

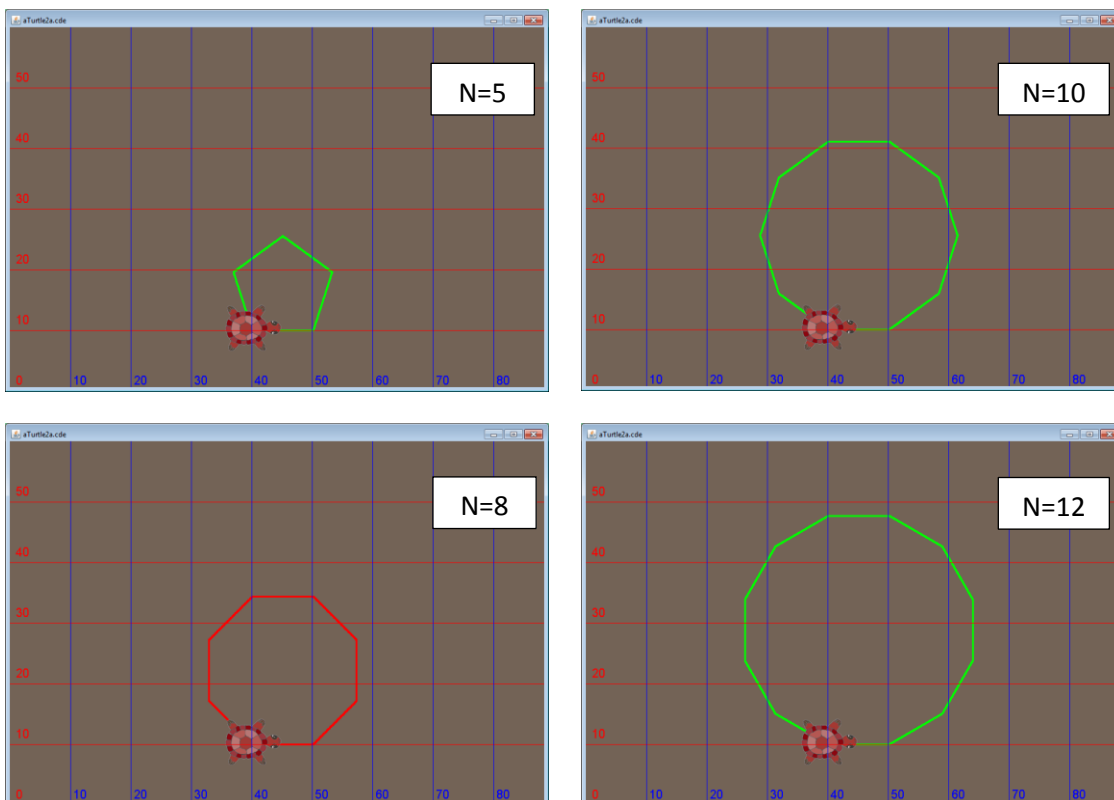
# Turtle Graphics with the WBEngine

## Rationale

In future research, we aim to ask the question “Does story-writing-coding (SWC) *explicitly* help children to learn to code?” To do this, we need a “non-SWC” activity, and we have chosen Turtle Graphics. The research design will randomly assign children in a class to the SWC and non-SWC activities and measure their performance. But that’s in the future. Nevertheless, it may be interesting for you to have a look at this approach, since it provides you with a non-SWC approach to teaching programming, more aligned with mathematics. Three examples are provided in the current code-drop; both use a “plan-view” rather than the usual WeeBee world.

<b>aTurtle.cde</b>	Demonstrates moving forward, rotation, pen-down (with optional color) and pen-up.
<b>aTurtleHouse.cde</b>	Two turtles collaborate to draw a house; one draws the walls, the second draws the roof. We feel that collaboration is an interesting application of the sequential/concurrent affordance of the WeeBee Engine.
<b>aTurtle2a.cde</b>	Draws a regular geometrical shape, where the user is asked for the number of sides to be drawn

Here’s some geometric shapes drawn by the turtle



All sorts of enquiry-based activities could be crafted with this code. Children could discover: (i) If the number of lines in the shape is divisible by 2, then the top and bottom lines are parallel. (ii) If the number of lines in the shape is divisible by 4, then the top and bottom lines are parallel, but so are the right and left lines. (iii) The larger the number of sides, the larger the shape. Children could calculate the perimeter lengths, and make actual measurements.

## Current Functionality

Please note, this has not been rigorously tested.

---

<b>turtle1.moveForward(20);</b>	moves forwards 20 units, in which ever direction the turtle is heading
---------------------------------	--

---

<b>turtle1.rotate(45);</b>	rotates 45-degrees from the current heading. Currently negative angle are anti-clockwise. This may need to be changed to conform to “standard” approaches.
----------------------------	--

---

<b>turtle1.setPenDown(true);</b>	enables drawing (default <b>red</b> )
----------------------------------	---------------------------------------

---

<b>turtle1.setPenDown(true,color);</b>	enables drawing with a specified color which uses the keywords <b>red blue black cyan orange yellow green</b>
--	---

---

<b>turtle1.setPenDown(false);</b>	pen is not down, so there is no drawing.
-----------------------------------	--

---

Please note the turtles have full access to the WeeBeeCode methods. The most useful of these could be **turtle1.says(<text>);** and **turtle1.thinks(<text>);**