

Angry Turkeys Part I

1. Intro screen
 - a. Write a script on the Stage using the “when green flag clicked” block.
 - b. When the green flag is clicked, broadcast a “start intro screen” message.
 - c. When the stage receives “start intro screen”, it changes to the “intro screen” backdrop. If you want you can do some drawing on the intro screen backdrop.
 - d. When all the sprites receive “start intro screen”, they hide, except the large turkey which shows. The large turkey slides in from the left side of the screen. Alternatively, you can use the “set ghost effect” block to make the large turkey fade in from invisible (100) to shown (0).
 - e. Create a sprite called “Title”. Put the sprite in Vector Mode and create a text object that says “Angry Turkeys by <your name>”. When the Title sprite receives the “start intro screen” message, it slides in from the right side until it meets the large turkey in the middle.
2. Level 1 screen
 - a. After the intro screen animation, wait three seconds, and then broadcast a “start level 1” message.
 - b. When the stage receives “start level 1”, it changes to the “level 1” backdrop.
 - c. When all the sprites receive “start level 1”, they show, and they go to their correct locations in the house. Make sure to test this by dragging the sprites to different places on the screen, and then click the green flag, and check that when level 1 starts all the sprites go to the right place.
3. Making the characters talk
 - a. For the Peter character, after the sprite shows after receiving “start level 1”, use a “forever” block to make Homer repeatedly wait 10 seconds, and then say “Man, am I stuffed!” for 2 seconds. Check your code to make sure Peter says that every 10 seconds.
 - b. Let’s make Peter more interesting by making him say a few different things. Instead of always saying the same thing, use the “random” operation to generate a random number from 1 to 3 and put it into a variable. Make a variable FOR THIS SPRITE ONLY (very import) in the Peter sprite called “choice”. Set the variable to a random number from 1 to 3. Then, if “choice” is 1, say “Man, am I stuffed!” If “choice” is 2, say “Eating turkey makes me sleepy!” If choice is 3, say “Let’s watch the game on TV!”
 - c. Advanced option using lists. If we wanted to make Peter say even more things, it could get annoying having to write so many “if” statements. A better way is to use a list. Make a new list FOR THIS SPRITE ONLY (very import) in the Peter sprite called “choices”. Add a “when green flag clicked” block for Peter and start with a “delete all of choices”. Then do 10 insert “whatever...” at last of the “choices” list to fill it with 10 statements. Then, when you want Peter to say something, pick a random number from 1 to 10, put it in the “choice” variable, and then use “item choice of choices” to grab one of the phrases out of the choices list and say it.

- d. Once you are happy with Peter's phrases, add phrases to the other characters. Make phrases for Homer, for the chef, and for the axeman. Those are all "bad guys". Make some phrases for the pigs that show the pig is innocent (the pig can say "how did I get here?" or "save me, Turkey!" or other things you can think of). As you're making the statements for all the sprites, remember the trick of dragging code from one sprite to another to make the programming faster.
4. Bypassing the intro screen
 - a. After a while it gets annoying waiting for the intro screen to play each time we test our game. Modify the script so that it starts level 1 right away. Later we can hook the intro screen back up.
5. Shooting the turkey without physics
 - a. Next week, we are going to start shooting the turkey with a realistic simulation of gravity. For now, let's just practice moving the turkey around with "glide" statements.
 - b. When the turkey receives "start level 1", after it goes to its start location, make it wait three seconds. Then it starts a sequence of "glide" statements. Do your best to make it look like the turkey:
 - i. Shoots up in the air and goes through the hole in the roof of the house.
 - ii. Bounces around the upper floor and kills the axeman and Homer.
 - iii. Bounces down to the second floor and kills Peter and the chef (but misses the pig).
 - iv. Bounces down to the ground floor and kills the other Homer 2 (but misses the pig).
 - v. Comes to rest, and then dies (see below for Turkey death).
 - c. None of these motions are going to look very realistic. Just do your best, and then when we use gravity physics you'll see how much better than looks, and how much easier it is.
6. Making the characters die
 - a. Make a second "when I receive start level 1" block for Peter. Use another forever loop to check over and over if Peter is touching the turkey. If so, Peter hides. (Also remember to show Peter when he receives "start level 1".)
 - b. Do the same for the other sprites so each hides when the turkey touches them.
7. Making the Turkey die
 - a. Just like in Angry Birds, after the Turkey has done his brave act, he dies. Instead of just making the Turkey disappear, use the a loop with "set whirl effect" and "set ghost effect" to make the Turkey look like it's whirling into spiral while it gets lighter and lighter until it's gone.

