

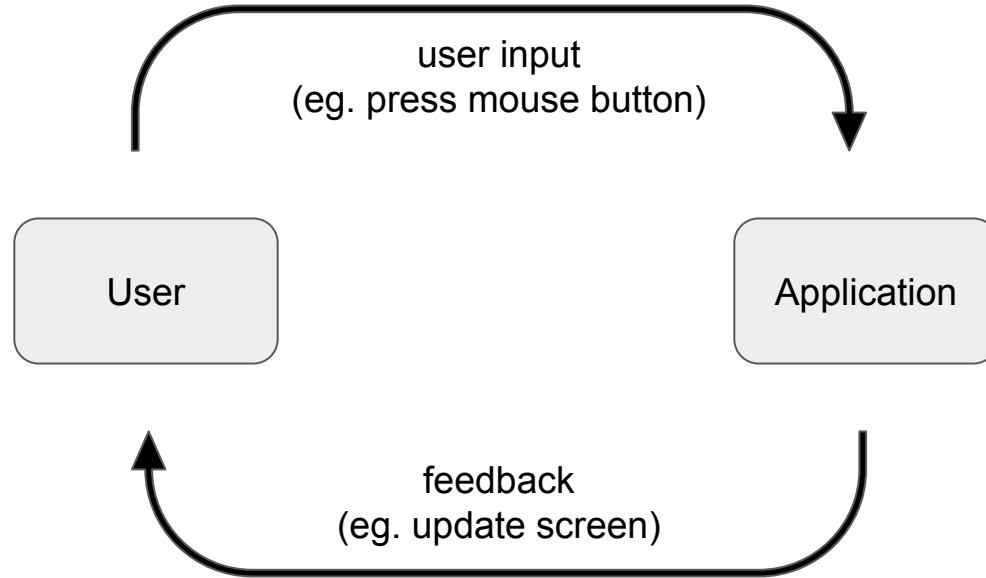
Building an Immediate Mode GUI (IMGUI) from scratch

Siddharth Bhatia (GitHub: @Sid-Bhatia-0)

<https://github.com/Sid-Bhatia-0/JuliaCon2022Talk>

<https://github.com/Sid-Bhatia-0/SimpleIMGUI.jl>

UI Loop



Two Paradigms of Interfacing with a UI Library

- Retained mode
- Immediate mode

Selected Resources

- Immediate-Mode Graphical User Interfaces - 2005
 - <https://www.youtube.com/watch?v=Z1qyvQsjK5Y>
- About the ImGui paradigm
 - <https://github.com/ocornut/imgui/wiki#about-the-ImGui-paradigm>

```
# ...  
window = GLFW.CreateWindow(#= ... =#)  
# ...
```

```
# ...  
window = GLFW.CreateWindow(#= ... =#)  
# ...  
user_input_state = UserInputState(#= ... =#)  
# ...
```

```
# ...  
window = GLFW.CreateWindow(#= ... =#)  
# ...  
user_input_state = UserInputState(#= ... =#)  
# ...  
user_interaction_state = UserInteractionState(#= ... =#)  
# ...
```

```
# ...  
window = GLFW.CreateWindow(#= ... =#)  
# ...  
user_input_state = UserInputState(#= ... =#)  
# ...  
user_interaction_state = UserInteractionState(#= ... =#)  
# ...  
function mouse_button_callback(window, #= ... =#)  
    # ...  
end  
# ...
```

```
# ...

window = GLFW.CreateWindow(#= ... =#)

# ...

user_input_state = UserInputState(#= ... =#)

# ...

user_interaction_state = UserInteractionState(#= ... =#)

# ...

function mouse_button_callback(window, #= ... =#)
    # ...
end

# ...

while !GLFW.WindowShouldClose(window)
    # ...
end

# ...
```



```

# ...

window = GLFW.CreateWindow(#= ... =#)

# ...

user_input_state = UserInputState(#= ... =#)

# ...

user_interaction_state = UserInteractionState(#= ... =#)

# ...

function mouse_button_callback(window, #= ... =#)
    # ...
end

# ...

while !GLFW.WindowShouldClose(window)
    # process widget interaction based on user input...
    # draw things on screen...
    # ...
end

# ...

```

```

# ...

window = GLFW.CreateWindow(#= ... =#)

# ...

user_input_state = UserInputState(#= ... =#)

# ...

user_interaction_state = UserInteractionState(#= ... =#)

# ...

function mouse_button_callback(window, #= ... =#)
    # ...
end

# ...

while !GLFW.WindowShouldClose(window)
    # process widget interaction based on user input...
    # draw things on screen...
    # ...

    GLFW.PollEvents() # update user_input_state using callbacks
end

# ...

```

```

# ...

window = GLFW.CreateWindow(#= ... =#)

# ...

user_input_state = UserInputState(#= ... =#)

# ...

user_interaction_state = UserInteractionState(#= ... =#)

# ...

function mouse_button_callback(window, #= ... =#)
    # ...
end

# ...

while !GLFW.WindowShouldClose(window)
    # process widget interaction based on user input...
    # draw things on screen...
    # ...

    GLFW.PollEvents() # update user_input_state using callbacks
end

# ...

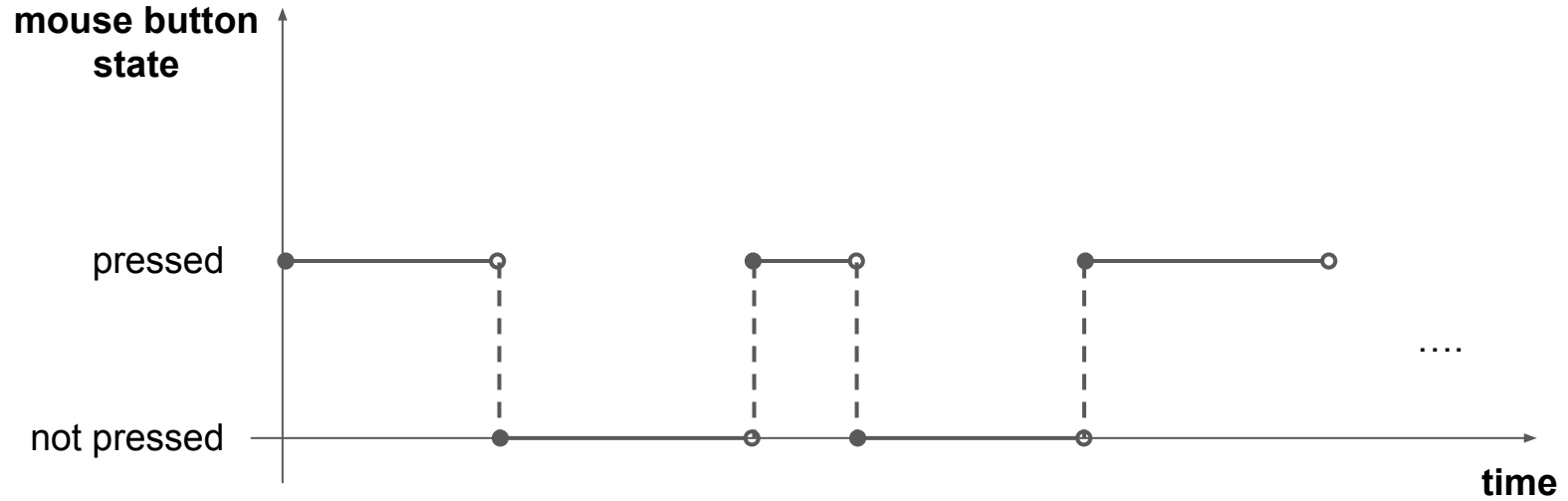
GLFW.DestroyWindow(window)

# ...

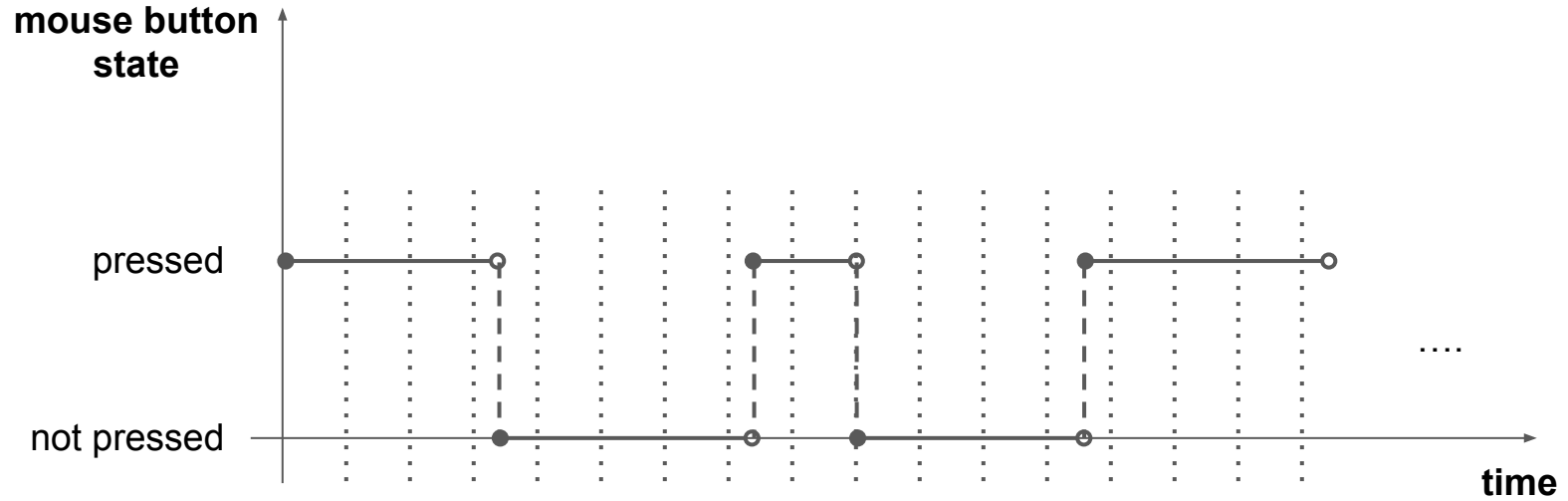
```

