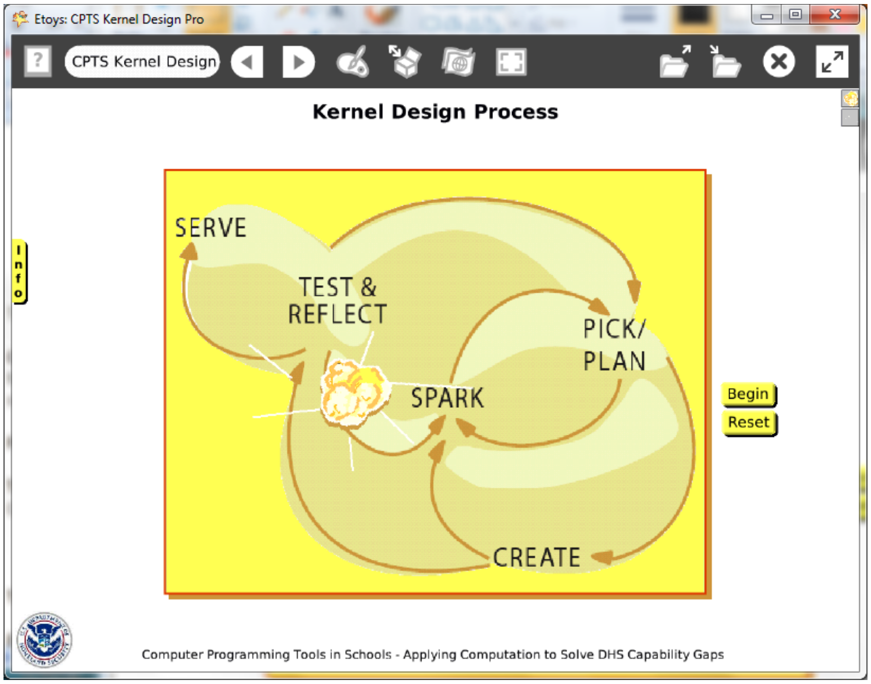
Project: Kernel Design Process
Difficulty: Level 1


Etoys: CPTS Kernel Design Pro
CPTS Kernel Design
Kernel Design Process
Info
SERVE
TEST & REFLECT
PICK/ PLAN
SPARK
CREATE
Begin
Reset
Computer Programming Tools in Schools - Applying Computation to Solve DHS Capability Gaps

Time: two 45 minute labs

Challenge:

Make a project that demonstrates the growth of an idea as it passes through the stages of a design process. This project uses a kernel of popcorn but any number of objects that grow could be used to good effect in this project.

Programming:

Conditional statements are used to increase the kernel's size when it is over the location of a word in the graphic. The sparkle uses a conditional test to show if the heading is divisible by 25 and to hide otherwise.

This project uses scripting category tiles to control when a script is ticking or paused. There are no Quick Guides for the scripting tiles; experiment with them and their effects. There *are* help balloons for each tile.

Things you'll need to know:

Quick Guides

- Paint Tools/ All
- Halo Handles/All
- Supplies: Text, Add a New Flap
- Object Catalog: Digital Images
- Script Tiles: Forward by, Turn by, Bounce, X and Y Tiles, Scale Factor, Hide and Show, Tests Category
- Menus: Button Fires a Script

Things to think about:

- What advantages and disadvantages are involved in the use of scripts that specify that the kernel increase in size at specific x and y locations? The art of programming is in knowing more than one solution and choosing among them for the effect desired and an efficient process to make it happen.

Extensions:

- Make a script that causes the kernel to bounce against the edge of the playfield and to increase in size with each bounce. Compare the scripts, the effect and the effort involved in making the scripts.
- Is there an easier way to show the development of the idea? Is it as effective?
- Make a script that uses the "obtrudes" tile in a test to affect the change in the kernel's size.
- Each kind of script has advantages and disadvantages. Discuss

NETS for Students:

http://www.iste.org/standards/nets-for-students/nets-student-standards-2007.aspx

1. Creativity and Innovation: a, b, c,
2. Communication and Collaboration: a, b
3. Research and Information Fluency: a, b
4. Critical Thinking, Problem Solving, and Decision Making: a, b
5. Digital Citizenship: b

6. Technology Operations and Concepts: a, b

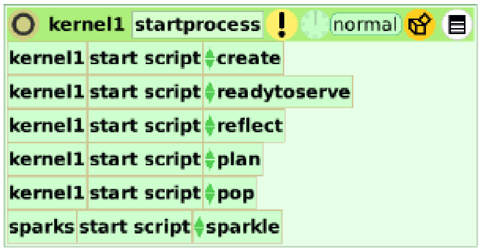
CSTA:
CSTA Level II: Objectives and Outline
http://csta.acm.org/Curriculum/sub/CurrFiles/L2-Objectives-and-Outlines.pdf
Level II objectives for middle school students are furthered through studying a programming language well enough that the student is proficient with it. Whether the language is Etoys, StarLogo TNG, or Scratch, it is the ability to use the language to express ideas that is valuable. A student skillful enough to use *any* programming language to express ideas, solve problems, model behaviors, simulate data, or to educate or entertain is an entitled person in today's society.
Topics of particular note are:
Topic 2: Problem Solving
Topic 6: Connections between Mathematics and Computer Science
Topic 11: Programming Languages
Topic 13: Multimedia

