

Gravity Exercises

IMPORTANT: In the exercises below, do not use “is touching” or “is touching color”. Instead, everything is checked by looking at x and y coordinates. Also, don't use any “move” functions. The “goto proposed x, proposed y” that's already in the code is the only thing that actually changes the penguin's coordinates.

1. Add code to the CheckKeys function. If the left arrow is down, the x velocity decreases. If the right arrow is down, the x velocity increases.
2. Make it so that when you press the J key the penguin puts on the jet pack, and when you press the J key again the penguin takes off the jet pack. When the penguin has the jet pack on and you hold down the up arrow, the jet pack fires and pushes the penguin upward.
3. Write code for the CheckMaxSpeeds function so that it only lets the x velocity get so big or so small. Similarly, keep the y velocity from getting too big or too small.
4. Write the CheckGround function. If the “on ground” variable is 0 (not on the ground) and the penguin (or actually the penguin's feet) are at or below the y value of the “ground” sprite, set the penguin's y value so that it's standing right on the ground, and set “on ground” to 1. When “on ground” is 1, gravity does not act. When “on ground” is 1 and the player presses the spacebar, make the penguin jump into the air, and set “on ground” to 0.
5. Write the ApplyLeftFan and ApplyUpFan functions. The way these work is if the penguin is more or less in line with the fan (and on the correct side of the fan), the fan blows the penguin (modifies its velocity) in the given direction.

Harder challenges:

6. Write CheckWrap. The idea is that if the penguin's proposed x would put it “off the screen” to the left, instead it “wraps” around the screen and comes out on the right side. Similarly, if it goes “off the screen” to the right, it wraps and comes out the left. For top and bottom wrapping, the ground already keeps the penguin from going off the bottom of the screen, but when the penguin goes off the top of the screen, you can (if you want) make it wrap to the bottom.
7. Write CheckHorizontalBar. The idea is that if the penguin moves onto the bar, it bounces the penguin back in the opposite direction. You can place the horizontal bar in different places on the screen to make sure it's working. Can you make it so the penguin bounces “up” off the bar if he hits it with a downward velocity from the top, but bounces “down” off the bar if he hits it with an upward velocity from the bottom?
8. Write CheckVerticalBar similar to CheckHorizontalBar, but bouncing in the x dimension instead of the y dimension.

Very hard challenge:

9. Make the magnet pull the penguin toward it. You will need to use sin and cos for this.