Angry Turkeys Part II

- These exercises are all going to be in the turkey_small script. We are going to make the turkey aim and then fly. Several blocks have already been created for you. You just need to add code to most of the blocks. Also, six variables have already been created in the turkey_small sprite.
8. Writing the Aim function. Here are the things the Aim function should do (in this order):
 - a. Position the small_turkey in the lower-left corner of the screen.
 - b. Set the small_turkey size to 100%. (You'll see why we need this later.)
 - c. Clear the small turkey's graphics effects. (Because he may have faded away when he died.)
 - d. Set the "flying" variable to 0. (Since we are aiming now, not flying.)
 - e. In a "repeat until" loop, until we are flying:
 - i. Point towards the mouse pointer.
 - ii. Call the calculate distance function to set the variable "distance" to the distance to the mouse.
 - iii. Use a "think" block to show the distance. (Use the "floor" operator to get rid of the decimal part.)
 - f. Think empty to clear the think.
 9. You can now test your Aim function. The small_turkey should always point toward the mouse, and the distance the turkey thinks should be the distance to the mouse.
 10. Writing the "when space key pressed" block. Do these, in order:
 - a. Set the "flying" variable to 1. (This stops the Aim function. Why?)
 - b. Set the small_turkey's size to 50%. We want to make him small so it's no so hard to aim him into the house.
 - c. Set the values of the xvelocity variable and the yvelocity variable. The xvelocity should be the x distance from the mouse to the small_turkey (how can you compute that?) divided by 10. The yvelocity should be the y distance from the mouse to the small_turkey divided by 10.
 - d. Call the Fly function to do the actual flying.
 - e. Broadcast the "start aim" message. (It's a little tricky to understand why we must broadcast "start aim" instead of just calling the Aim function. Ask me or another mentor if you want to understand why.)
 11. You can test the "when space key pressed" block now. But since we haven't written the Fly function yet, all pressing the key should do is make the turkey stop aiming (stop following the mouse).
 12. Writing the Fly function. Do these, in order:
 - a. The whole function is a "repeat until" loop that repeats until we are no longer flying. Inside this "repeat until" loop:

- i. (optional) Use the “stamp” block to stamp a picture of the turkey.
 - ii. Change the turkey’s x position by the value of the xvelocity variable.
 - iii. Change the turkey’s y position by the value of the yvelocity variable.
 - iv. Change the yvelocity variable by -0.5 (negative one half). That’s gravity.
 - v. Call the CheckBounce function.
 - vi. Call the CheckDeath function.
13. You can test this function. It should make the turkey fly and fall. The turkey is not yet checking for death so it will just fall to the bottom of the screen and you’ll need to press the stop sign to restart the game.
14. What I hope you find somewhat amazing is that steps ii., iii., and iv. above are all that’s needed to make the turkey fly with realistic gravity. Those three lines are the heart of “Angry Birds”.
15. Write the CheckDeath function. Do these, in order:
 - a. Use an “if” statement to check if the turkey has gone lower than y position of -160. If so, set “flying” to 0 and call the Die function.
 - b. Use an “if” statement to check if the turnkey is touching the color red. If so, set “flying” to 0 and call the Die function. If you want to be fancy, you can combine these two “if” statements into just one, using an “or” block.
16. You should now be able to test an almost working game. You can aim the turkey and start it flying, and it flies until it hits red or gets to the bottom of the screen.
17. Write the CheckBounce function. This function is used to make the turkey bounce off the blue and green walls. It uses two “if” statements:
 - a. Check if the turkey is touching the color blue. If so, bump the turkey back “x velocity” pixels (why do we need to do that?) and then set “x velocity” to be negative “x velocity” divided by 2. By setting it to the negative of itself, we make it bounce the turkey. By dividing it by 2, we make the “bounce” slowly get less and less, kind of like it would do in real life.
 - b. Check if the turkey is touching the color green. If so, bump the turkey back “y velocity” pixels and then set “y velocity” to be negative “y velocity” divided by 2.