Common Core Standards Mathematics:
http://www.corestandards.org/the-standards/mathematics
6. EE.2, 6.EE.2c, 6.EE.6, 6.NS.6
8. G.3

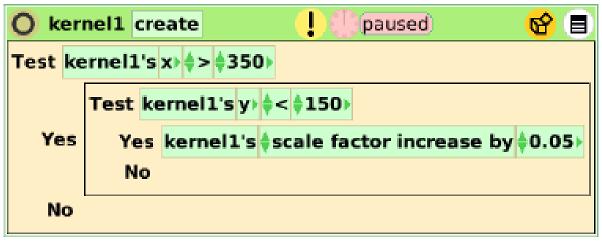
Teacher Notes:
Materials: The internet contains a large number of images/examples of the design process as imagined by many designers. A quick survey and discussion will help students start to visualize new ideas and ways to show the process.

Comments: Objects - Scripts – Decisions
Give students time to develop their idea of a design process, it could be to invent a machine or develop a new procedure in topics ranging from medicine to sports.

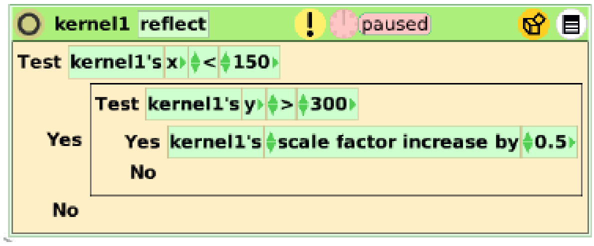
Example Scripts:



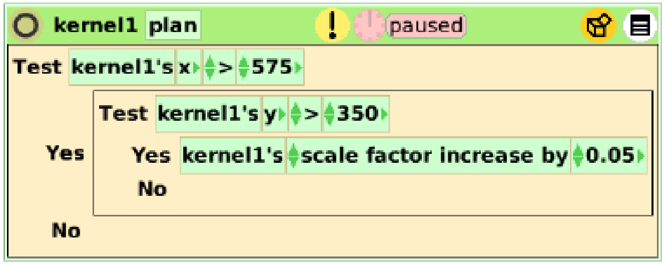
```
kernel1 startprocess ! normal
kernel1 start script create
kernel1 start script readytoserve
kernel1 start script reflect
kernel1 start script plan
kernel1 start script pop
sparks start script sparkle
```



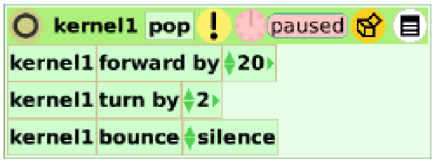
```
kernel1 create ! paused
Test kernel1's x > 350
Yes   Test kernel1's y < 150
      Yes kernel1's scale factor increase by 0.05
      No
No
```



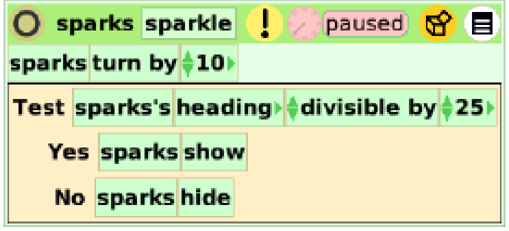
```
kernel1 reflect ! paused
Test kernel1's x < 150
Yes   Test kernel1's y > 300
      Yes kernel1's scale factor increase by 0.5
      No
No
```



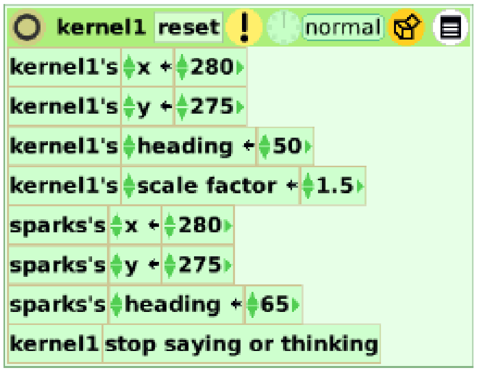
```
kernel1 plan ! paused
Test kernel1's x > 575
Yes   Test kernel1's y > 350
      Yes kernel1's scale factor increase by 0.05
      No
No
```



```
kernel1 pop ! paused
kernel1 forward by 20
kernel1 turn by 2
kernel1 bounce silence
```



```
sparks sparkle ! paused
sparks turn by 10
Test sparks's heading divisible by 25
  Yes sparks show
  No sparks hide
```



```
kernel1 reset ! normal
kernel1's x ← 280
kernel1's y ← 275
kernel1's heading ← 50
kernel1's scale factor ← 1.5
sparks's x ← 280
sparks's y ← 275
sparks's heading ← 65
kernel1 stop saying or thinking
```

Student Notes:
None provided.