تطبيقات علوم الحاسوب
وتكنولوجية المعلومات
في تدريس العلوم

عمادة العلوم الدقيقة

أكاديمية القاسمي

2015-2016

تطوير التفكير الخوارزمي من خلال البرمجة المرئية

د. خالد أسعد* و د. مؤنس طيبي**

مقدمة:

يعتبر موضوع البرمجة وكتابة الحلول الخوارزمية من أهم المواضيع في علم الحاسوب. وبفضل مختصين ومحترفين في وضع الحلول الخوارزمية والبرمجة دخل الحاسوب في كافّة مجالات الحياة في المجتمع؛ وجراء ذلك ظهرت في الحواسيب التي في البيوت وأماكن العمل والمؤسسات والمدارس عشرات الآلاف من البرمجيات والتطبيقات التي تخدم الناس على اختلاف وظائفهم.

نظرا لاتساع استخدام التطبيقات المحوسبة في كافّة نواحي الحياة والمجتمع، بدأ الكثير من الباحثين بالاقتناع بضرورة إدخال تدريس علم الحاسوب والبرمجة في المدارس من جيل مبكّر ليشمل المرحلة الابتدائية أيضا. تعود أسباب هذا المطلب إلى أن تدريس مهارات البرمجة في سنّ مبكّرة يساهم في تخفيف صعوبات تعلّم البرمجة في المستقبل ويساعد على تنمية مهارات متعددة كحل المشكلات، والتفكير الخوارزمي، والعمل بالمشاركة وغيرها من المهارات الحيوية المطلوبة من كل الخريجين في هذا العصر التكنولوجي المتقدم.

يهدف هذا المقال إلى استعراض موضوع البرمجة في بيئات تعلم مرئية تناسب المبتدئين في المراحل الابتدائية والإعدادية. وقبل أن نستعرض الموضوع نقدم له بالتعريفات التالية:

ما هي الخوارزمية وما هو البرنامج؟

الخوارزمية هي كلمة مشتقة من اسم العالم الرياضي محمد بن موسى الخوارزمي الذي عاش في بغداد بين سنة 780هـ و 850هـ. الخوارزمية (Algorithm) عبارة عن مجموعة نهائية من الأوامر والخطوات التي تمثل أو تصف طريقة حل مشكلة معطاة أو محددة. أما البرنامج (Program) فهو سلسلة من الأوامر والتعليمات يزوّد بها الحاسوب لكي يؤدي مهمة معيّنة. هذه الأوامر والتعليمات هي خاصة ومتميزة وتسمى بمجموعها لغة برمجة (Programming Language). وهناك العشرات من لغات البرمجة المشهورة كلغة سي (C) وجافا (Java).

---

* رئيس قسم علوم الحاسوب في أكاديمية القاسمي

** رئيس قسم علوم الحاسوب في بيت بيرل ومحاضر في أكاديمية القاسمي

ما هو التفكير الخوارزمي وماذا يتطلب؟

هو القدرة على كتابة حلول خوارزمية لقضايا ومشكلات حياتية يحتاجها الإنسان. ينبغي على من يكتب الحلول الخوارزمية أن يحلل القضية الحياتية المطلوب برمجتها، وأن يعرف الأوامر البرمجية الملائمة لحل هذه القضية، وأن يملك مهارة دمجها في حل خوارزمي متسلسل (تفكير خوارزمي). فالمبرمج يجمع بين القدرة على تحليل القضايا الحياتية والفهم للغات البرمجة والقدرة على التفكير الخوارزمي لكتابة الحلول الخوارزمية.

البرمجة والخوارزميات المرئية

من أجل تعليم طلاب المدارس التفكير الخوارزمي والبرمجة في جيل مبكر يوصى بأن تكون مواضيع البرمجة قريبة من بيئة الطلاب وحياتهم، وأن تكون بيئات تعليم البرمجة بيئات تفاعلية (Interactive) ومرئية (Visual) في نفس الوقت. وهكذا يتفاعل الطالب مع الكود البرمجي (الشيفرة) وما ينتج عنه مباشرة، بحيث يكون الكود البرمجي محسوسا (مجسدا) ومرئيا (Visual) هو وما ينتج عن تنفيذه. وهكذا يبني الطالب الكود البرمجي وكأنه يركب مجسما وبأسلوب تفاعلي مباشر. والجدير بالذكر أن العديد من هذه البيئات تعتمد الألعاب لدمج التسلية والتحدي في آن واحد؛ الأمر الذي يسهل على الطالب ممارسة البرمجة واكتساب مهارات كتابة الحلول الخوارزمية من خلال اللعب.

تنمية التفكير الخوارزمي والبرمجة في بيئات عمل مرئية

في السنوات الأخيرة بدأت تظهر العديد من بيئات البرمجة المرئية (Visual Programming Environments) والتي من شأنها أن تخدم عملية تعليم البرمجة وتنمية التفكير الخوارزمي للمبتدئين وللأجيال الصغيرة. من المهم ذكره أن أغلبية هذه البيئات موجودة على شبكة الإنترنت كبيئات مفتوحة للبرمجة وللمشاركة، الشيء الذي يجعلها متوفرة للطلاب عبر أي حاسوب موصول بالشبكة. فيما يلي نستعرض ثلاثة أنواع من هذه البيئات وما يميز كل منها.

