
- ◯ مفتاح A.
- ◯ أي مفتاح عدا مفتاح a.



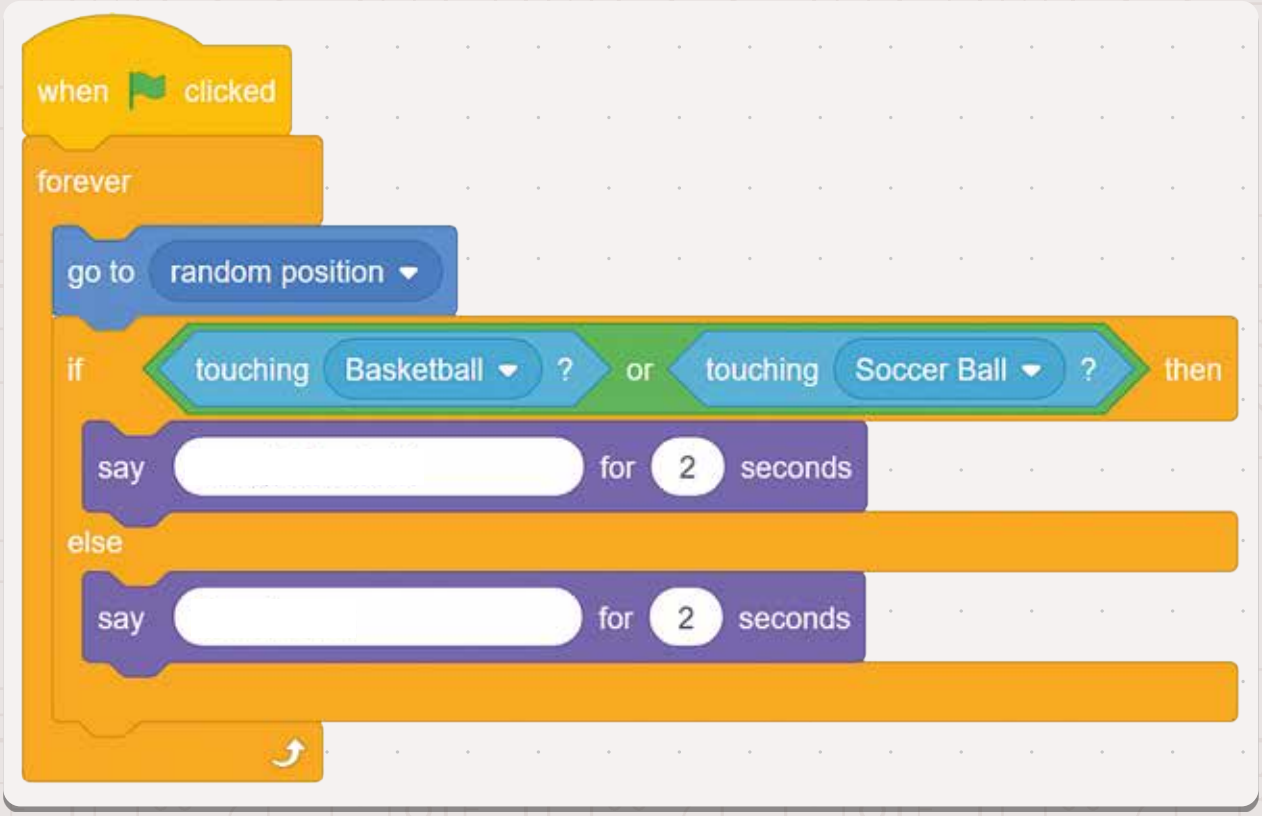
- ◯ أي مفتاح.
- ◯ مفتاحا a و b.
- ◯ مفتاح a أو b.

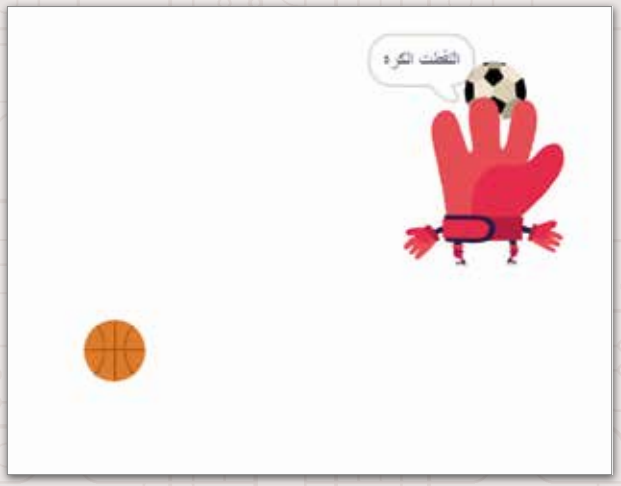


- ◯ مفتاحا a وb.
- ◯ مفتاحا b و a.
- ◯ جميع ما سبق.

تدريب ٣

أكمـل كل حقـل فـارغ بالعبـارة المفقـودة وفقًـا للصـورتين أدنـاه. انتقـل إلى تطبيـق Scratch للتحقـق من إجاباتك.





تدريب ٤

أنشئ مشروعًا في تطبيق Scratch يوضّح تفاعلًا بين كائنين باستخدام البث، حيث يطرح أحد الكائنات سؤالًا، ويجيب عنه كائن آخر عند استقباله للبث.

١. الكائن ١: يقول "ما ناتج جمع ٥ + ٣؟" لمدة ثانيتين، ثم يبث الرسالة.

٢. الكائن ٢: عندما يستقبل الكائن ٢ الرسالة يقول "الإجابة هي ٨" لمدة ثانيتين.

تدريب ٥

أنشئ لعبة في تطبيق Scratch، واستخدم البث لتغيير الخلفيات.

١. أضف خلفيتين وكائنين إلى مشروعك.

٢. ابدأ اللعبة بالخلفية الأولى.

٣. بعد ثلاث ثوانٍ، يُرسل الكائن الأول رسالة بث.

٤. عندما يتلقى الكائن الثاني الرسالة يقول:

"لقد استلمت الرسالة"، أو أي جملة تراها مناسبة، ثم تتغير الخلفية إلى الخلفية الثانية.

الدرس ٤-١:

الآلة الحاسبة

العمليات الحسابية

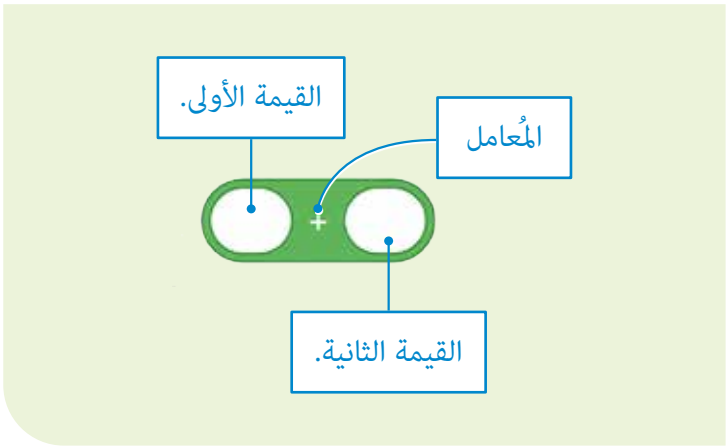
في البرمجة، تُستخدم الرموز الآتية لتمثيل العمليات الحسابية (Calculations). حيث يختلف عن الرياضيات رمزا القسمة والضرب فقط كما هو موضح أدناه.

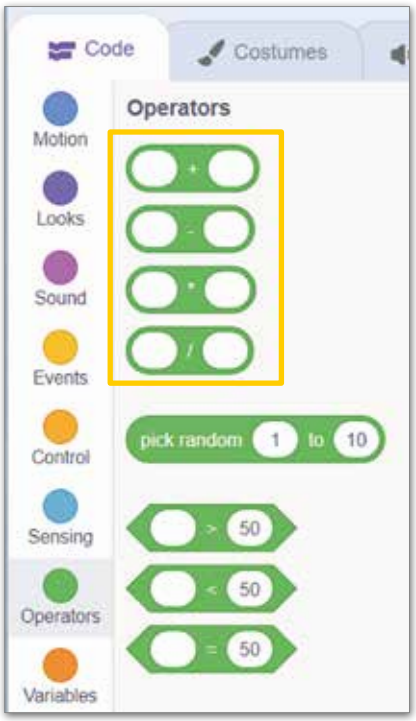
العمليات الحسابية في تطبيق Scratch

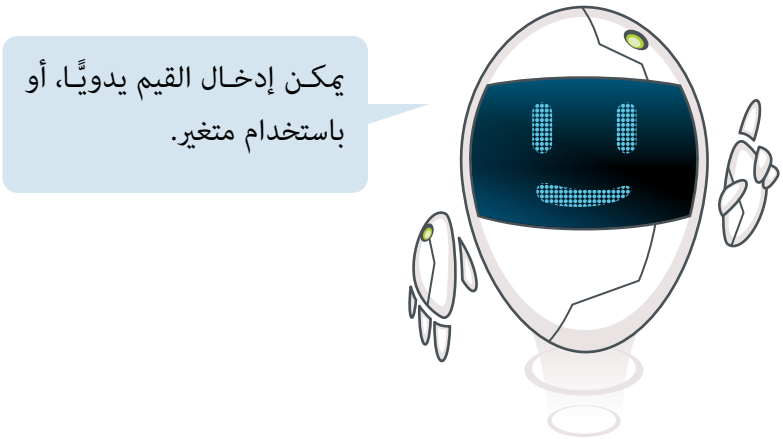
في تطبيق Scratch، يمكنك استخدام لبنات خاصة لتنفيذ العمليات الحسابية، مثل الجمع، والطرح، والضرب، والقسمة وغيرها.

يُطلق على الرمز الذي يعبّر عن عملية حسابية معيّنة اسم المُعامل. على سبيل المثال: علامة الجمع (+) هي المُعامل الذي يُمثل عملية الجمع.

في البرمجة تُسمّى الرموز التي تستخدمها لإجراء العمليات الحسابية المعاملات الرياضية، ويمكنك العثور عليها في فئة لبنات **Operators** (**العمليات**).







العمليات الحسابية باستخدام المتغيرات

أَنشئ مقطعًا برمجيًّا يُجري عمليات حسابية بين رقمين، يمثل كل رقم كائنًا يتغيّر مظهره عشوائيًّا كلما نقر المستخدم عليه، وكل مظهر عبارة عن رقم مختلف، ويُخزَّن هذا الرقم في متغير.

وعند ضغط المستخدم على مفتاح المسافة، يتغير المُعامل، فكل مظهر عبارة عن مُعامل مختلف. يتم تخزين النتيجة لكل عملية حسابية بين الرقمين وعرضها في متغير خاص بالنتيجة.

يتيح هذا المقطع البرمجي للاعب:

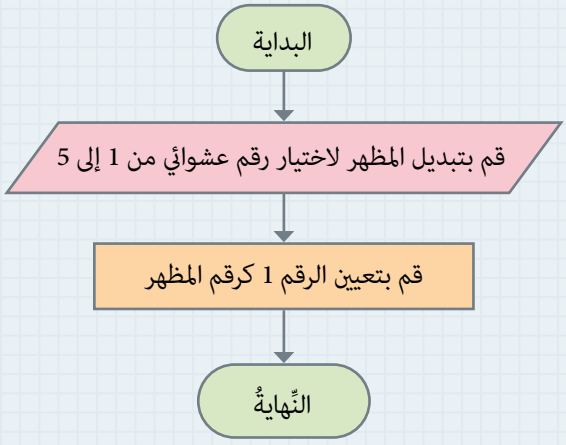
- الضغط على مفتاح المسافة للتبديل بين العمليات الحسابية المختلفة.
- إجراء عمليات حسابية بين رقمين باستخدام العملية المحددة.
- عرض النتيجة في متغير منفصل.

لإنشاء مشروع العملية الحسابية:

١ افتح مشروعًا جديدًا.

٢ احذف الكائن (Cat) وأضف الكائن (Glow-1).

المخطط الانسيابي



لإنشاء كائنين من الأرقام:

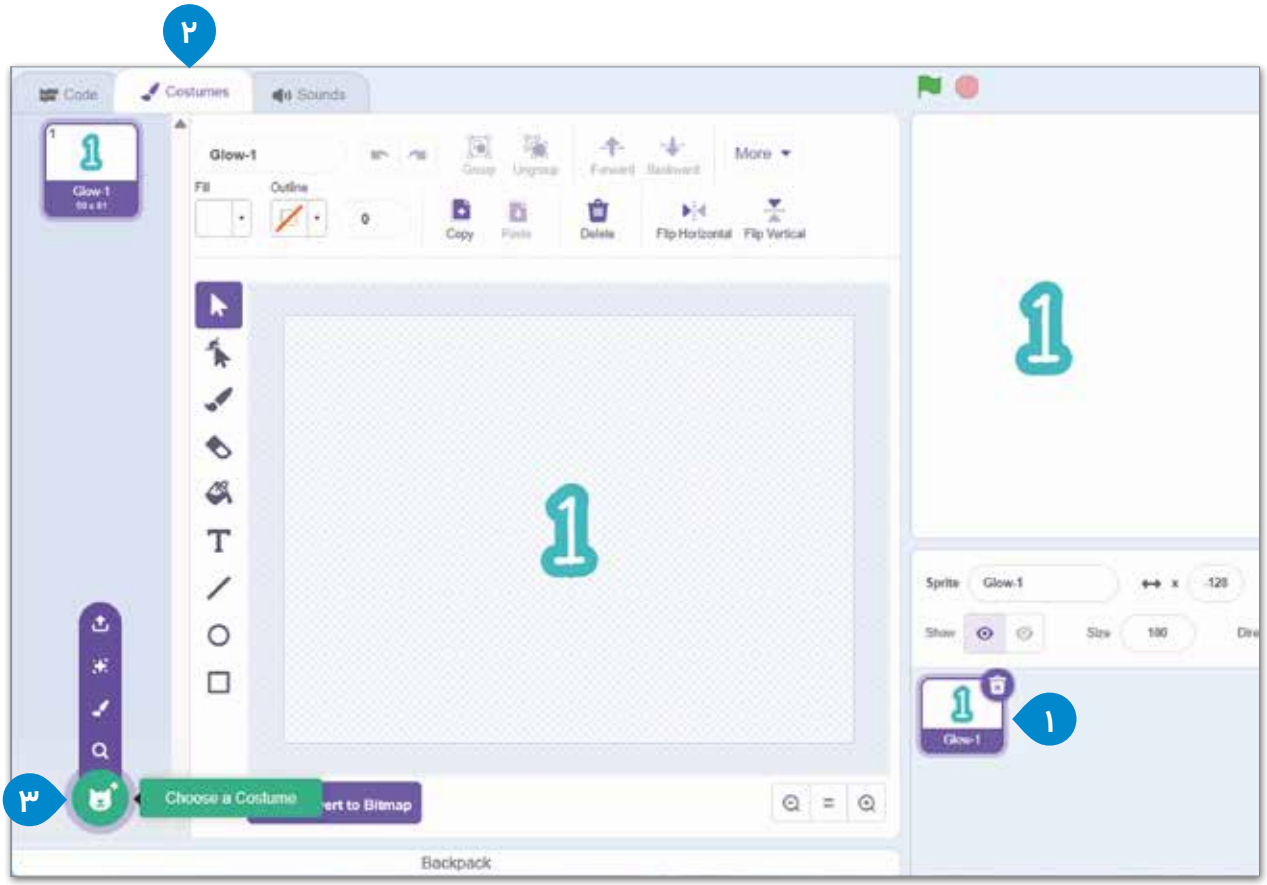
> ١ اختر الكائن (Glow-1).

> ٢ اختر تبويب **Costumes** (المظاهر).

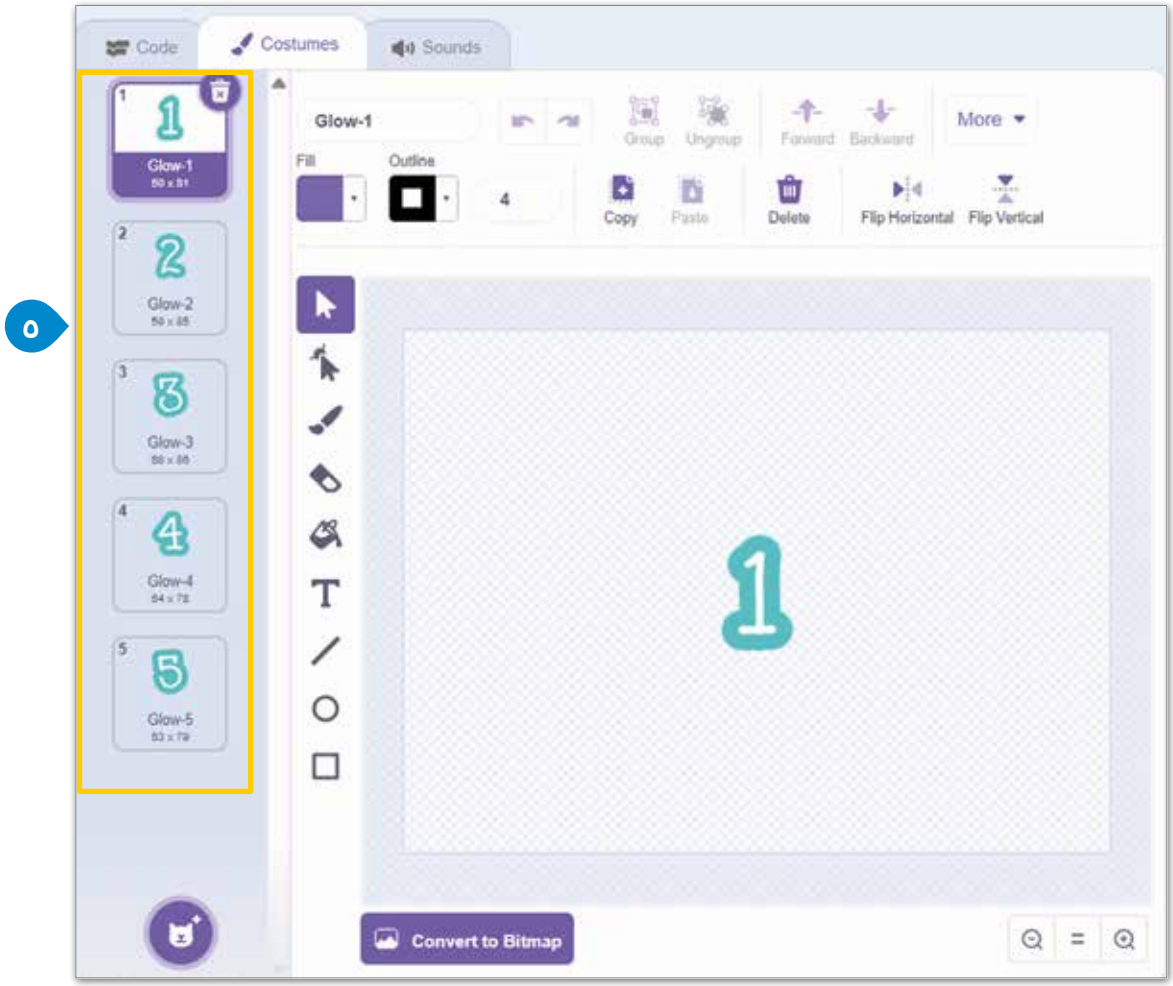
> ٣ انقر على **Choose a Costume** (اختيار مظهر).

> ٤ اختر الكائن (Glow-2).

> ٥ كرّر الخطوتين الثالثة والرابعة، واختر جميع الكائنات التي تحتاجها.

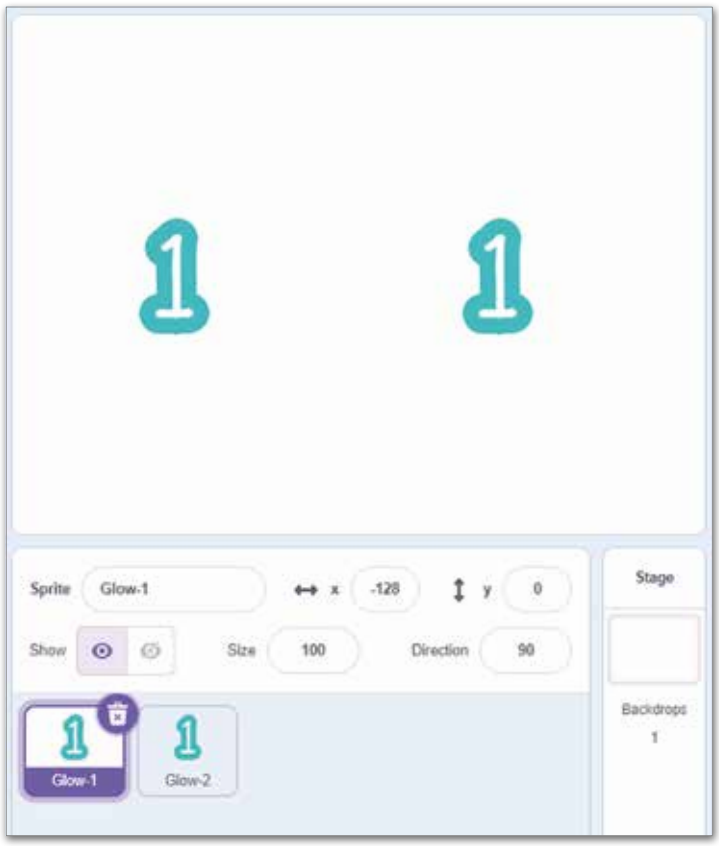


Back
Choose a Costume
Search
All
Animals
People
Fantasy
Dance
Music
Sports
Block-t
Block-u
Block-v
Block-w
Block-y
Block-z
Glow-0
Glow-1
Glow-2
Glow-3
Glow-4
Glow-5
Glow-6
Glow-7
٤

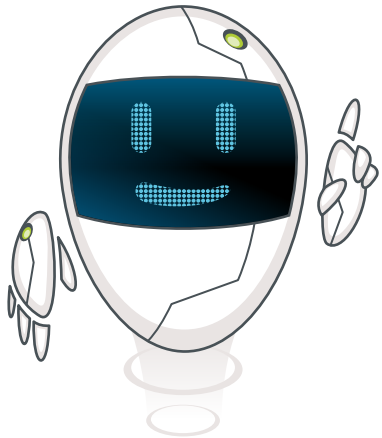

Code
Costumes
Sounds
Glow-1
Fill
Outline
4
Group
Ungroup
Forward
Backward
More
Copy
Paste
Delete
Flip Horizontal
Flip Vertical
Glow-2
Glow-3
Glow-4
Glow-5
Convert to Bitmap
٥

لإنشاء الكائن الثاني:

١ ضاعف الكائن (Glow-1).



عنـد مضاعفـة الكائن الأول، تُنسـخ معه جميع مظاهره تلقائيًا.



لحفظ الأرقام وعرضها على الشاشة:

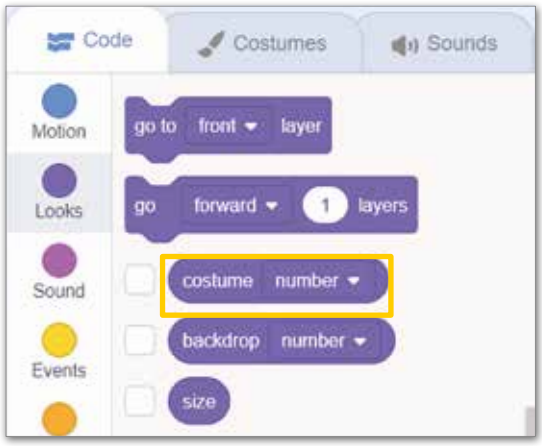
١ أنشئ متغيرًا لكل كائن باسم number1 (الرقم ١) و number2 (الرقم ٢).

٢ انقر على خانة الاختيار المجاور للمتغيرات لعرضها على المنصة.

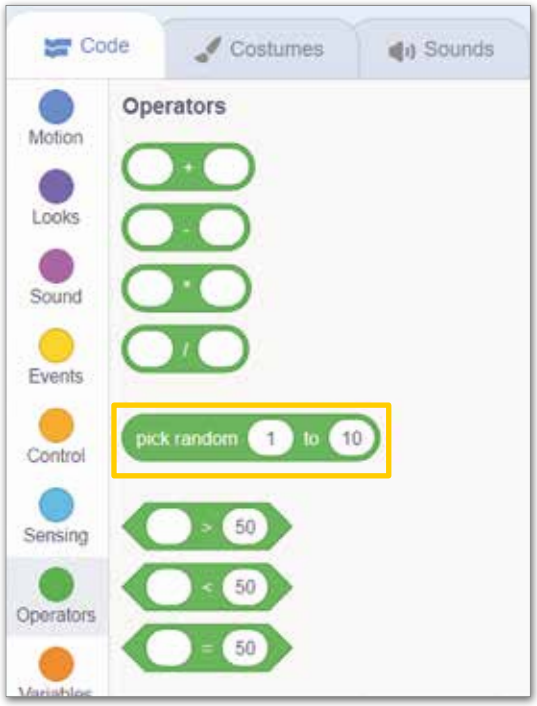
٣ ضع المتغيرات أعلى كل كائن.

لبنات المُراسِل في تطبيق Scratch

في تطبيق Scratch، تُستخدم لبنات المُراسِل (Reporter) كمدخلات داخل لبنات أخرى؛ للحصول على قيم معينة مثل الأرقام أو النصوص. ولا يمكن وضع هذه اللبنات بمفردها أو توصيلها بلبنات أخرى.



لبنة ( ) **Costume** (( ) **المظهر**): إحدى لبنات المُراسِل (Reporter) الموجودة في فئة **Looks** **(الهيئة)**، وهي متاحة فقط في الكائنات، حيث تعطي رقم المظهر أو اسمه.



تُعطي اللبنة رقمًا عشوائيًّا يتراوح بين الرقم الأول والثاني، بما في ذلك رقما البداية والنهاية. إذا كان كلا الرقمين عددًا صحيحًا، فسيكون الرقم المُعطَى عددًا صحيحًا أيضًا.

لإنشاء مقطع برمجي للكائنات:

> ١ من فئة لبنات **Events** (الأحداث)، اختر لبنة **when this sprite clicked (عند نقر هذا الكائن)**.

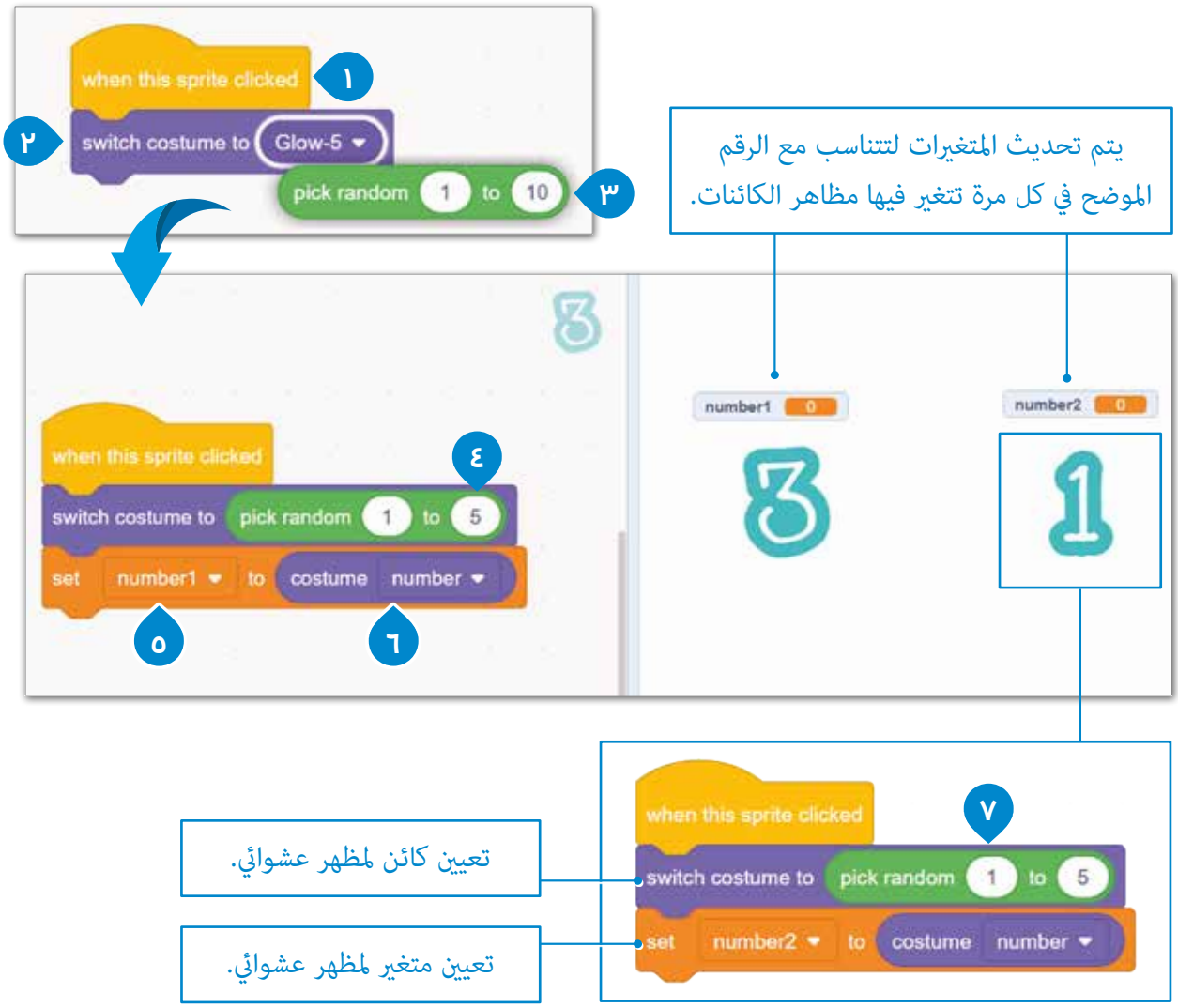
> ٢ من فئة لبنات **Looks** (الهيئة) أضف لبنة **switch costume to ( ) (غيّر المظهر إلى ( ))**.