UserInputState

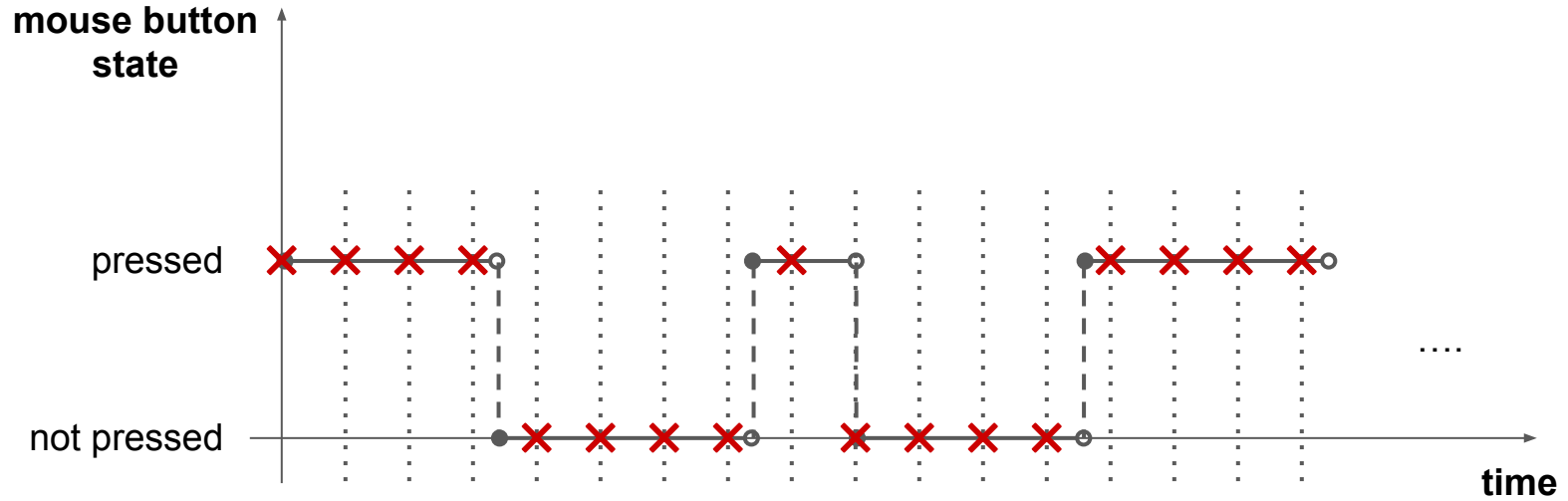
Mouse Button



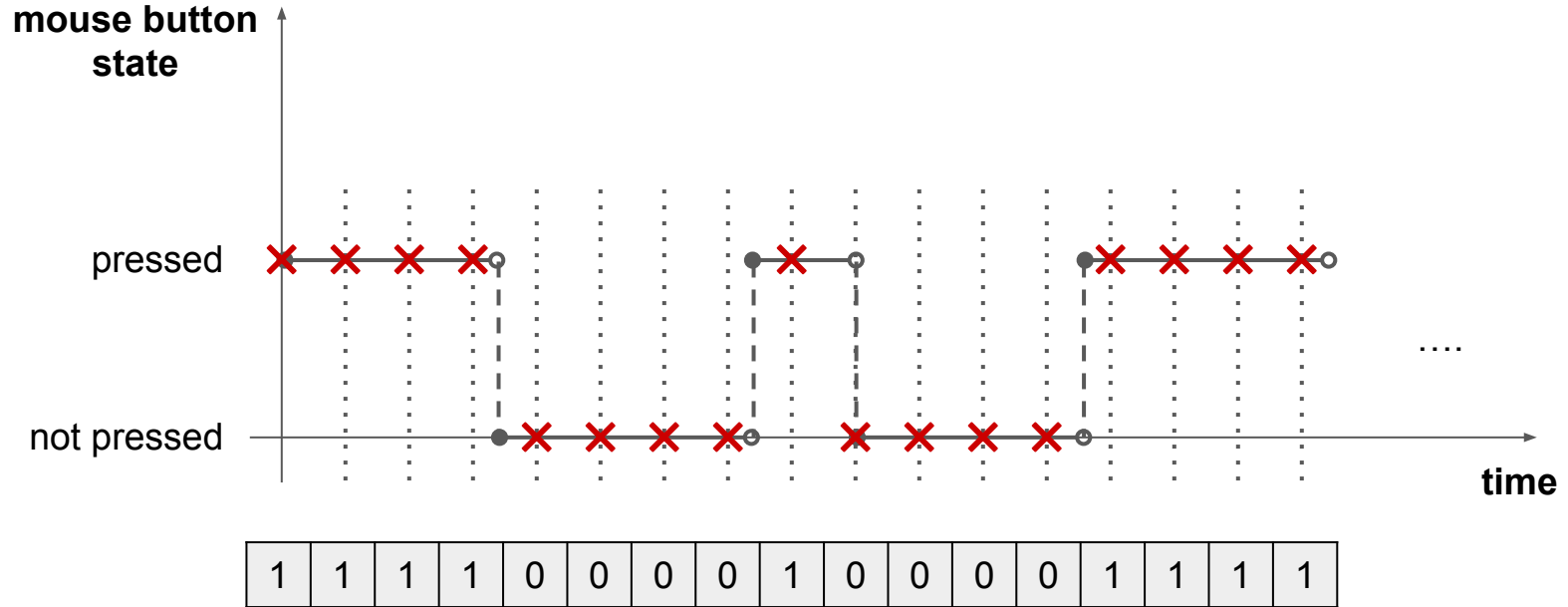
Mouse Button



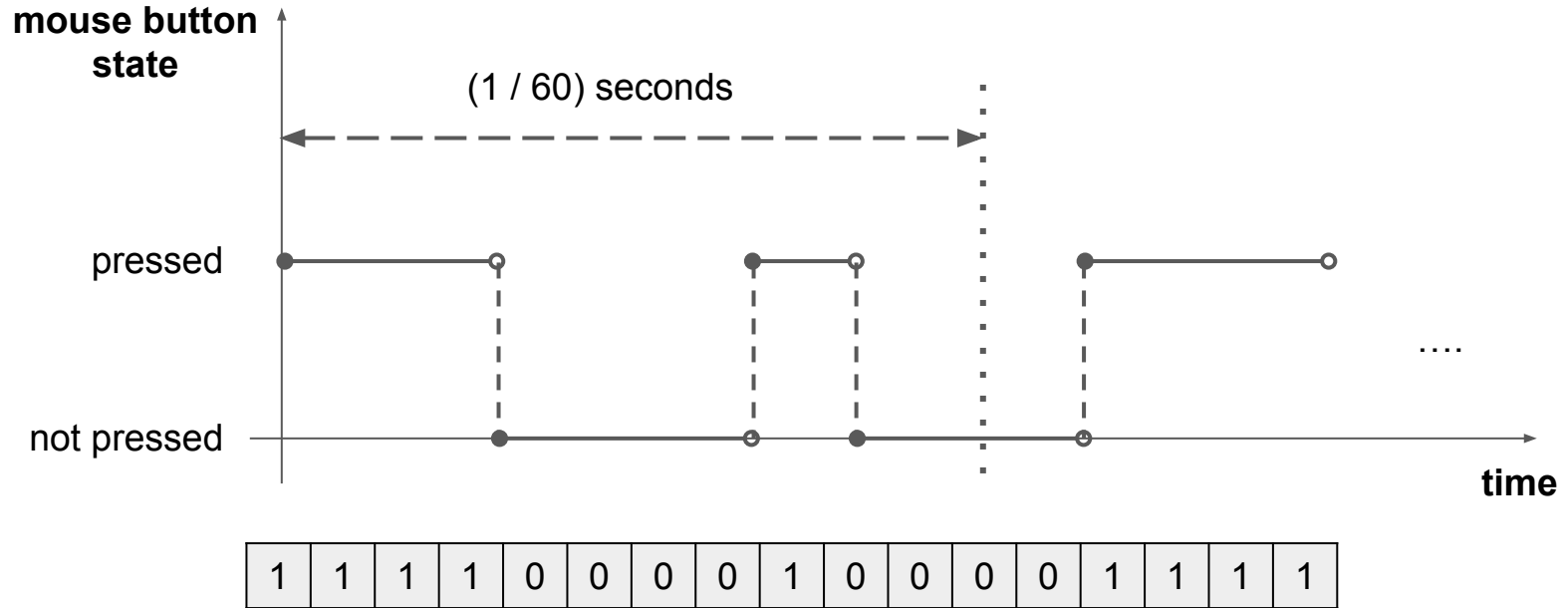