- بيئات تفاعلية لتنمية التفكير الخوارزمي بألعاب المنطق وحل الأحاجي
- بيئات برمجية وتفاعلية لتنمية التفكير الخوارزمي (السلحفاة + خان أكاديمي)
- بيئات برمجية وتفاعلية مرئية تجسد الكود البرمجي وما ينتج عنه لتنمية التفكير الخوارزمي

1. بيئات تفاعلية لتنمية التفكير الخوارزمي بألعاب المنطق وحل الأحاجي

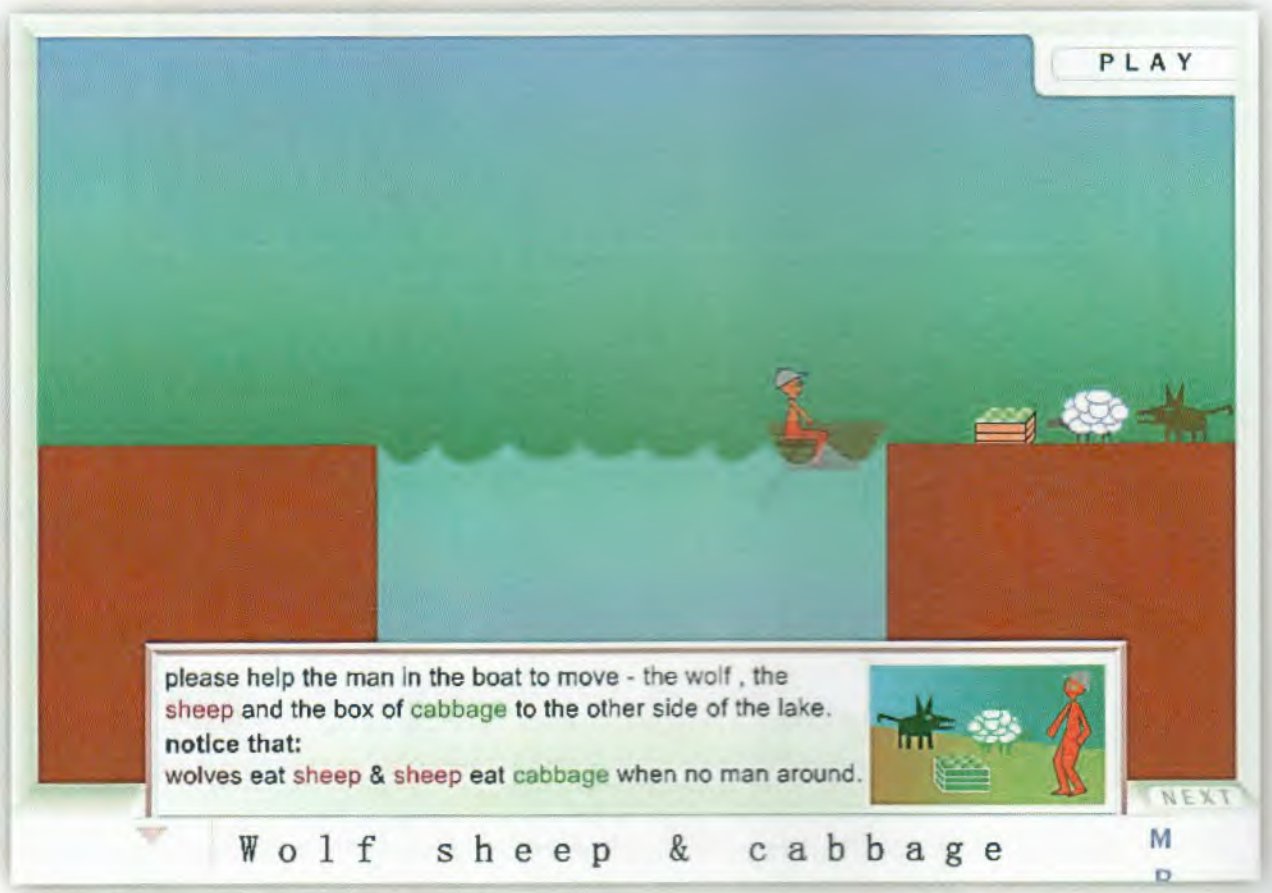
هذه البيئات من النوع الذي يساعد على تنمية التفكير الخوارزمي لدى الطالب عن طريق تجسيد بعض الأحاجي ليتمكن الطالب من حلّها بصورة تفاعليّة مرئيّة. هذه البيئات تجمع بين التسلية وبين التجربة والخطأ للوصول للحل. ومن المواقع التي تشمل مثل هذه البيئة نذكر موقع بالستلينا لألعاب المنطق والأحاجي. في هذا الموقع نجد العديد من ألعاب المنطق والأحاجي التي على الطالب أن يحلها بصورة تفاعلية. انظر صورة 1 للصفحة الأولى لموقع بالستلينا على الرابط التالي: www.plastelina.net