> ٣ في لبنة **switch costume to ( ) (غيّر المظهر إلى)** أدخل لبنة **pick random ( ) to ( ) (عدد عشوائي بين ( ) و ( ))** التابعة لفئة لبنات **Operators** (العمليات)، ٤ ثم اضبطها من **1** إلى **5** (لأن هناك خمسة مظاهر).

> ٥ من فئة لبنات **Variables** (المتغيرات) أضف لبنة **set ( ) to ( ) (اجعل ( ) مساويًا ( ))** واضبطها إلى **number1**.

> ٦ في لبنة **set ( ) to ( ) (اجعل ( ) مساويًا ( ))** أدخل **لبنة costume number (رقم المظهر)**.

> ٧ كرّر الخطوات لإنشاء مقطع برمجي للكائن (Glow-2).

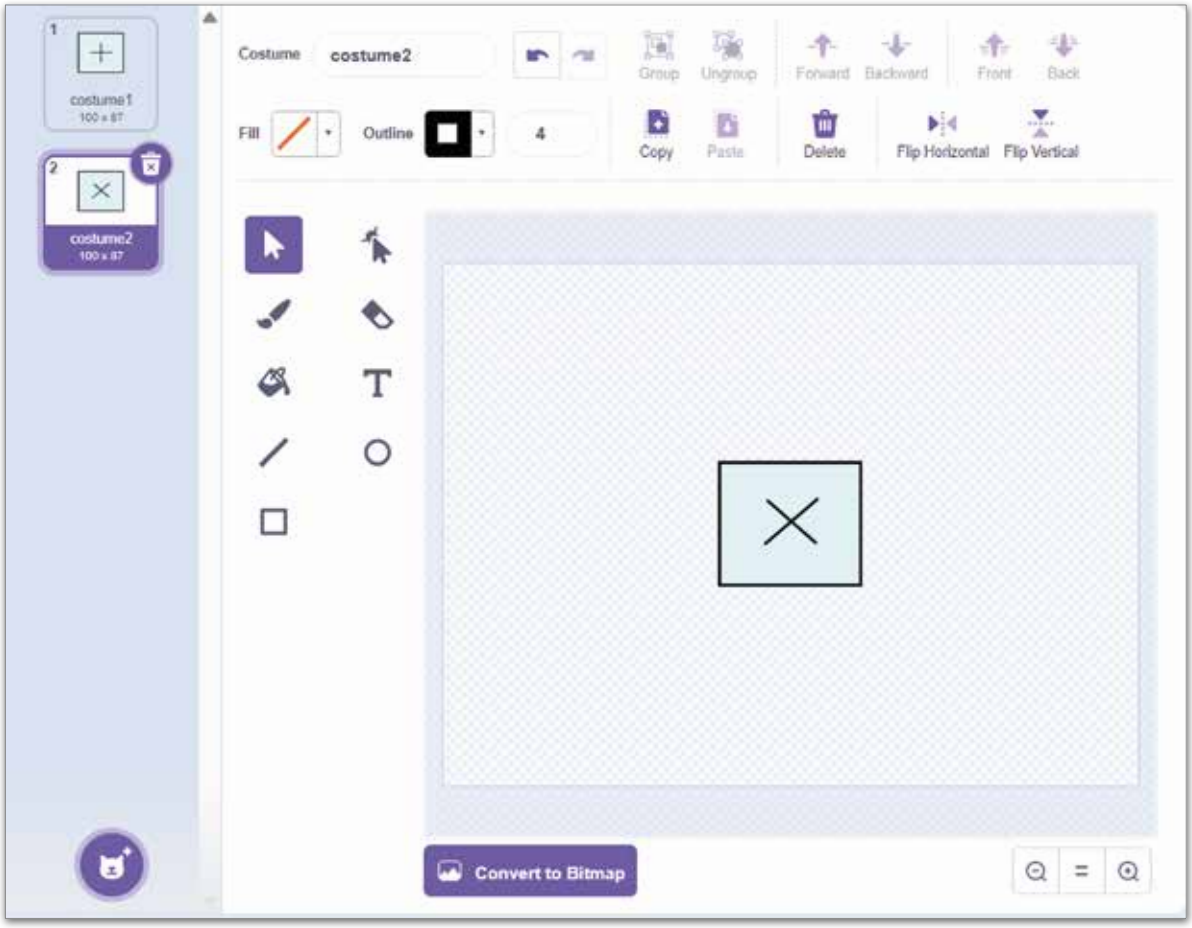


لإنشاء كائن يمثل مُعاملًا عدديًّا:

١ انقر على **Choose a Sprite** (اختيار كائن).

٢ انقر على **Paint** (رسم).

٣ أنشئ مظهرين، مظهر لمُعامل الجمع ومظهر لمُعامل الضرب.



المظهر 1 ← الجمع

المظهر 2 ← الضرب

لحفظ النتيجة وعرضها على الشاشة:

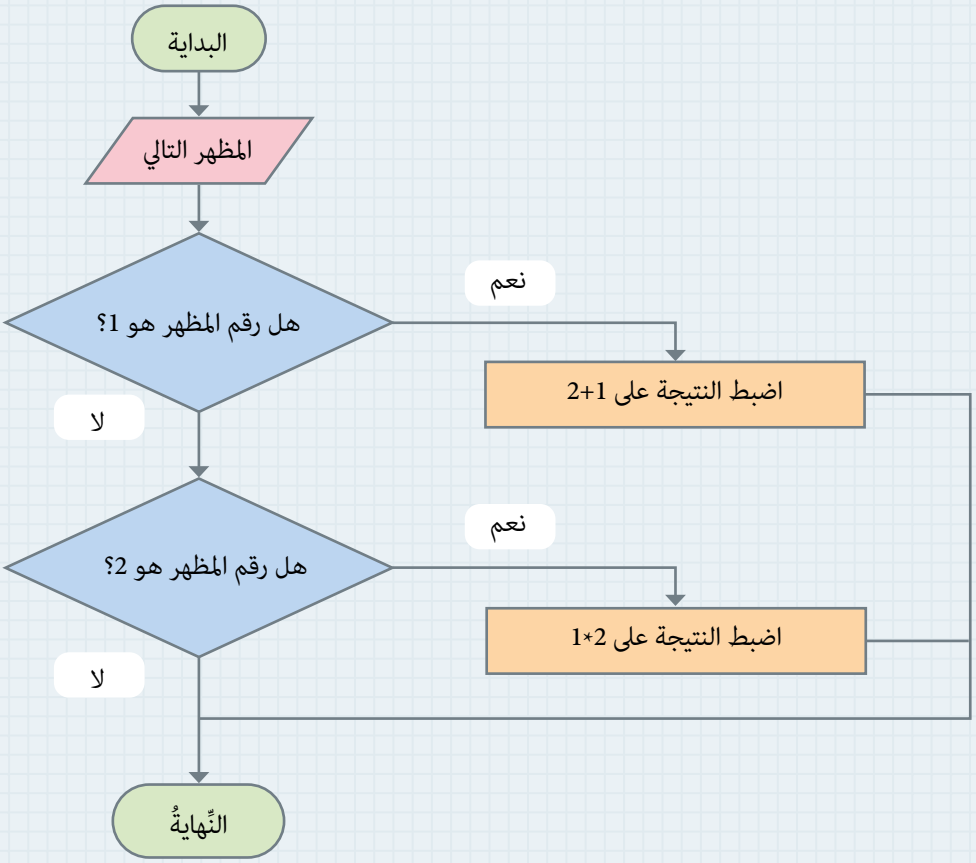
١ أنشئ متغيراً للكائن باسم result (النتيجة).

٢ انقر على خانة الاختيار بجوار المتغير result (النتيجة) لعرضه على المنصة .

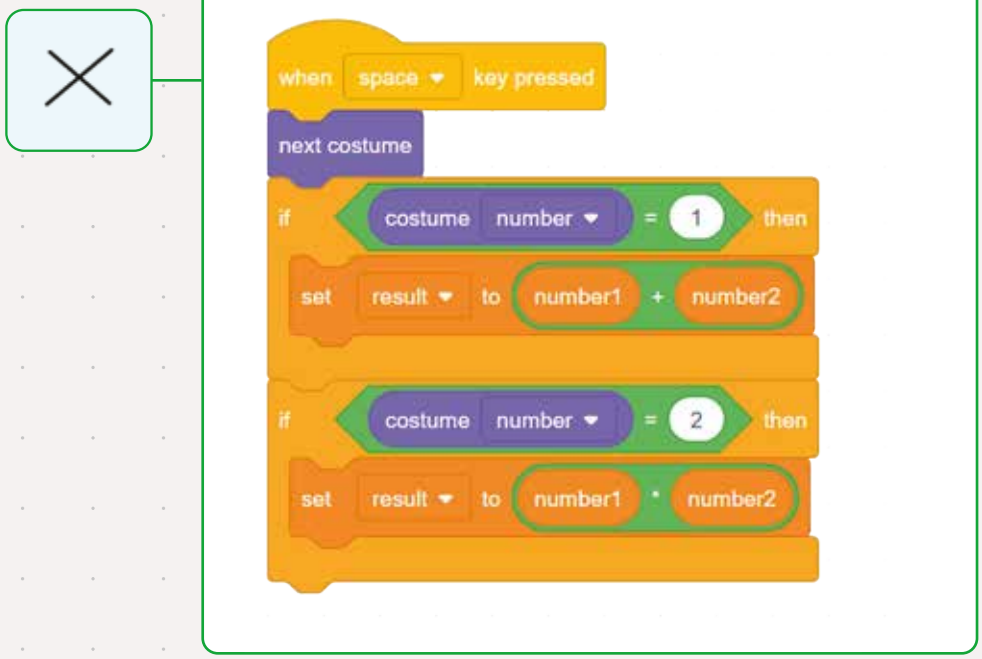
٣ ضع المتغير أسفل الكائن الذي أنشأته.

متغير (result): يخزن مخرجات العملية الحسابية.

المخطط الانسيابي



الآن، ستنشـئ المقطـع البرمجـي الخـاص بالكائن (Sprite1). سـيُغيّر هـذا الكائن مظهره، ثم يُنشـئ مُعاملًا يتـم حفظـه في مُتـغير النتيجـة. سـيعمل المقطـع البرمجي كآلة حاسـبة بسـيطة تُجـري عمليتين حسـابيتين هما: الجمع والضرب.

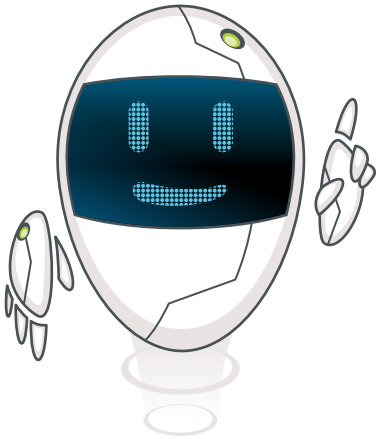


فيما يلي كيفية عمل المقطع البرمجي:

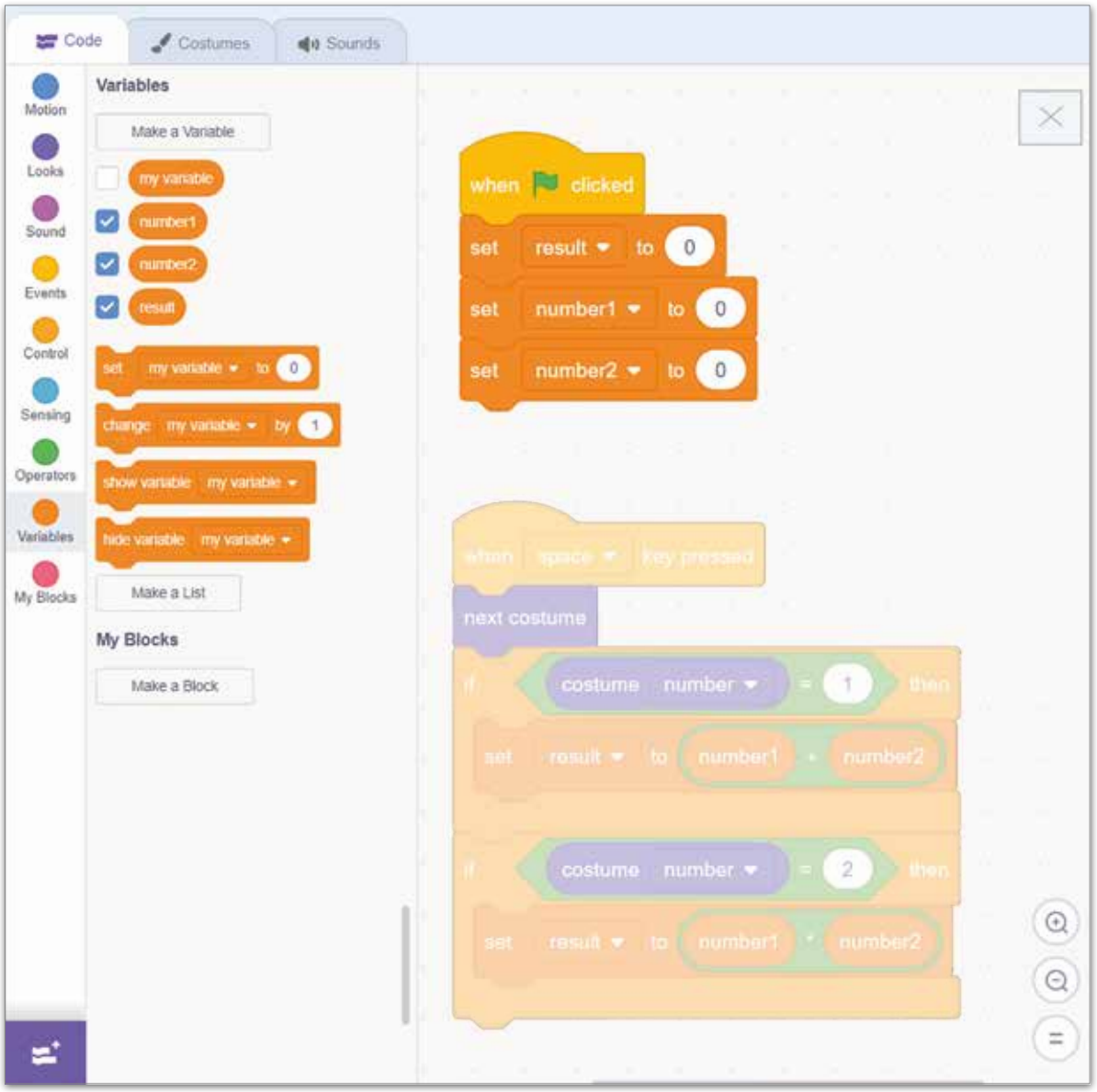
١ عندما يضغط اللاعب على زر المسافة يتغير الكائن (Sprite1) إلى المظهر التالي.

٢ إذا كان المظهر هو (costume1)، فسيتم تعيين المتغير (result) إلى جمع المتغيرين (number1) و (number2).

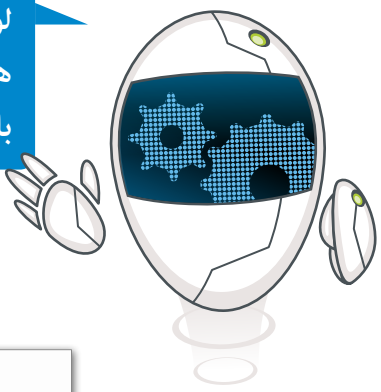
إذا كان المظهر هو (costume2)، فسيتم تعيين المتغير (result) إلى ضرب المتغيرين (number1) و (number2).

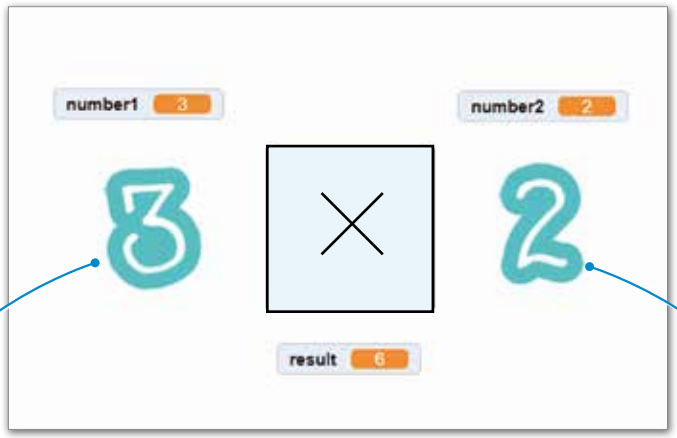


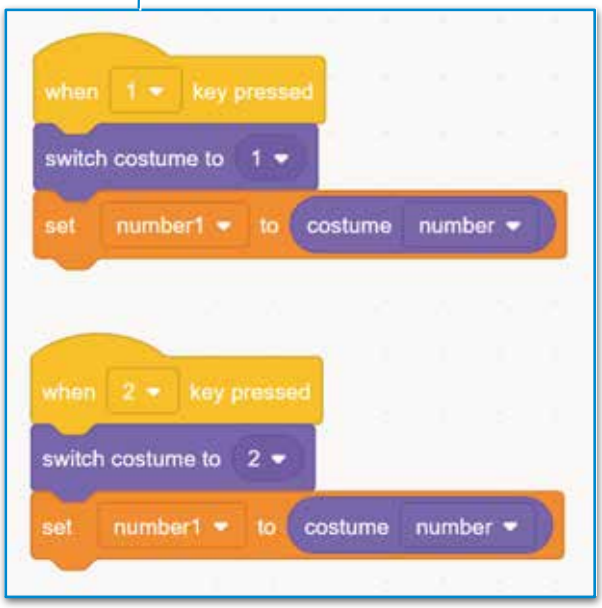
مـن الجيـد ضبـط المتغيرات الثلاثـة إلى 0 عند النقـر على العلم الأخضر، ولكن هذا ليس ضروريًّا لهذا المقطع البرمجي.

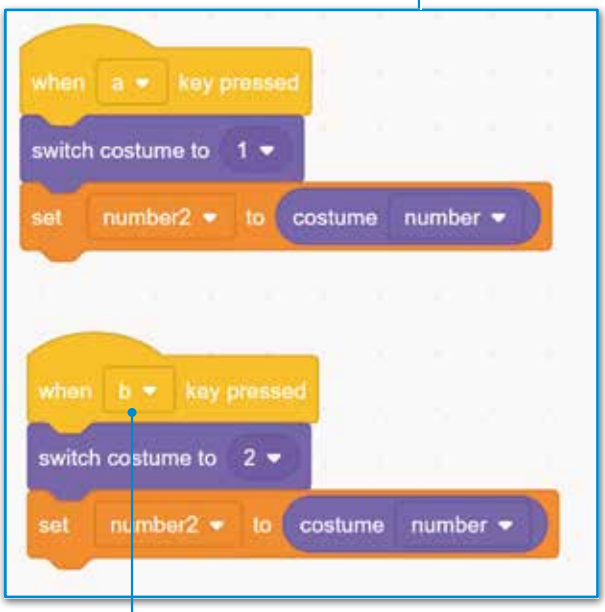


غيّر اللعبة بحيث لا يتم توليد الأرقام عشوائيًا. بدلًا من ذلك، اجعل الكائنين (number1) و (number2) يغيران مظهرهما عند الضغط على مفاتيح لوحة المفاتيح. كل مفتاح سيُغيّر مظهر الكائن إلى الرقم المُطابق، وسيتم حفظ هذا الرقم نفسه في مُتغيّر. انظر إلى المثال أدناه وأكمل باقي المقاطع البرمجية بالطريقة نفسها.







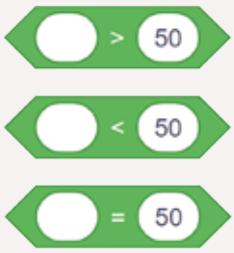
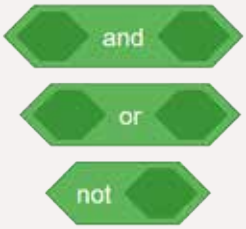


استخدم مفاتيح مختلفة للتحكم في الرقم الثاني، بحيث يتغير كل رقم بشكل مستقل ويتجنب تكرار القيم – على سبيل المثال: استخدم 1 و 2 للرقم الأول، و A و B للرقم الثاني.

لنطبق معًا

تدريب ١

اكتب فئة المُعامل في الجدول الآتي.

فئة المُعامل	اللبنات
.....	
.....	
.....	

تدريب ٢

تحقق من الجمل الآتية لمعرفة ما إذا كانت صحيحة أو خاطئة، ثم صحِّح العبارات الخاطئة.

	صحيحة	خاطئة
١. يمكن كتابة عملية القسمة في البرمجة باستخدام رمز الشرطة المائلة (/).	◯	◯
٢. يمكن كتابة عملية الضرب في البرمجة باستخدام الرمز x.	◯	◯
٣. المُعامل هو رمز يُستخدم لإجراء عملية حسابية مثل الجمع أو الطرح.	◯	◯

تدريب ٣

أنشئ المقطع البرمجي الآتي، واختبره عن طريق كتابة أرقام مختلفة في كل مرة.

اكتب الأرقام التي اخترتها والنتيجة التي حصلت عليها.

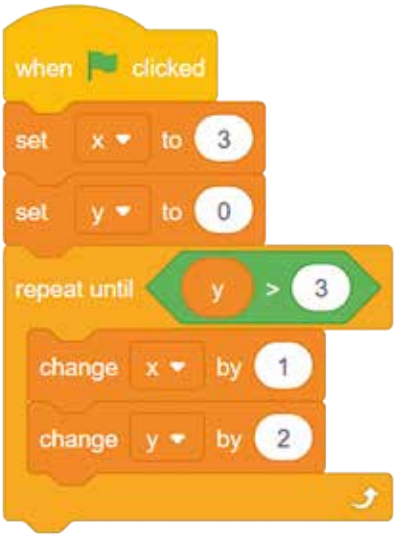
x = _____, y = _____

x = _____, y = _____

x = _____, y = _____

تدريب ٤

تتبّع تغيّر قيم المتغيرين x و y أثناء تنفيذ المقطع البرمجي، ثم اختر القيم النهائية لهما بعد انتهاء تنفيذه.



x = 3, y = 4 ○

x = 5, y = 4 ○

x = 6, y = 5 ○

تدريب ٥

أنشئ مقطعًا برمجيًّا لحساب عمرك.

- سيطلب المقطع البرمجي سنة ميلادك.
- ثم يطلب السنة الحالية.
- سيجري عملية طرح، ثم يعرض الكائن النتيجة.

الدرس ٥-١:

المشروع

١. في هذا المشروع، ستقوم بإنشاء لعبة يتحرك فيها كائن على الشاشة باستخدام مفاتيح الأسهم.

٢. يجب على اللاعب تحريك الكائن ليلامس الكائنات الأربعة الأخرى. عندما يلامس الكائن الرئيس أي كائن آخر:

- ستظهر رسالتان تطلب كل منهما من اللاعب إدخال رقم.
- سيتم تخزين هذين الرقمين في متغيرين.
- سيقوم المقطع البرمجي الذي أنشأته بحساب نتيجة عملية حسابية (مثل الجمع، أو الطرح، أو الضرب، أو القسمة)، حيث إن كل كائن في اللعبة مسؤول عن إحدى هذه العمليات الحسابية.
- سيتم عرض النتيجة على الشاشة.

الفكرة

التخطيط

أهداف اللعبة:

الكائنات الرئيسة:

خلفية اللعبة:

قواعد اللعبة:

عناصر التحكم:

تصميم النموذج الأولي

يمكنك استخدام المساحة أدناه لتصميم نموذج أولي.
ارسم الكائن الذي يتحرك باستخدام الأسهم.

ارسـم أو اكتـب أسماء الكائنـات الأربعـة التي تحتـوي على المقاطع البرمجيـة الخاصة بالعمليات الحسـابية الأربـع. بجانـب كل كائـن، اكتب العملية الحسـابية التي سـيقوم بها (على سـبيل المثال: الجمـع، أو الطرح، أو الضرب، أو القسمة).

التنفيذ

١. ارسم المخطط الانسيابي.

٢. برمج لعبتك باستخدام تطبيق Scratch.

الاختبار

اختبر لعبتك، واكتب هنا إذا قمت بإجراء أي تحسينات.

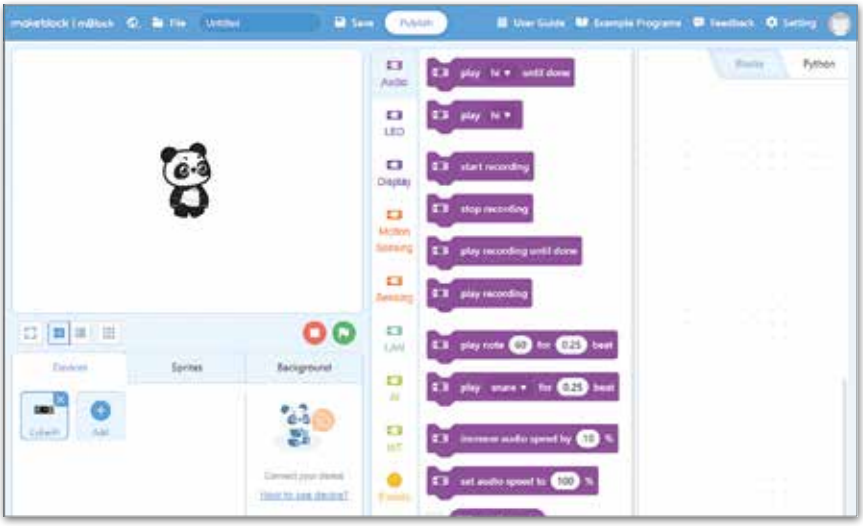
برامج أخرى

هوبسكوتش (Hopscotch)

يمكنك تنزيل تطبيق Hopscotch (هوبسكوتش) على جهاز الآيباد، وإنشاء مقاطع برمجية، ومن ثَمَّ تشغيلها على أجهزة أخرى.

أم بلوك (mBlock)

يُساعد تطبيق mBlock (أم بلوك) الطلبة على توفير تجربة تعليمية محسنة، تمكِّنهم من تطوير مهاراتهم في البرمجة.