Mouse Button



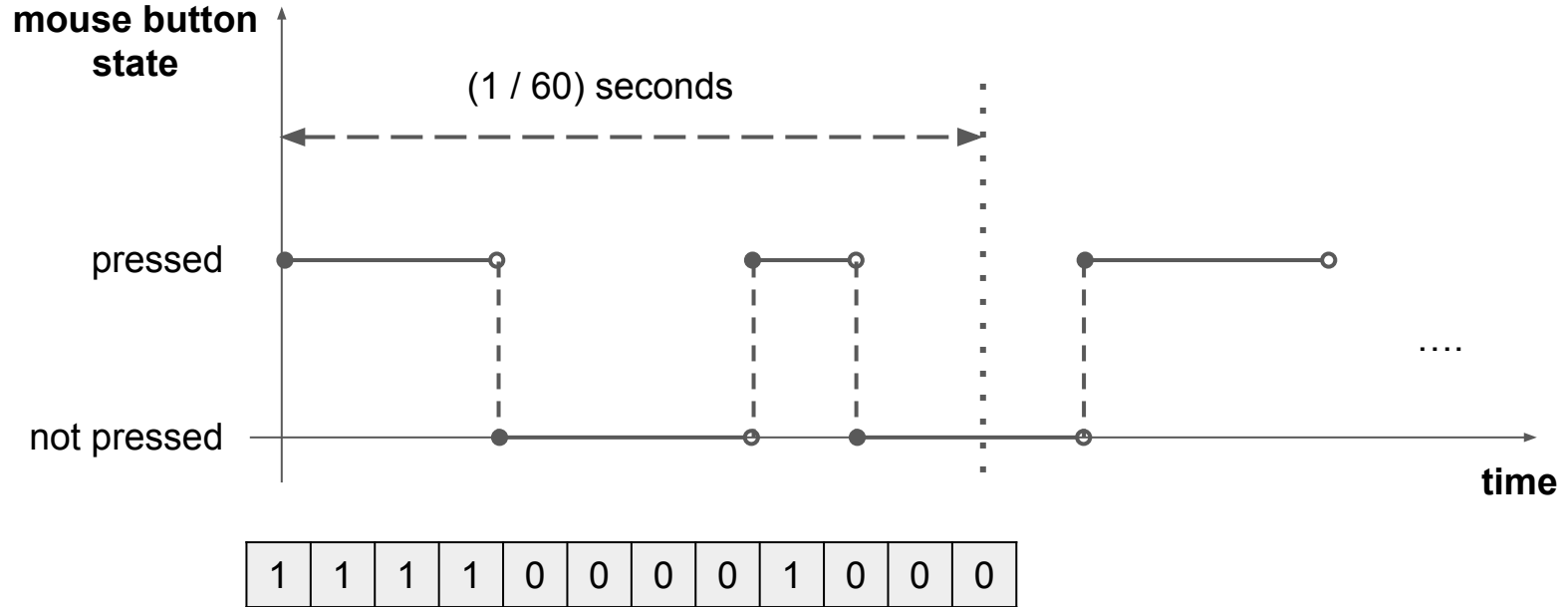
Mouse Button



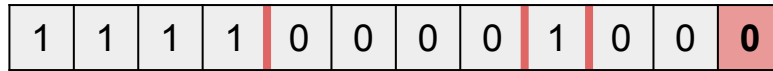
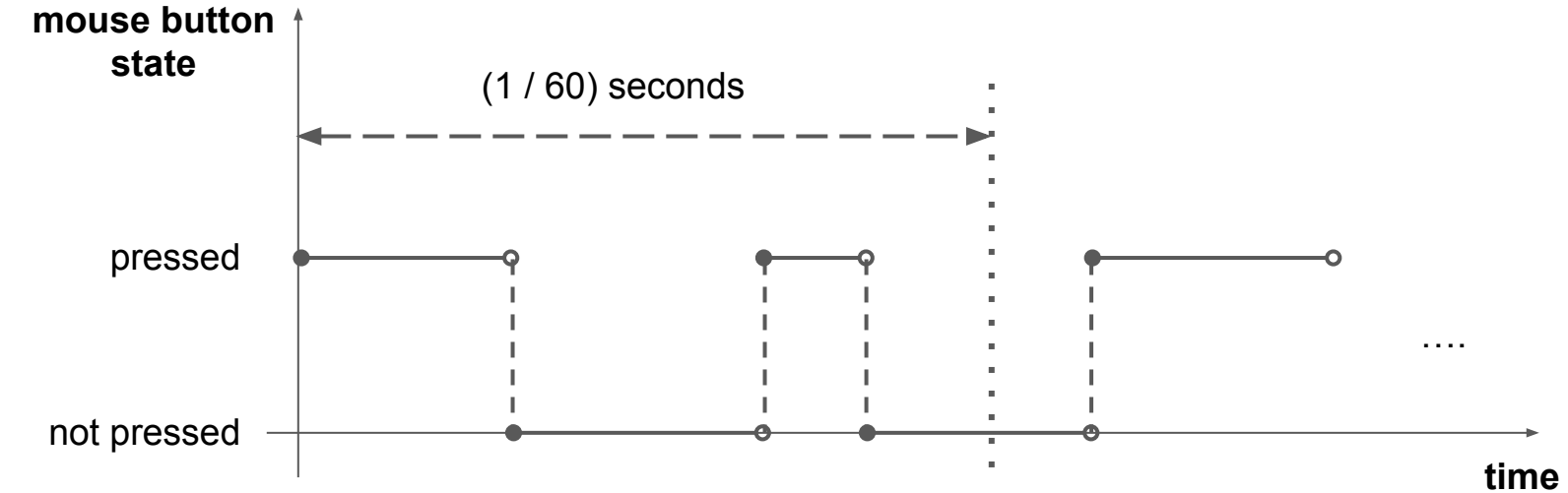
Mouse Button