صورة 1 – الصفحة الأولى لموقع بالستلينا لألعاب المنطق www.plastelina.net

إليكم مثال لإحدى هذه الأحاجي في هذه البيئة:

أحجية الفلاح والذئب والخروف والملفوف

في هذه الأحجية مطلوب من الطالب أن يساعد الفلاح لعبور نهر ونقل ثلاثة أشياء هي **ذئب وخروف وملفوف** إلى الضفة الأخرى للنهر بقارب صغير لا يتسع إلى للفلاح وأحد هذه الأشياء. على الطالب أن يقترح طريقة يضمن فيها النقل الآمن بحيث لا يترك الملفوف مع الخروف ولا الخروف مع الذئب ولا يحمّل في القارب أكثر من شيء واحد. يستطيع الطالب أن يحل هذه الأحجية حلا تفاعليا في الحاسوب. انظر صورة 2 وصورة 3.



صورة 2- صفحة البداية لأحجية الفلاح والذئب والخروف والملفوف، قبل الضرب على زر PLAY

صورة 3- مرحلة من مراحل حل أحجية الفلاح والذئب والخروف والملفوف

وهكذا، فإن موقع بالستلينا يحوي العديد من الأحاجي وألعاب المنطق المجسدة من أجل أن يحلها الطالب بصورة تفاعلية. فمن ناحية يستمتع الطالب بحلها ومن ناحية أخرى يكتسب وينمي التفكير الخوارزمي لديه وهو ما يحتاجه لكتابة الحلول الخوارزمية والبرمجة.

2. بيئات برمجية وتفاعلية لتنمية التفكير الخوارزمي (السلحفاة + خان أكاديمي)

هذه البيئات من النوع الذي يساعد على تنمية التفكير الخوارزمي لدى الطالب عن طريق جعل عملية البرمجة عملية تفاعلية بحيث يقوم الطالب بكتابة الأوامر وتلقي النتائج أمامه مباشرة في شاشة الحاسوب. فيجمع فيها بين التحدي وبين أسلوب التجربة والخطأ للوصول للحل المطلوب. يفضّل في هذه الحالة أن يزود الطالب بكراس عمل يشتمل على مهام متدرجة ومهام فيها تحدي وأخرى مفتوحة كمشاريع عالية المستوى.

فيما يلي مثالين لهذه البيئات:

أ. البرمجة في أكاديمية السلاحف

الدخول إلى موقع أكاديمية السلاحف والبيئة البرمجية

http://turtleacademy.com/playground/en

هذه البيئة هي بيئة تفاعلية للرسم عن طريق البرمجة. تتمثل هذه البيئة بوجود سلحفاة على شكل مثلث تظهر في وسط صفحة الرسم تنتظر الأوامر والتعليمات من الطالب لكي ترسم. بحسب التعليمات التي يكتبها الطالب تقوم السلحفاة بالرسم في الصفحة مباشرة بعد كل أمر (انظر صورة 4 لشاشة صفحة الرسم).

صورة 4 – صفحة الرسم في وسطها سلحفاة على هيئة مثلث وفي أسفلها مربع لكتابة الأوامر

استعرض بعض الأمثلة والفعاليات في هذه البيئة:

مثال (1) هيا نرسم مستطيل

سلسلة الأوامر التالية (مقطع برنامج) ترشد السلحفاة كيف ترسم مستطيل:

البرنامج (سلسلة الأوامر)	عمل السلحفاة

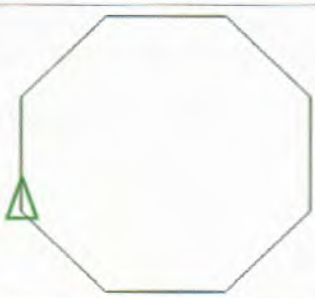
`> fd 100`
`> rt 90`
`> fd 60`
`> rt 90`
`> fd 100`
`> rt 90`
`> fd 60`
`> rt 90`	

شرح الأوامر

- الأمر (`fd 100`) يعني أن تتقدم السلحفاة للأمام وترسم خطًا بطول 100 نقطة والأمر مختصر كلمة **forward**.
- الأمر (`rt 90`) يعني أن تدور السلحفاة في مكانها لجهة اليمين بمقدار 90 درجة. والأمر مختصر كلمة **right**.
- تقوم السلحفاة بدورها بتنفيذ التعليمات واحدة تلو الأخرى مرتبة فنحصل على الرسمة المعروضة.

مثال (2) بأقل عدد من الأوامر هيا نرسم مضلعا منتظما مكونا من ثمانية أضلاع طول كل ضلع 50 نقطة

المربع هو مضلّع منتظم له أربعة أضلاع متساوية. المثمّن هو مضلع منتظم له ثمانية أضلاع متساوية. سلسلة الأوامر التالية ترشد السلحفاة كيف ترسم مثمّنا منتظما بتكرار أوامر بسيطة:

البرنامج (سلسلة الأوامر)	عمل السلحفاة
`repeat 8 [fd 70 7 rt 45]`	

شرح الأوامر

- الأمر (**fd 70**) يعني أن تتقدم السلحفاة للأمام وترسم خطًا بطول 70 نقطة .
- الأمر (**rt 45**) يعني أن تدور السلحفاة في مكانها لجهة اليمين بزاوية مقدارها 45 درجة.
- الأمر (**repeat 8 [..]**) يأمر السلحفاة أن تكرر تنفيذ ما في الأقواس 8 مرات لنحصل على الرسمة المعروضة.