في الختام

مخرجات التعلُّم

المخرج التعليمي	مستوى الإتقان	
	أتقن	لم أتقن
١. تعريف نظام الإحداثيات واستخدامه لتحريك الكائنات.		
٢. استخدام المتغيرات لتخزين البيانات وعرضها وتحديثها.		
٣. تصميم وبرمجة لعبة تفاعلية تستجيب لإجراءات المستخدم.		
٤. إنشاء أكثر من مظهر للكائن.		
٥. التمييز بين المعاملات المنطقية.		
٦. استخدام اللبنات الشرطية وتطبيق بث الرسائل لإنشاء تفاعلات بين الكائنات والأحداث.		
٧. استخدام لبنات المُراسِل.		
٨. استخدام المعاملات الرياضية لإجراء العمليات الحسابية داخل المقطع البرمجي.		
٩. برمجة آلة حاسبة تُجري العمليات الحسابية بناءً على مدخلات المستخدم.		

المصطلحات

نظام الإحداثيات / Coordinate System

إحداثيات (س) و (ص) / X and Y coordinates

المتغير / Variable

البث / Broadcast

مُعامل شرطي / Conditional operator

مُعامل منطقي / logical operator

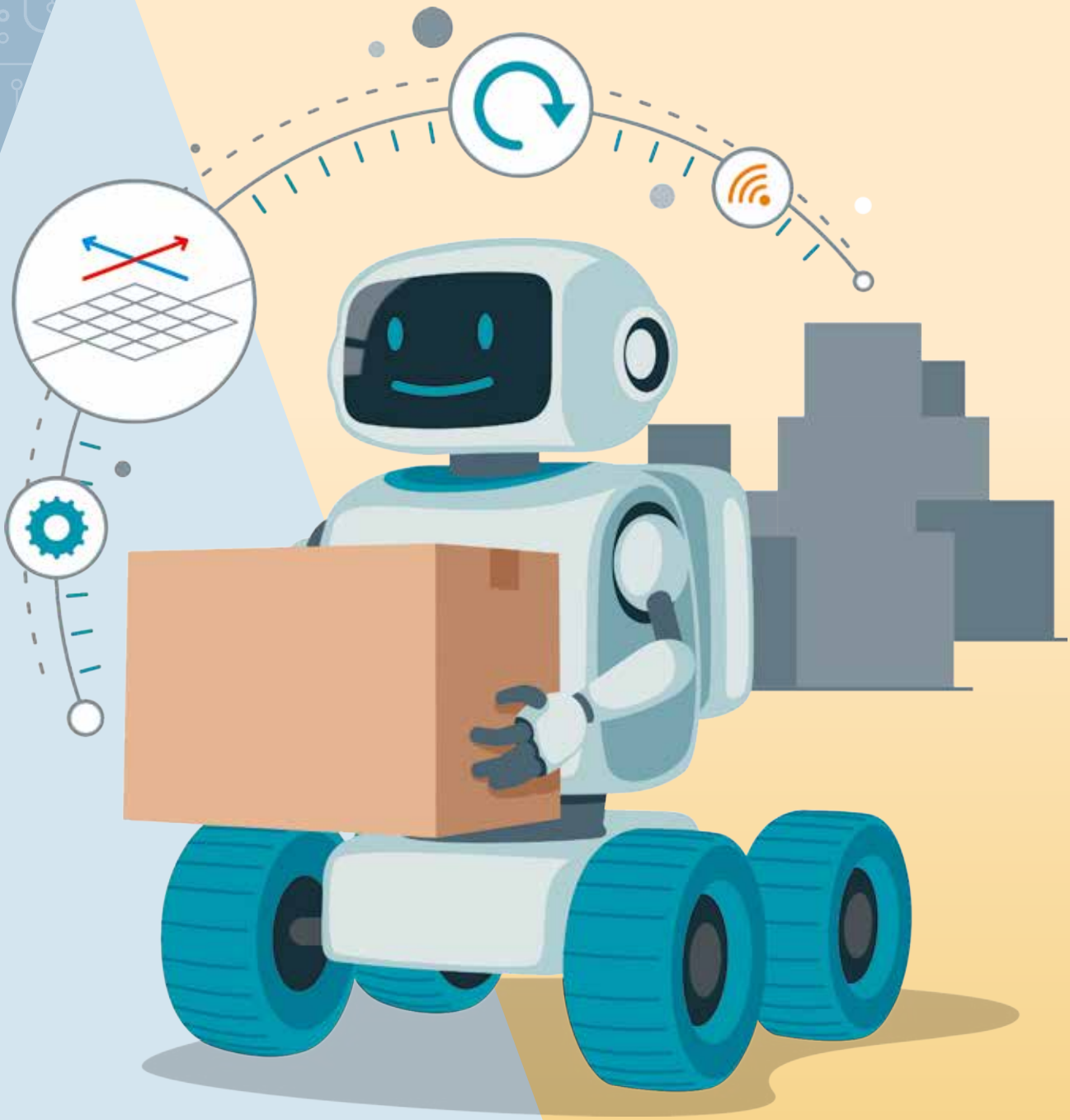
العمليات الحسابية / Mathematical operations

مظهر / Costume

لبنات المُراسِل / Reporter blocks

الوحدة ٢:

الروبوتات



المقدمة

ستتعرَّف في هذه الوحدة إلى الروبوتات وكيف نستخدمها في حياتنا اليومية، وستتعلَّم أيضًا استخدام منصة VEXcode VR للتحكّم في الروبوت داخل بيئة افتراضية، كما ستتعرف إلى لبنات البرمجة وستستخدمها لجعل الروبوت يتحرّك ويدور ويرسم أشكالًا مختلفة، وستستكشف مستشعرات مثل مستشعر الاصطدام ومستشعر الانعطاف وستسخدمها لزيادة دقة حركة الروبوت وتمكينه من التفاعل مع البيئة المحيطة، كما ستتعلم كيفية استخدام المتغيرات لتخزين معلومات مثل الوقت والنقاط، واستخدام الجمل الشرطية لاتخاذ القرارات وجعل الروبوت يتصرّف بناءً على ما يحدث في بيئته المحيطة.

أهداف التعلُّم

ستتعلَّم في هذه الوحدة:

- مفهوم الروبوتات وأنواعها.
- استخدامات الروبوتات في حياتنا اليومية.
- مفهوم الروبوتات الافتراضية.
- مكونات وأدوات بيئة تطبيق VEXcode VR.
- التمييز بين طرق العرض المختلفة لساحة اللعب.
- لبنات التكرار في برمجة الروبوت.
- التمييز بين مستشعر الاصطدام ومستشعر الانعطاف في الروبوت الافتراضي.
- الجمل الشرطية في برمجة الروبوت.
- المتغيرات في برمجة الروبوت.

الأدوات

- VEXcode VR

المهارات

ستتمكن بعد هذه الوحدة من:

- تغيير ساحة اللعب في تطبيق VEXcode VR.
- برمجة الروبوت ليتحرك على شكل مربع.
- معاينة حركة الروبوت باستخدام طرق عرض مختلفة.
- التحكم في سرعة الروبوت.
- استخدام لبنات التكرار لتِكرار الإجراءات في برمجة الروبوت.
- برمجة الروبوت ليرسم أشكالًا مختلفة.
- توجيه الروبوت باستخدام الإحداثيات.
- تحريك الروبوت باستخدام المستشعرات.
- استخدام الجمل الشرطية للتحكم في حركة الروبوت.
- استخدام المتغيرات للتحكم في حركة الروبوت.

الدرس ١-٢:

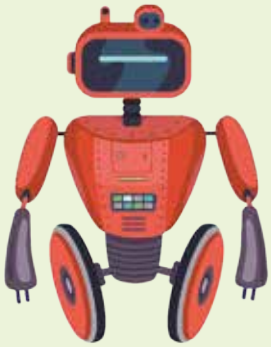
الروبوتات في حياتنا

الروبوت هو آلة تجمع البيانات من حولها، ثم تعالج تلك البيانات وتستخدمها لاتخاذ القرار المناسب، وبعد ذلك تنفّذ الأوامر للقيام بمهمة معينة. وتتضمّن هذه العملية ثلاث خطوات رئيسة: الاستشعار، ثم المعالجة، ثم التنفيذ.

أنواع الروبوتات

هناك نوعان من الروبوتات: الروبوتات الثابتة (Fixed Robots)، والروبوتات المتنقلة (Mobile Robots).

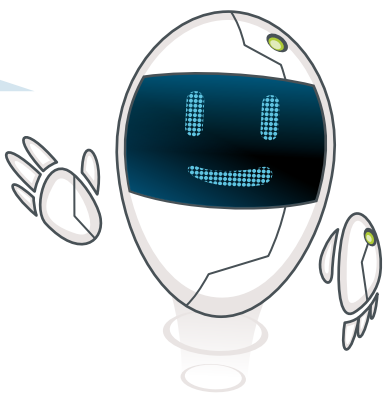
الروبوتات الثابتة: روبوتات يتم تثبيتها في موقع واحد لأداء مهام محددة، وتستخدم بشكل واسع في المصانع؛ لأنها أسرع وأكثر قوة من الروبوتات المتنقلة، فعلى سبيل المثال: يمكن لهذا النوع من الروبوتات إنتاج سيارة بوقت أسرع مما يستغرقه البشر، كما يمكن لهذه الروبوتات رفع الأوزان الثقيلة جدًّا.



الروبوتات المتنقلة: روبوتات تتنقل في الأرض أو البحر أو الجو، وهي تعتمد على المحركات في حركتها. يمكن للبشر التحكم فيها لاسلكيًّا أو يمكنها التنقل بمفردها من خلال أجهزة الاستشعار الخاصة بها.

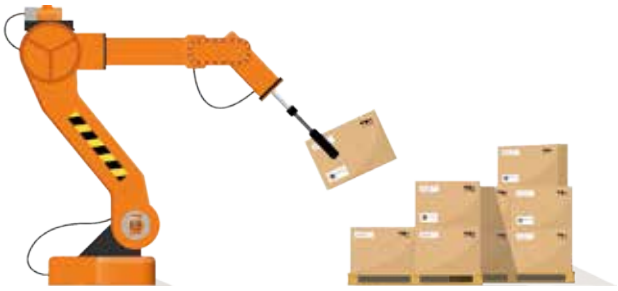
الروبوتات في الحياة اليومية

الروبوتات الثابتة والمتنقلة موجودة في كل مكان حولنا ويمكن العثور عليها في العديد من المنازل، حيث يمكن لبعضها طهي الطعام أو تنظيف الأرضية أو نقل الأشياء من مكان إلى آخر، كما يمكن العثور على الروبوتات في المطاعم والفنادق والمستشفيات. هذه الآلات تجعل الحياة اليومية أسهل وتمنح الناس مزيدًا من الوقت للقيام بأنشطة أخرى.



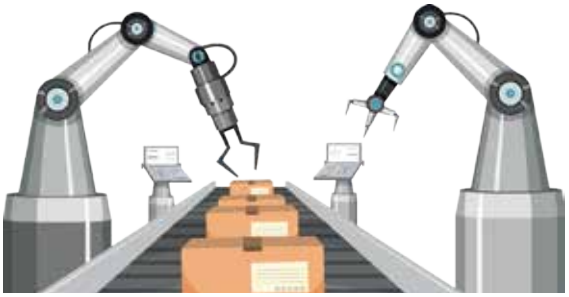
استخدامات الروبوتات

تُعدُّ الروبوتات أدوات مهمة في حياتنا اليومية، فهي تساعد في إنجاز الأعمال بدقة وسرعة. ولهذا تُستخدم اليوم في العديد من المجالات، مثل الصناعة، والتعليم، والزراعة، والطب، لحل بعض مشكلات العالم الحقيقي وتسهيل حياة الناس.



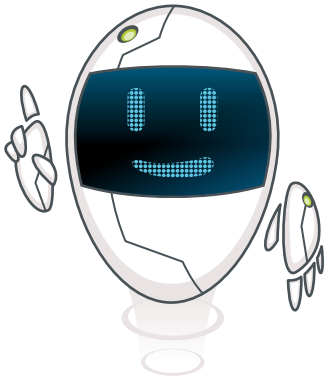
التعليم: تُستخدم الروبوتات في التعليم لتنفيذ تجارب تعليمية تفاعلية تُنمّي مهارات حل المشكلات والبرمجة والتفكير النقدي.

الطب: يستخدم الأطباء الروبوتات في المستشفيات لإجراء بعض العمليات الجراحية التي تحتاج إلى دقة وثبات كبيرين. فالأذرع الروبوتية تتحرك بثبات أكبر من يد الإنسان، مما يساعد الأطباء على أداء العمليات بدقة وأمان.



الصناعة: تُستخدم الروبوتات في المصانع لإنتاج البضائع بسرعة عالية ودقة كبيرة، مما يجعل عملية التصنيع أكثر كفاءة، كما تساعد الروبوتات في نقل الأشياء الثقيلة داخل المخازن، أو ترتيب البضائع في المتاجر بشكل أسرع.

الزراعة: تدعم الروبوتات الزراعة من خلال تحسين الإنتاجية وتنفيذ مجموعة من المهام مثل مراقبة المحاصيل، والري الدقيق، وجمع البيانات، كما تُستخدم في الحصاد الآلي لتقليل الجهد وزيادة الكفاءة.



في عام ٢٠٢٥م، نجح المستشفى السُّلطاني في تنفيذ أول عمليّة جراحيّة باستخدام الروبوت الجراحي الذي يعتمد على أذرع روبوتية ورؤية ثلاثية الأبعاد، ويُعدّ هذا الإنجاز خطوة مهمة تُظهر تقدّم القطاع الصحي في سلطنة عُمان.

مزايا وعيوب استخدام الروبوتات

يمكن أن يؤثر استخدام الروبوتات في حياتنا اليومية بطريقة إيجابية، وبالرغم من ذلك فإن للروبوتات سلبيات تحد من استخدامها.

المزايا	العيوب
أداء مهام صعبة أو خطرة على الإنسان.	بعض الروبوتات لا يمكنها اتخاذ القرارات بأنفسها، فهي مُبرمجةٌ للقيام بأشياء مُحددة.
دقيقة للغاية.	هناك بعض المهن التي تتطلب تفكيرًا إبداعيًّا ونقديًّا لا تستطيع الروبوتات القيام بها.
تعمل لفترات طويلة.	تحتاج إلى طاقة لكي تعمل.
يمكن إصلاحها.	يحتاج إصلاحها في بعض الحالات إلى الكثير من الوقت.
تعمل بشكل سريع.	تكلفتها عالية.

لنطبق معًا

تدريب ١

تحقق من العبارات الآتية لمعرفة ما إذا كانت صحيحة أو خاطئة، ثم صحِّح العبارات الخاطئة.

	صحيحة	خاطئة
١. تُستخدم الروبوتات في المصانع لأداء المهام المتكررة بسرعة ودقة تفوق سرعة ودقة البشر.	○	○
٢. الروبوتات المتنقلة تبقى في مكان واحد وتُستخدم فقط لأداء مهمة محددة.	○	○
٣. يُمكن التحكم في الروبوتات عن بُعد باستخدام اتصال لاسلكي.	○	○
٤. يستخدم الأطباء الروبوتات لتحقيق نتائج أفضل في العمليات الجراحية.	○	○

تدريب ٢

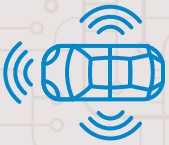
طابق أنواع الروبوتات بأمثلة حولها.

١. الروبوتات الثابتة •

٢. الروبوتات المتنقلة •

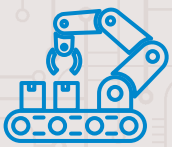
• أ. الطائرات المُسيرة.

• ب. السيارات ذاتية القيادة.



• جـ. روبوتات التنظيف.

• د. الأذرع الروبوتية في المصانع.



تدريب ٣

تؤثر الروبوتات في حياتنا ووظائفنا. اقرأ العبارات الآتية، وحدّد ما إذا كانت تنطبق على البشر، أو الروبوتات، أو كليهما.

	البشر	الروبوتات
١. الحاجة إلى طاقة للعمل.	○	○
٢. الحاجة إلى النوم.	○	○
٣. التعب خلال العمل.	○	○
٤. استبدال قطع الغيار.	○	○
٥. اتخاذ القرارات بنفسها.	○	○

تدريب ٤

اختر الإجابة الصحيحة.

ما المهمة التي لا تستطيع الروبوتات القيام بها بكفاءة أعلى من البشر؟	
١. الكتابة الإبداعية.	○
٢. رفع الأوزان الثقيلة.	○
٣. تنظيم البيانات بسرعة.	○
٤. حساب الأرقام بدقة.	○

لنطبق معًا

أيٌّ مما يلي يُعدّ تأثيرًا سلبيًّا للروبوتات؟

١. تعمل بسرعة كبيرة.

٢. دقيقة جدًّا.

٣. لا تتعب أبدًا.

٤. تكلفتها عالية.

أيٌّ من هذه العبارات لا يصف الروبوتات بشكل صحيح؟

١. يمكن إصلاحها.

٢. يمكنها اتخاذ قرارات مستقلة.

٣. تحتاج إلى طاقة للعمل.

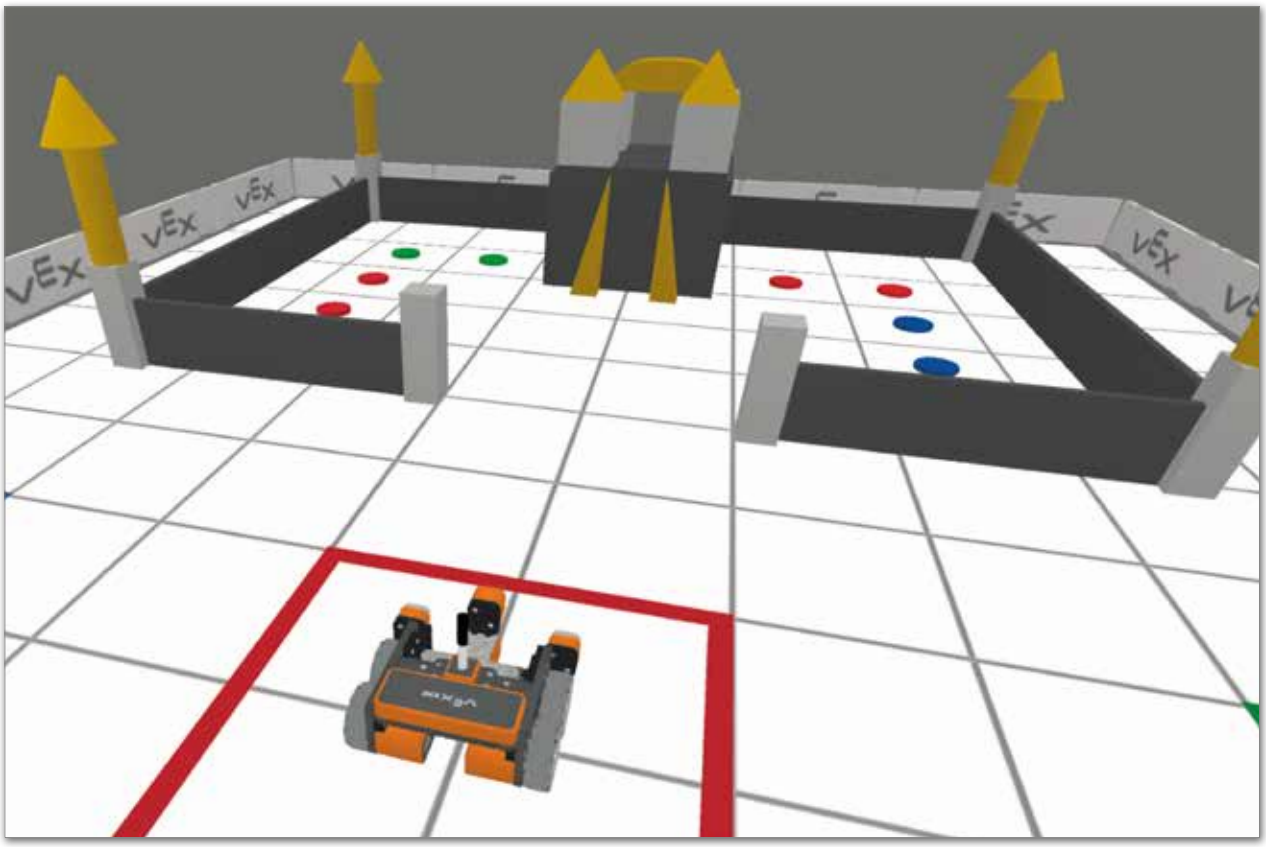
٤. قد تتعطل وتتوقف عن العمل لأيام.

الدرس ٢-٢:

الروبوتات الافتراضية

يُعدّ امتلاك روبوت حقيقي أمرًا مفيدًا لتعلم البناء والبرمجة، لكن ذلك قد لا يكون ممكنًا دائمًا؛ لذلك نستخدم الروبوتات الافتراضية لمحاكاة عمل الروبوتات الحقيقية. تتضمن الروبوتات الافتراضية استخدام روبوتات مُحاكاة لتطوير واختبار المقاطع البرمجية بسرعة وأمان، مع تقليل خطر تلف المعدات أو فقدانها، كما تتيح الوصول إلى مجموعة واسعة من المكوّنات وأنواع الروبوتات التي قد لا تكون متاحة فعليًّا، مما يتيح تنفيذ مشاريع أكثر إبداعًا وتطورًا.

توجد العديد من التطبيقات التي تتيح لنا برمجة الروبوتات في بيئة افتراضية، أحدها تطبيق VEXcode VR الذي سنتعرف عليه في هذا الدرس.



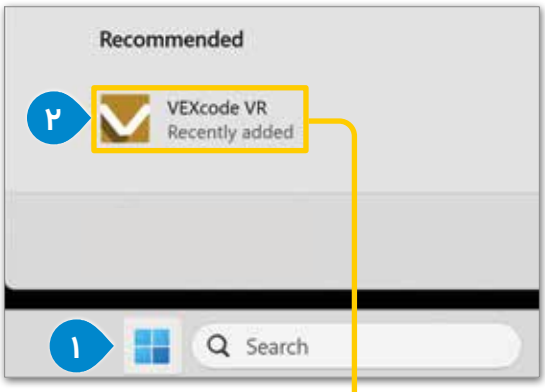
تطبيق VEXcode VR

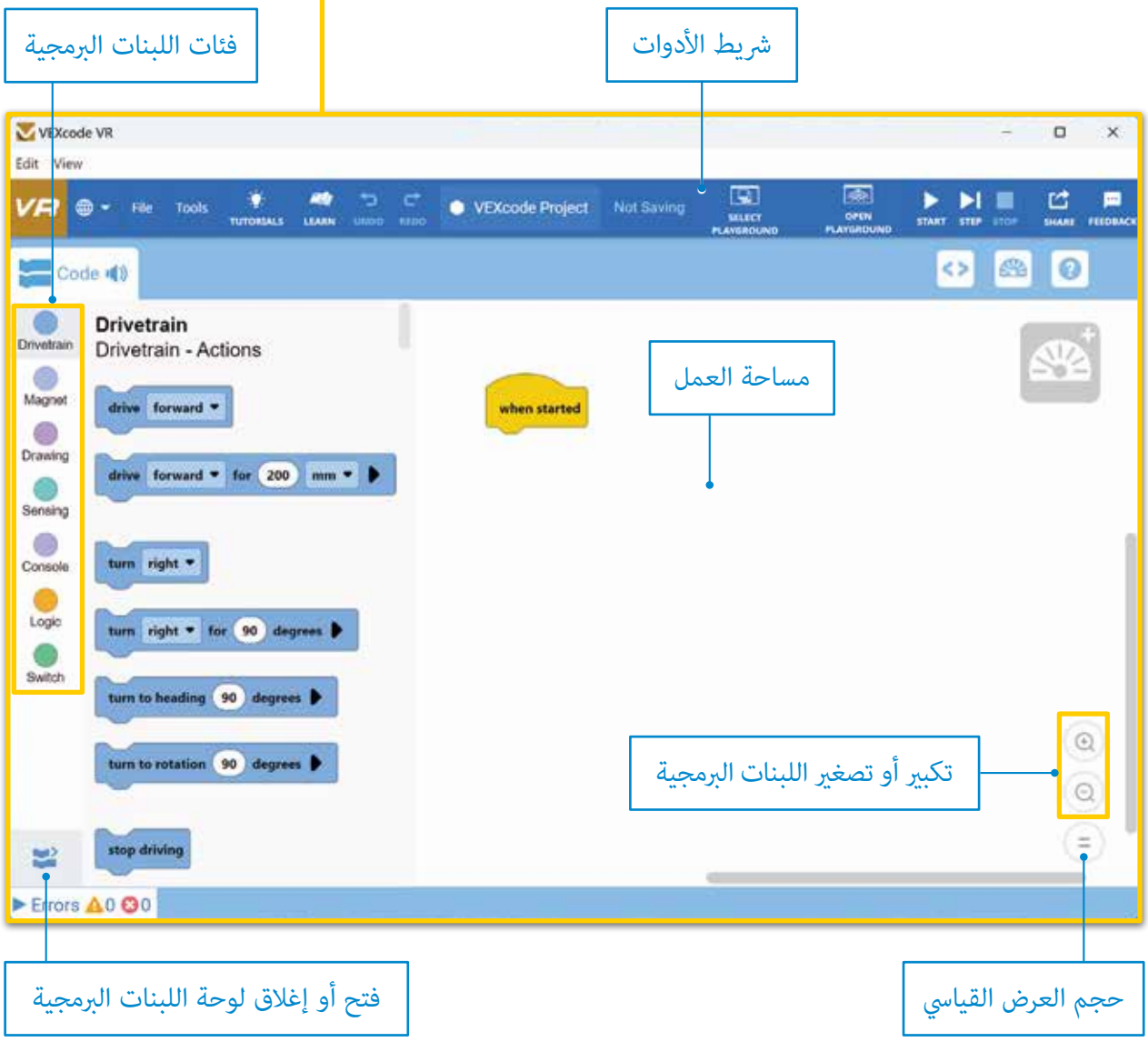
تطبيـق قائـم على اسـتخدام اللَّبِنـات البرمجيـة، ومشـابه لتطبيـق Scratch، تتميـز واجهـة بيئـة البرمجـة بالبسـاطة وسـهولة الاسـتخدام، حيث يمكنك إنشـاء المقاطع البرمجية دون كتابة تعليمات برمجية معقدة، فكل ما عليك فعله هو سحب اللبنات البرمجية إلى مساحة العمل وتوصيلها معًا.

لفتح تطبيق VEXcode VR:

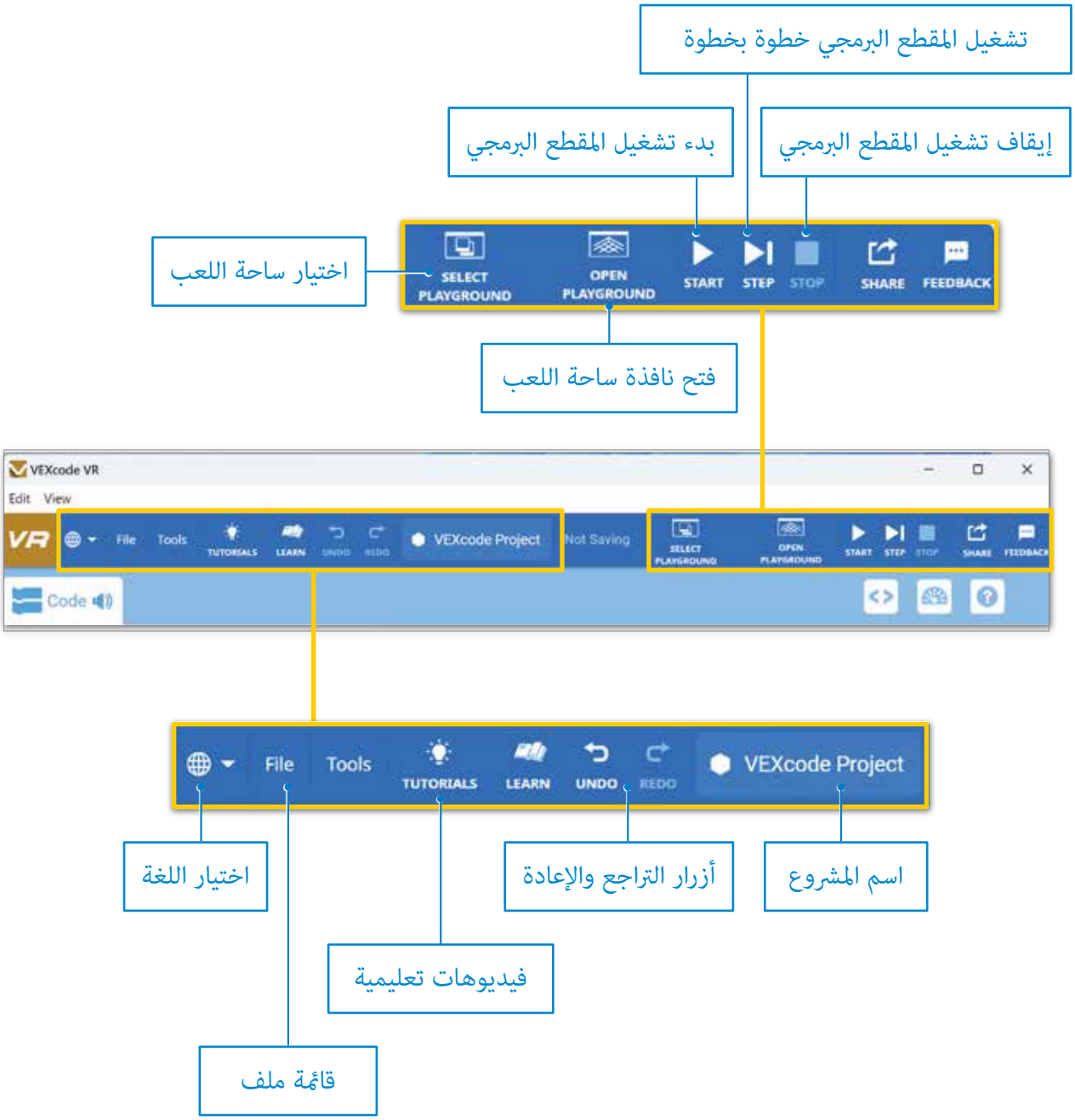
> ١ من شريط المهام، انقر على زر **Start** (البدء).