Mouse Button



Mouse Button



num_transitions = 3

ended_down = false

```
struct InputButton
    ended_down::Bool
    num_transitions::Int
end
```

```
struct InputButton
    ended_down::Bool
    num_transitions::Int
end
```

```
went_down(ended_down, num_transitions) = (num_transitions >= 2) ||
                                           ((num_transitions == 1) && ended_down)
```

```
went_down(input_button) = went_down(input_button.ended_down, input_button.num_transitions)
```

```
went_up(ended_down, num_transitions) = (num_transitions >= 2) ||
                                          ((num_transitions == 1) && !ended_down)
```

```
went_up(input_button) = went_up(input_button.ended_down, input_button.num_transitions)
```

```
struct InputButton
    ended_down::Bool
    num_transitions::Int
end
```

```
went_down(ended_down, num_transitions) = (num_transitions >= 2) ||
                                           ((num_transitions == 1) && ended_down)
```

```
went_down(input_button) = went_down(input_button.ended_down, input_button.num_transitions)
```

```
went_up(ended_down, num_transitions) = (num_transitions >= 2) ||
                                           ((num_transitions == 1) && !ended_down)
```

```
went_up(input_button) = went_up(input_button.ended_down, input_button.num_transitions)
```

```
struct InputButton
    ended_down::Bool
    num_transitions::Int
end
```

```
went_down(ended_down, num_transitions) = (num_transitions >= 2) ||
    ((num_transitions == 1) && ended_down)
```

```
went_down(input_button) = went_down(input_button.ended_down, input_button.num_transitions)
```

```
went_up(ended_down, num_transitions) = (num_transitions >= 2) ||
    ((num_transitions == 1) && !ended_down)
```

```
went_up(input_button) = went_up(input_button.ended_down, input_button.num_transitions)
```

```
mutable struct UserInputState
    cursor::Point
    mouse_left::InputButton
    mouse_right::InputButton
    key_up::InputButton
    key_down::InputButton
    # ...
end
```


UserInteractionState

```

# ...

user_interaction_state = UserInteractionState(#= ... =#)

# ...

while !GLFW.WindowShouldClose(window)

    # ...

    button1_id = # ...
    button1_value = do_widget!(BUTTON, user_interaction_state, user_input_state, button1_id, #=
... =#)
    if button1_value
        # ...
    end

    # draw button1...

    # other widgets...

    # ...

end

# ...

```

```
# ...  
user_interaction_state = UserInteractionState(#= ... =#)  
# ...  
while !GLFW.WindowShouldClose(window)  
    # ...  
  
    button1_id = # ...  
    button1_value = do_widget!(BUTTON, user_interaction_state, user_input_state, button1_id, #=  
... =#)  
    if button1_value  
        # ...  
    end  
    # draw button1...  
  
    # other widgets...  
  
    # ...  
end  
# ...
```

```

# ...

user_interaction_state = UserInteractionState(#= ... =#)

# ...

while !GLFW.WindowShouldClose(window)

    # ...

    button1_id = # ...

    button1_value = do_widget!(BUTTON, user_interaction_state, user_input_state, button1_id, #=
... =#)

    if button1_value

        # ...

    end

    # draw button1...

    # other widgets...

    # ...

end

# ...

```

```
# ...

user_interaction_state = UserInteractionState(#= ... =#)

# ...

while !GLFW.WindowShouldClose(window)

    # ...

    button1_id = # ...

    button1_value = do_widget!(BUTTON, user_interaction_state, user_input_state, button1_id, #=
... =#)

    if button1_value
        # ...
    end

    # draw button1...

    # other widgets...

    # ...

end

# ...
```

```
# ...

user_interaction_state = UserInteractionState(#= ... =#)

# ...

while !GLFW.WindowShouldClose(window)

    # ...

    button1_id = # ...
    button1_value = do_widget!(BUTTON, user_interaction_state, user_input_state, button1_id, #=
... =#)
    if button1_value
        # ...
    end
    # draw button1...

    # other widgets...

    # ...

end

# ...
```

```
# ...

user_interaction_state = UserInteractionState(#= ... =#)

# ...

while !GLFW.WindowShouldClose(window)

    # ...

    button1_id = # ...
    button1_value = do_widget!(BUTTON, user_interaction_state, user_input_state, button1_id, #=
... =#)
    if button1_value
        # ...
    end
    # draw button1...

    # other widgets...

    # ...

end

# ...
```

```
struct WidgetID
```

```
    line::Int
```

```
end
```

```
# file.jl
```

```
# ...
```

```
button1_id = WidgetID(@__LINE__) # WidgetID(102)
```

```
# ...
```

```
button2_id = WidgetID(@__LINE__) # WidgetID(124)
```

```
# ...
```



```
# file1.jl
# ...
button1_id = WidgetID(@__LINE__) # WidgetID(102)
# ...
```

```
# file2.jl
# ...
button2_id = WidgetID(@__LINE__) # WidgetID(102)
# ...
```

```
struct WidgetID
    file::String
    line::Int
end
```

```
# file1.jl
```

```
# ...
```

```
button1_id = WidgetID(@__FILE__, @__LINE__) # WidgetID("/path/to/file1.jl", 102)
```

```
# ...
```

```
# file2.jl
```

```
# ...
```

```
button2_id = WidgetID(@__FILE__, @__LINE__) # WidgetID("/path/to/file2.jl", 102)
```

```
# ...
```

```
# file.jl
# ...

for i in 1:5
    button_i_id = WidgetID(@__FILE__, @__LINE__) # WidgetID("/path/to/file.jl", 102)
    # ...
end

# ...
```

```
struct WidgetID
    file::String
    line::Int
    instance::Int
end

# file.jl
# ...

for i in 1:5
    button_i_id = WidgetID(@__FILE__, @__LINE__, i) # WidgetID("/path/to/file.jl", 102, i)
    # ...
end

# ...
```

```

# ...

user_interaction_state = UserInteractionState(#= ... =#)

# ...

while !GLFW.WindowShouldClose(window)

    # ...

    button1_id = WidgetID(@__FILE__, @__LINE__, 1) # WidgetID("/path/to/file.jl", 102, 1)
    button1_value = do_widget!(BUTTON, user_interaction_state, user_input_state, button1_id, #=
... =#)

    if button1_value

        # ...

    end

    # draw button1...

    # other widgets...

    # ...

end

# ...