مثال (3): استخدام أمر التكرار لكي نرسم وردة مكونة من 8 خطوط منتظمة.

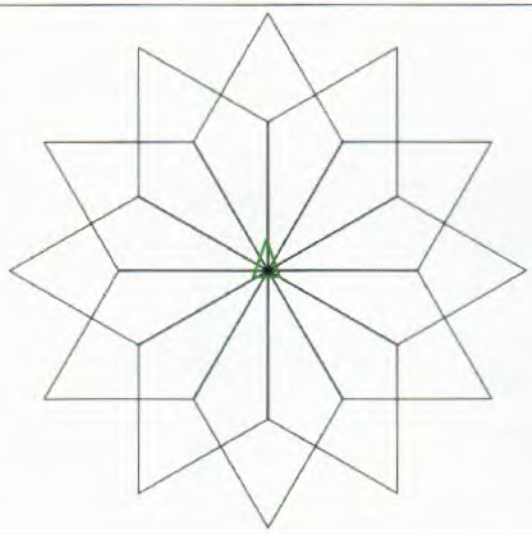
نريد أن نرسم خطوطا مستقيمة في 8 اتجاهات.

عمل السلحفاة	البرنامج (سلسلة الأوامر)
	`repeat 8 [fd 100 bk 100 rt 45]`

شرح البرنامج

- الأمر (**fd 100**) يعني أن تتقدم السلحفاة للأمام وترسم خطًا بطول 100 نقطة .
- الأمر (**rt 45**) يعني أن تدور السلحفاة في مكانها لجهة اليمين بزاوية مقدارها 45 درجة.
- الأمر (**bk 100**) يعني أن ترجع السلحفاة إلى الوراء وترسم خطًا بطول 100 نقطة والأمر مختصر الكلمة **backward**.
- الأمر (**repeat 8 [..]**) يأمر السلحفاة أن تكرر تنفيذ ما في الأقواس 8 مرات لنحصل على الرسمة المعروضة.

مثال (4): استخدام أمر التكرار لكي نرسم وردة مكونة من 8 متوازيات أضلاع منتظمة.

البرنامج (سلسلة الأوامر)	عمل السلحفاة
`repeat 12 [repeat 2 [fd 100 rt 60 fd 100 rt 120] rt 30]`	

شرح الأوامر

- الأمر (`repeat 2 [ .. ]`) الداخلي وما فيه يرسم متوازي أضلاع واحد
- الأمر (`repeat 12 [ .. ]`) الخارجي يكرر رسم متوازي الأضلاع 12 مرة مع دوران بسيط لليمين `rt 30` بعد كل مرة لنحصل على الرسمة المعروضة.

ب. البرمجة في "أكاديمية خان"

يمكن الدخول إلى "أكاديمية خان" والبيئة البرمجية التابعة لها من خلال الرابط التالي:

https://www.khanacademy.org/hour-of-code/hour-of-code-tutorial/p/intro-to-drawing

أو من خلال استخدام كود QR التالي:

من خلال هذه البيئة يستطيع الطالب أن يرسم أشكالاً هندسية أساسية ومركبة وكذلك أن يبرمج صورًا متحركة وأمورًا أخرى وذلك عن طريق إدخال أوامر (أو تعليمات) بلغة البرمجة والتي ستجعل الحاسوب يرسم هذه الأشكال وفق المقاييس التي يتم إدخالها.

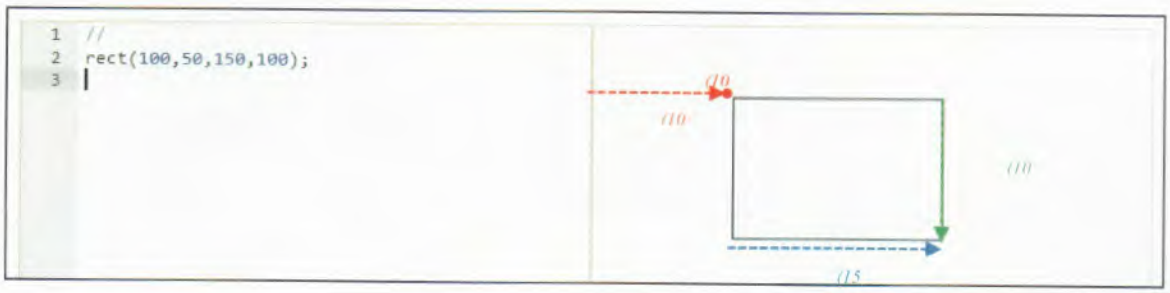
المركبات الأساسية للشاشة:

لنتمعن بعض الأمثلة في هذه البيئة:

مثال (1) هيا نرسم مستطيلا - الأمر rect

السطر التالي يؤدي إلى رسم مستطيل:

```
rect(100,50,150,100);
```



لنتمعن في مركبات هذا السطر.