> ٢ من قائمة **Start** (البدأ)، انقر **VEXcode VR**.





يحتوي شريط الأدوات على عدة خيارات كالآتي:



ساحة اللعب

بيئة تفاعلية افتراضية خاصة بالروبوت الافتراضي، تُمكِّنك من تنفيذ مقاطعك البرمجية بسيناريوهات مختلفة.

فئات اللَّبِنات البرمجية

توجد مجموعة واسعة من فئات اللبنات البرمجية التي يمكن استخدامها لإنشاء مقطع برمجي، وتتميز كل فئة بلون محدد، وتُجمع اللبنات معًا في فئات محددة طبقًا لنوعها واستخدامها.

فئات اللبنات البرمجية ووظائفها

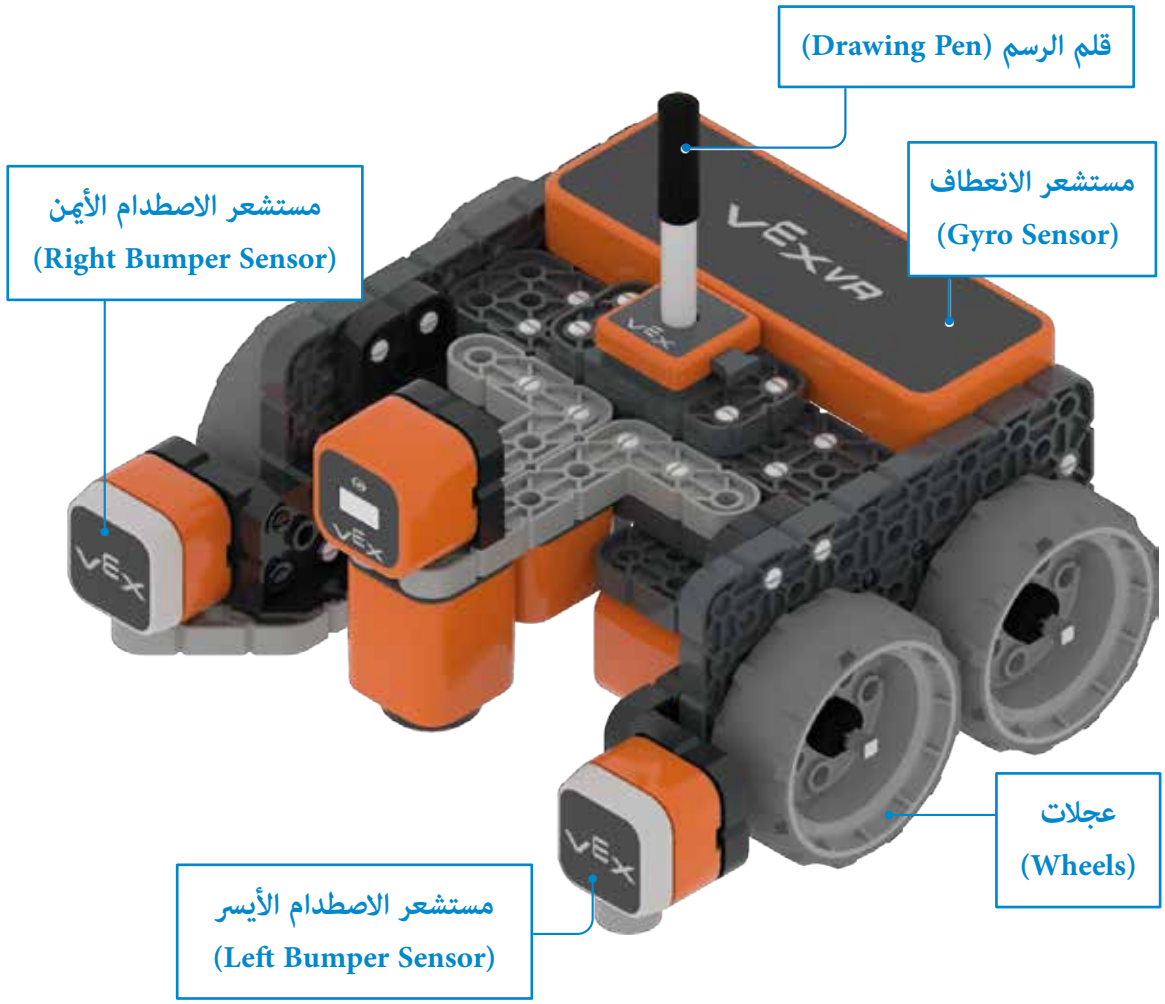
فئة اللبنات	الوظيفة
Drivertrain (نظام نقل الحركة)	تتحكم في حركة الروبوت في ساحة اللعب.
Magnet (مغناطيس)	تستخدم لالتقاط الأقراص في ساحات لعب معينة.
Drawing (رسم)	تستخدم للتحكم في قلم الروبوت.
Sensing (الاستشعار)	تستخدم لقراءة قيم مستشعرات الروبوت.
Logic (المنطق)	تستخدم للتحكم بعملية اتخاذ القرار لجعل الروبوت يتفاعل بذكاء.

معلومة تقنية

تُربط اللبنات البرمجية ببعضها البعض ويتم تنفيذها بواسطة الروبوت وفقًا لترتيبها. يُعرف هذا المفهوم باسم "تسلسل العمليات"، فعند تشغيل المقطع البرمجي يتم تنفيذ اللبنات البرمجية المتصلة ببعضها فقط.

روبوت VEXcode الافتراضي

ستستخدم روبوتًا افتراضيًّا تم إنشاؤه سابقًا مجهزًا بعجلات للحركة وبه عدَّة مستشعرات مدمجة تُمكنه من التفاعل مع بيئته، ويحتوي أيضًا على قلم يُمكّنه من رسم خطوط أو أشكال متنوعة في ساحات اللعب المختلفة.



استكشف بيئة برمجة تطبيق VEXcode VR، وتعرّف على أهم ميزاتها وتصميمها، وأدواتها، ثم قارنها ببيئة تطبيق Scratch التي استخدمتها سابقًا.

- ما أوجه التشابه بين التطبيقين في طريقة استخدام اللبنات أو تشغيل المقاطع البرمجية؟
- ما الاختلافات التي تلاحظها في الواجهة أو الأدوات المتاحة في كل منهما؟

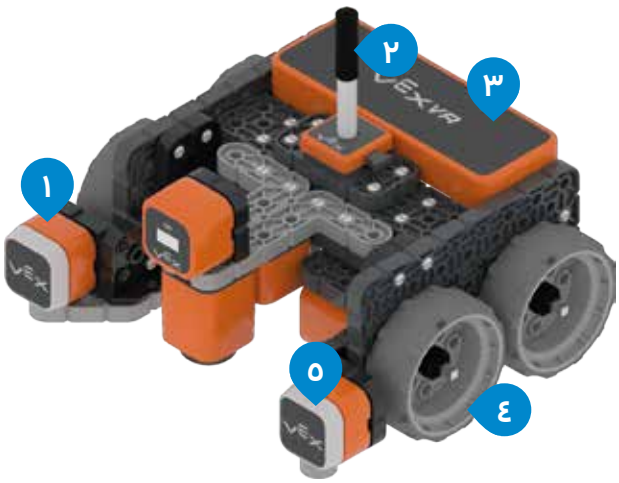
لنطبق معًا

تدريب ١

ما فوائد استخدام الروبوتات الافتراضية؟

تدريب ٢

سمِّ الأجزاء المشار إليها بالأرقام لروبوت VEXcode VR.



١ .....

٢ .....

٣ .....

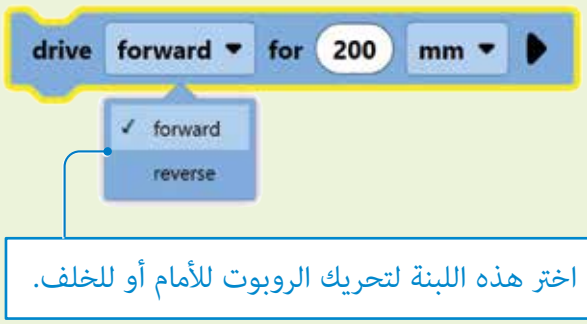
٤ .....

٥ .....

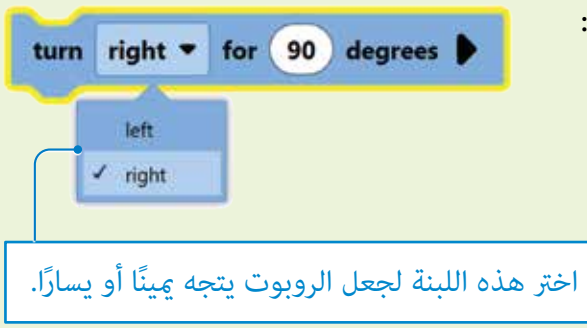
الدرس ٣-٢:

حركة الروبوت

لتتحكم في حركة روبوتك الافتراضي فإنك بحاجة إلى استخدام لبنات من فئة **Drivetrain (نظام نقل الحركة)** التي تشبه فئة لبنات **Motion (الحركة)** التي استخدمتها سابقًا في Scratch.



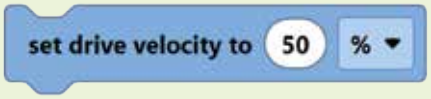
لبنة **drive ( ) for ( ) (محرك ( ) لـ ( ) ( ))**:
تُحرك الروبوت مسافة محددة.



لبنة **turn ( ) for ( ) degrees (أدر ( ) لمدة ( ) درجة)**:
تُنفذ انعطاف الروبوت بقيمة محددة من الدرجات.

لبنة **stop driving (توقف عن القيادة)**: تُوقف حركة الروبوت.

لبنة( ) ( ) **set drive velocity to** (ضبط سرعة القيادة إلى ( ) ( )):
تضبط سرعة حركة الروبوت.



لبنة ( ) ( ) **set turn velocity to** (ضبط سرعة الدوران إلى ( ) ( )):
تضبط سرعة انعطاف الروبوت.

معلومة تقنية

تقبل اللبنتان الخاصتان بسرعة الروبوت قيمًا تتراوح بين 0% إلى 100%.

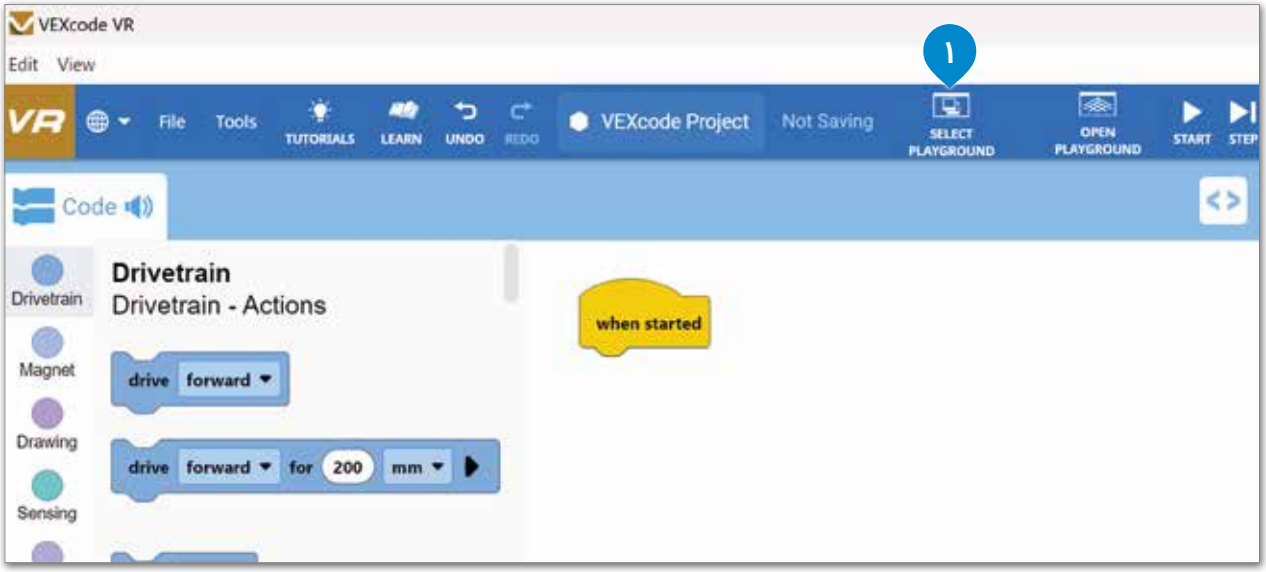
اختيار ساحة اللعب

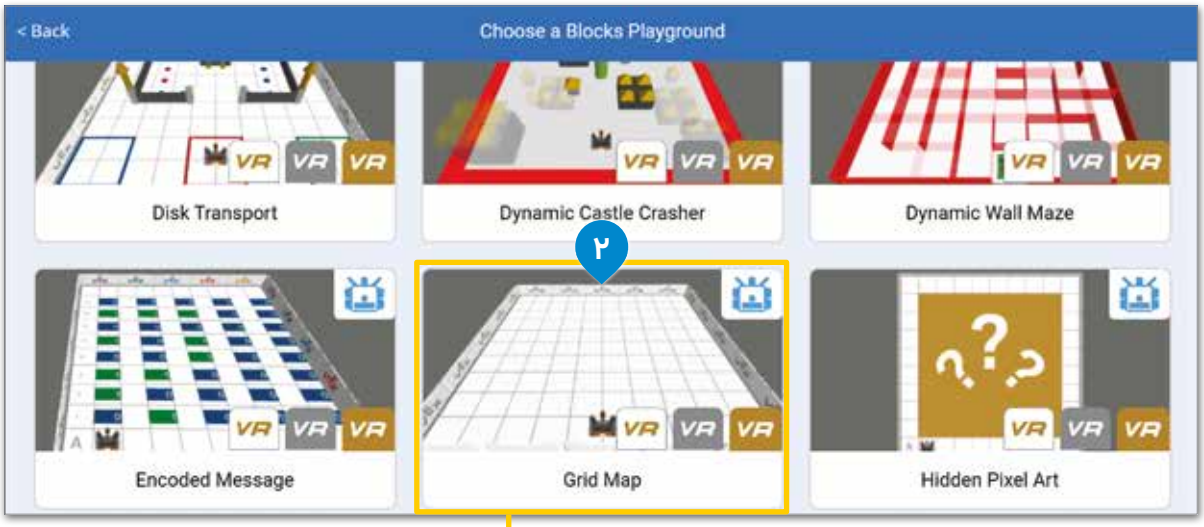
يوفر تطبيق VEXcode VR مجموعة من ساحات اللعب، ستستخدم ساحة لعب Grid Map (خريطة الشبكة) لينفذ الروبوت المقطع البرمجي، حيث توفر هذه الساحة تصميمًا شبكيًّا بسيطًا يساعدك على قياس حركة الروبوت والتحكم فيها بدقة، وهي مناسبة لتجربة أوامر الحركة، وقياس المسافة، والانعطاف.

لتغيير ساحة اللعب:

١ من شريط الأدوات، انقر **SELECT PLAYGROUND** (حدِّد الملعب).

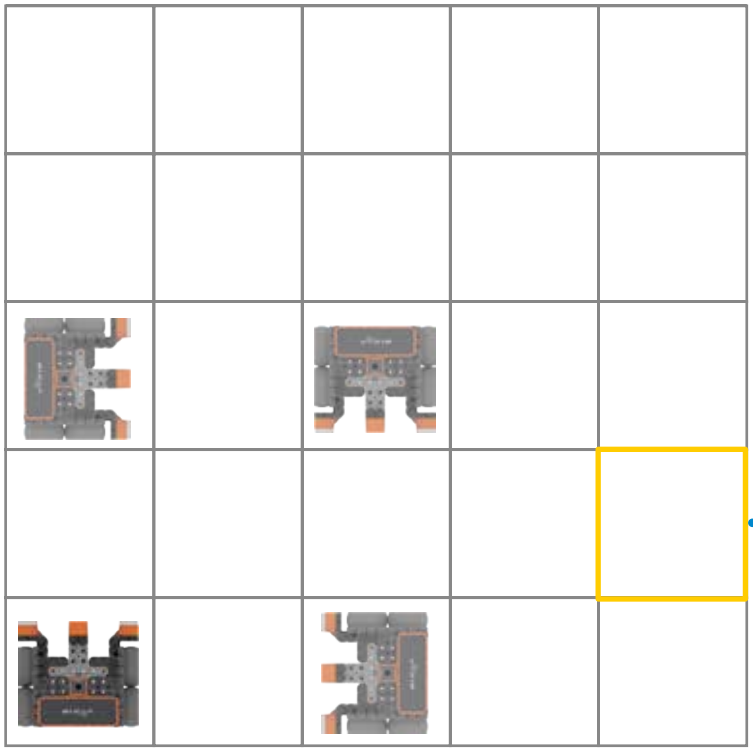
٢ من نافذة **Block Playground** (ملعب الكتل)، انقر **Grid Map** (خريطة الشبكة).





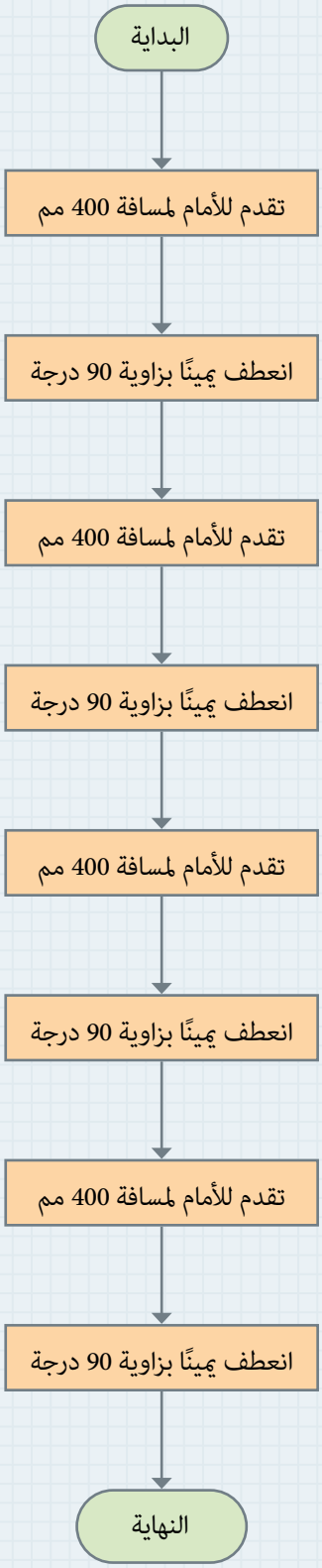
تحريك الروبوت

لتحريك الروبوت من نقطة البداية بحيث يسير في مسار على شكل مربع بحجم (3×3)، ستستخدم لبنات من فئة **Drivetrain (نظام نقل الحركة)**.



تذكَّر أن طول كل ضلع في مربعات ساحة اللعب **Grid Map** (**خريطة الشبكة**) هو 200 مليمتر.

المخطط الانسيابي



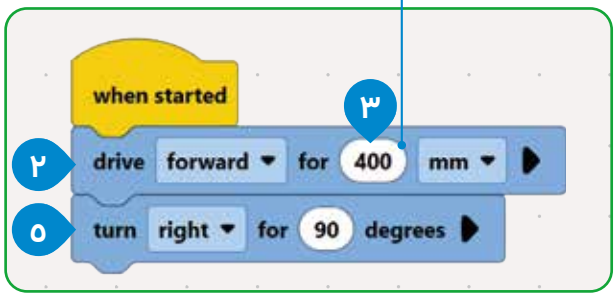
لنبدأ أولًا بتحريك الروبوت وتشكيل الضلع الأول من المربع، ثم الانعطاف لتكوين الزاوية الأولى من الشكل.

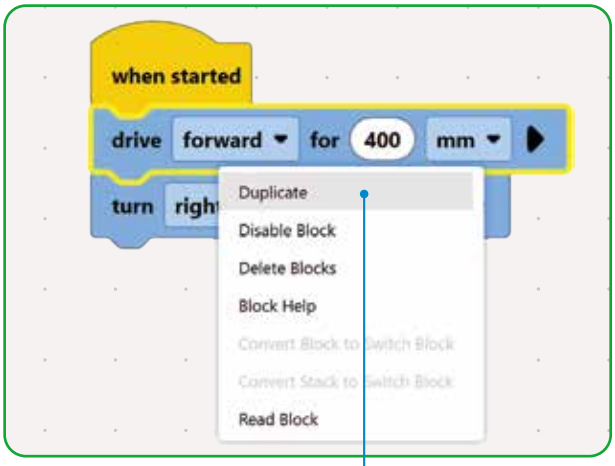
لتحريك الروبوت:

> ١ من فئة لبنات **Drivetrain** (نظام نقل الحركة)، ٢ اسحب وأفلت لبنة( ) ( ) **for** ( ) **drive** (**محرك** ( ) **لـ** ( ) ( )) داخل لبنة **when started** (**عندما بدأت**)، ٣ واضبط المسافة لـ (**400**).

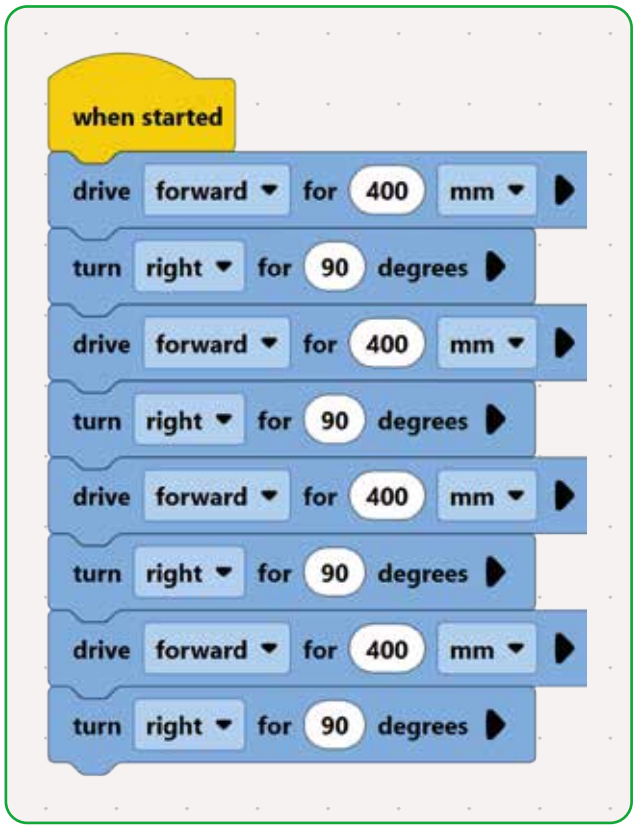
> ٤ من فئة لبنات **Drivetrain** (نظام نقل الحركة)، ٥ اسحب وأفلت لبنة **degrees** ( ) **for** ( ) **turn** ( ) (**أدر** ( ) **لمدة** ( ) **درجة**) داخل لبنة ( ) ( ) **for** ( ) **drive** (**محرك** ( ) **لـ** ( ) ( )).

سيتحرك الروبوت في هذا المثال مربعين للأمام، لذا ستكون المسافة الإجمالية التي سيقطعها الروبوت 400 مليمتر.



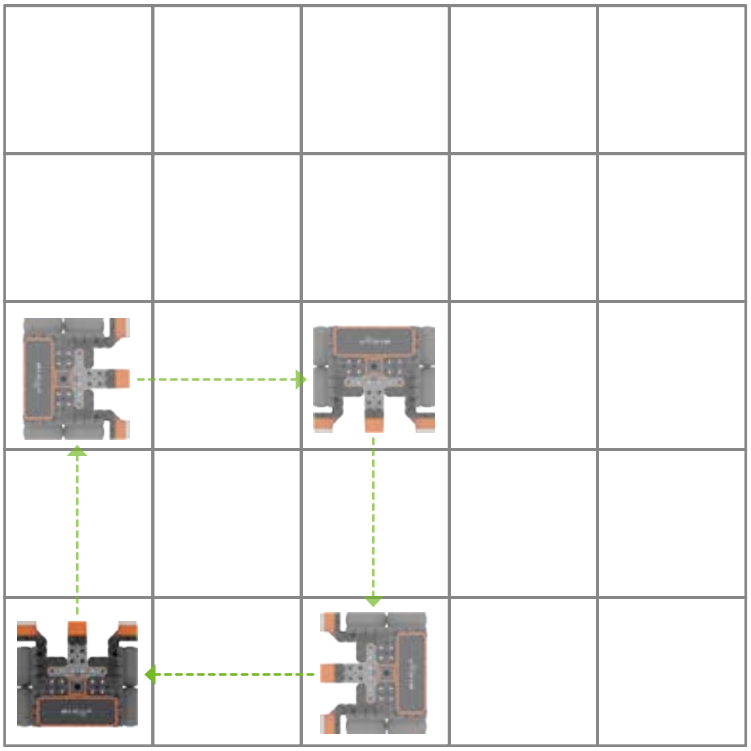


لنسخ اللبنات، انقر بزر الفأرة الأيمن على اللبنة العلوية من التسلسل الذي تريد نسخه، ثم اختر **Duplicate** (**ينسخ**)، وسيتم نسخ اللبنة المختارة وجميع اللبنات المرتبطة بها أو الموجودة بداخلها.



لإكمال المربع، كرر عملية النسخ للأضلاع والزوايا الثلاث المتبقية، ثم انقر على زر Start (البدء) لاختبار المقطع البرمجي.

شغّل مقطعك البرمجي وتأكد من أن الروبوت يتحرك في مسار على شكل مربع.