So you can see what those two “if” statements do. The first makes the turkey bounce in the x (horizontal) direction. The second makes the turkey bounce in the y (vertical) direction.

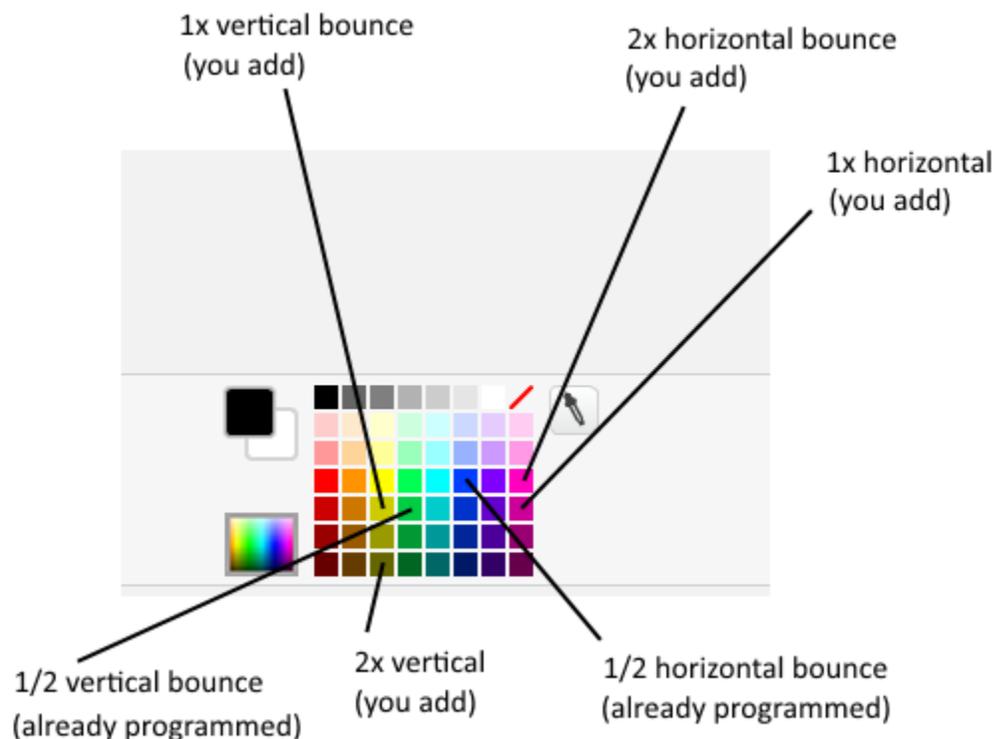
Now we have a fully playable game. We are going to add more levels next time. For now, you might want to add a variable to keep track of how many shots it takes you to clear the house. (But try not to kill the innocent pigs.)