- الكلمة **rect** (اختصار للكلمة rectangle وتعني مستطيل) وتعتبر أمرًا يستخدم لرسم مستطيل. نسميه أمرًا لأننا من خلاله نأمر الحاسوب بتنفيذ عمل محدّد – في هذا المثال العمل هو رسم مستطيل.
- كل أمر نكتبه ضمن هذه اللغة يجب أن ينتهي بفاصلة منقوطة (;) والتي تدل الحاسوب على نهاية الأمر.

الشرح: عندما نطلب من شخص أن يرسم مستطيلاً على ورقة بيضاء وضعت أمامه علينا أن نزوّده بأمور أخرى مثل: في أي نقطة عليه أن يبدأ برسم المستطيل وما هي أطوال أضلاعه. تعبّر الأرقام الموجودة داخل الأقواس في الأمر أعلاه عن هذه الأمور. الرقم الأول والثاني يحدّدان نقطة الانطلاق (الإحداثيات) لرسم المستطيل والرقمان 150 و 100 يحدّدان أطوال أضلاع المستطيل، أي العرض والارتفاع.

تحديد لون التعبئة للمستطيل – الأمر fill:

يستخدم الأمر fill لتحديد لون التعبئة. على سبيل المثال، عندما نرغب برسم مستطيل مع لون تعبئة أزرق نكتب الأمر fill ونحدد اللون الأزرق ومن ثم نكتب في السطر التالي أمرًا لرسم المستطيل (أو أي شكل آخر) مما يؤدي إلى رسم هذا المستطيل مع لون التعبئة الذي تمّ اختياره.

الشيفرة	النتيجة

`fill(255, 0, 0);`
`rect(50,50,200,150);`
`fill(0, 0, 255);`
`rect(75,75,150,100);`	

ملاحظة:

عند اختيار اللون بالفأرة نحصل على ثلاثة أرقام ضمن الأقواس. كل واحد من هذه الأرقام يمثل درجة لونيّة معيّنة من الألوان الرئيسية الثلاثة وهي الأحمر، الأخضر والأزرق. الرقم الأول يمثل درجة اللون الأحمر والثاني يمثل درجة اللون الأخضر والثالث يمثل درجة اللون الأزرق. وهكذا فاللون الذي اخترناه معرف كمزيج من ثلاثة ألوان بدرجات متنوعة. والنظام المسؤول عن تحقيق هذا المزيج ويدعى نظام الألوان RGB (اختصار الكلمات Red, Green, Blue).

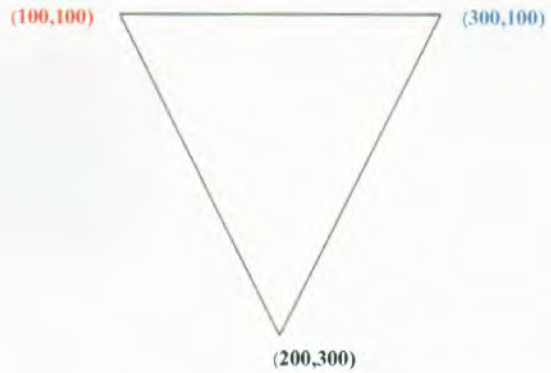
هناك بالطبع أوامر أخرى عديدة مثل لأمر إزالة لون التعبئة ;()noFill. وأمر تحديد لون خلفية الشاشة التي نرسم عليها ;()background والكثير من الأوامر التي يمكن التعرّف عليها من خلال النقر على الوصلة Documentation الظاهرة أسفل الجزء الذي نكتب فيه شيفرة البرنامج.

مثال (2) هيا نرسم مثلث – الأمر triangle

لرسم مثلث على الشاشة نستخدم الأمر التالي الذي يحدّد مواقع نقاط رؤوس المثلث:

```
triangle(100, 100, 300, 100, 200, 300);
```

في هذا المثال لدينا مواقع الرؤوس التالية: (100, 100)، (300, 100)، و (200, 300) كما هو موضح في الشكل التالي:



مثال:

الشيفرة	النتيجة

`fill(255, 0, 0);`
`triangle(100, 200, 300, 200, 200, 300);`
`fill(0, 0, 255);`
`triangle(100, 200, 300, 200, 200, 100);`	

مثال (3) هيا نرسم خط – الأمر line:

لرسم خط على الشاشة نستخدم الأمر التالي الذي يحدّد موقع النقطة الأولى وموقع النقطة الثانية اللتين سيمتد بينهما الخط:

```
line(100, 50, 200, 50);
```

مثال:

الشيفرة	النتيجة

`line(100, 50, 300, 50);`
`stroke(0, 255, 0);`
`strokeWeight(5);`
`line(100, 60, 300, 60);`
`stroke(0, 0, 255);`
`strokeWeight(15);`
`line(100, 80, 300, 80);`	

توضيح: في هذا المثال تمّ استخدام الأمر **stroke** لتحديد لون الخط والأمر **strokeWeight** لتحديد سمك الخط المرسوم كما هو مبيّن في النتيجة.