طرق العرض المختلفة

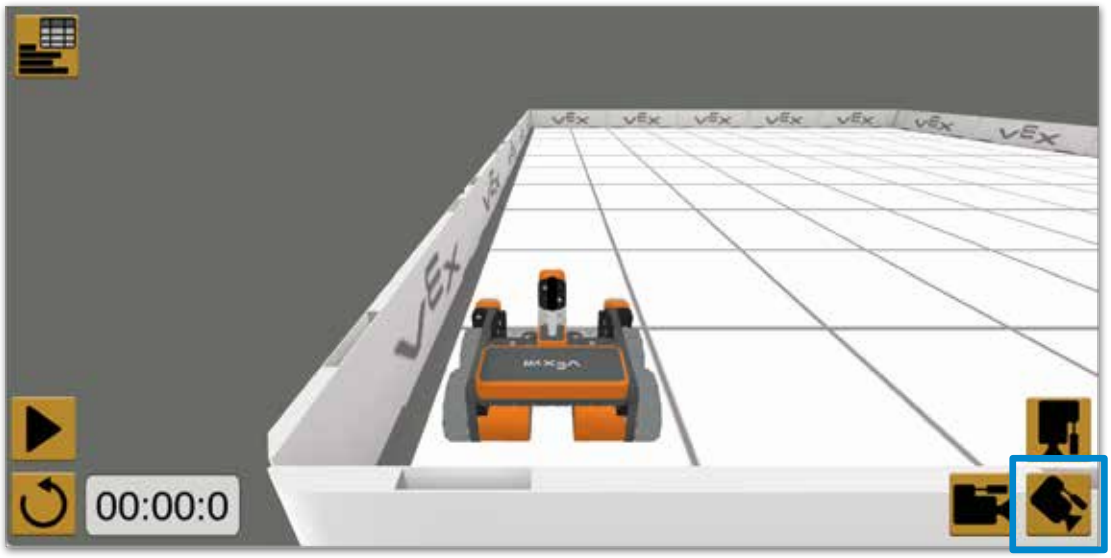
يمكنك الاستفادة من طرق عرض الكاميرا المختلفة المتاحة عند إنشاء المقاطع البرمجية في تطبيق VEXcode VR واختبارها في ساحات اللعب، حيث يُمكنك معاينتها بصورة أفضل.

الكاميرا العلوية (Top Camera)

إن وضع الكاميرا العلوية هو الوضع الافتراضي للكاميرا عند فتح نافذة ساحة اللعب، حيث تعرض ساحة اللعب من الأعلى بشكل كامل.

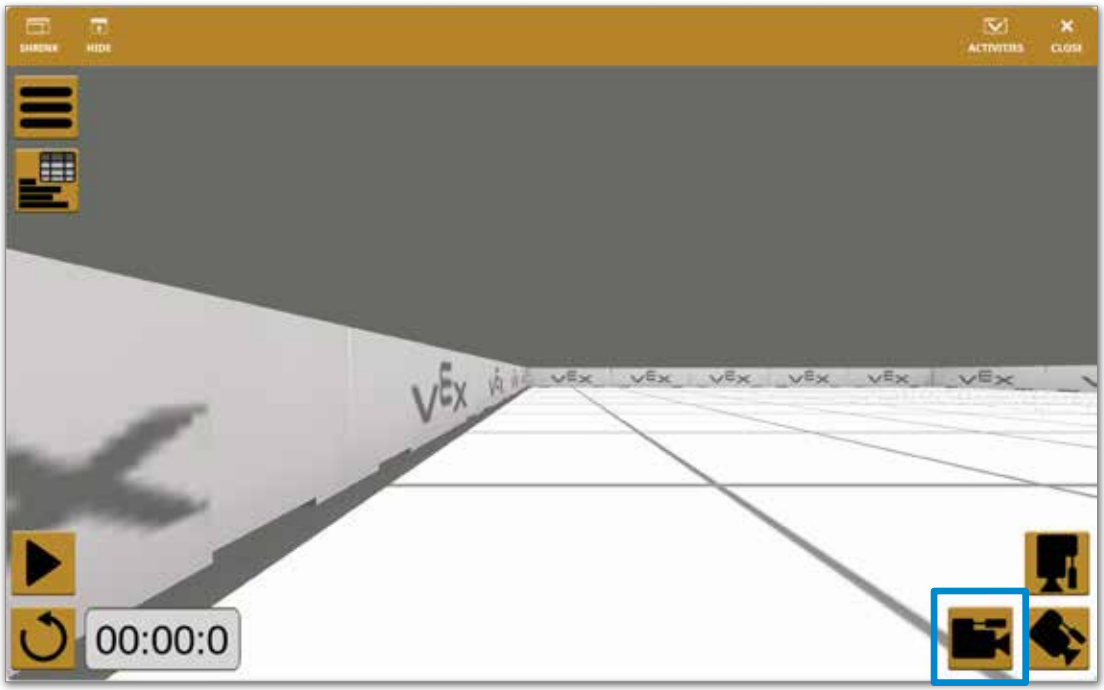
كاميرا التتبع (Chase Camera)

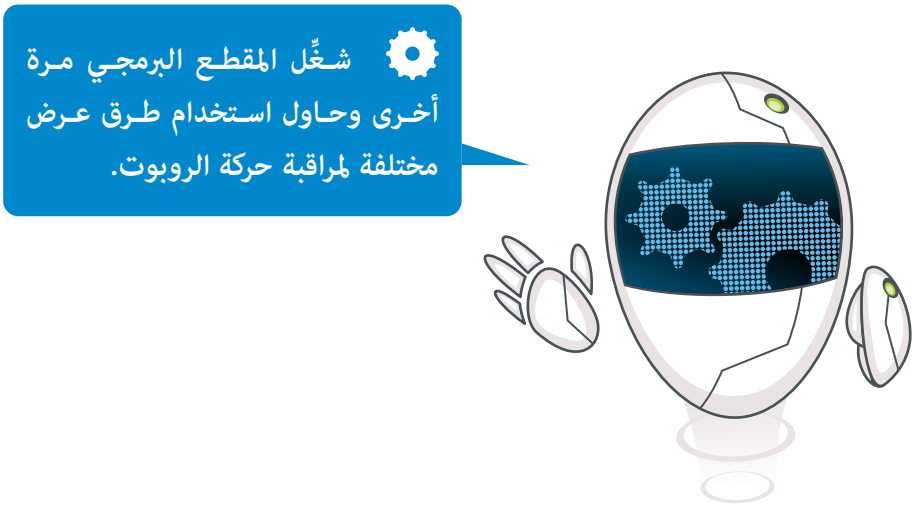
طريقة عرض كاميرا التتبع (العرض ثلاثي الأبعاد) تُمكِّنك من استخدام الضغط والسحب بالفأرة للتنقل والتكبير والتصغير باستخدام عجلة تمرير الفأرة مما يتيح لك رؤية أقرب أو منظورًا أوسع حسب الحاجة.



كاميرا الشخص الأول (First-Person Camera)

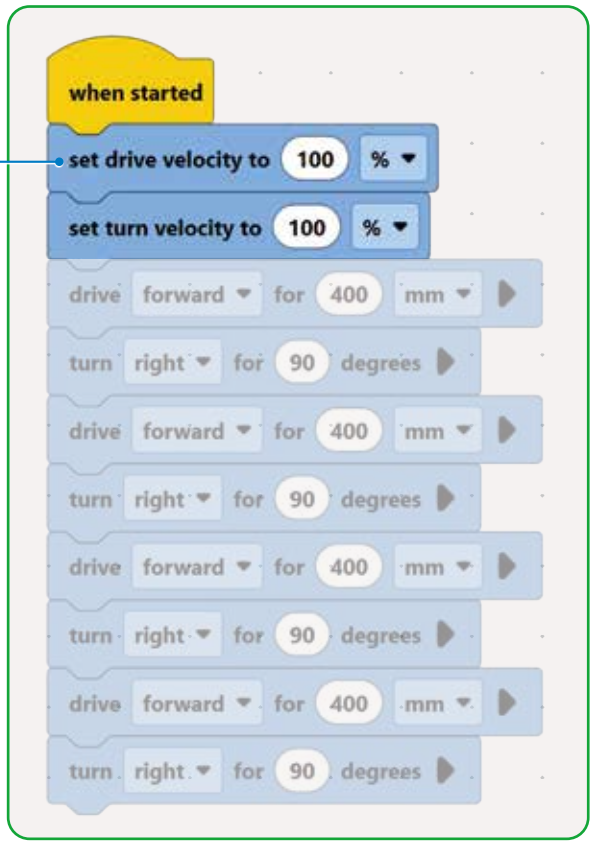
تعرض كاميرا الشخص الأول ساحة اللعب وكأنك تقود الروبوت من داخله.





التحكم في سرعة الروبوت

يمكنك التحكم في سرعة حركة الروبوت ودورانه عن طريق تغيير إعدادات السرعة، بإضافة لبنة **set drive velocity to ( ) ( ) (ضبط سرعة القيادة إلى ( ) ( ))** ولبنة **set turn velocity to ( ) ( ) (ضبط سرعة الدوران إلى ( ) ( ))** في بداية المقطع البرمجي. زيادة هذه القيم تساعد الروبوت على إكمال حركته في وقت أقل، ويسمح لك بمقارنة تأثير السرعة على أدائه.

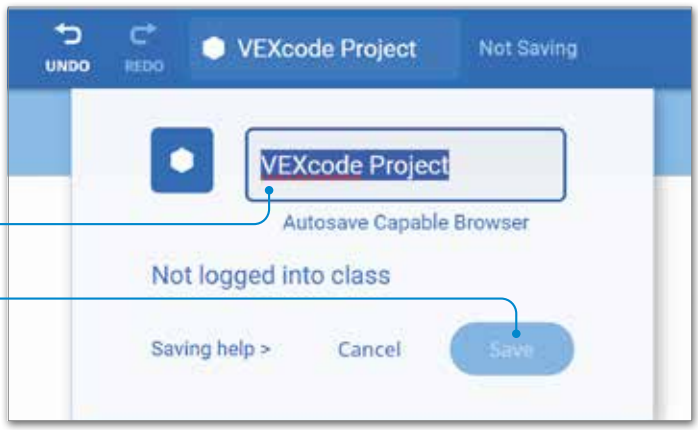


إن سرعة الروبوت الافتراضية عند إنشائه هي 50%، لتغيير سرعته، أضف لبنتي سرعة القيادة المحددة وسرعة الدوران المحددة في بداية المقطع البرمجي الخاص بك.

شغِّل المقطع البرمجي، هل تلاحظ أي اختلاف عن السابق؟

تغيير اسم المشروع

يمكنك تغيير اسم المشروع الذي تعمل عليه، من خلال الانتقال إلى شريط الأدوات، والنقر على VEXcode Project، ثم كتابة اسم جديد لمشروعك والنقر على Save (حفظ).



اكتب الاسم الذي ترغب فيه لمشروعك.

انقر **Save (حفظ)** لحفظ مشروعك بالاسم الجديد.

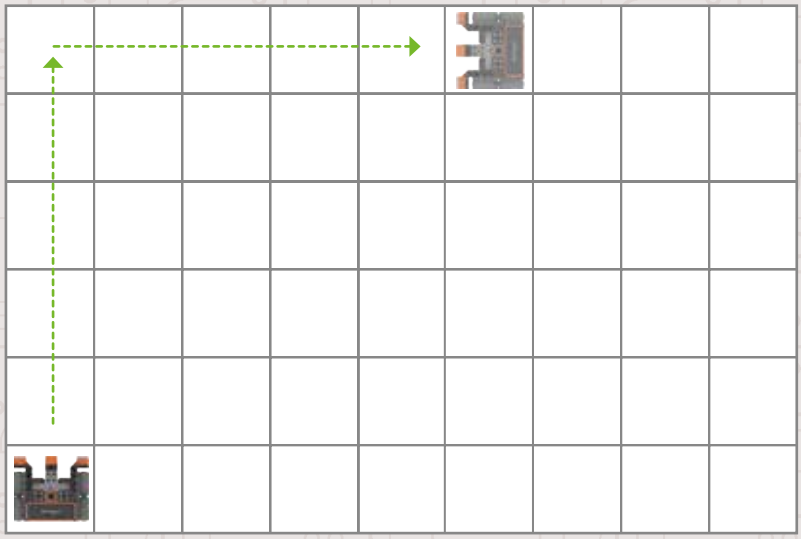
لنطبق معًا

تدريب ١

عدِّد طرق العرض المختلفة التي يمكنك استخدامها في تطبيق VEXcode VR.

تدريب ٢

أنشئ مقطعًا برمجيًّا يتحرك فيه الروبوت خمسة مربعات للأمام، ثم ينعطف 90 درجة إلى اليسار، ثم يتحرك خمسة مربعات للخلف. اضبط السرعة على 20% عندما يتحرك الروبوت و50% عندما ينعطف. ادرس الصورة المرفقة لفهم الهدف النهائي.



تدريب ٣

أنشئ مقطعًا برمجيًّا يُشكّل فيه الروبوت مثلثًا. ادرس الصورة المرفقة لحساب الدرجات المطلوبة لكل انعطاف يقوم به الروبوت.

الدرس ٤-٢:

التَّكرارات ورسم الأشكال

أوامر التَّكرار

قد ترغب في بعض الأحيان بإنشاء مقطع برمجي ينفذ نفس التعليمات البرمجية عدة مرات. يمكن استخدام **Loop commands** (**أوامر التكرار**) لتنفيذ ذلك.

تُعدُّ لبنات التكرارات: ( ) **repeat** (**كرر** ( ))، و **forever** (**للأبد**)، و ( ) **repeat until** (**كرر حتى** ( ))، هي الأكثر استخدامًا في تطبيق VEX code VR. تنتمي هذه اللبنات إلى فئة لبنات **Logic** (**المنطق**)، وتتحكم بسير المقطع البرمجي.

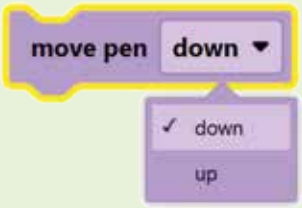
لبنات التكرار في تطبيق VEXcode VR

لقد أنشأت في الدرس السابق مقطعًا برمجيًّا يُمكِّن الروبوت من تشكيل مربع. حاول الآن إعادة البرمجة، ولكن باستخدام لبنة ( ) repeat (كرر ( )).

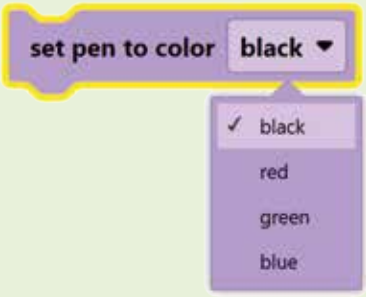
يجب ضبط سرعة القيادة والانعطاف مرة واحدة فقط في بداية المقطع البرمجي وخارج لبنة التكرار.

رسم الأشكال

لرؤية مسار الروبوت بشكل أوضح أثناء تحركه، يمكنك استخدام قلم الروبوت، فهو يترك أثرًا يساعدك في متابعة الحركة بدقة. يوجد هذا القلم في وسط الروبوت. لتنفيذ ذلك يمكن استخدام فئة لبنات **Drawing (رسم)**.



لبنة **move pen ( ) (حرك القلم ( ))**: تستخدم لتحريك أداة القلم (أسفل) ليتمكن الروبوت من الرسم في ساحة اللعب، أو تحريكه (فوق) لإيقاف الرسم.



لبنة **set pen to color ( ) (اضبط القلم على اللون ( ))**: تستخدم لتغيير لون القلم.

رسم دائرتين

في هـذا المشروع سيرسـم الروبـوت دائـرتين، ولكي يرسـم الروبـوت دائرة، يجب أن يتحرك مسـافة قصيرة، ثم ينعطف زاوية صغيرة في كل مرة.

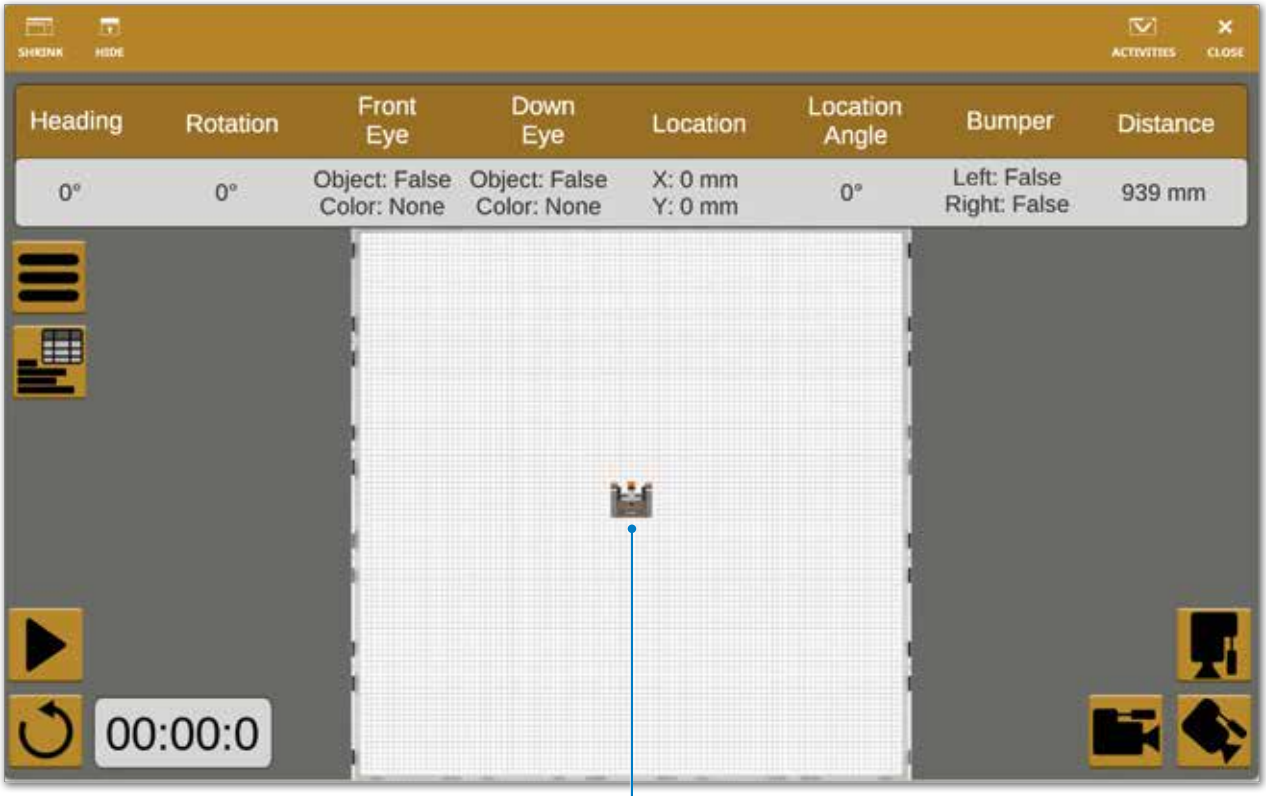
فـعلى سـبيل المثـال: إذا تحـرك الروبـوت 50 مليـمتر ثـم انعطـف 10 درجـات بشـكل متكـرر، فسـيتكوّن خط دائري.

ولمعرفة عدد مرات التكرار، تذكّر أن الدائرة الكاملة = 360 درجة.

وبما أن الروبوت ينعطف 10 درجات كل مرة، تُحسب عدد التكرارات كالآتي:

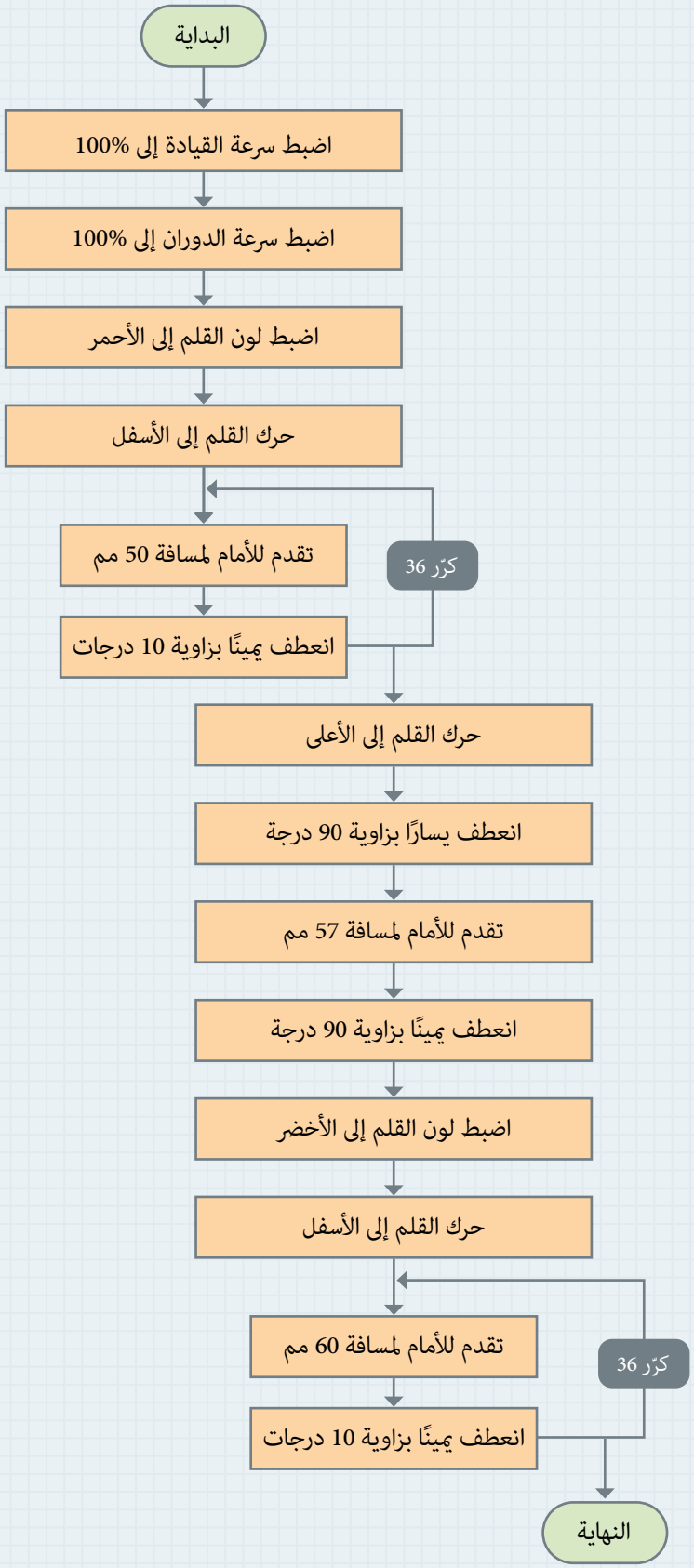
360 ÷ 10 = 36 مرة

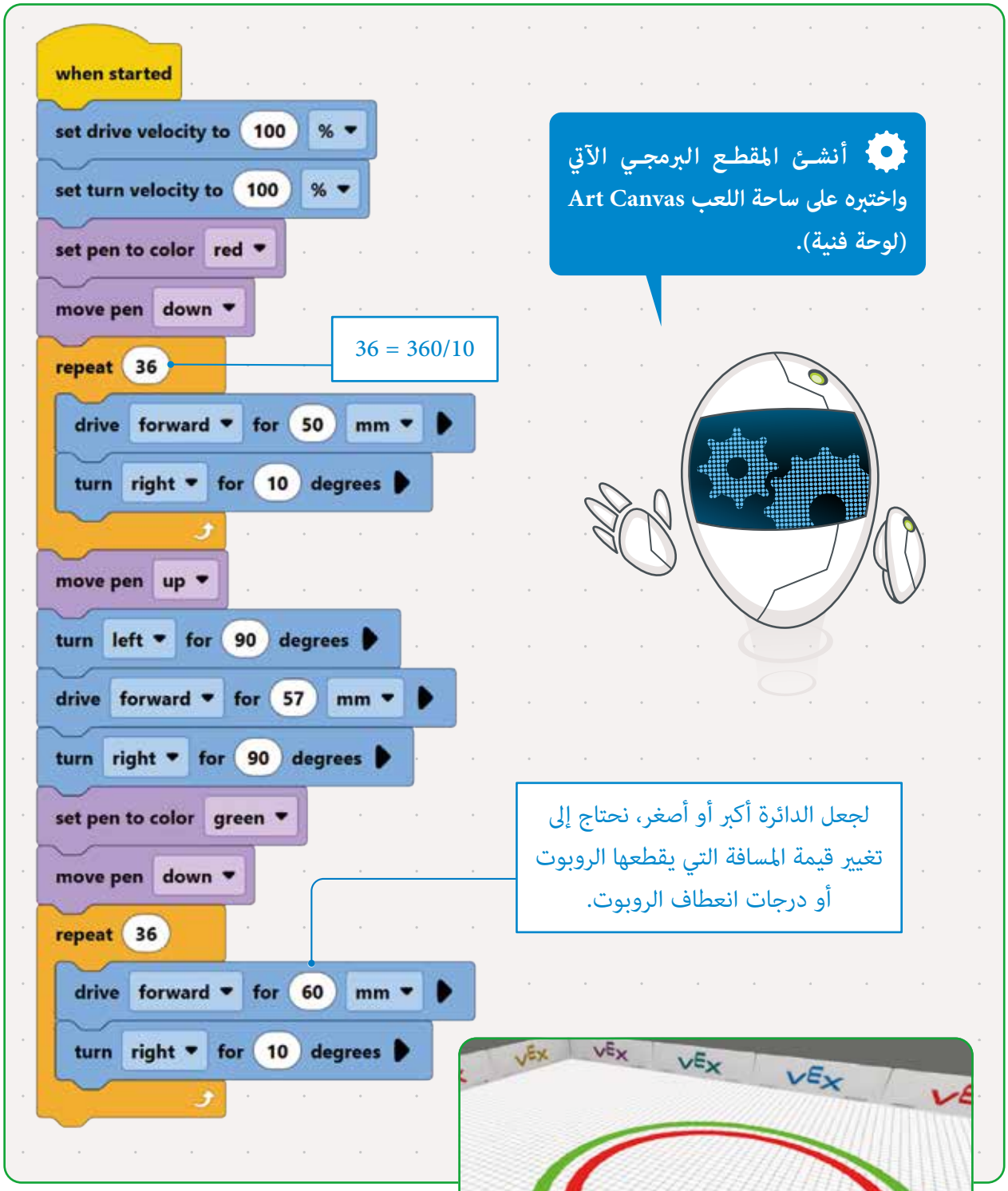
أي أن الروبـوت يحتـاج إلى 36 تكـرارًا ليكمـل دائـرة كاملـة. لتنفيـذ ذلـك ستسـتخدم سـاحة اللعـب Art Canvas (لوحة فنية).



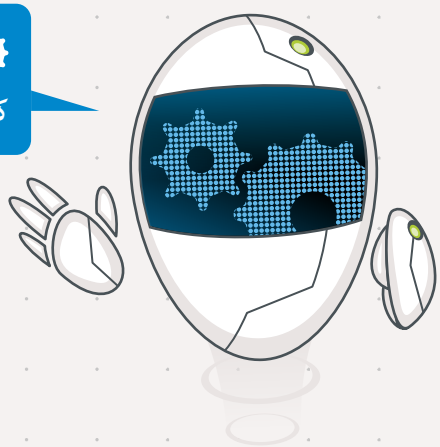
تمثل **Art Canvas (لوحة فنية)** ساحة اللعب الافتراضية في تطبيق VEXcode VR، حيث تُفتح تلقائيًّا عند بدء مشروع جديد. في هذه الساحة يكون الموضع الافتراضي للروبوت عند النقطة X: 0 مليمتر، و Y: 0 مليمتر، ويتم تقسيم المساحة إلى مربعات صغيرة طول ضلعها 20 مليمتر.

المخطط الانسيابي




when started
set drive velocity to 100 %
set turn velocity to 100 %
set pen to color red
move pen down
repeat 36
drive forward for 50 mm
turn right for 10 degrees
move pen up
turn left for 90 degrees
drive forward for 57 mm
turn right for 90 degrees
set pen to color green
move pen down
repeat 36
drive forward for 60 mm
turn right for 10 degrees
36 = 360/10
أنشئ المقطع البرمجي الآتي واختبره على ساحة اللعب Art Canvas (لوحة فنية).
لجعل الدائرة أكبر أو أصغر، نحتاج إلى تغيير قيمة المسافة التي يقطعها الروبوت أو درجات انعطاف الروبوت.
استخدم Chase Camera (كاميرا التتبع) ولاحظ أن القلم الموجود في وسط الروبوت يتغير لونه بحسب لون الخط الذي يرسمه.

أضـف الأجـزاء الآتية إلى المقطـع البرمجي، ولاحظ كيف سيبدو الرسم.