```

```
mutable struct UserInteractionState
    hot_widget::WidgetID
    active_widget::WidgetID
    null_widget::WidgetID
end

const NULL_WIDGET_ID = WidgetID("", 0, 0)
```

null_widget	button1	button2	button3
HOT			
ACTIVE			

null_widget	button1	button2	button3
HOT			
ACTIVE			



mouse_over_button1

null_widget	button1	button2	button3
HOT			
ACTIVE			

null_widget	button1	button2	button3
	HOT		
ACTIVE			

↓
mouse_over_button1

null_widget	button1	button2	button3
HOT			
ACTIVE			

null_widget	button1	button2	button3
	HOT		
ACTIVE			

↓ mouse_over_button1

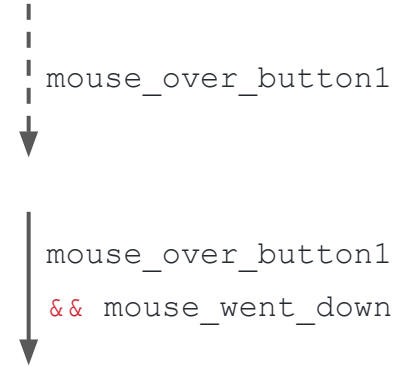
```

function try_set_hot_widget(hot_widget, active_widget, null_widget, this_widget, condition)
    if (hot_widget == null_widget) && (active_widget == null_widget) && condition
        return this_widget
    else
        return hot_widget
    end
end
end

```

null_widget	button1	button2	button3
HOT			
ACTIVE			

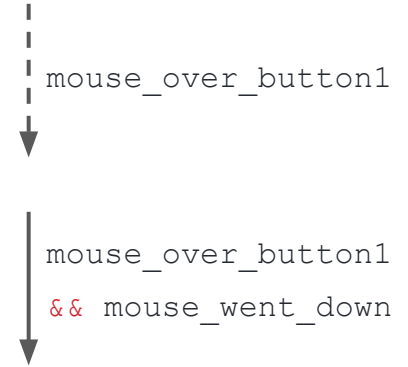
null_widget	button1	button2	button3
	HOT		
ACTIVE			



null_widget	button1	button2	button3
HOT			
ACTIVE			

null_widget	button1	button2	button3
	HOT		
ACTIVE			

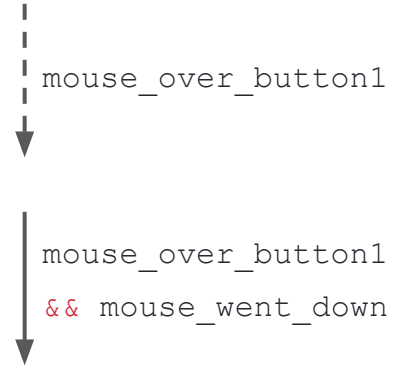
null_widget	button1	button2	button3
	HOT		
	ACTIVE		



null_widget	button1	button2	button3
HOT			
ACTIVE			

null_widget	button1	button2	button3
	HOT		
ACTIVE			

null_widget	button1	button2	button3
	HOT		
	ACTIVE		



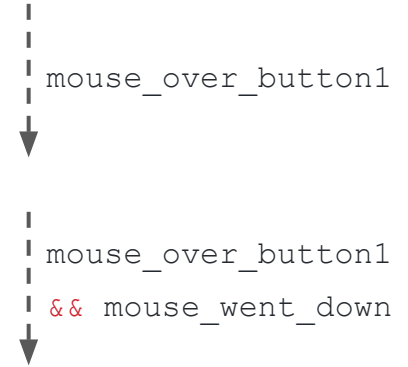
```

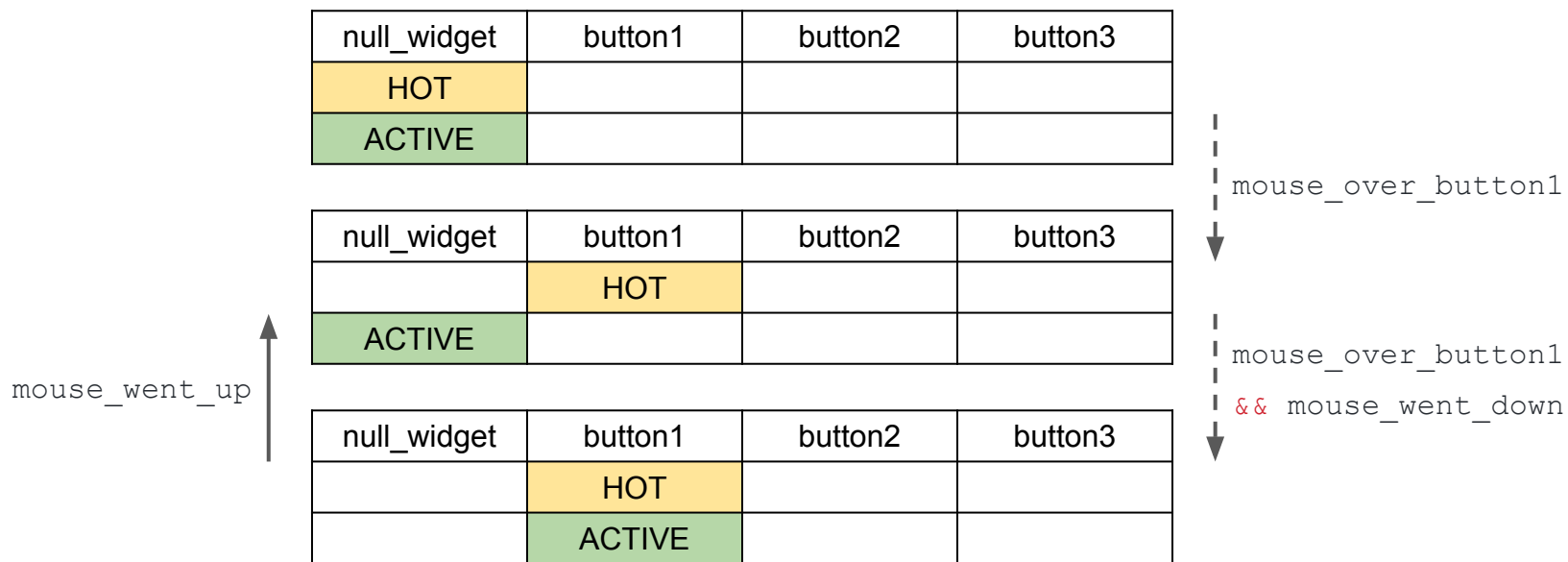
function try_set_active_widget(hot_widget, active_widget, null_widget, this_widget, condition)
  if (hot_widget == this_widget) && (active_widget == null_widget) && condition
    return this_widget
  else
    return active_widget
  end
end
end
  
```

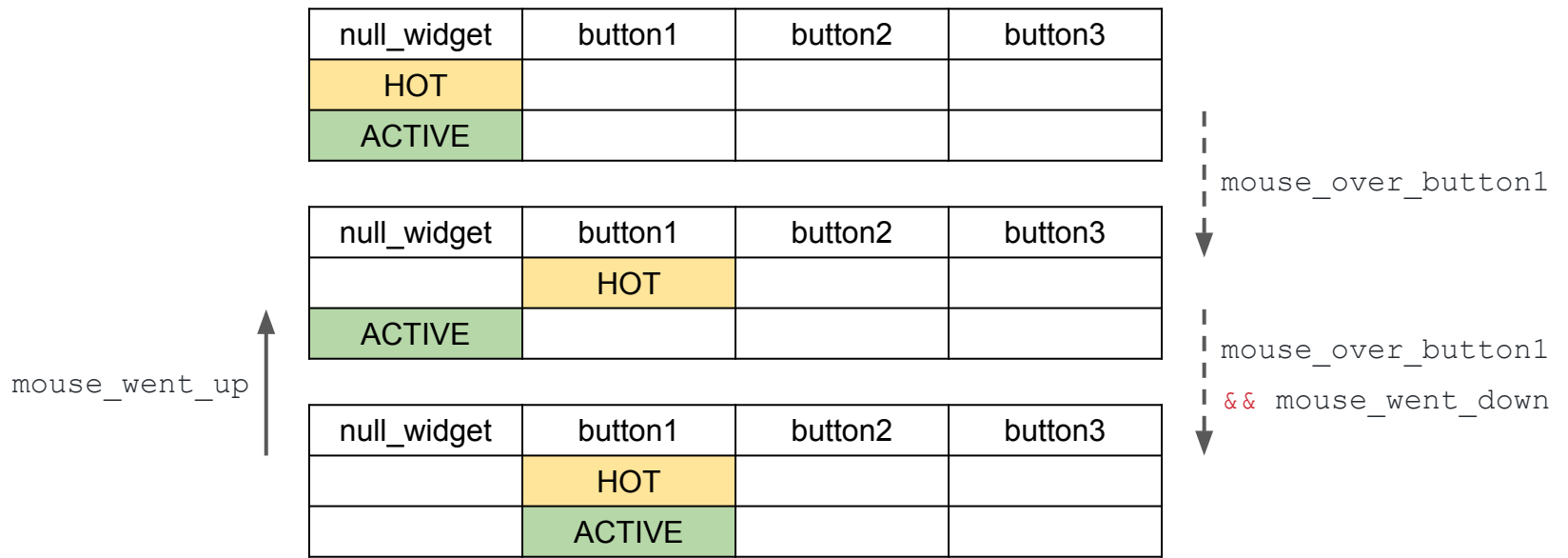
null_widget	button1	button2	button3
HOT			
ACTIVE			