Angry Turkeys Part III

18. As it stands, the large turkey broadcasts “start level 1” when the intro animation is done. Let’s make it so that instead of starting level 1 right away, it shows a menu where you can choose to play level 1, level 2, or level 3.
 - a. In the turkey_large and Title sprites, remove the code after the glide (that is, the “repeat 25” fade and the hide) and put it under a new “when I receive fade intro screen” block. (I’ve already done this for you.)
 - b. In the turkey_large, instead of broadcasting “start level 1”, broadcast “show level menu”. (I’ve already done this for you.)
 - c. Make three new sprites to be the buttons. Each button will be very similar. (If you want to be fancy, you can try using one sprite with cloning instead of three sprites.)
 - d. Make the “Level 1” button first. Draw the button using VECTOR MODE. Put the button in the lower-left part of the screen. Add the following code to the button:
 - i. When the sprite receives “start intro screen”, it hides.
 - ii. When the sprite receives “show level menu”, it shows.
 - iii. When the sprite is clicked, it broadcasts “fade intro screen”.
 - iv. When the sprite receives “start intro screen”, it hides, waits 2 seconds, and then broadcasts “start level 1”.
 - e. Test button 1 to see that it works.
 - f. Make “Level 2” button and “Level 3” button sprites, which are just like the Level 1 button sprite, except they have a different screen position and they broadcast “start level 2” and “start level 3”.
19. Limiting the number of turkeys. As it stands, you can shoot as many turkeys as you want. Let’s make it so there is a limit.
 - a. Make a variable in turkey_small called “num shots” (make the variable FOR ALL SPRITES).
 - b. In “when I receive start level 1” and “when I receive start level 2” and “when I receive start level 3”, set “num shots” to some number appropriate for the level. (Try 10 as the number of shots for level 1. But you can change it if you find that number is too big or too small.)
 - c. Add to the “when I receive start aim” block to check if there are any shots left. If there are not, instead of calling the Aim function, make the turkey_small say “Game Over” for 5 seconds, and stop all scripts.
 - d. Make it so that when the turkey hits any of the bad guys (axeman, chef, etc.) it adds 3 to “num shots”.
 - e. Make it so that when the turkey hits a pig it subtracts 1 from “num shots”.
20. Winning. Make a variable called “num bad guys alive”. When the level starts, set “num bad guys alive” to the number of shown bad guys. Each time the turkey kills a bad guy, subtract 1 from “num bad guys alive”. When “num bad guys alive” gets to zero, make the turkey say “You Win!” for 5 seconds, and stop all scripts.

21. Making super bounces. If you look in the CheckBounce function of the turkey_small, you will see that we are already checking for blue and green bounces. When the turkey hits blue, it does a horizontal (left/right) bounce. When it hits green, it does a vertical bounce (up/down). You can see that to make it bounce we negate the velocity (the “x velocity” for horizontal, the “y velocity” for vertical) and we also divide by 2. The division by 2 makes the turkey bounce less and less, kind of like a bouncing ball bounces less high each time it hits the ground. Let’s add some more colors that have different bounce powers.

- Make it so that each of the colors in the picture below give the indicated kind of bounce. We already have the 1/2 horizontal and 1/2 vertical. Add an “if” statement to CheckBounce to make the 1x horizontal bounce (1x means it reverses the “x velocity” but doesn’t increase or decrease it). Make the 2x horizontal bounce (2x means it reverses the horizontal and multiplies it by 2, so it’s actually a “super bounce” where the turkey bounces off twice as fast as it comes in). Make the 1x and 2x vertical bounces.
- In all cases, use the colors indicated below. Note that you often have to actually draw something on the backdrop of that color so that you can then choose that color for your “if” statement.



Angry Turkeys Part IV

22. Lose animation. When you run out of shots and haven't gotten all the bad guys, the "game lost" message is broadcast. Go the turkey_large script. You will see that when the turkey large receives this message it shows. Your job is to add to this script to make the turkey say something and/or look sad to indicate that you lost. Some ideas:
 - a. Make a new costume for the turkey and modify the eyebrows and mouth to make it look like the turkey is sad.
 - b. Make the turkey say something sad.
 - c. Make tears come down from the turkey's eyes.

23. Win animation. When you get all the bad guys, the "game won" message is broadcast. Go the turkey_large script. You will see that when the turkey large receives this message it shows. Your job is to add to this script to make the turkey say something and/or look happy to indicate that you lost. Some ideas:
 - a. Make a new costume for the turkey and modify the eyebrows and mouth to make it look like the turkey is happy.
 - b. Make the turkey say something happy.
 - c. Make the turkey jump up and down.
 - d. Make the turkey say something special if you saved one or both of the pigs.

24. Making your own levels. This is the creative part, where you get to design level 2 and level 3. Use the red "kill" color and all six of your bounce colors (see previous page for the colors) to draw the backdrops for levels 2 and 3. Remember to stay in vector mode and keep your lines fairly thick. Below are some pictures which may give you some ideas for levels you could make.

VERY IMPORTANT: Make sure all your vertical bounce lines (the three shades of green) go side-to-side, and all your horizontal bounce lines (blue, pink, purple) go up-and-down. See level 1 for an example.

