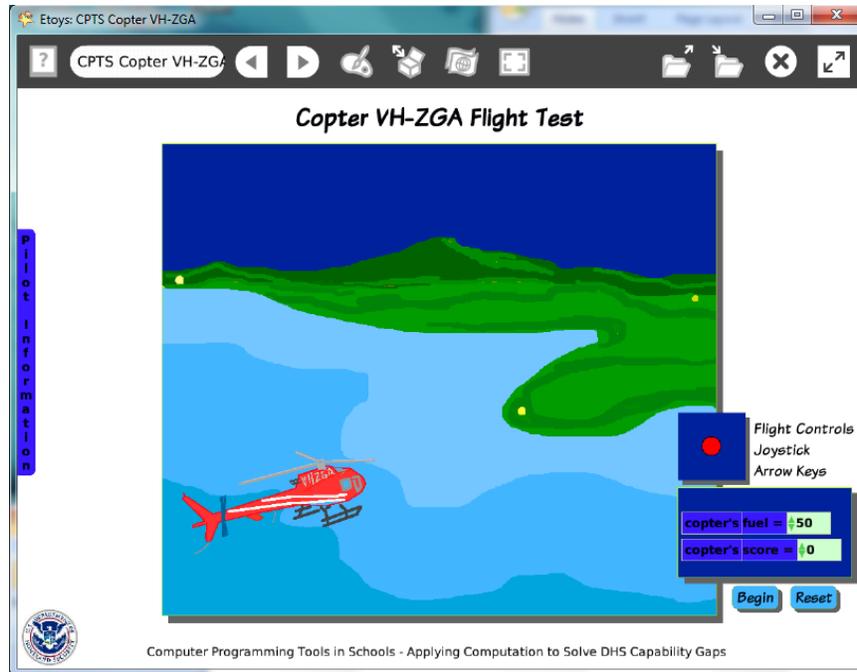


## Project: Copter VH-ZGA Flight Test Difficulty: Level 2



Time: Five 45 minute labs

**Challenge:** Control the motion of an aircraft to simulate a pilot flying from point to point. Use a joystick to control up down and left right. Use the keyboard arrows to change the scale factor of the aircraft so when it is going up and down on the screen the scale factor decreases and increases modeling our perception of near and far. Use two views of the aircraft and control which view shows using the left and right arrow keys. Create two variables: fuel and score to give the game verisimilitude.

### Programming:

This project uses nested conditional statements, variables and random numbers. There are numerous scripts. Give students time to write a scenario describing the game plan, objects in the game and scripts to control them. They should include a reset script and one to start the game.

Students should research aircraft images and select two or more views of the same aircraft.

NB: There are useful commands in the Halo's white menu including commands to keep objects in specific layers, resist being deleted, resist being picked up, and be locked. This project makes use of the Scripting Category of tiles to start, pause, and stop scripts based on events while the project is running. There are no Quick Guides for scripting tiles. There *are* Help balloons that are useful.

### **Things you'll need to know:**

#### Quick Guides

- Paint Tools/ All
- Halo Handles/All
- Supplies: Object Catalog, Text, Joystick Control, Players Tool, Add a New Flap
- Object Catalog: Digital Images
- Script Tiles: Forward by, X and Y Tiles, Heading, Scale Factor, Hide and Show, World Input, Random Numbers, Tests Category, Two Color Test
- Menus: Normal Ticking, Viewer Icons, Set, Scriptor Icons Set, Grab Me Reveal Me, Watchers, Bring to Front, Drop Shadow

### **Things to think about:**

- How responsive should the copter be? The size of the joystick's square changes the level of difficulty. Should more points accrue for more distant city locations?
- Should points accrue just once for each event or continuously while the aircraft hovers?

### **Extensions:**

- Control the location of the city points with random x and y values.
- Change the size of the city markers to make hovering easier or harder.

### **NETS for Students:**

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

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

### **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. NS.5, 6, 7, 8

6. EE.2

### **Teacher Notes:**

Materials: Give students time to research aircraft images and to select a pair of images facing in opposite directions. This example has left/right facing images but front and back view would work as well. Hand drawing the image by tracing the photo gives students time to think about the parts of the machine

### Comments: Objects - Scripts – Decisions

This topic may be of interest to students who want to simulate actions with real world applications. Discuss what elements can be modeled successfully, what can be alluded to, and what cannot be shown realistically. Flight simulators in games are becoming very sophisticated. See for example the iPad app Apache HD. Look at Google Earth's Flight Simulator.

### Example Scripts:

The 'copter landingzone' script starts with a 'ticking' timer. It contains two main decision blocks. The first block tests 'copter's color sees color'. If 'Yes', it tests 'copter's fuel > 0'. If 'Yes', it shows 'safe show' and 'copter pause script scorelandings'. If 'No', it tests 'copter's fuel < 0'. If 'Yes', it shows 'crash show', decreases 'copter's score' by 0, and then 'copter pause script directionchange' and 'copter pause script fueluse'. If 'No', it shows 'zone show'.

The 'copter scorelandings' script is 'paused'. It contains three decision blocks based on 'copter's color sees color'. If 'Yes' and color is red, 'copter's score increase by 100'. If 'Yes' and color is yellow, 'copter's score increase by 200'. If 'Yes' and color is green, 'copter's score increase by 300'. All 'No' paths do nothing.

The 'copter fueluse' script is 'paused'. It starts with 'copter's fuel decrease by 1'. It then tests 'copter's fuel < 10'. If 'Yes', it changes 'warninglight's color' to red, shows 'warning show', and 'zone show'. If 'No', it does nothing.

The 'copter begin' script is 'normal'. It contains four 'start script' blocks: 'copter start script fueluse', 'copter start script scorelandings', 'copter start script directionchange', and 'copter start script landingzone'.

```
copter forward by 1
copter's x increase by Joystick's left right
copter's y increase by Joystick's up down
Test copter's last keystroke = <right>
  Yes copter look like right
  copter's last keystroke ← q
  No
Test copter's last keystroke = <left>
  Yes copter look like left
  copter's last keystroke ← q
  No
Test copter's last keystroke = <up>
  Yes copter's scale factor decrease by 0.01
  No
Test copter's last keystroke = <down>
  Yes copter's scale factor increase by 0.01
  No
```

```
copter's scale factor ← 0.50
copter's fuel ← 50
copter's x ← 75 + random (700)
copter's y ← 75 + random (500)
copter's score ← 0
warning hide
warninglight's color ← color
zone's x ← 100 + random (500)
zone's y ← 460 - random (400)
zone hide
safe hide
crash hide
```

**Student Notes:**

None provided