

# Code Clinic Tips

This page includes helpful tips based on our experience in helping students in the "Code Clinic" ([interactivepython@online.rice.edu](mailto:interactivepython@online.rice.edu)). Be sure and take a look at these tips if you get stuck.

## Tip #1

For those of you struggling with controlling your paddles, remember this. You should update the paddle's position in the draw handler so it's updated continuously. The paddle's velocity is updated in the key handlers when a key is pressed or released. Use both the keyup and keydown handlers to control the paddle's velocity.

## Tip #2

`paddle1_pos`, `paddle2_pos`, `paddle1_vel` and `paddle2_vel` should be numbers, NOT lists. Using lists causes needless complexity and is highly UNrecommended. Just use the numbers `paddle1_pos` and `paddle2_pos` to build expressions to draw the paddles.

If you are having trouble visualizing how their values relate to the geometry of Pong, please review the Pong video and sit down with pencil and paper to sketch out the Pong frame and label it with quantities associated with your variables.

## Tip #3 - Very important

I can't stress this enough. Read the mini-project description and walk through. They are there to help you avoid wandering off into dead ends. If you are stuck and haven't read the description carefully, you are probably headed for hard times.

## Tip #4

Stuck paddles are the bane of many Pong programmers trying to keep them on the screen. Remember that you update your paddle's position via something like

```
paddle1_pos += paddle1_vel
```

Ideally, you'd like to add something like

```
if paddle1_pos >= HALF_PAD_HEIGHT:  
    paddle1_pos += paddle1_vel
```

The problem with this solution is that the paddle sticks because when the condition in the if becomes **True**, the paddle can't move anymore, ever! The trick is to replace **paddle1\_pos** in the condition for the if by **paddle1\_pos + paddle1\_vel**. Now, this condition check to see if the next movement for the paddle puts it off the edge. If so, the move is disallowed. Now, you can move the paddle back down.

## Tip #5 - Keyboard issue in CodeSkulptor

In Pong (and later in Spaceship/RiceRocks), we will use the keyboard to enter input for your mini-projects. This page describes a couple of the common keyboard issues that have arisen in previous IIPP sessions.

- A common issue with many keyboards is "ghosting", where a keyboard fails to recognize three or more simultaneous key presses. For this class, we have designed the rubrics to focus on testing the behavior of your mini-projects using single keys presses in isolation so ghosting should not be an issue during peer assessment.
- In Firefox, if the option "Search for text when I start typing" is enabled, the "w" and "s" key presses may be captured by Firefox. The option "Always use cursor keys to navigate within pages" may also cause issues by capturing arrow keys. (These options are off by default.) In Windows, these options can be disabled via Tools -- Options -- Advanced. Another alternative is simply to use Chrome.
- Some Firefox users report that the up/down arrows keys do not work properly. One possible cause is being in "caret browsing" mode. Try pressing F7 key (before running the program) to switch out of this mode.<http://forums.mozillazine.org/viewtopic.php?t=424900>
- Some students have observed that the frame created by SimpleGUI for Pong does not automatically recognize keyboard input. Browser windows (including SimpleGUI frames) do not recognize keyboard input unless the window is selected and has "focus". To get Pong to recognize keyboard input, click anywhere on the frame. The top of the window will highlight and now keyboard input will be recognized.
- A few students have reported issues with keyboard input on Chromebooks, especially on the keyboard echo example. Try clicking on the canvas (not just the window) to ensure that the frame has focus.