مثال (4) هيا نرسم دائرة – الأمر ellipse:

لرسم دائرة نستعين بالأمر ellipse الذي ينتظر إدخال أربع قيم: حيث تحدّد القيمتان الأولى والثانية موقع نقطة مركز الدائرة وأما القيمتان الثالثة والرابعة فتهدف كل منهما إلى تحديد العرض والارتفاع على التتالي.

مثال:

الشيفرة	النتيجة
`fill(0, 255, 0);` `ellipse(100, 100, 50, 50);`	

مثال:

```
fill(0, 255, 255);
ellipse(130, 150, 80, 80);

fill(255, 0, 255);
ellipse(180, 150, 80, 80);
```

```
fill(255, 0, 0);
ellipse(100, 100, 80, 80);

fill(0, 255, 0);
ellipse(150, 100, 80, 80);

fill(0, 0, 255);
ellipse(200, 100, 80, 80);
```

نرى من خلال هذه الأمثلة البسيطة طريقة كتابة الشيفرة (الكود البرمجي)، أي الأوامر التابعة للغة البرمجة، من أجل الحصول على ما نرغب. تختلف هذه الطريقة من البرمجة المبنيّة على كتابة الشيفرة عن تلك المعروضة في الفرع التالي والتي تتعامل مع تركيب مكونات البرنامج بشكل مرئي وتفاعلي أكثر، ولذا يفضّل أن تأتي في مرحلة لاحقة ضمن سيرورة التعلم لكي يكون التعامل معها أسهل للطالب وبعد أن اكتسب بعض الأسس في هذا المجال. من الجدير ذكره أن موقع "خان أكاديمي" يحتوي على شرح مفصل لجميع أوامر اللغة وكذلك العديد من الأمثلة التوضيحية والمشاريع، الأمر الذي يجعل عملية دراسة الموضوع أكثر سهولة وإمتاعا.

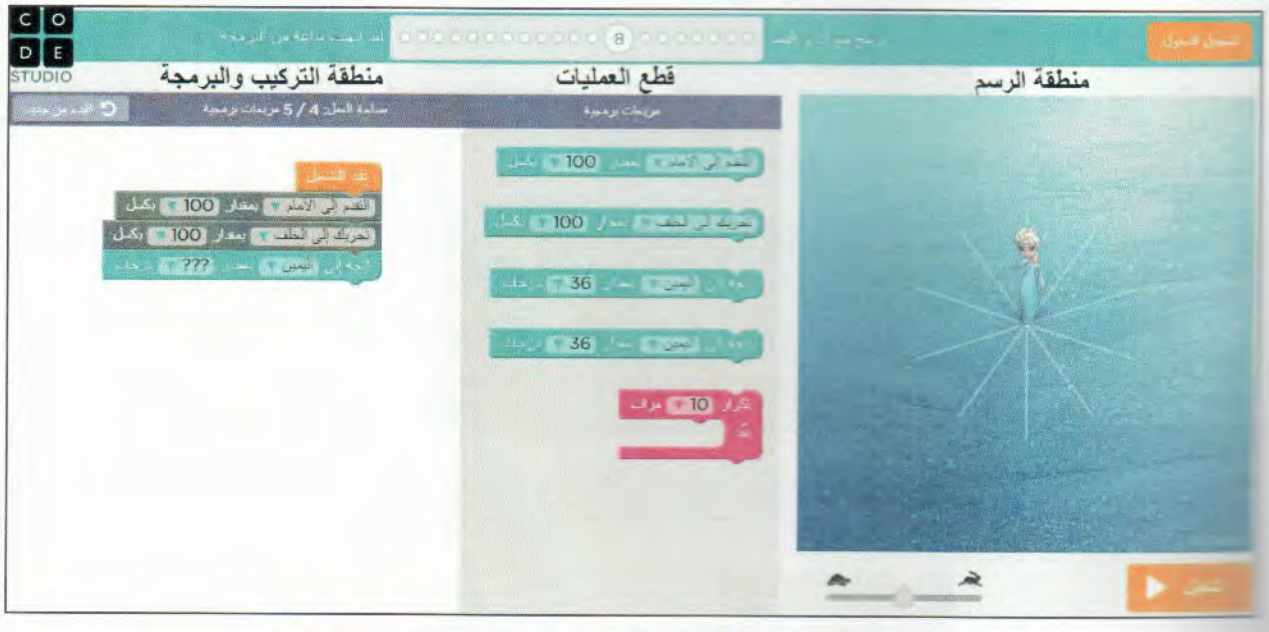
3. بيئات برمجيّة وتفاعليّة مرئيّة تجسّد الكود البرمجيّ وما ينتج عنه لتنمية التّفكير الخوارزميّ

