The Bricks Game in Auto Mode:

Concepts used in this program:

- Algorithms
- Clones
- Conditional looping (repeat until)
- Conditionals (IF)
- Conditions: YES/NO questions
- Costume-based animation
- Lists
- Logic operators (AND, OR, NOT)
- Nested looping
- Procedures with inputs
- Relational operators (=, <, >)
- Simple looping (repeat, forever)
- Stopping scripts
- String operations (join, letter, length of)
- String variables
- Synchronization using broadcasting
- User input (ASK)
- Variables - local/global scope
- Variables – numbers
- Variables as sliders

The Bricks Program:

I am going to use the following program as my starting point:

https://scratch.mit.edu/projects/104716140/

It implements the Bricks game which you play with a ball and a flat paddle. A number of bricks are lined up at the top of the screen. As the ball bounces up and down you use the paddle to hit the bricks and score points. The ball must not touch the ground: after 3 such touches you lose the game. If you hit all the bricks you win the game. You can control the difficulty level of the game by changing the speed of the ball.

To play the game:

1. Click on the "Green flag": everything is reset to original state.
2. Set ball speed using the slider.
3. Press SPACE BAR to start the game.

New Features:

We would like add the following features to this program:

1. Make the computer play the game, i.e. instead of the player moving the paddle, the computer will move the paddle and play the game.
2. Since the computer is much better at playing this game (as you will realize after doing the above), have 3 balls simultaneously bouncing up and down that the computer must juggle.

Version 1:

Feature idea 1:

Make the computer move the paddle.

This is a simple change. Instead of tracking "mouse x", track "x position" of the ball.

Feature idea 2:

Have 3 balls simultaneously bouncing up and down that the computer must play with.

Let's do this in 3 steps:

Step 1: Have 3 ball sprites.
Duplicate the ball sprite to have 3.

Step 2: Make all 3 balls follow the same game rules.
Add 2 additional "when I start as a clone" scripts to the "brick" sprite. Each script will wait to touch a different ball.

Each "ball" sprite will respond to "when I receive ball touch" message.

Each ball must enter the game at a random time and at a random place:
Add a random wait, and "go to random x".

Step 3: Remove # of lives feature since we don't need it.
Just remove the relevant code, sprites, and variables.

Step 4: The paddle will track the "y positions" of all 3 balls and pick the one that is closest.
See algorithm the below:

```
Compare y positions of ball1 and ball2
If ball1 is lower
      lowest = 1
Else
      lowest = 2
Endif
```

```
Compare y positions of ball3 and "lowest"
If ball3 is lower
      lowest = 3
Endif
If lowest = 1
      Track ball1
If lowest = 2
      Track ball2
If lowest = 3
      Track ball3
```

Note: "Tracking" can be implemented as "go to" or "glide to", the latter having increased chances of the computer losing.

Solution: bricks-auto-1.sb2
https://scratch.mit.edu/projects/318160664/

Version 2:

Feature idea 3:
Repeat the game 3 times since it gets over quickly.

This is just a matter of repeating part of the code in "stage" 2 more times.

Solution: bricks-auto-2.sb2
https://scratch.mit.edu/projects/318161277/

Author: Abhay B. Joshi (abjoshi@yahoo.com)
Last updated: 2 July 2019