null_widget	button1	button2	button3
	HOT		
ACTIVE			

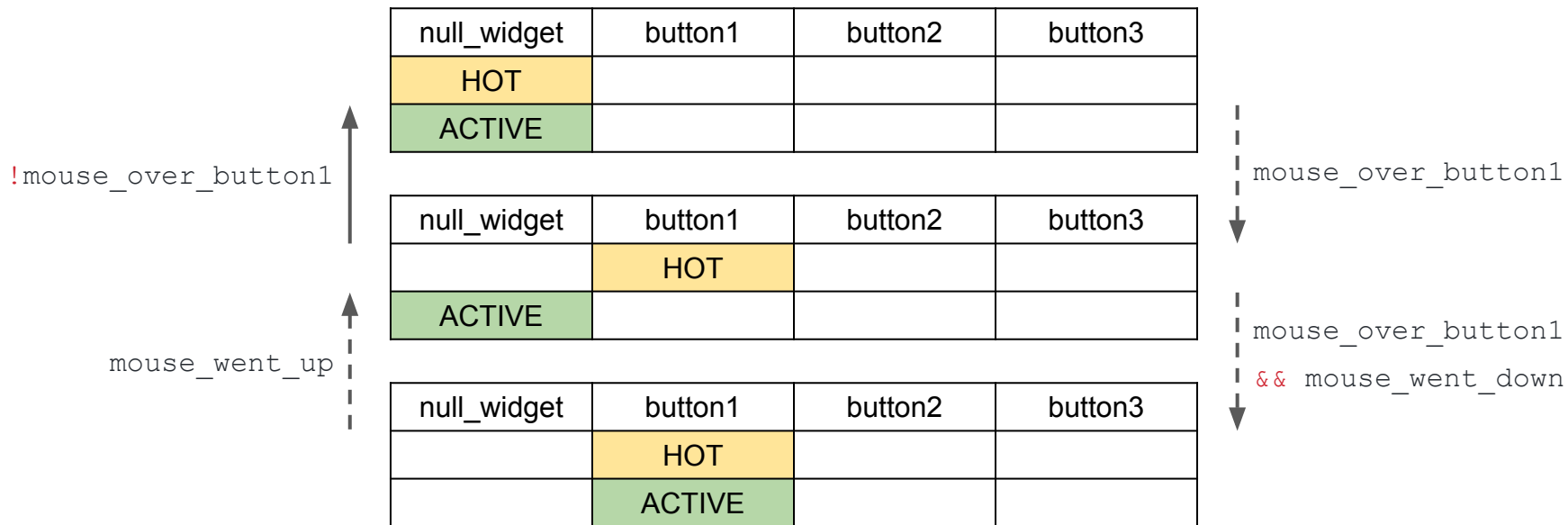
null_widget	button1	button2	button3
	HOT		
	ACTIVE		

