Snap! Project helicopter

UBMS Snap! Programming Summer 2023

October 31, 2023

Project helicopter

Figure 1: BPMN diagram for the helicopter navigation

- Building the minimal helicopter (in class):
  1. Background = variable
  2. Costumes = helicopter, base structure
  3. Sprites = helicopter, landing pad
  4. Navigation = use arrow keys
  5. Gravity = when helicopter not operated, it moves down
- Optional improvements (home assignment = bonus points):

1. Costumes/navigation: Helicopter points in the direction it is flying
2. Navigation: physical gravity = helicopter accelerates down
3. Costumes: Indication that the helicopter engine is running (could be spinning fan, blinking lights, engine sounds)
4. Background: Moving clouds
5. Background: Landing pad and starting place separated (helicopter does not just sit in mid-air at the start).
6. Navigation/costumes: if helicopter falls below ground level or runs into rock, it explodes!
7. Background: Title screen with instructions, final screen with 'Yay'

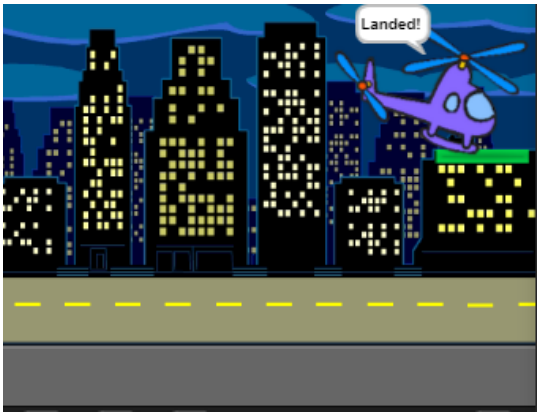
Minimal helicopter

Link to the finished project.

Link to the description (PDF): tinyurl.com/ProjectHelicopter.

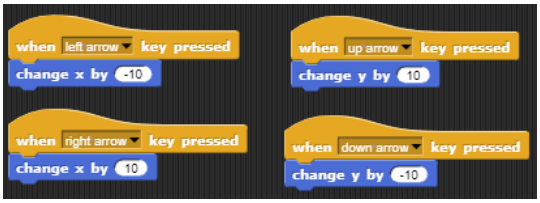
1. Background: standard background `Night city with street`.
2. Costumes: download the `helicopter` costume from the library.
3. Sprites:
   - The helipad is the green `paddle` costume from the standard library. It is placed on a house on the right side of the stage:

   - The exact position is immaterial since we control successful landing with the `touching Sprite` command: this must be in a `forever` loop so that the loop block action is continuously checked:

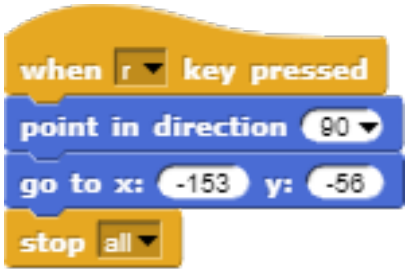


4. Navigation:

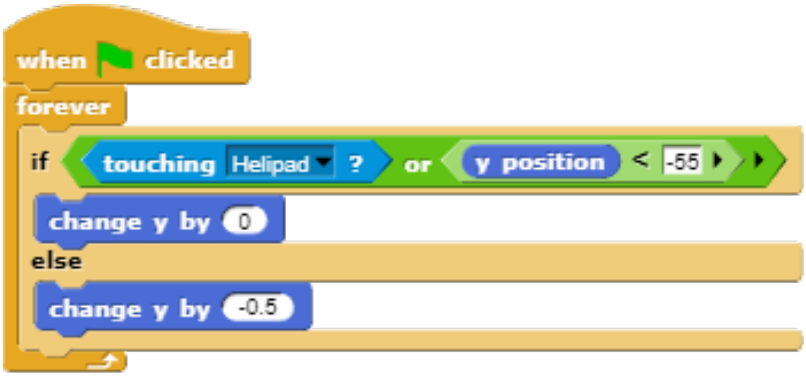
   - Arrow keys change x and y coordinates by 10 steps per activation:



   - The default starting position on street is (-153,-56). The reset script resets to that position:



5. Gravity: whenever we are not on the landing pad or on the ground, we want to lose height. This means that we `forever` reduce `y` by a small amount unless we're on the landing pad or `y` is ground level (-55 in my case). This got to be a *concurrent* script:



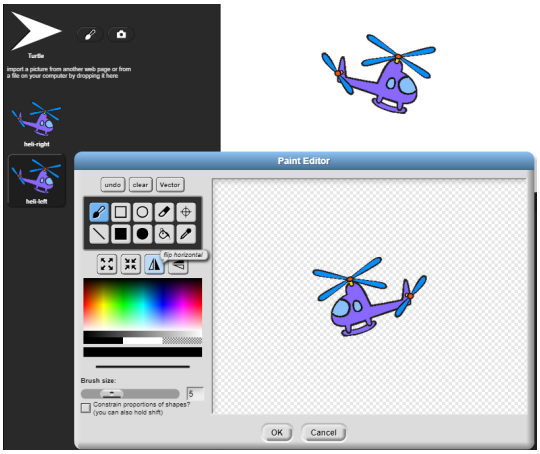
Improved helicopter

Link to the (unfinished) project.