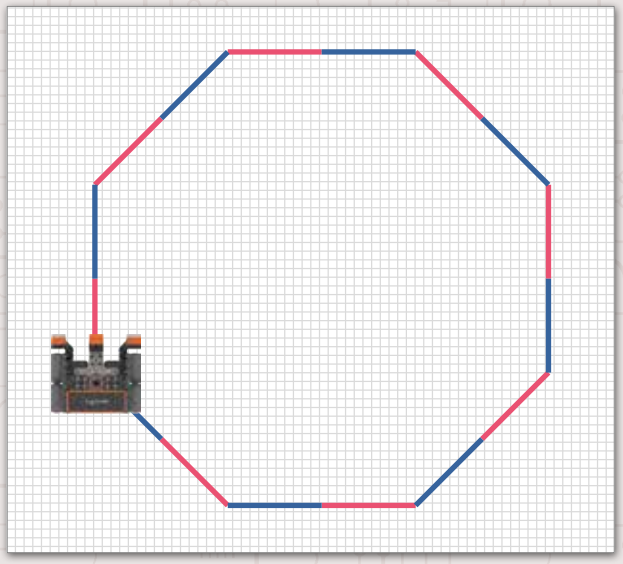


```
when started
set drive velocity to 100 %
set turn velocity to 100 %
move pen down
repeat 36
  drive forward for 40 mm
  turn right for 10 degrees
move pen up
turn left for 90 degrees
drive forward for 57 mm
turn right for 90 degrees
set pen to color red
move pen down
repeat 36
  drive forward for 50 mm
  turn right for 10 degrees
move pen up
turn left for 90 degrees
drive forward for 57 mm
```

```
turn right for 90 degrees
set pen to color green
move pen down
repeat 36
  drive forward for 60 mm
  turn right for 10 degrees
move pen up
turn left for 90 degrees
drive forward for 57 mm
turn right for 90 degrees
set pen to color blue
move pen down
repeat 36
  drive forward for 70 mm
  turn right for 10 degrees
```

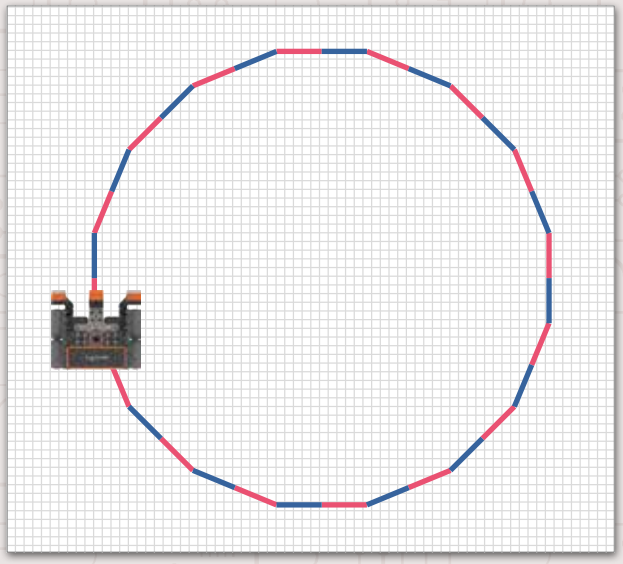
تدريب ١

استخدم ساحة لعب Art Canvas (لوحة فنية) لإنشاء مقطع برمجي يُمكّن الروبوت من تشكيل مضلع منتظم بثمانية أضلاع وزوايا متساوية. يجب أن يكون كل ضلع من المضلع نصف أحمر ونصف أزرق. يمكنك استخدام الصورة الآتية لحساب الدرجات التي يحتاجها الروبوت لكل دورة. يتحرك الروبوت مسافة 150 مم لرسم الجانب الأول.



تدريب ٢

بناءً على التدريب السابق، أنشئ مقطعًا برمجيًّا لرسم مضلع بضعف عدد أضلاع المضلع السابق. إذا ضاعفتَ عدد الأضلاع، فما زاوية الدوران التي يحتاجها الروبوت هذه المرة؟ يتحرك الروبوت مسافة 150 مم لرسم الجانب الأول.



لنطبق معًا

تدريب ٣

قم بتحليل المقطع البرمجي ورسم النتيجة بعد أن ينفذ الروبوت المقطع البرمجي.

الدرس ٥-٢:

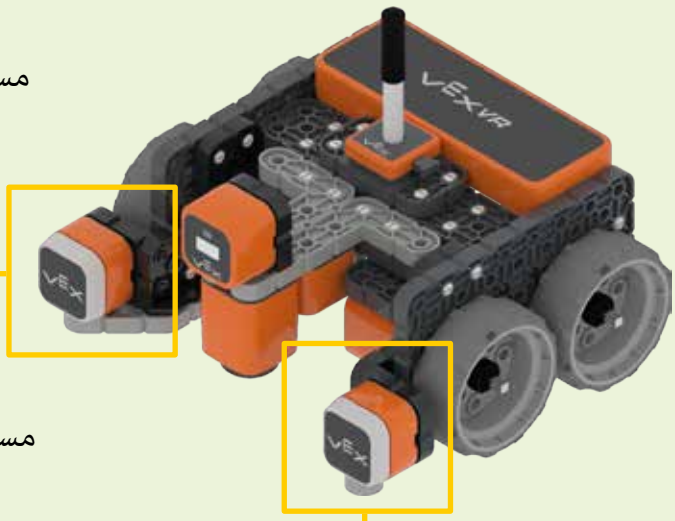
المستشعرات

تُزوَّد الروبوتات بأجهزة استشعار متخصصة تُمكّنها من التنقل في محيطها. تُساعدها هذه الأجهزة على تحديد موقعها، ورصد الأجسام القريبة، والتفاعل مع الروبوتات الأخرى والبشر. على سبيل المثال: تعتمد السيارات ذاتية القيادة على تقنيات الروبوتات، وتحتوي على نفس نوع أجهزة الاستشعار الموجودة في الروبوت.

مستشعر الاصطدام (Bumper Sensor)

الروبوت الافتراضي مزود بمستشعرَي Bumper (اصطدام)، أحدهما في الجهة اليمنى والآخر في الجهة اليسرى. يعمل كل مستشعر مثل زرّ صغير يُضغط عندما يلامس الروبوت جسمًا أمامه، فيُرسل إشارة تُخبر الروبوت بحدوث اصطدام.

مستشعر Right Bumper (الاصطدام الأيمن)

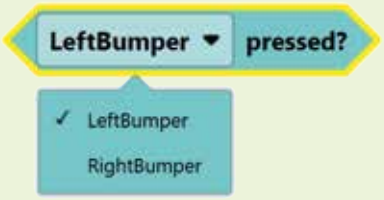


مستشعر Left Bumper (الاصطدام الأيسر)

تحريك الروبوت باستخدام المستشعرات

ستستكشف كيف يمكن للروبوت اتخاذ قرارات اعتمادًا على المستشعرات عندما يكتشف جسمًا أمامه. فالروبوتات تحتاج أحيانًا إلى تغيير طريقها أو تعديل حركتها حسب ما يحدث حولها. ستستخدم ساحة لعب Grid Map (خريطة الشبكة) لتمكين الروبوت من تغيير اتجاهه عند اصطدامه بجدار. ولتحقيق ذلك، سنتعرّف على لبنتين مهمتين تساعدان الروبوت على تنفيذ هذا السلوك.

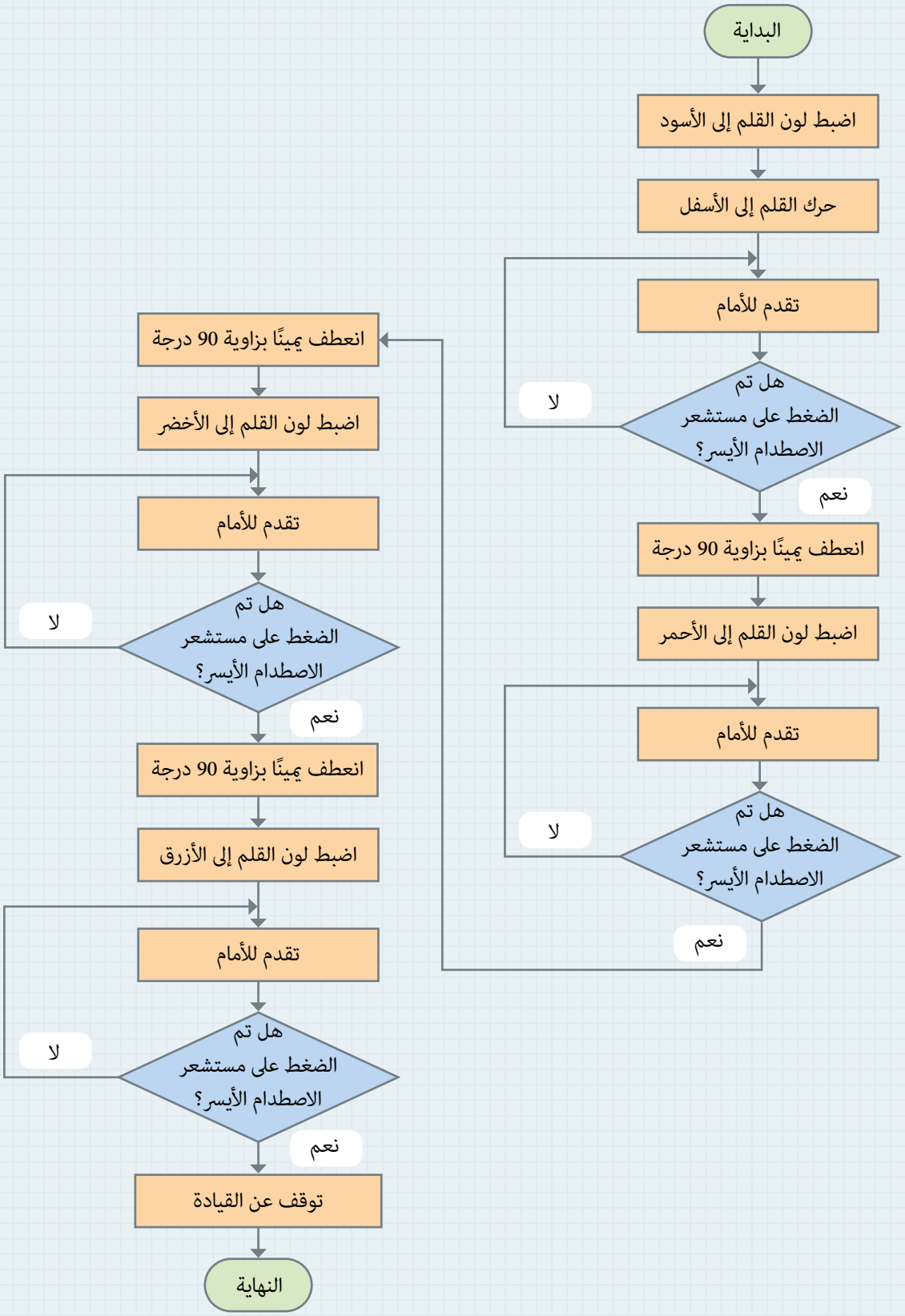
لبنة **wait until** ( ) (**انتظر حتى** ( )): تُستخدم لإيقاف المقطع البرمجي مؤقتًا حتى يتحقق شرط معين. هذا يعني أن الروبوت سيتوقف وينتظر حتى وقوع الحدث المحدد قبل الانتقال إلى الأمر التالي. تنتمي هذه اللبنة إلى فئة لبنات **Logic** (**المنطق**).

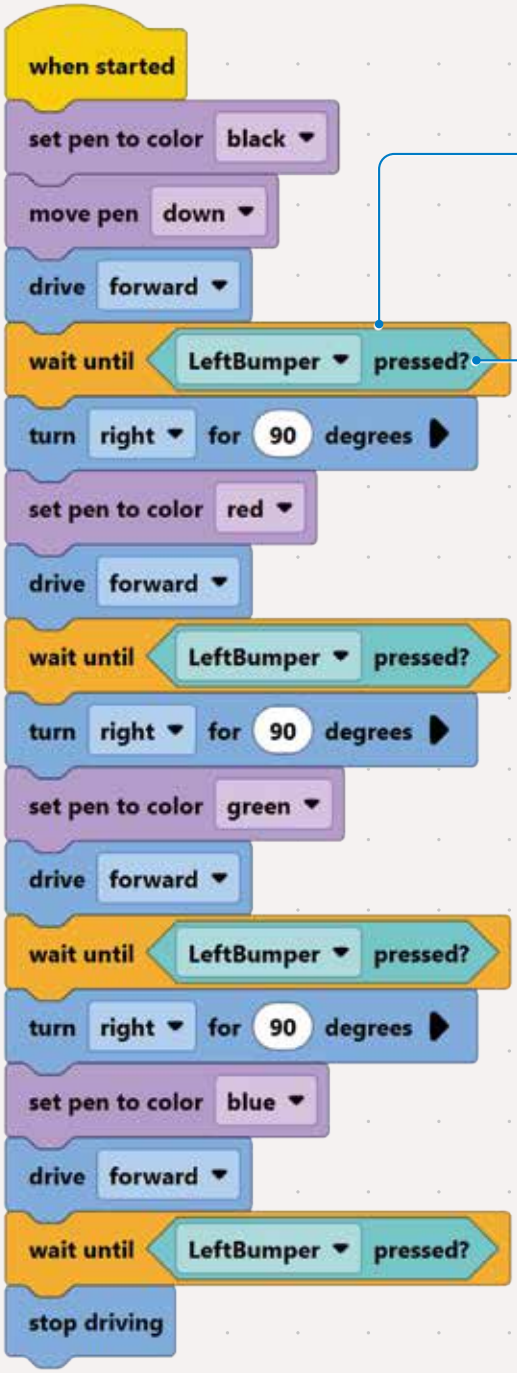


لبنة **pressed** ( ) (( ) **مضغوط؟**): لبنة شرطية تُستخدم للتحقق مما إذا كان مستشعر الروبوت Bumper (الاصطدام) الأيسر أو الأيمن قد تم الضغط عليه. تُرجع القيمة "صحيح" إذا تم الضغط على المستشعر، والقيمة "خطأ" إذا لم يتم الضغط عليه. تنتمي هذه اللبنة إلى فئة لبنات **Sensing** (**الاستشعار**).

غيّر ساحة اللعب إلى Grid Map (خريطة الشبكة)، ثم استخدم مستشعر Bumper (الاصطدام) وأداة القلم وحاول رسم مربع كبير.

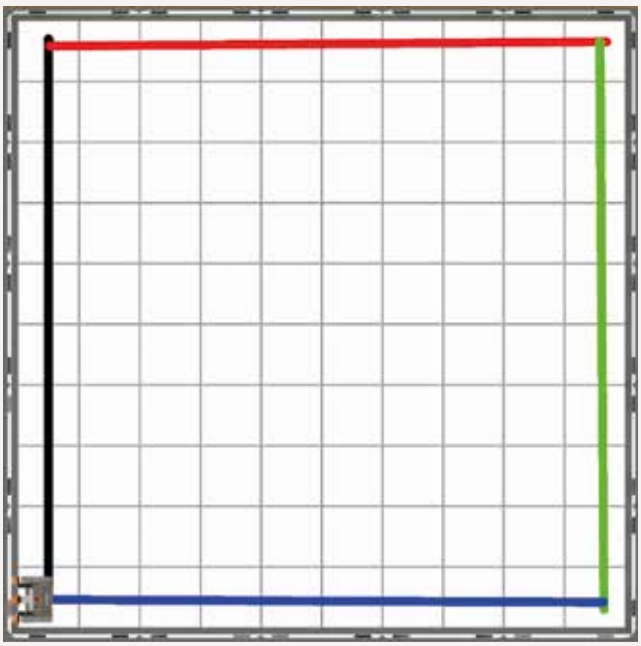
المخطط الانسيابي





عندما تجمع لبنتي **wait until** ( ) (**انتظر حتى** ( )) و ( ) **pressed** (( ) **مضغوط؟**)، سيوقف الروبوت حركته مؤقتًا ولن يستأنفها إلا عندما يلامس الزر جدارًا أو جسمًا.

يمكنك إيجاد لبنة **Left Bumper pressed?** (**الاصطدام الأيسر مضغوط؟**) في فئة لبنات **Sensing** (**الاستشعار**).

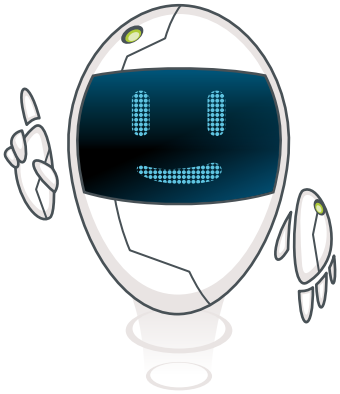


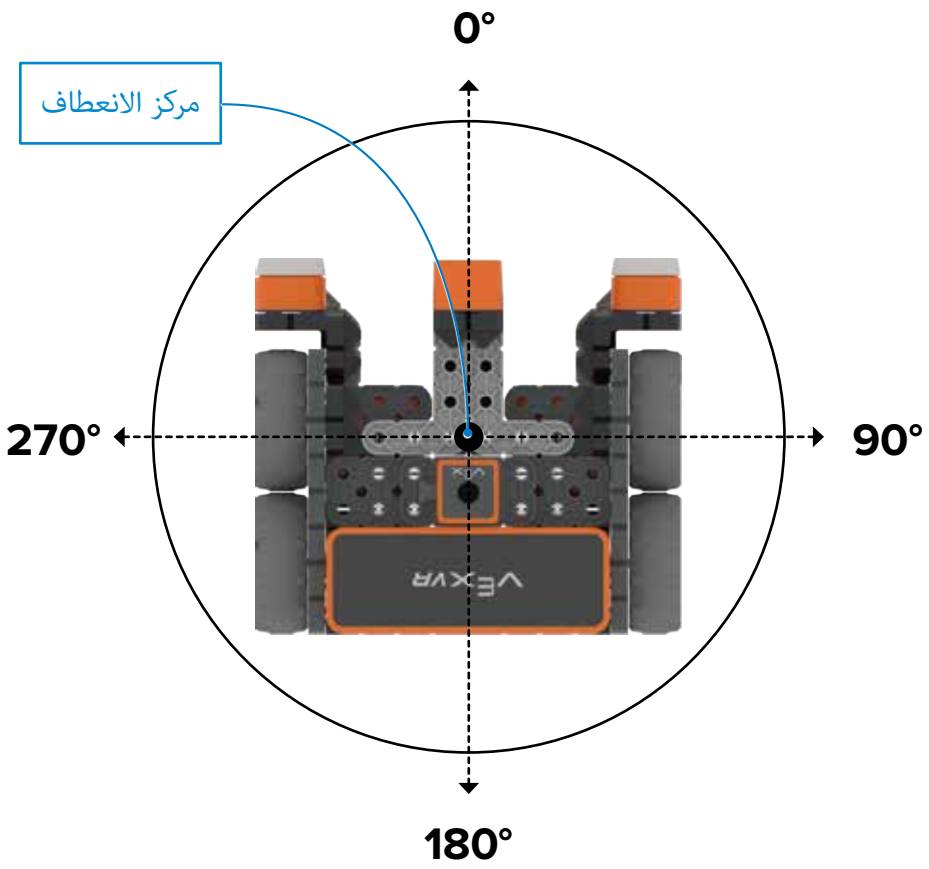
مستشعر الانعطاف (Gyro Sensor)

يوجد مستشعر الانعطاف في الجزء الخلفي من الروبوت، وهو المسؤول عن تمكينه من القيادة في خط مستقيم والقيام بانعطافات دقيقة، ويعتمد الروبوت في تحديد موقعه الافتراضي على مركز الانعطاف، وهو نفس موضع القلم المستخدم في التتبع.

يساعد مستشعر الانعطاف في تحديد موقع الروبوت من خلال قياس اتجاهه وسرعة وانحراف دورانه أثناء الحركة، ويُستخدم هذا المستشعر في العديد من الروبوتات والأجهزة، مثل الطائرات المسيّرة، حيث يسهم في الحفاظ على ثبات الكاميرا حتى عند انحراف الطائرة يمينًا أو يسارًا.

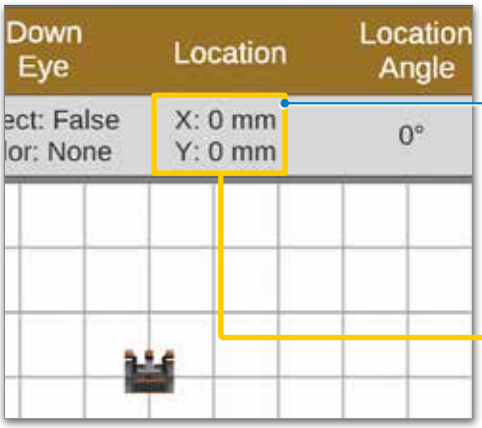
يمكن لمستشعر الانعطاف تحديد الاتجاه ومسافة انعطاف الروبوت عن نقطة البداية. واكتشاف ما إذا كانت الحركة باتجاه عقارب الساعة أو عكسها.



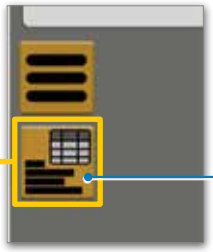


نظام الإحداثيات في تطبيق VEXcode VR

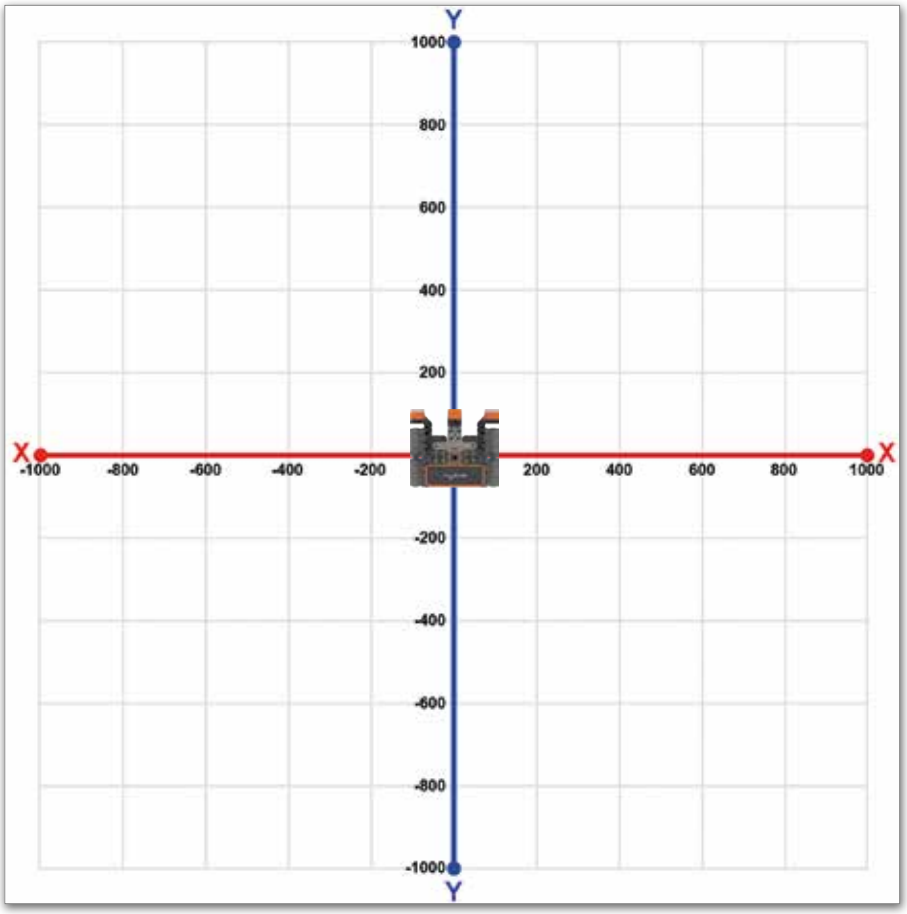
يُستخدم نظام الإحداثيات المعتمد على القيمتين (x,y) في تطبيق VEXcode VR لتحديد موقع الروبوت أو موقع أي نقطة في ساحة اللعب، مما يساعد على توجيه الروبوت بدقة إلى الموقع المطلوب.



يمكن العثور على موقع X و Y المحدد للروبوت الافتراضي على لوحة المعلومات.



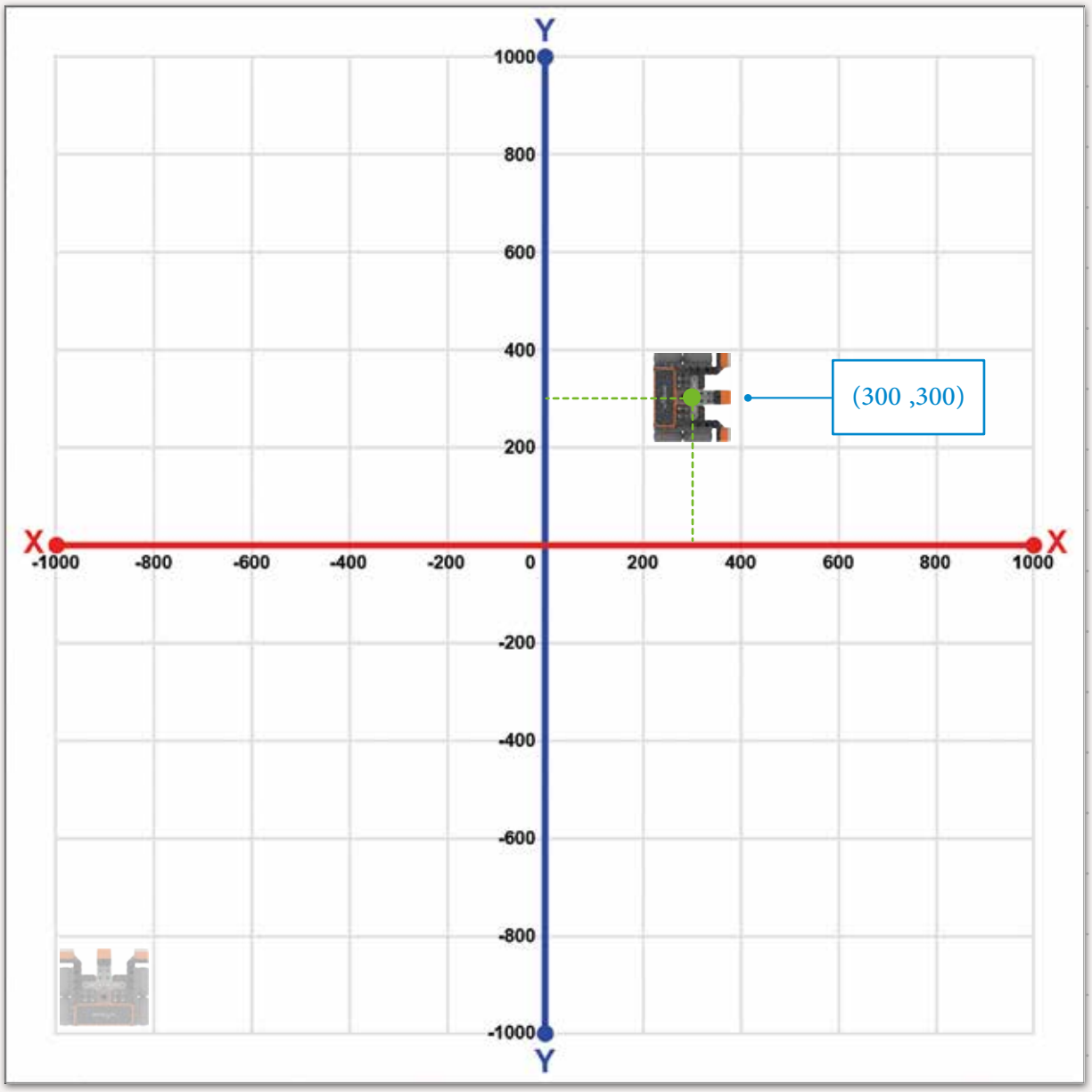
انقر زر لوحة المعلومات في الزاوية العلوية اليسرى من نافذة ساحة اللعب لفتح لوحة المعلومات.



معلومة تقنية

يتراوح نطاق ملاعب VEXcode VR من 1000- مم إلى 1000 مم لكلٍّ من المحورين x وy.

نقطة البداية الافتراضية للروبوت في ساحة اللعب (Grid Map) هي (900- ,900-). وهذا يعني أنه لتحريك الروبوت إلى الإحداثيات (0, 0)، يجب برمجته وفقًا لذلك. حاول إنشاء المقطع البرمجي ليتمكن الروبوت من الوصول إلى الإحداثيات (300, 300).