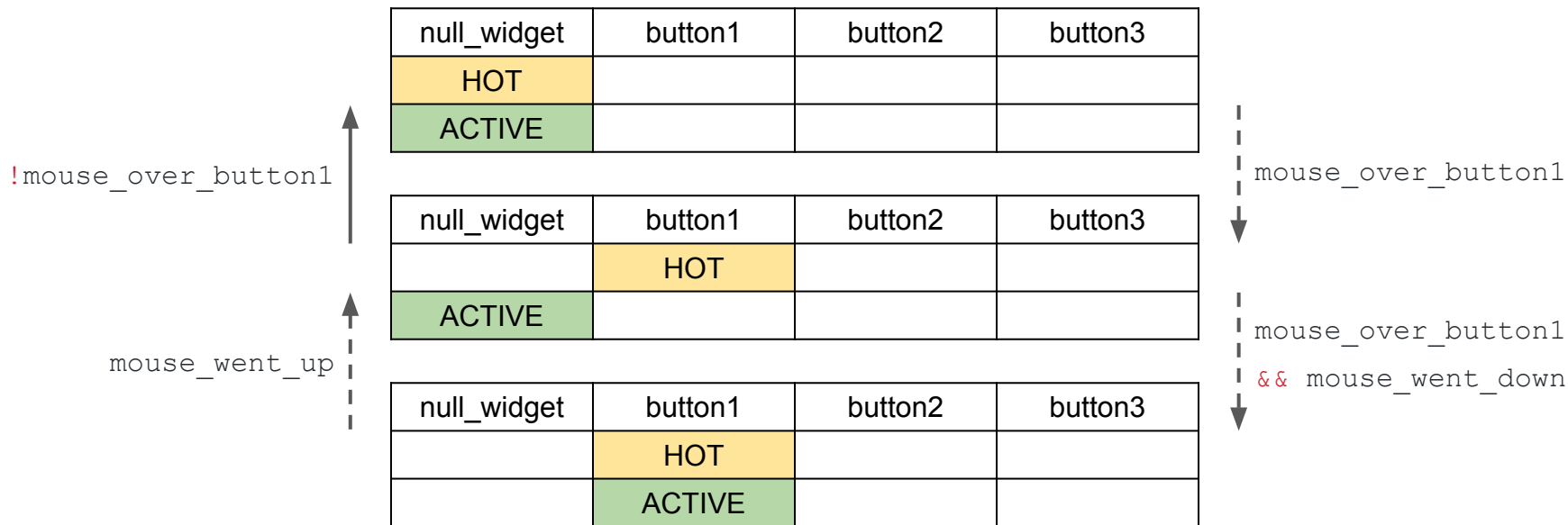




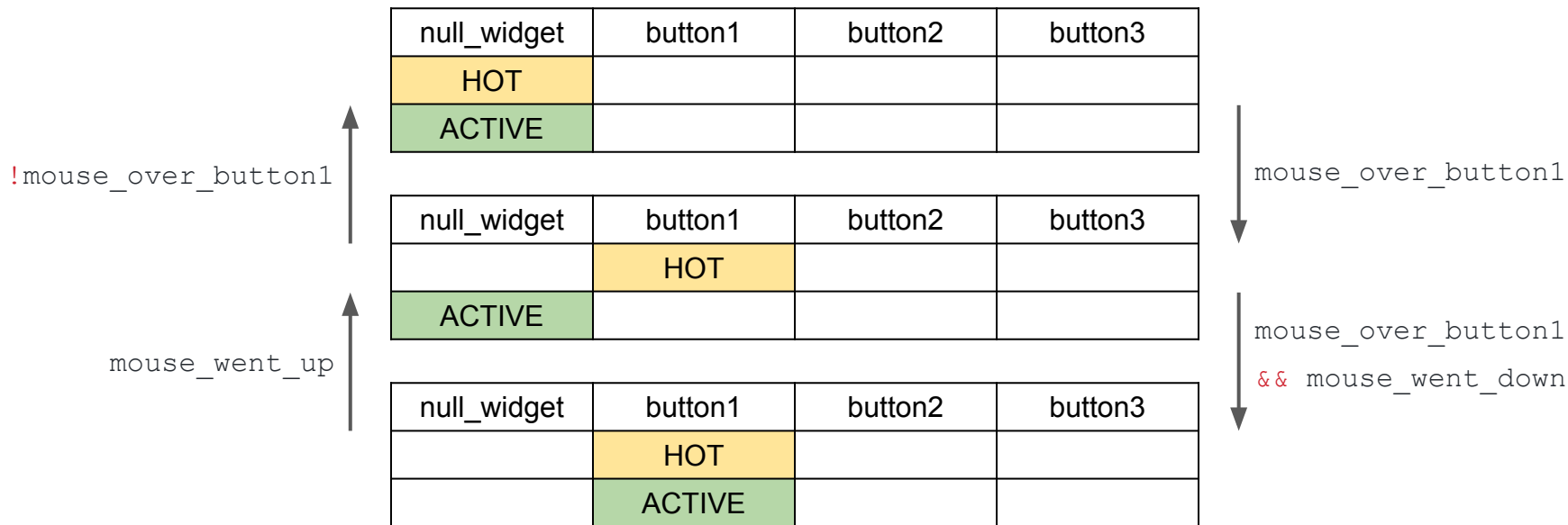


```
function try_reset_active_widget(hot_widget, active_widget, null_widget, this_widget, condition)
    if (hot_widget == this_widget) && (active_widget == this_widget) && condition
        return null_widget
    else
        return active_widget
    end
end
```

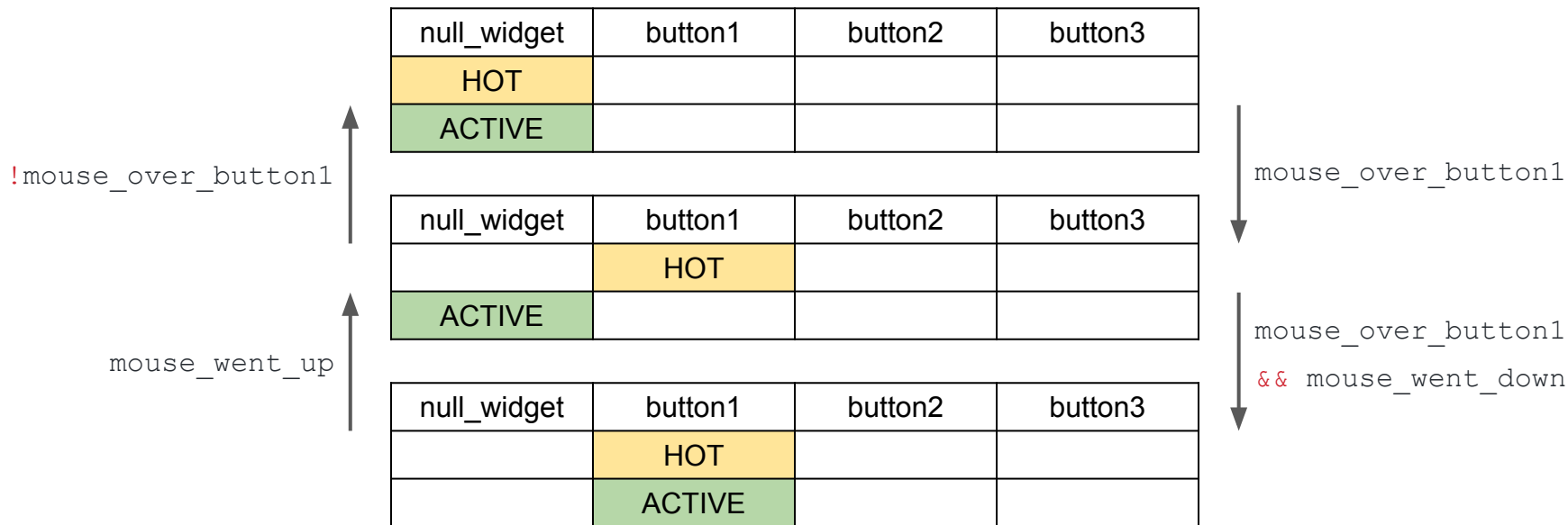





```
function try_reset_hot_widget(hot_widget, active_widget, null_widget, this_widget, condition)
  if (hot_widget == this_widget) && (active_widget == null_widget) && condition
    return null_widget
  else
    return hot_widget
  end
end
```



```
function get_widget_value(::Button, hot_widget, active_widget, this_widget, condition)
    if (hot_widget == this_widget) && (active_widget == this_widget) && condition
        return true
    else
        return false
    end
end
```



```
function get_widget_value(::Button, hot_widget, active_widget, this_widget, condition)
    if (hot_widget == this_widget) && (active_widget == this_widget) && condition
        return true
    else
        return false
    end
end
end
```

↑
`mouse_over_button1`
`&& mouse_went_up`

```
function do_widget(widget_type::Button, hot_widget, active_widget, null_widget, this_widget,  
i_min, j_min, i_max, j_max, i_mouse, j_mouse, ended_down, num_transitions)  
  
    # ...  
end
```

```
function do_widget(widget_type::Button, hot_widget, active_widget, null_widget, this_widget,  
i_min, j_min, i_max, j_max, i_mouse, j_mouse, ended_down, num_transitions)  
  
    mouse_over_widget = (i_min <= i_mouse <= i_max) && (j_min <= j_mouse <= j_max)  
  
    # ...  
end
```

```
function do_widget(widget_type::Button, hot_widget, active_widget, null_widget, this_widget,  
i_min, j_min, i_max, j_max, i_mouse, j_mouse, ended_down, num_transitions)  
  
    mouse_over_widget = (i_min <= i_mouse <= i_max) && (j_min <= j_mouse <= j_max)  
    mouse_went_down = went_down(ended_down, num_transitions)  
    mouse_went_up = went_up(ended_down, num_transitions)  
  
    # ...  
end
```

```
function do_widget(widget_type::Button, hot_widget, active_widget, null_widget, this_widget,  
i_min, j_min, i_max, j_max, i_mouse, j_mouse, ended_down, num_transitions)  
  
    mouse_over_widget = (i_min <= i_mouse <= i_max) && (j_min <= j_mouse <= j_max)  
    mouse_went_down = went_down(ended_down, num_transitions)  
    mouse_went_up = went_up(ended_down, num_transitions)  
  
    hot_widget = try_set_hot_widget(hot_widget, active_widget, null_widget, this_widget,  
mouse_over_widget)  
  
    # ...  
end
```



```
function do_widget(widget_type::Button, hot_widget, active_widget, null_widget, this_widget,  
i_min, j_min, i_max, j_max, i_mouse, j_mouse, ended_down, num_transitions)  
  
    mouse_over_widget = (i_min <= i_mouse <= i_max) && (j_min <= j_mouse <= j_max)  
    mouse_went_down = went_down(ended_down, num_transitions)  
    mouse_went_up = went_up(ended_down, num_transitions)  
  
    hot_widget = try_set_hot_widget(hot_widget, active_widget, null_widget, this_widget,  
mouse_over_widget)  
  
    active_widget = try_set_active_widget(hot_widget, active_widget, null_widget, this_widget,  
mouse_over_widget && mouse_went_down)  
  
    # ...  
end
```

```

function do_widget(widget_type::Button, hot_widget, active_widget, null_widget, this_widget,
i_min, j_min, i_max, j_max, i_mouse, j_mouse, ended_down, num_transitions)

    mouse_over_widget = (i_min <= i_mouse <= i_max) && (j_min <= j_mouse <= j_max)
    mouse_went_down = went_down(ended_down, num_transitions)
    mouse_went_up = went_up(ended_down, num_transitions)

    hot_widget = try_set_hot_widget(hot_widget, active_widget, null_widget, this_widget,
mouse_over_widget)

    active_widget = try_set_active_widget(hot_widget, active_widget, null_widget, this_widget,
mouse_over_widget && mouse_went_down)

    value = get_widget_value(widget_type, hot_widget, active_widget, this_widget,
mouse_over_widget && mouse_went_up)

    # ...
end