هذه البيئات من النوع الذي يساعد على تنمية التفكير الخوارزمي لدى الطالب عن طريق جعل عملية البرمجة عملية تفاعلية، بحيث يقوم الطالب بحل المسائل عن طريق تركيب قطع تمثل عمليات وأوامر برمجية، ويشغلها ويتلقى النتائج أمامه مباشرة في شاشة الحاسوب. في مثل هذه البيئات يجمع الطالب بين التحدي وبين التجربة والخطأ للوصول للحل المطلوب. إن أغلب المواقع التي تشمل مثل هذه البيئات تحتوي على سلسلة من المهام المتدرّجة التي تحتوي على تحديات. إليك مثالا لمثل هذه البيئات:

أ) موقع البرمجة أرسم مع إلسا ELSA

الدخول إلى موقع بيئة البرمجة المرئية مع إلسا Elsa (انظر صورة 5)

https://studio.code.org/s/frozen/stage/1/puzzle/1

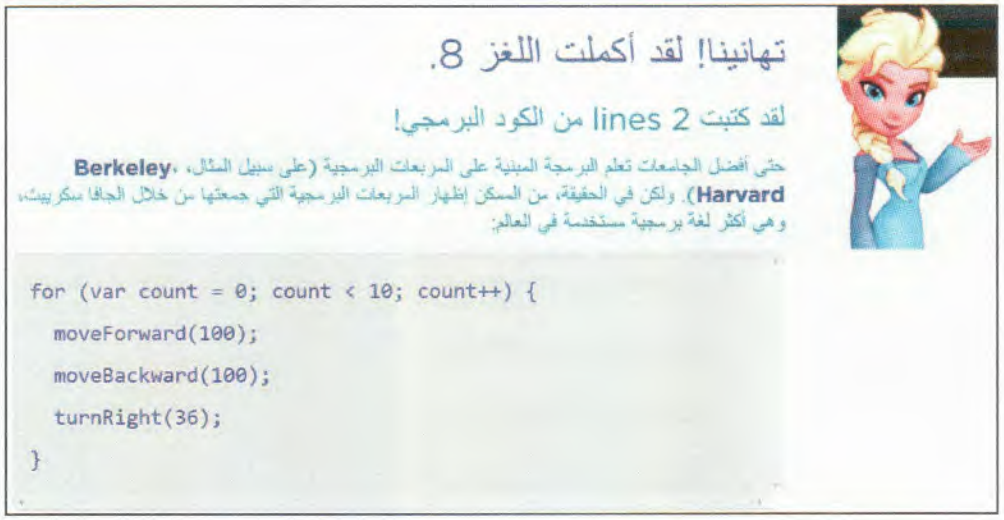


صورة 5: شاشة موقع البرمجة في البيئة التفاعلية المرئية – برمج مع إلسا

في هذه البيئة على الطالب أن يساعد "إلسا" الرسامة لرسم عدة أشكال ورسومات، وذلك بتحديد العمليات التي عليها أن تتبعها من أجل أن ترسم الأشكال المطلوبة. تظهر العمليات على شكل قطع ملونة. على الطالب أن يركب من هذه

القطع حلا خوارزميا (برنامجا) لرسم الأشكال المطلوبة من إلسا. بعد الانتهاء من التركيب يضرب على زر تشغيل لتبدأ إلسا بالرسم. على الطالب أن يجتاز عشرين مرحلة متدرجة الصعوبة وفي كل مرحلة يطلب منه أشكال جديدة.

من مميزات هذه البيئة أنها غنيّة بالتفاعل، وغنية بالإرشادات باللغة التي يختارها الطالب. فباختيار اللغة العربية مثلا تتحول البرمجة والقطع والشروح كلها إلى العربية كما يظهر في الصور 5 و 6 و 7. ومن مميزاتها أن الطالب يستطيع تحويل البرنامج الذي ركّبه من القطع البرمجية إلى برنامج حقيقي بأوامر مكتوبة. فبالضرب على زر **إظهار الكود البرمجي** يظهر البرنامج مكتوبا بلغة برمجة حقيقية. انظر صورة 6. بالإضافة إلى كل ما سلف، تتيح هذه البيئة للطالب إمكانية مشاركة مبرمجين آخرين في العالم بالكود البرمجي الذي بناه والرسمة التي حصل عليها.