موقع الاستشعار

تستخدم لبنات ( ) **in** ( ) **position** (**الموضع** ( ) **في** ( )) و **position angle in degrees** (**زاوية الموضع بالدرجات**) مع مستشعر Gyro. توجد هذه اللبنات في فئة لبنات **Sensing** (**الاستشعار**) في قسم Location Sensing (موقع الاستشعار).

لبنة ( ) **in** ( ) **position** (**الموضع** ( ) **في** ( )): ترتبط مع اللبنات الأخرى، لتحديد موضع إحداثيات x و y للروبوت الافتراضي بالمليمتر (mm) أو بالبوصة (inch).

لبنة **position angle in degrees** (**زاوية الموضع بالدرجات**): ترتبط مع اللبنات الأخرى لحساب الاتجاه الحالي للروبوت الافتراضي بالدرجات.

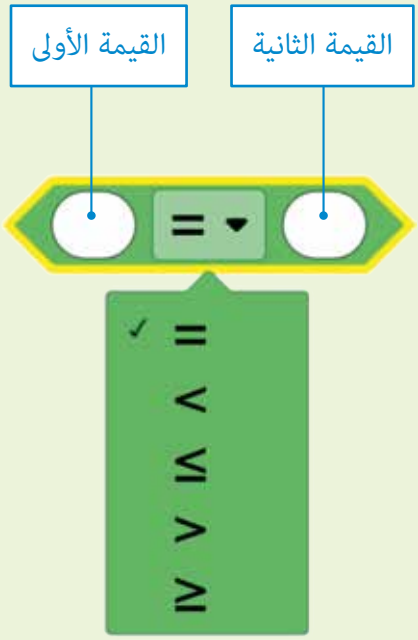
لاحظ أن لبنة ( ) in ( ) position ( ) (الموضع ( ) في ( )) تستخدم لتحديد موقع حركة الروبوت الافتراضي في ساحة اللعب، بينما تستخدم لبنة position angle in degrees (زاويةُ الموضع بالدرجات) لتحديد الانعطافات التي يقوم بها.

الجمل الشرطية

يُعدُّ اتخاذ القرارات جزءًا مهمًّا من الحياة اليومية، فأنت تتخذ القرارات بناءً على ما تلاحظه أو بما تعتقد بأنه صواب. في تطبيق VEXcode VR يمكنك استخدام لبنة **wait until** ( ) (**انتظر حتى** ( )) لإيقاف المقطع البرمجي حتى تنفيذ الشرط.

عندما تمطر السماء فسنستخدم المظلة، فالشروط هي السبب ولها نتيجة معينة. في الواقع لا يمكن للروبوت أن يقرر بنفسه كيفية الاستجابة لأحداثٍ أو ظروفٍ معينة، ولذلك تستخدم الجمل الشرطية التي تُخبر الروبوت بما يجب أن يقوم به ومتى يفعل ذلك.

المعاملات الشرطية في تطبيق VEXcode VR



عند كتابة الجمل الشرطية، يمكنك استخدام المعاملات للمقارنة بين القيم ويتخذ الروبوت الإجراء المناسب بناءً على نتيجة هذه المقارنة. وتكون نتيجة الشرط دائمًا إمّا True (صواب) أو False (خطأ). توجد ثلاث لبنات للمعاملات الشرطية:

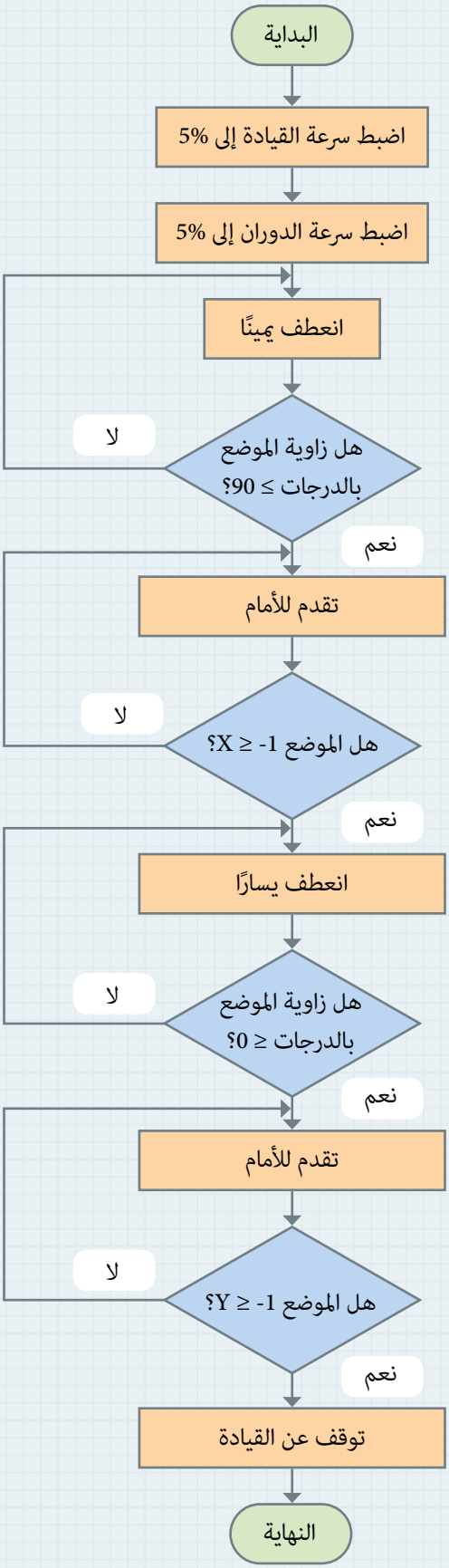
- لبنة **greater than** ( ) (**أكبر من** ( ) > ( )) >
- لبنة **less than** ( ) (**أصغر من** ( ) < ( )) >
- ولبنة **equal to** ( ) (**يساوي** ( ) = ( )) >

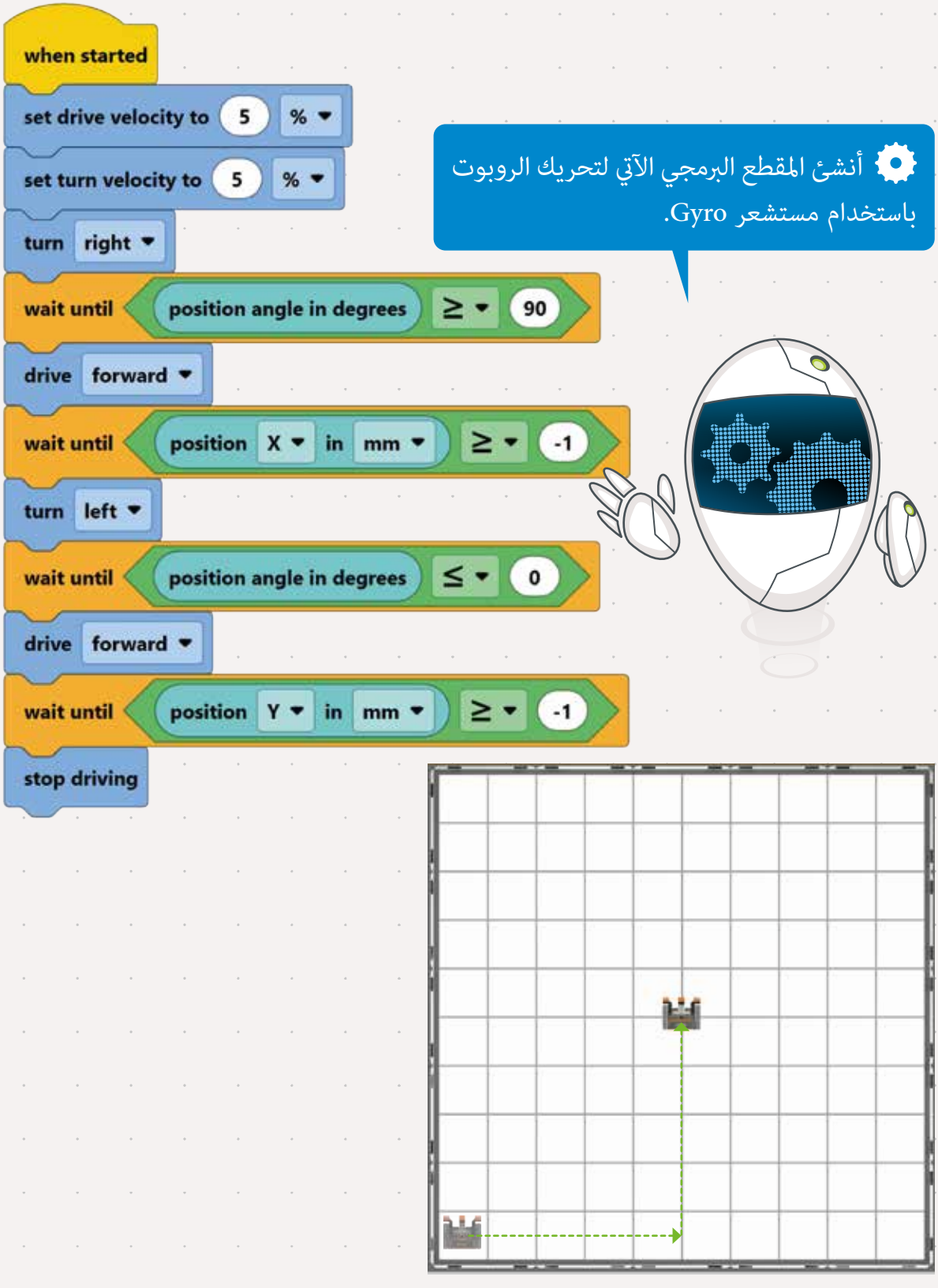
تحتوي كل لبنة على خانتي إدخال تكتب فيهما نصًا أو تضع قيمة معينة (مثل لبنة الإجابة). يمكن العثور على جميع هذه اللبنات في فئة لبنات **Logic** (**المنطق**).

إنشاء مقطع برمجي

ستنشئ مقطعًا برمجيًّا مستخدمًا ساحة اللعب Grid Map (خريطة الشبكة) لجعل الروبوت يصل إلى النقطة X: 0 مم، Y: 0 مم، ثم يتوقف.

المخطط الانسيابي





معلومة تقنية

يسهم ضبط السرعة على سرعة أقل في اكتشاف مستشعر Gyro الحركات بدقة أكبر.

لنطبق معًا

تدريب ١

اشرح كيف يساعد مستشعر Gyro الروبوت الافتراضي على التحرك بشكل أكثر دقة.

تدريب 2

استخدم مستشعر Gyro وأنشئ مقطعًا برمجيًّا لجعل الروبوت يتوقف عند مروره بالمحور Y.

تدريب 3

يُظهر موضع لبنة الروبوت إحداثيات قيمتي X وY للروبوت على ساحة اللعب Grid Map (خريطة الشبكة).

١. إذا بدأ الروبوت من (900- ,900-) وتقدم للأمام على طول المحور X، فكيف ستتغير قيمة X؟

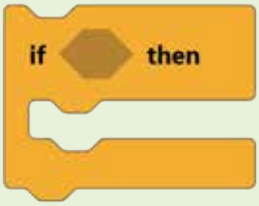
٢. إذا تحرك الروبوت للأمام على طول المحور Y، فكيف ستتغير قيمة Y؟

الدرس ٦-٢:

اتخاذ القرارات

اتخاذ القرارات

تسمح الجمل الشرطية للمقطع البرمجي بتنفيذ إجراءات مختلفة بناءً على شروط منطقية. يتم تنفيذ جزء محدد من المقطع البرمجي اعتمادًا على ما إذا كان الشرط صحيحًا أو خاطئًا.



لبنة **if ( ) then (إذا كان ( ) إذًا)**: تتحقق أولًا من شرطها؛ فإذا كان الشرط صحيحًا، فسيتم تنفيذ الأوامر الموجودة بداخلها، وإذا كان الشرط خاطئًا، فسيتم تجاهل هذا الجزء من المقطع البرمجي. تنتمي هذه اللبنة في تطبيق VEXcode VR إلى فئة لبنات **Logic (المنطق)**.

كيفية إعادة ضبط الاتجاه والانعطاف

يُعدُّ تحديد موقع واتجاه الروبوت أثناء تحركه في ساحة اللعب أمرًا مُهمًّا للغاية، وتساعد هذه المعلومات على نقل الروبوت إلى موقع آخر. فعلى سبيل المثال: إذا أردت الذهاب إلى مدرستك، فإنك ستتوجه إلى مدخل المنزل، ثم ستفتح الباب وتتحرك لتصل إلى رصيف الشارع، ثم ستتابع التقدم وتنعطف باتجاه مدرستك وستستمر بهذا الأمر حتى الوصول إلى المدرسة.

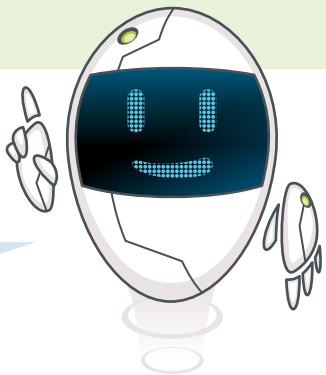
يمكن القيام بهذا الأمر باستخدام الروبوت من خلال استخدام فئة لبنات **Drivetrain (نظام نقل الحركة)** وبالاستعانة بفئة لبنات **Sensing (الاستشعار)**.

لبنة **set drive heading to ( ) degrees (ضبط اتجاه القيادة إلى ( ) درجة)**: تُحدد اتجاه الروبوت إلى قيمة محددة من اختيارك، بالاستعانة بوضع الزاوية الحالي لمستشعر gyro.

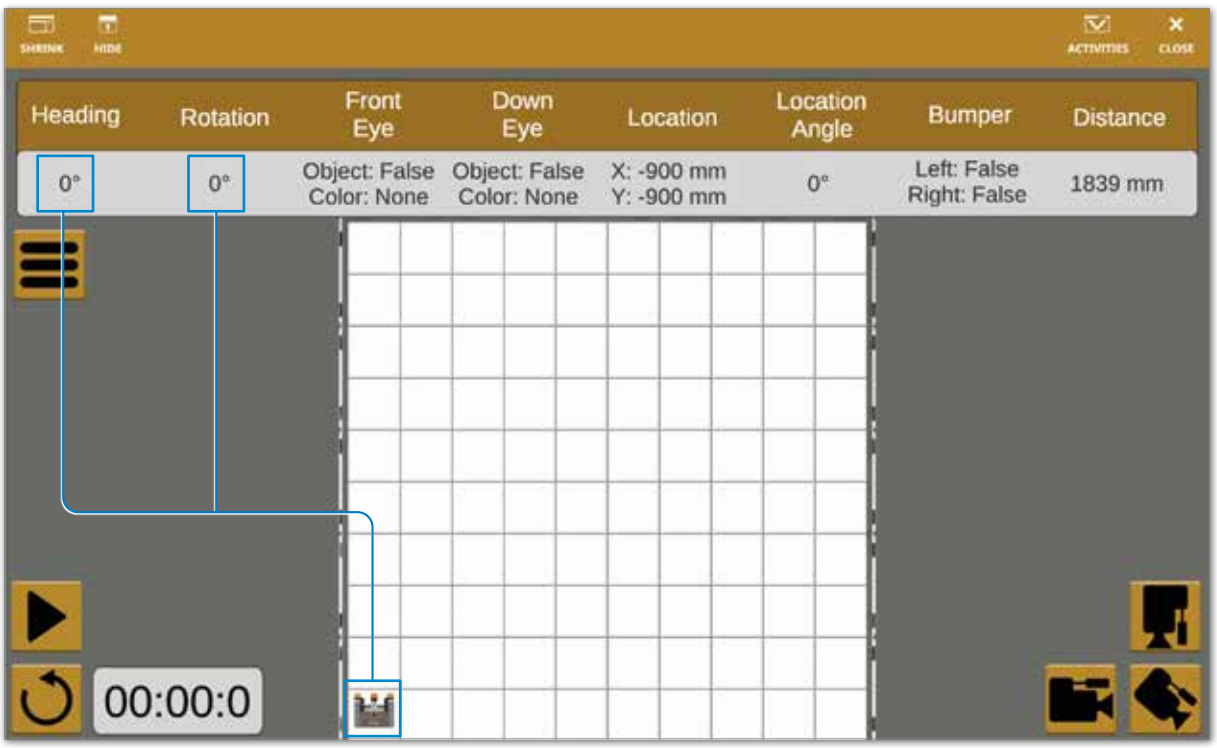
لبنة اضبط **set drive rotation to ( ) degrees** (**ضبط دوران المحرك إلى ( ) درجة**): تُحدد زاوية انعطاف الروبوت أثناء قيادته إلى قيمة محددة من اختيارك.

لبنة **drive heading in degrees** (**اتجاه القيادة بالدرجات**): تُحدد اتجاه نظام قيادة الروبوت بالاستعانة بوضع الزاوية الحالي لمستشعر gyro.

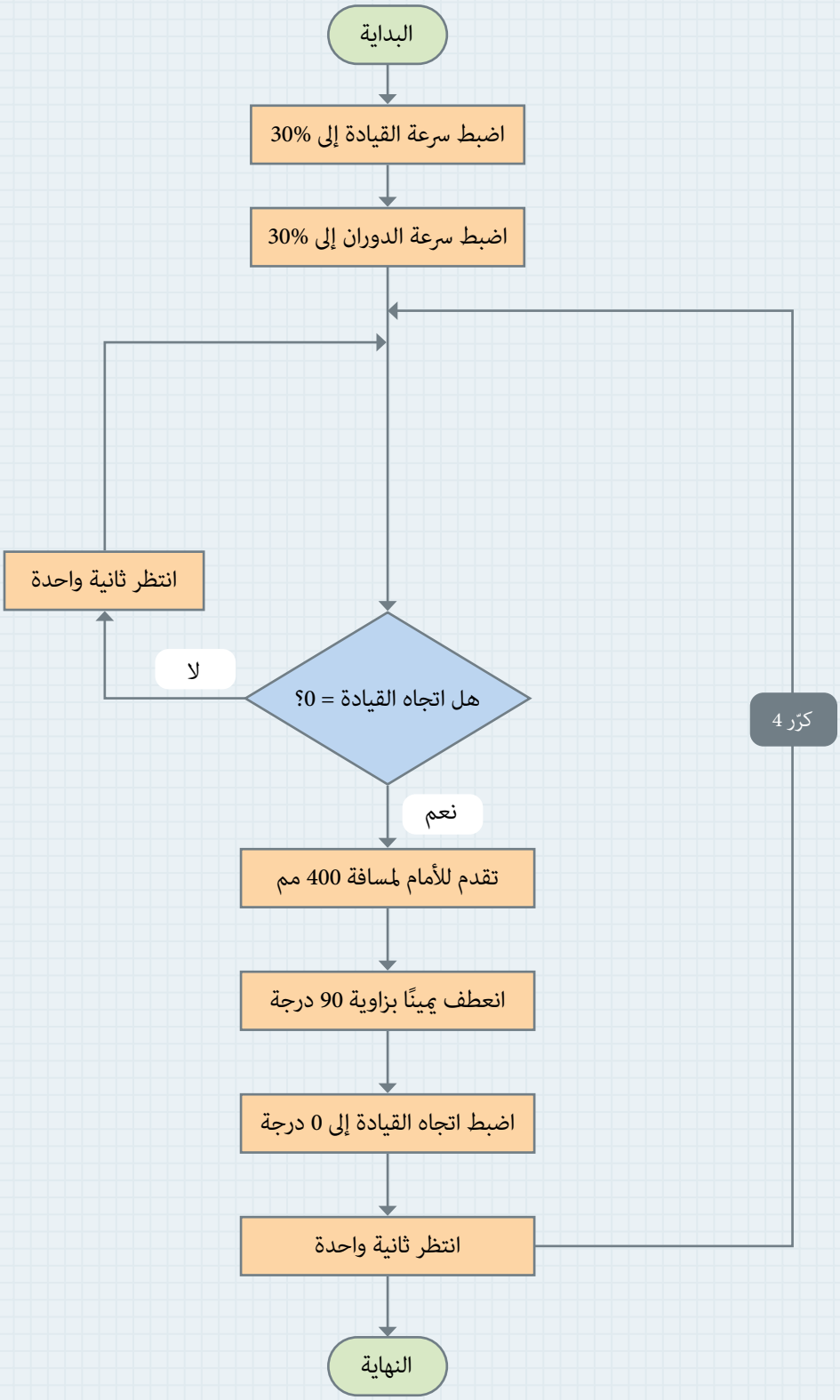
لبنة **drive rotation in degrees** (**دوران المحرك بالدرجات**): تُحدد زاوية انعطاف نظام قيادة الروبوت عند ضبطه باستخدام مستشعر gyro.



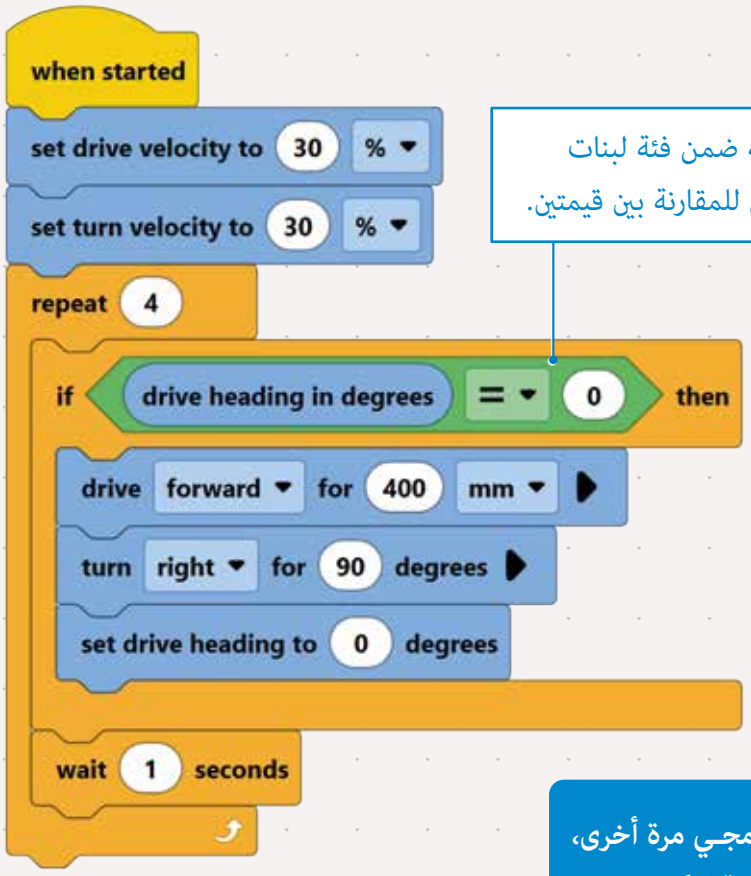
يمكنك دائمًا تتبع اتجاه الروبوت الافتراضي وعدد الانعطافات التي قام بها من خلال لوحة المعلومات.



المخطط الانسيابي

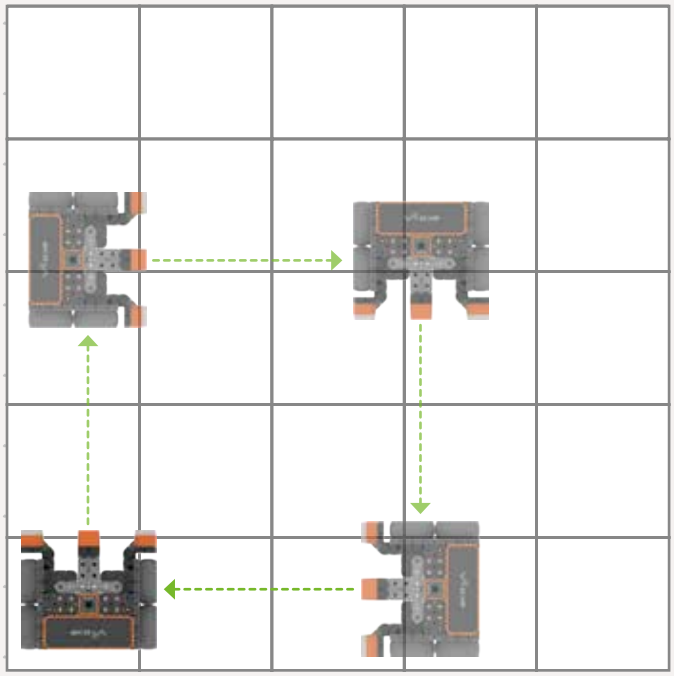


اختر ساحة اللعب Grid Map (خريطة الشبكة)، ثم أنشئ المقطع البرمجي الآتي وشغله.



توجد لبنة عامل المقارنة ضمن فئة لبنات **Logic (المنطق)**، وتُستخدم للمقارنة بين قيمتين.

جرّب تشغيل المقطع البرمجي مرة أخرى، ولكن هذه المرة، اضبط السرعة لتكون ١٠٪. كيف سيؤثر ذلك في النتيجة برأيك؟



لنطبق معًا

تدريب ١

تحقق من العبارات الآتية لمعرفة ما إذا كانت صحيحة أو خاطئة.

	صحيحة	خاطئة
١. تسمح لبنة if ( ) then (إذا كان ( ) إذًا) للروبوت باتخاذ قرارات بناءً على صحة الشرط أو خطئه.	○	○
٢. يقيس مستشعر gyro المسافة التي قطعها الروبوت بالمليمتر.	○	○
٣. تعيد لبنة set drive heading to ( ) degrees (ضبط اتجاه القيادة إلى ( ) درجة) ضبط اتجاه الروبوت إلى اتجاهه الافتراضي.	○	○
٤. تُظهر لبنة drive heading in degrees (اتجاه القيادة بالدرجات) الاتجاه الحالي للروبوت.	○	○
٥. تقيس لبنة drive rotation in degrees (دوران المحرك بالدرجات) المسافة الإجمالية التي قطعها الروبوت على ساحة اللعب Grid Map (خريطة الشبكة).	○	○

تدريب ٢

يقوم مستشعر gyro بقياس اتجاه الروبوت بالدرجات.

١. ماذا يعني اتجاه 0 درجة؟

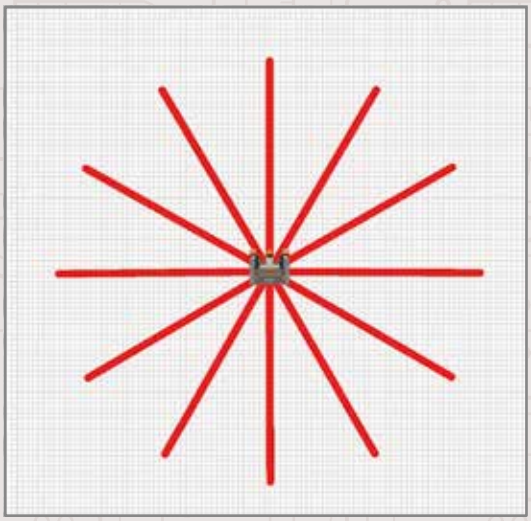
٢. إذا انعطف الروبوت يمينًا 90 درجة، فما هو اتجاهه؟

٣. إذا انعطف الروبوت يسارًا 180 درجة، فما هو اتجاهه؟

تدريب ٣

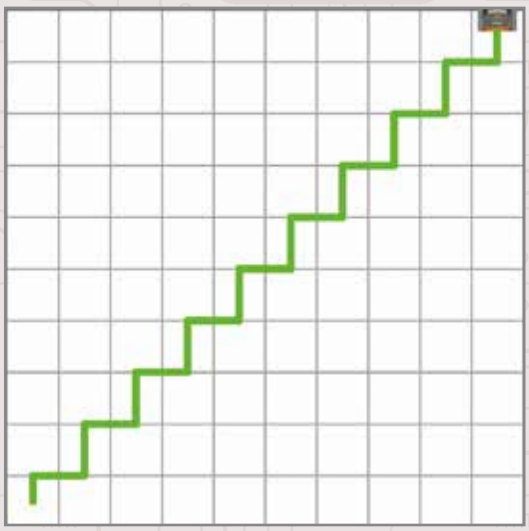
أنشئ مقطعًا برمجيًّا يستخدم لبنة drive heading in degrees (اتجاه القيادة بالدرجات) لرسم النمط الموضح في الصورة أدناه. اختر لون قلم أحمر وحرك القلم إلى الأسفل لبدء الرسم. استخدم حلقة تكرار ليكرر الروبوت نفس التسلسل عدة مرات.

