Jasper Rutherford Project 3 Writeup

Explicit List of Features I Implemented:

- **Multiple Arm IK**
  - The monkey's feet independently reach for two respective bananas
- **Single Arm IK**
  - Multiple Arm IK counts for Single Arm IK
- **Moving IK**
  - The monkey traverses a series of vines
- **Motion Planning**
  - The Monkey uses an RRT to navigate the vines

Breakdown of Files Submitted

- **/src/**
  - Folder containing all files necessary to run the simulation.
    - /Images/ - File containing images used, along with image credits
    - /PROJ3/ - File containing all .pde files
- **Monkey.mp4**
  - A video that showcases various features implemented.
    - Split up into 11 sections labeled A – K (in alphabetical order)

Video Description:

- **Note:** Sometimes in the video it looks like the monkey makes some motions that are a little jumpy, or not as smooth as they should be. I've attributed that to my recording software missing frames/poor video quality, as I have been unable to recreate any choppy behavior in the actual simulation when I run it and watch it.
- **A & D** (0:00 & 0:10):
  - The monkey has grabbed one banana but is unable to reach the second banana.
  - Note: I had to specifically change the simulation to make this happen. Typically the simulation only reaches for the bananas if both are close enough to be reached, but for demonstration purposes I set the simulation to allow this to happen.
- **B & C** (0:01 & 0:05):
  - The monkey successfully moves to and grabs both bananas.
- **J & K** (0:50 & 0:54):
  - The monkey is unable to reach either banana. Again, I had to edit the simulation to enable this to happen for demonstration purposes. The version I turned in will always be able to reach the bananas*
- **E & F** (0:24 & 0:27):
  - The monkey successfully moves to and grabs both bananas with the RRT display enabled
- **G** (0:36):
  - *Here the stars aligned and several of the bones in the monkey's arm perfectly faced the opposite directions of each other, perfectly in line with the direction they wanted to face. This is the only time I saw this happen (the total time I've watched this simulation is likely over an hour at this point), and I happened to get

it on video! So technically the simulation doesn't grab the bananas 100% of the time, but I've declared the probability of this happening to be negligible. This could likely be avoided if joint limits were implemented.

- **H & I** (0:40, 0:45):
  - The monkey successfully moves to and grabs both bananas with debug colors on.
  - You can see which limb each bone is associated with over time.
    - H goes through configurations A -> B -> C -> E
    - I goes through configurations A -> B -> D -> E
    - (what this means is explained further on in this doc)

Code I Wrote vs Code I Didn't:

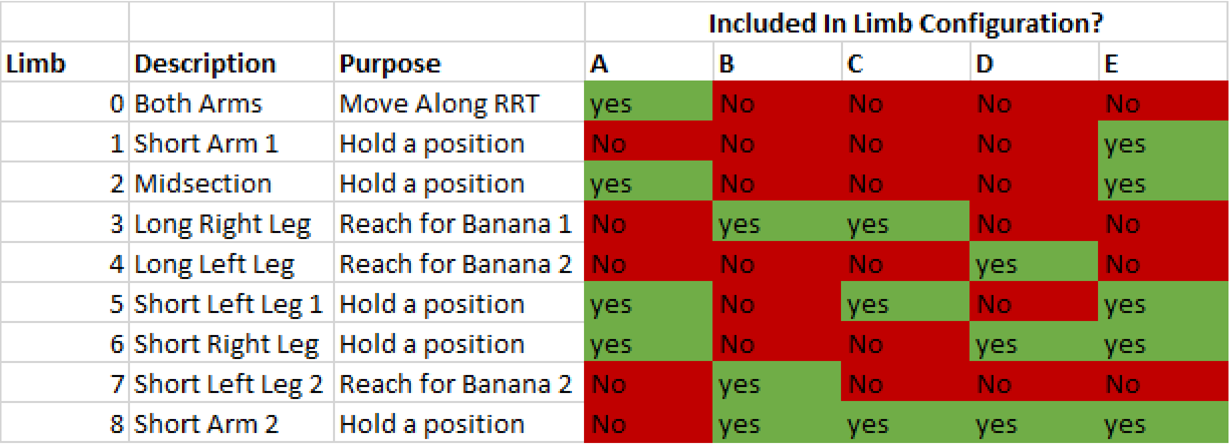
- **Code I Wrote:**
  - Bone.pde, Joint.pde, Map.pde, PROJ3.pde, VineNode.pde

- **Grey Area:**
  - Limb.pde
    - Largely written by me, but any limb that moves toward a goal (rather than just moving to/holding a predetermined position) uses code that is very heavily based off the "IK_Exercise" in class activity.
  - RRT.pde & Node.pde
    - Almost completely copied from the RRT.pde & Node.pde files I submitted for project 1 (so 100% written by me, but some graders might not like reusing stuff from previous projects)
- **Code I Did Not Write:**
  - Vec2.pde
    - Although there is one function I wrote, that function is not used in this project.
    - Also includes two functions from "IK_Exercise" in class activity.