1. Costumes/navigation: helicopter points in the direction it is flying. We need to turn it around and allow for an angle when the helicopter is rising or falling.

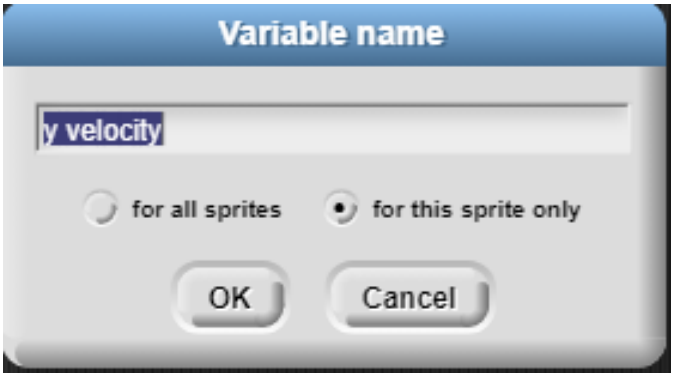
   Let's begin with the direction of the helicopter:

   - Import helicopter costume from the costume library and name it `heli_right`
   - Duplicate costume, edit it in the paint editor, flip it horizontally and rename it to `heli-left`.



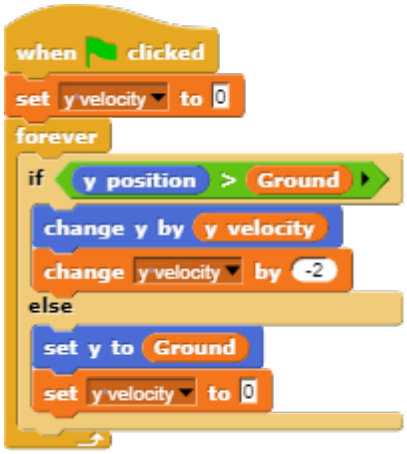
2. Improved gravity (demo script):

   - Open a new project `Gravity` in your browser to test this.
   - Two new concepts: variables (hold numeric, string or Boolean values), and locality (where a variable is known to the program).
   - Change the sprite's turtle `costume` to an `airplane`.
   - Import the `reset` script and set its coordinates to (0,110).
   - In the `Variables` menu, `Make a variable` called `y velocity` which is `for this sprite only` and `show` it on the stage:

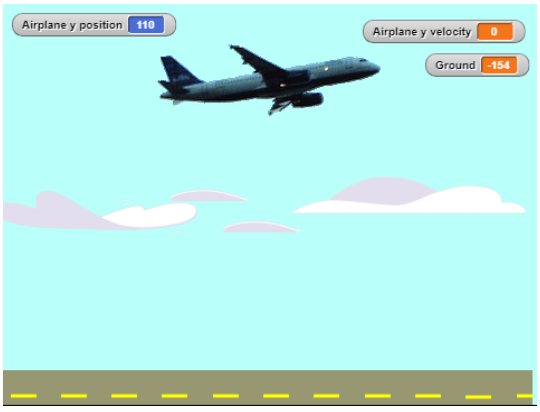


- When velocity is a positive number, the plane moves up, when it is negative, it moves down.
- Gravity makes objects accelerate downwards: the speed at which it moves must change negatively *while the program is running*:

- When you start the script, the `y velocity` is set to `0` (the object stands still). Then it is `forever` (until the script is stopped) increased in the downwards direction.
- You can compute the acceleration a = Δ velocity / Δ time and show it on the stage. In reality, the acceleration due to Earth's gravity is constant: g = 9.8 m/s/s.
- The terminal velocity of an object falling to Earth is bounded by multiple factors: shape, size, air density, mass. It can be computed to be ca. 53 m/s (ca. 120 miles/hr), which is when the air density breaking effect is balanced by the Earth's gravity.
- You also want to add some ground-level code so that the object doesn't fall forever:



- For this to work, you have to `make` and `set` the `Ground` variable: it's global variable (that is, it is known to all sprites).



- When you transfer this script to the helicopter, you'll have to re-balance the navigation values and/or change the events.

3. Engine is running when the helicopter script is active:
    - When script starts, send `broadcast` to run sound script.

Links / Sources

- Ride of the Valkyries by Richard Wagner / Apokalypse Now (Francis Ford Coppola's adaptation of Joseph Conrad's Heart of Darkness)
- Solution in the Snap! cloud by Tasikass2024
- Compare with: video solution in Scratch (not Snap!)
- Turn the helicopter into a multilevel game (Scratch, not Snap!)