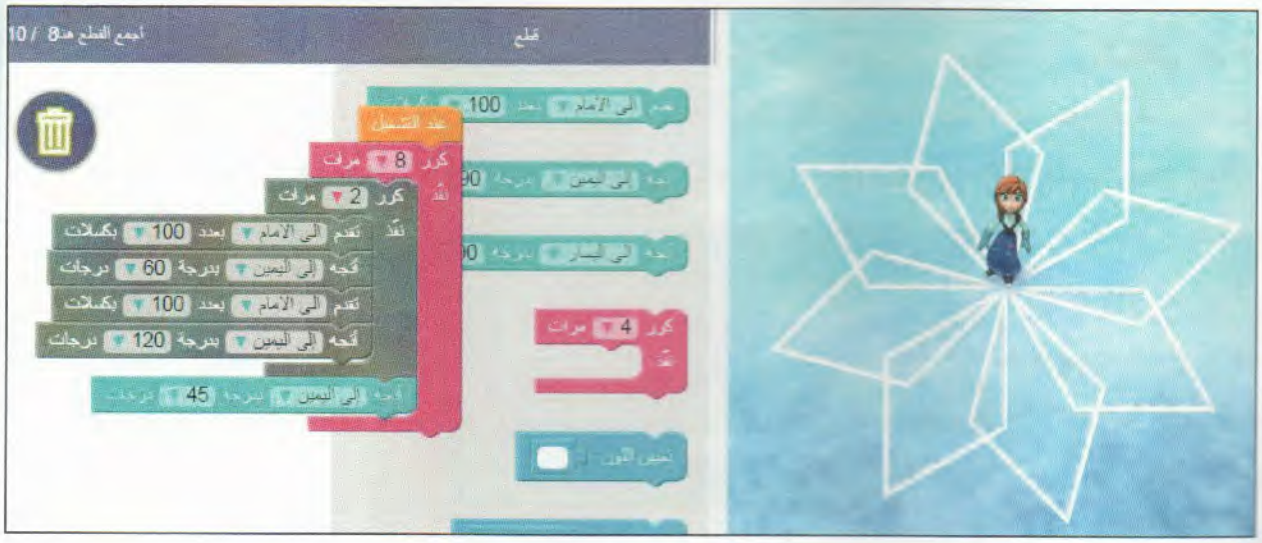


صورة 6: شاشة التهنئة بعد رسم أحد الأشكال في بيئة "برمج مع إلسا" مع إظهار الكود البرمجي الملائم لقطع البرنامج

في مثال (1) نعرض طريقة لبرمجة رسم أحد الأشكال بالقطع المرئية:

فيما يلي بعض الأمثلة:

مثال (1): هيا نرسم وردة مكونة من 8 متوازيات أضلاع منتظمة



شرح البرنامج

- نرسم ضلعين من متوازي الأضلاع بمساعدة تركيب أربع قطع

- نكررها مرتين لرسم شكل المتوازي الأضلاع كاملا باستخدام القطعة البرمجية **تكرار ؟؟؟ مرات**، هكذا:

- لتكملة الشكل المطلوب نكرر رسم 10 من متوازيات الأضلاع وفي كل مرة نقوم بالدوران لليمين بزاوية مقدارها
درجة (=360/10). وذلك باستخدام القطعة البرمجية **اتجه إلى اليمين بمقدار ؟؟؟ درجات.**
- من أجل التكرار 10 مرات نستخدم قطعة تكرار إضافية

بعد إتمام تركيب القطع، نضرب على زر تشغيل لفحص العمل، وفي حال النجاح نحصل على الإعلا

صورة 8 – إعلان النجاح في برمجة الشكل المطلوب وإتاحة الفرصة لمشا

إن الإعلان الذي يظهر في صورة 8 يعطي الطالب فرصة لكي يطبع الرسمة التي أنتجها

مبرمجين آخرين حول العالم الرسمة التي أنتجها والكود البرمجي الذي كوّنه. في المثال السابق مشاركة الرسمة تكون بالرابط التالي: https://studio.code.org/c/51803719. عند فتح هذا الرابط تعرض الرسمة في متصفح الإنترنت ويُعطى الآخرون فرصة لكي يستعيدوا الكود البرمجي وأن يعملوا على تكملته أو تطويره من جديد ويشاركوا عملهم للآخرين أيضا. إن هذه العملية التي تشمل مشاركة النتائج البرمجية عبر الشبكة وطرحها للغير من أجل تكملتها أو تطويرها تسمى بالـ(Remix). وقد شاعت هذه العملية في العديد من مواقع التطوير والبرمجة، وخاصة في بيئات ومواقع تعليم البرمجة للأولاد لما فيها من المنفعة.

أخيرا، من الجدير ذكره أن هناك تجارب عديدة ناجحة في إدخال عالم الروبوتات في تعليم البرمجة للأولاد في السنوات الأخيرة. في هذه التجارب يقوم الطالب باستخدام تطبيق على أجهزة الحاسوب اللوحي ليكتب كودا برمجيا ويتحكم بروبوت صغير عن بعد. من أمثلة هذه الروبوتات التعليمية نذكر الروبوت Vortex وزوج الروبوتات Dash & Dot (انظر صورة 9) وغيرها.

صورة 9 - زوج الروبوتات *Dash & Dot* لتعليم البرمجة للأولاد

استعرضنا في هذا المقال بيئات للبرمجة المرئية من الممكن دمجها بسهولة في عملية التعليم من خلال دروس الحاسوب لطلاب المدارس الابتدائية والإعدادية. من الواضح أن هذه البيئات تساعد على تطوير التفكير الخوارزمي بآليات سهلة وقريبة إلى عالم الطالب. من هنا، نوصي بإنشاء وحدات تعليمية وكراسات عمل ملائمة للعديد من هذه البيئات ونوصي بوضع مناهج تدريس لدمج مثل هذه البيئات في منهاج تدريس علوم الحاسوب.