Simon Says Rubric:

_____ Start button 5
> Start button to start a new game. No need to run program again to play again.
> Player can press start at any time.
> Pressing start while playing ends the current game and starts a new one
> All buttons should resize if window resizes

_____ Continuous play/ success detection 10
> No need to click any button to signify completion of sequence.

_____ Flash Timing 15
> Longer sequence should begin flashing after 1 sec of player successfully completing previous sequence
> Flashing buttons should be 0.5 sec flash (depressed)
> So each color/button should take 0.5 sec being lit. Also wait approx 0.25s between flashes so same button flashed multiple times can be observed.

_____ Sequence should not change 10
> Sequence should be the same as before only add one new button(color) to the end
> Sequence should start with 1 button and increase by one on each successful completion

_____ Error Detection 15
> Error press should be detected immediately and not at the end of sequence
> Error if wrong button is pressed. Game over.
> Error if player does not press any buttons for 5 sec. Game over.

_____ Score 5
> Save high score and name(omit)
> Display score at end of game
> Score equals last level completed successfully

_____ Sounds 10
> Copy sounds from original game from supplied wav file
> Colors correct sound
> Error sound
> Use Audacity to cut sounds

_____ Pressing buttons should be easily noticeable 10
> Buttons should have the same pressed/selected color but in a lighter or darker shade, so player can easily detect a press
> Button should stay lit while left mouse is depressed and turn off upon release (max 5 secs depressed and then error)

Coding style: 20

_____ Object Oriented Programming style

_____ use of meaningful variable names
        avoid meaningless names
        names should not be very long ie. Horizontalposition

_____ comments
        comment lines which require explanation
        do not comment every line
        do not write verbose comments
        comments should be short and to the point (concise)
        not in sentence form
        will YOU be able to understand your code 1 year later
        function docstrings required (purpose, accepted args, return values)

_____ efficient code
        prefer loops instead of code repitition
        prefer functions for code that has a specific purpose

_____ readable and logical code
        easy to understand
        avoid convoluted/complicated code
        blank lines to separate logical blocks