How It Works:

- **Map Generation**
  - Between 20 – 30 vines are spawned, each vine between 100 – 350 pixels long
  - From the top of each vine, every 40 pixels down the length of the vine a point is saved as a point that the monkey can grab, with an additional point always spawning on the end of the vine.
  - Each point then loops through and tracks what points are closer than 75% of the arm span of the monkey (their "neighbors")
  - Two bananas are spawned
    - Banana 1 on a random point from the list
    - Banana 2 on a random neighbor of that point
  - The list of points is fed into the RRT code I wrote for Project 1 (loosely adapted to fit this project), and a map of ideal paths are generated
  - The monkey is spawned with the end of his left hand rooted to the node closest to (200, 200)

- **The Monkey**
  - The Monkey is represented as a series of limbs, with each limb represented as a series of joints and bones.
  - There are 9 possible limbs, and each limb either
    - tries to reach for a goal or
    - holds a position and waits for other limbs to do their thing.
  - When a limb accomplishes its goal, it rearranges the bones/joints of the monkey into a different configuration of limbs with different purposes.
  - The various limbs and limb configurations are shown here:



			Included In Limb Configuration?				
Limb	Description	Purpose	A	B	C	D	E
0	Both Arms	Move Along RRT	yes	No	No	No	No
1	Short Arm 1	Hold a position	No	No	No	No	yes
2	Midsection	Hold a position	yes	No	No	No	yes
3	Long Right Leg	Reach for Banana 1	No	yes	yes	No	No
4	Long Left Leg	Reach for Banana 2	No	No	No	yes	No
5	Short Left Leg 1	Hold a position	yes	No	yes	No	yes
6	Short Right Leg	Hold a position	yes	No	No	yes	yes
7	Short Left Leg 2	Reach for Banana 2	No	yes	No	No	No
8	Short Arm 2	Hold a position	No	yes	yes	yes	yes

- **The Limb Configurations** (A huge problem whose solution I am pleased with)
  - Configuration A - Limbs 0, 2, 5, and 6:
    - Limbs 2, 5, and 6 hold a predetermined position while Limb 0 moves along the RRT toward the bananas.
    - When Limb 0's root is close enough to the two bananas that the bananas can be reached by the feet, the monkey is rearranged into configuration B
  - Configuration B - Limbs 3, 7, and 8:
    - Limb 8 moves to and holds a predetermined position while Limb 3 reaches for banana 1 and Limb 7 reaches for banana 2.
    - If Limb 7 reaches banana 2 before Limb 3 reaches banana 1, then the monkey is rearranged into configuration C.
    - If Limb 3 reaches banana 1 before Limb 7 reaches banana 2, then the monkey is rearranged into configuration D.
  - Configuration C - Limbs 3, 5, and 8:
    - Limbs 5 and 8 move to and hold predetermined positions while Limb 3 reaches for banana 1.
    - When Limb 3 reaches banana 1, the monkey is rearranged into configuration E.
  - Configuration D – Limbs 4, 6, and 8:
    - Limbs 6 and 8 move to and hold predetermined positions while Limb 4 reaches for banana 2.
    - When Limb 4 reaches banana 2, the monkey is rearranged into configuration E.

- Configuration E – Limbs 1, 2, 5, 6, and 8:
    - All five limbs move to and hold predetermined positions.
    - If Auto-Restart is on and all five limbs are in their predetermined positions, then the simulation will restart with new random values.

Hotkeys:

- **A:** Toggle Auto-Restart. On by default. Automatically restarts the simulation with new random generation when the monkey enters rest position after grabbing the bananas

- **B:** Toggle Debug Colors. Off by default. Renders each bone's color depending on which limb it is a part of. Bones of the same color are part of the same limb.

- **G:** Toggle the RRT graph. Off by default. Renders every node and path between two nodes in the RRT.

- **R:** Restart the simulation with new random values.