حاول تعديل زاوية الدوران أو عدد التكرارات لترى كيف يُغير ذلك التصميم.



تدريب ٤

ستُنشئ مقطعًا برمجيًّا يُمكّن الروبوت الافتراضي من رسم شكل درج باستخدام لبنة drive rotation in degrees (دوران المحرك بالدرجات). اختر لون القلم الأخضر، وحرّك القلم إلى الأسفل لبدء الرسم. يجعل المقطع البرمجي الروبوت يتقدم إلى الأمام أولًا لمسافة قصيرة، ثم ينعطف يمينًا. داخل حلقة تكرار، يتناوب الروبوت بين التقدم والانعطاف، مستخدمًا لبنة turn to rotation ( ) degrees (الدوران إلى ( ) درجة) للتحكم في كل انعطاف بدقة.

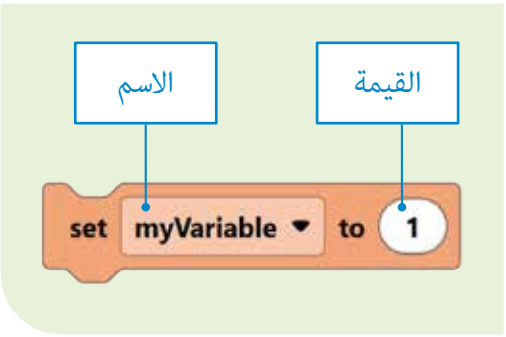


الدرس ٧-٢:

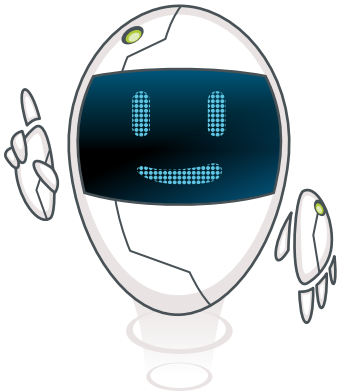
المتغيِّرات

المتغيِّر هو اسم تُخزَّن بداخله قيمة يريد الحاسوب الاحتفاظ بها أثناء تنفيذ البرنامج، ويعمل مثل حاوية يمكن وضع الأرقام أو الحروف أو البيانات المختلفة فيها.

يمكنك العثور على لبنات المتغيرات في فئة لبنات **Logic** (**المنطق**).



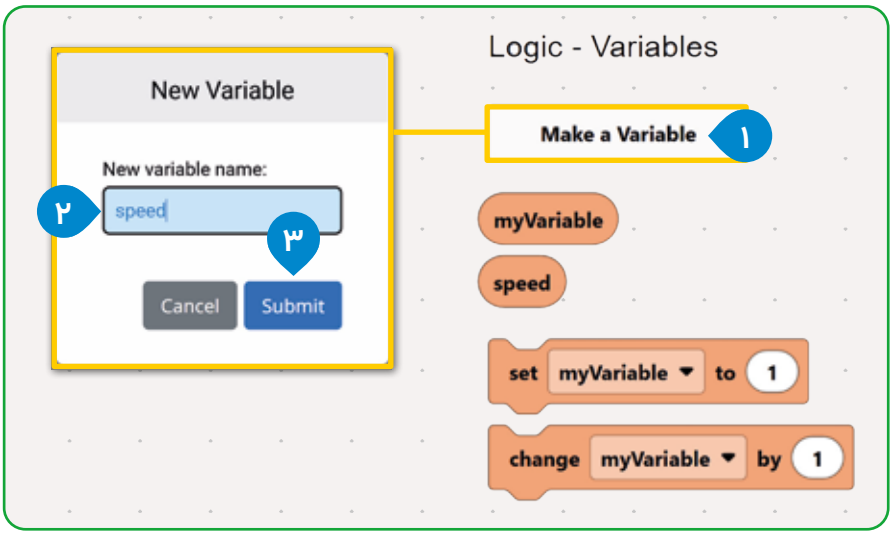
يحتوي تطبيق VEXcode VR على متغير محدد مسبقًا، يسمى (myVariable)، وهو متوفر للاستخدام في المقاطع البرمجية.



لإنشاء متغير:

> ١ من فئة لبنات **Logic** (المنطق)، انقر **Make a Variable** (إنشاء متغير).

> ٢ اكتب الاسم الذي تريده في مربع **New Variable** (متغير جديد) ، وليكن (Speed)، ٣ وانقر **Submit** (إرسال).



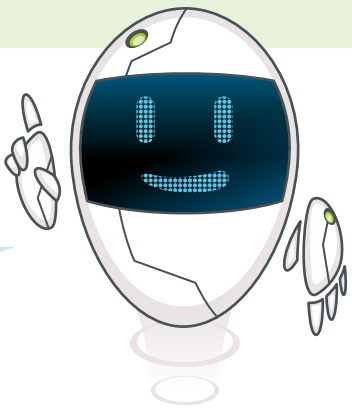
معلومة تقنية

يمكن أن تساعدك المتغيرات في متابعة النتائج أو المسافات أو أي بيانات متغيرة أخرى في مقطعك البرمجي.

لبنة MyVariable: تعرض القيمة الرقمية المخزنة داخل متغير ما، مما يسمح للمقاطع البرمجية بتتبع البيانات المتغيرة واستخدامها.

لبنة ( ) **to** ( ) **set** (**اضبط** ( ) **إلى** ( )): تُستخدم لتعيين أو تحديث قيمة المتغير.

لبنة ( ) **by** ( ) **change** (**تغيير** ( ) **بمقدار** ( )): تُستخدم لتغيير قيمة المتغير المخزنة بمقدار محدد.



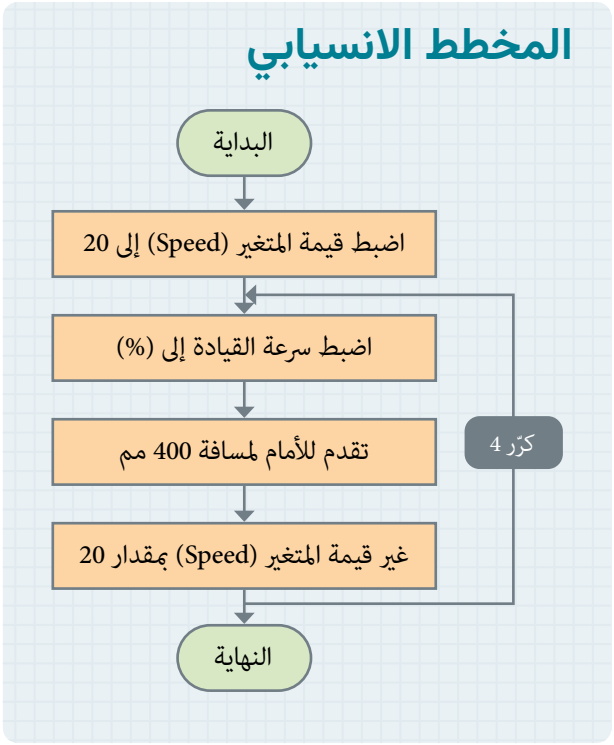
لا يمكن للمتغيرات أن تحتوي إلا على قيمة واحدة في كل مرة.

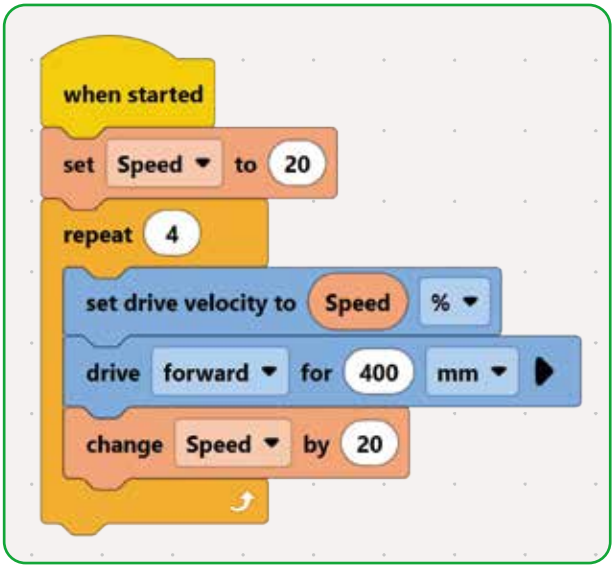
كيفية استخدام المتغيرات للتحكم في حركات الروبوت الافتراضي

يمكنك إنشاء متغيرات مختلفة للتحكم في السرعة ودرجات الدوران والمسافة التي يقطعها الروبوت في الواقع الافتراضي وغيرها.

ابدأ بإنشاء المقطع البرمجي الآتي واختبره على ساحة اللعب Grid Map (خريطة الشبكة). لاحظ كيف يؤثر هذا المقطع في حركة الروبوت؟ ثم غيّر القيمة في لبنة change (speed) by (20) (تغيير (السرعة) بمقدار (20)) . اختبر المقطع البرمجي مرة أخرى، وناقش كيف تؤثر التغييرات التي أجريتها على حركة الروبوت؟

المخطط الانسيابي





لبنة repeat until (كرّر حتى)

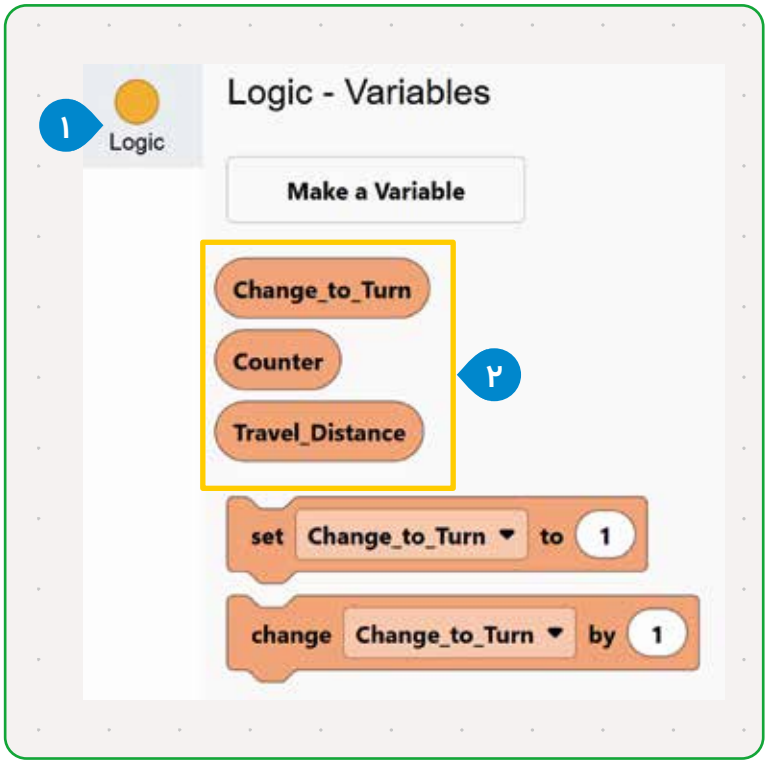
قد تحتاج إلى تنفيذ نفس سطور التعليمات البرمجية عدة مرات في مقطع برمجي، حيث تُوفر حلقات التكرار حلًّا فعالًا لهذه المشكلة، إذ تسمح بتكرار الأوامر. في هذا الدرس، ستستخدم لبنة **repeat until** (**كرّر حتى**) مع متغير تُنشئه.

لبنة **repeat until** (**كرّر حتى**): تكرر تنفيذ المقطع البرمجي الموجود بداخلها طالما أن الشرط خاطئ. عندما يصبح الشرط صحيحًا، يتوقف تنفيذ المقطع البرمجي.

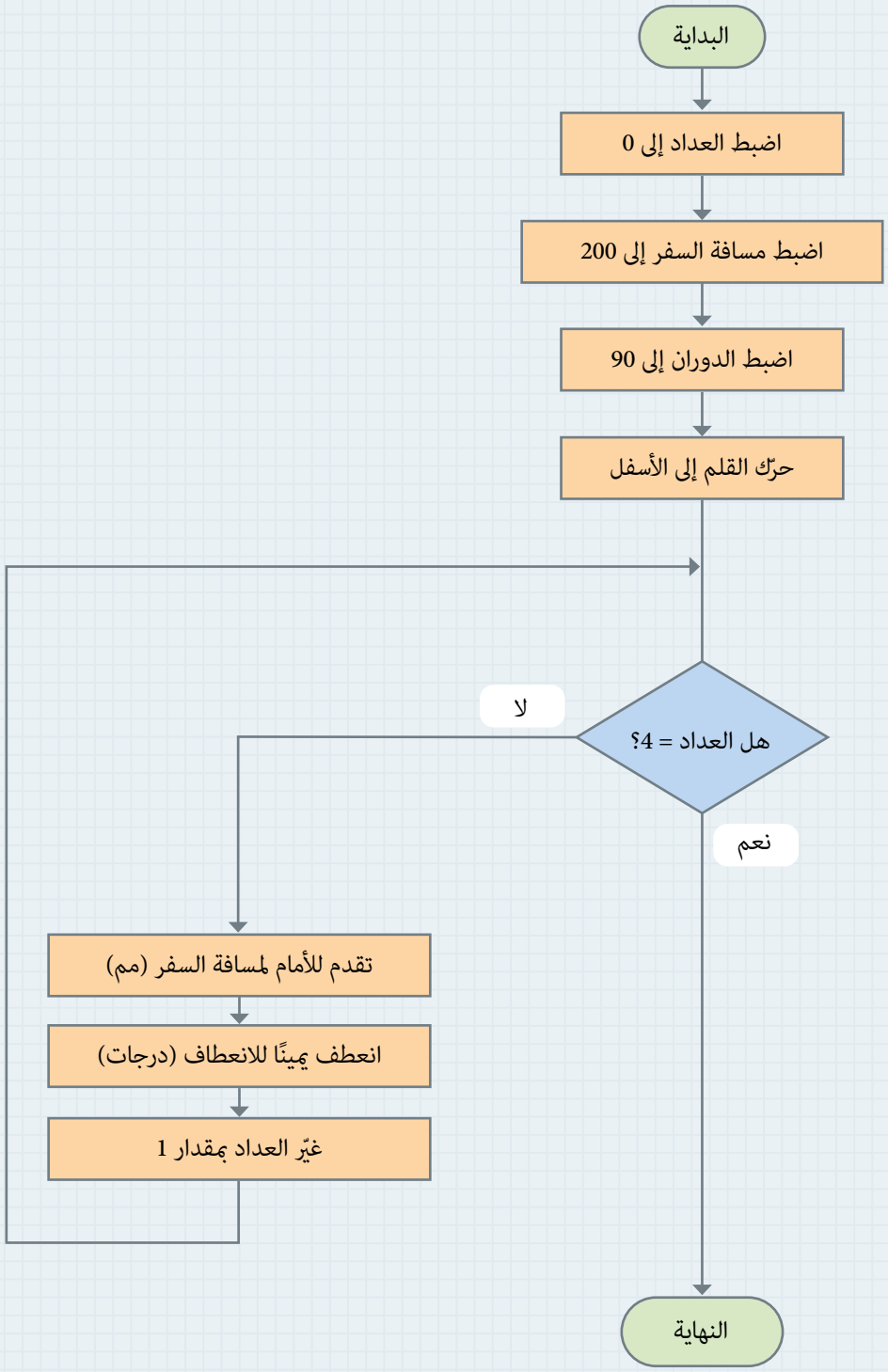
أنشئ مقطعًا برمجيًّا لجعل الروبوت الافتراضي يرسم مربعًا باستخدام المتغيرات.

لإنشاء ثلاثة متغيرات:

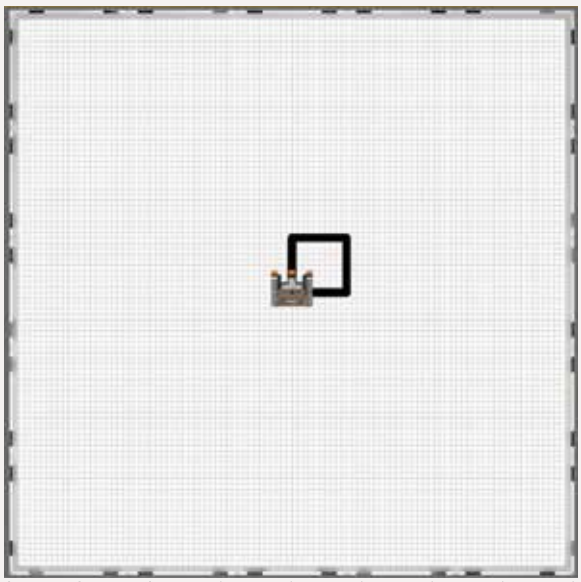
> ١ من فئة لبنات **Logic** (المنطق)، ٢ أنشئ ثلاثة **Variables** (متغيرات).



المخطط الانسيابي



أنشئ المقطع البرمجي واختبره على ساحة اللعب Art Canvas (لوحة فنية). ماذا يفعل الروبوت الافتراضي؟ ستستخدم في هذا المقطع البرمجي متغيرات للتحكم في مسافة الروبوت ودرجات دورانه. كيف يعرف الروبوت متى يتوقف عن تنفيذ الخطوات داخل حلقة التكرار؟



يرسم الروبوت حاليًا مربعًا صغيرًا. عدّل وأكمل المقطع البرمجي ليتمكن الروبوت الافتراضي من رسم مربعٍ أكبر فوق المربع الصغير.

لنطبق معًا

تدريب ١

تحقق من العبارات الآتية لمعرفة ما إذا كانت صحيحة أو خاطئة.

	صحيحة	خاطئة
١. يمكن للمتغير تخزين أكثر من قيمة في آنٍ واحد.	○	○
٢. تُعطي لبنة set ( ) to ( ) (اضبط ( ) إلى ( )) المتغير قيمته الابتدائية.	○	○
٣. تقوم لبنة change ( ) by ( ) (تغيير ( ) بمقدار ( )) بتحديث قيمة المتغير عن طريق إضافة أو طرح رقم.	○	○
٤. يستمر تشغيل لبنة repeat until (كرّر حتى) التكرار حتى يتحقق الشرط.	○	○
٥. يمكن أن تتضمن أسماء المتغيرات أرقامًا وشرطات سفلية في تطبيق VEXcode VR.	○	○

تدريب ٢

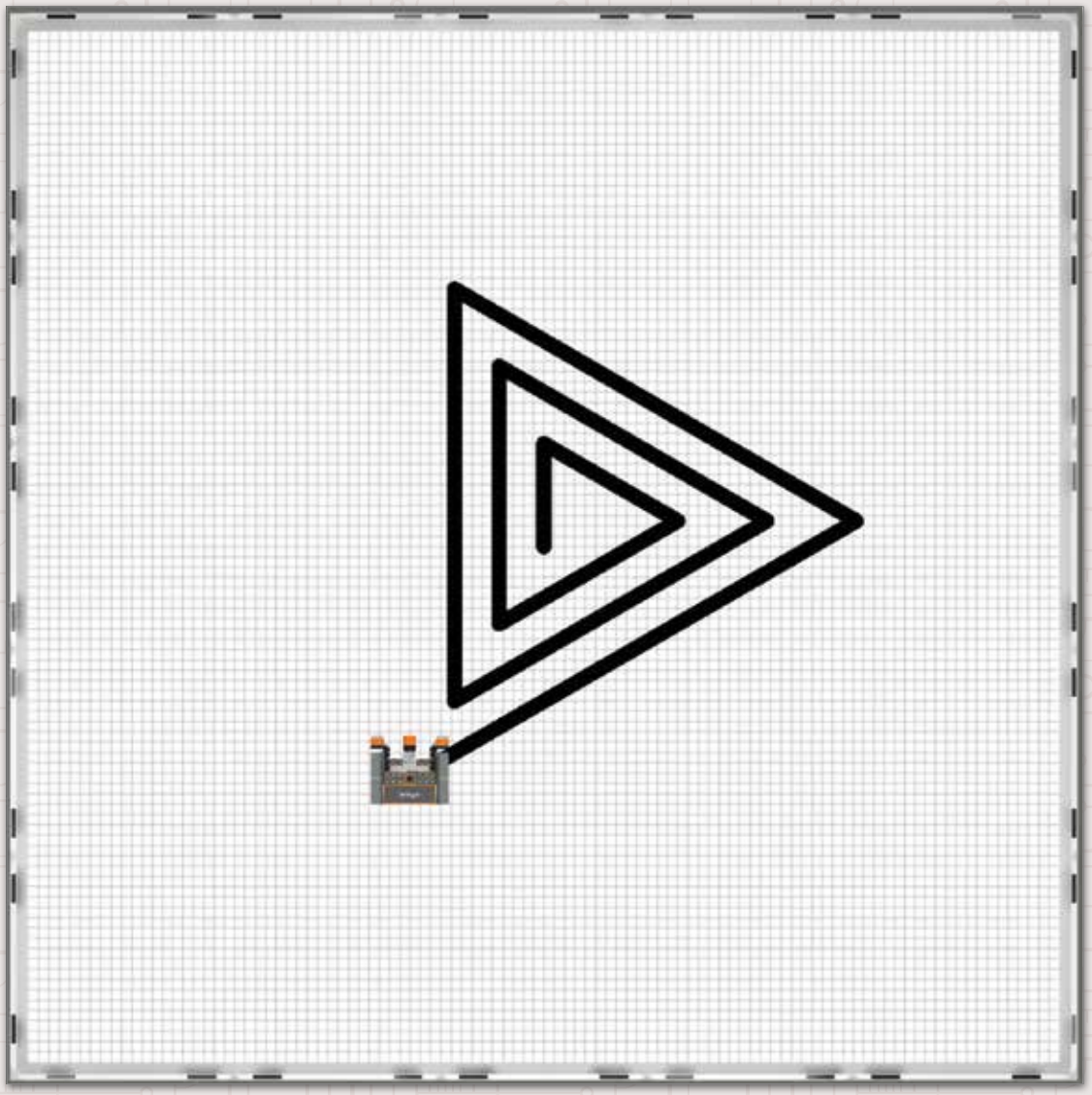
اشرح المقصود بالمتغير، ووضّح دوره وأهميته في البرمجة.

تدريب ٣

استخدم ساحة اللعب Art Canvas (لوحة فنية)، وأنشئ مقطعًا برمجيًّا يُمكّن الروبوت من تشكيل ثلاثة مثلثات لولبية، كما في الصورة المرفقة.

١. استخدم متغيرًا للتكرار.

٢. تذكّر أنه في كل مرة يرسم الروبوت ضلعًا جديدًا، يجب أن يكون أكبر من الضلع السابق.



الدرس ٨-٢:

المشروع

في هـذا المشروع، سـتقوم بإنشـاء مقطـع برمجـي لجعـل الروبوت الافتراضي يرسـم منزلًا بسـيطًا على ساحة اللعب Art Canvas (لوحة فنية).

١. أنشئ مشروع VEXcode VR جديدًا، واختر ساحة اللعب Art Canvas (لوحة فنية).

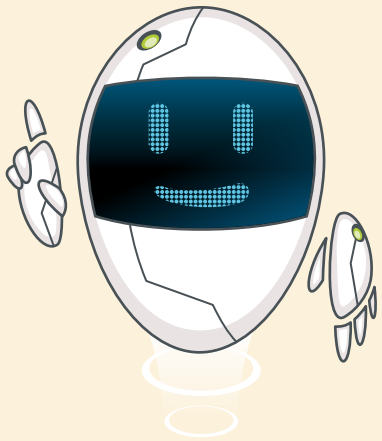
٢. لرسم جدران المنزل، استخدم لبنات القلم وقم بتغيير لون القلم بناءً على الموضع.

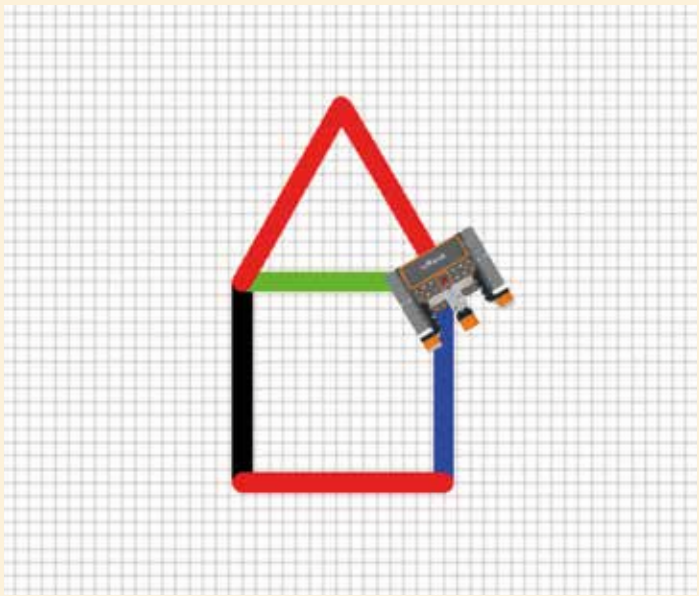
٣. استخدم لبنة drive heading in degrees (اتجاه القيادة بالدرجات) لإجراء منعطفات دقيقة.

٤. استخدم متغيرًا للتحكم بحجم جدران المنزل.

٥. لرسم السقف، استخدم زاوية الموضع بالدرجات لتعيين كل منعطف.

شـغّل المقطـع البرمجـي الذي أنشـأته، ثم اختبر حركـة الروبوت ولاحظ كيف ينفّذ الأوامر خطوةً بخطوة.





برامج أخرى

أوبن روبيرتا لاب (Open Roberta Lab)

بيئة برمجة قائمة على الويب تتيح للمتعلمين إنشاء المقاطع البرمجية الخاصة بهم للروبوت واختبارها وتشغيلها مباشرةً من خلال المتصفح. وهي تدعم أنواعًا متعددة من الروبوتات وتستخدم نظام ترميز مرئيًّا قائمًا على اللبنات؛ مما يجعل البرمجة سهلة وممتعة للمبتدئين.

مجموعة أدوات الروبوت الافتراضية (virtual robotics toolkit)

تطبيق محاكاة مهم يتيح للمتعلمين بناء وبرمجة واختبار روبوتات LEGO® في بيئة افتراضية ثلاثية الأبعاد. يعمل التطبيق بسلاسة مع نظام LEGO® MINDSTORMS ويتيح للمستخدمين ممارسة مفاهيم البرمجة والروبوتات دون الحاجة إلى مجموعة أدوات مادية.

في الختام

مخرجات التعلُّم

المخرج التعليمي	مستوى الإتقان	
	أتقن	لم أتقن
١. تعريف الروبوتات وأنواعها.		
٢. تحديد استخدامات الروبوتات في حياتنا اليومية.		
٣. تعريف الروبوتات الافتراضية.		
٤. التنقل في بيئة تطبيق VEXcode VR.		
٥. التعرف إلى المستشعرات الموجودة في الروبوت الافتراضي وشرح وظائفها.		
٦. استخدام الإحداثيات لتوجيه حركة الروبوت في ساحة اللعب.		
٧. استخدام لبنات التكرار لتَكرار الإجراءات بكفاءة في المقاطع البرمجية.		
٨. استخدام الجمل الشرطية لجعل الروبوت يستجيب لمختلف المواقف.		
٩. برمجة الروبوت للتحرك والرسم والتفاعل داخل ساحات اللعب.		
١٠. استخدام مستشعر الانعطاف في المقاطع البرمجية لتحسين دقة الحركة.		
١١. استخدام المتغيرات للتحكم في حركات الروبوت وتعديلها.		

المصطلحات

الروبوتات الثابتة / Fixed Robots

الروبوتات المتنقلة / Mobile Robots

الروبوتات الافتراضية / Virtual Robotics

ساحة اللعب / Playground

الكاميرا العلوية / Top camera

كاميرا التتبع / Chase camera

كاميرا الشخص الأول / First-Person camera

مستشعر / Sensor

مستشعر الاصطدام / Bumper sensor

مستشعر الانعطاف / Gyro sensor

اتجاه / Heading

المتغير / Variable

رقم الإيداع: ٢٠٢٥/١٠٢٤٣

طبع بمطبعة الزاوية الزرقاء ش.م.م - هاتف: ٢٤٨١٨٣٥٦-٩٦٨+

ISBN 978-99992-1-373-8

www.moe.gov.om