```

```

function do_widget(widget_type::Button, hot_widget, active_widget, null_widget, this_widget,
i_min, j_min, i_max, j_max, i_mouse, j_mouse, ended_down, num_transitions)

    mouse_over_widget = (i_min <= i_mouse <= i_max) && (j_min <= j_mouse <= j_max)
    mouse_went_down = went_down(ended_down, num_transitions)
    mouse_went_up = went_up(ended_down, num_transitions)

    hot_widget = try_set_hot_widget(hot_widget, active_widget, null_widget, this_widget,
mouse_over_widget)

    active_widget = try_set_active_widget(hot_widget, active_widget, null_widget, this_widget,
mouse_over_widget && mouse_went_down)

    value = get_widget_value(widget_type, hot_widget, active_widget, this_widget,
mouse_over_widget && mouse_went_up)

    active_widget = try_reset_active_widget(hot_widget, active_widget, null_widget, this_widget,
mouse_went_up)

    # ...
end

```

```

function do_widget(widget_type::Button, hot_widget, active_widget, null_widget, this_widget,
i_min, j_min, i_max, j_max, i_mouse, j_mouse, ended_down, num_transitions)

    mouse_over_widget = (i_min <= i_mouse <= i_max) && (j_min <= j_mouse <= j_max)
    mouse_went_down = went_down(ended_down, num_transitions)
    mouse_went_up = went_up(ended_down, num_transitions)

    hot_widget = try_set_hot_widget(hot_widget, active_widget, null_widget, this_widget,
mouse_over_widget)

    active_widget = try_set_active_widget(hot_widget, active_widget, null_widget, this_widget,
mouse_over_widget && mouse_went_down)

    value = get_widget_value(widget_type, hot_widget, active_widget, this_widget,
mouse_over_widget && mouse_went_up)

    active_widget = try_reset_active_widget(hot_widget, active_widget, null_widget, this_widget,
mouse_went_up)

    hot_widget = try_reset_hot_widget(hot_widget, active_widget, null_widget, this_widget,
!mouse_over_widget)

    # ...
end

```

```

function do_widget(widget_type::Button, hot_widget, active_widget, null_widget, this_widget,
i_min, j_min, i_max, j_max, i_mouse, j_mouse, ended_down, num_transitions)

    mouse_over_widget = (i_min <= i_mouse <= i_max) && (j_min <= j_mouse <= j_max)
    mouse_went_down = went_down(ended_down, num_transitions)
    mouse_went_up = went_up(ended_down, num_transitions)

    hot_widget = try_set_hot_widget(hot_widget, active_widget, null_widget, this_widget,
mouse_over_widget)

    active_widget = try_set_active_widget(hot_widget, active_widget, null_widget, this_widget,
mouse_over_widget && mouse_went_down)

    value = get_widget_value(widget_type, hot_widget, active_widget, this_widget,
mouse_over_widget && mouse_went_up)

    active_widget = try_reset_active_widget(hot_widget, active_widget, null_widget, this_widget,
mouse_went_up)

    hot_widget = try_reset_hot_widget(hot_widget, active_widget, null_widget, this_widget,
!mouse_over_widget)

    return hot_widget, active_widget, null_widget, value
end

```

```

function do_widget(widget_type::Button, hot_widget, active_widget, null_widget, this_widget,
i_min, j_min, i_max, j_max, i_mouse, j_mouse, ended_down, num_transitions)

    mouse_over_widget = (i_min <= i_mouse <= i_max) && (j_min <= j_mouse <= j_max)
    mouse_went_down = went_down(ended_down, num_transitions)
    mouse_went_up = went_up(ended_down, num_transitions)

    hot_widget = try_set_hot_widget(hot_widget, active_widget, null_widget, this_widget,
mouse_over_widget)

    active_widget = try_set_active_widget(hot_widget, active_widget, null_widget, this_widget,
mouse_over_widget && mouse_went_down)

    value = get_widget_value(widget_type, hot_widget, active_widget, this_widget,
mouse_over_widget && mouse_went_up)

    active_widget = try_reset_active_widget(hot_widget, active_widget, null_widget, this_widget,
mouse_went_up)

    hot_widget = try_reset_hot_widget(hot_widget, active_widget, null_widget, this_widget,
!mouse_over_widget)

    return hot_widget, active_widget, null_widget, value
end

```

```

function do_widget(widget_type::Button, hot_widget, active_widget, null_widget, this_widget,
i_min, j_min, i_max, j_max, i_mouse, j_mouse, ended_down, num_transitions)

    mouse_over_widget = (i_min <= i_mouse <= i_max) && (j_min <= j_mouse <= j_max)
    mouse_went_down = went_down(ended_down, num_transitions)
    mouse_went_up = went_up(ended_down, num_transitions)

    hot_widget = try_set_hot_widget(hot_widget, active_widget, null_widget, this_widget,
mouse_over_widget)

    active_widget = try_set_active_widget(hot_widget, active_widget, null_widget, this_widget,
mouse_over_widget && mouse_went_down)

    value = get_widget_value(widget_type, hot_widget, active_widget, this_widget,
mouse_over_widget && mouse_went_up)

    active_widget = try_reset_active_widget(hot_widget, active_widget, null_widget, this_widget,
mouse_went_up)

    hot_widget = try_reset_hot_widget(hot_widget, active_widget, null_widget, this_widget,
!mouse_over_widget)

    return hot_widget, active_widget, null_widget, value
end

```

```

function do_widget(widget_type::Slider, hot_widget, active_widget, null_widget, this_widget,
i_min, j_min, i_max, j_max, i_mouse, j_mouse, ended_down, num_transitions, last_value)

    mouse_over_widget = (i_min <= i_mouse <= i_max) && (j_min <= j_mouse <= j_max)
    mouse_went_down = went_down(ended_down, num_transitions)
    mouse_went_up = went_up(ended_down, num_transitions)

    hot_widget = try_set_hot_widget(hot_widget, active_widget, null_widget, this_widget,
mouse_over_widget)

    active_widget = try_set_active_widget(hot_widget, active_widget, null_widget, this_widget,
mouse_over_widget && mouse_went_down)

    value = get_widget_value(widget_type, hot_widget, active_widget, this_widget, clamp(j_mouse -
j_min + one(j_min), zero(j_min), j_max - j_min + one(j_min)), last_value)

    active_widget = try_reset_active_widget(hot_widget, active_widget, null_widget, this_widget,
mouse_went_up)

    hot_widget = try_reset_hot_widget(hot_widget, active_widget, null_widget, this_widget,
!mouse_over_widget)

    return hot_widget, active_widget, null_widget, value
end

```



```

function do_widget(widget_type::Slider, hot_widget, active_widget, null_widget, this_widget,
i_min, j_min, i_max, j_max, i_mouse, j_mouse, ended_down, num_transitions, last_value)

    mouse_over_widget = (i_min <= i_mouse <= i_max) && (j_min <= j_mouse <= j_max)
    mouse_went_down = went_down(ended_down, num_transitions)
    mouse_went_up = went_up(ended_down, num_transitions)

    hot_widget = try_set_hot_widget(hot_widget, active_widget, null_widget, this_widget,
mouse_over_widget)

    active_widget = try_set_active_widget(hot_widget, active_widget, null_widget, this_widget,
mouse_over_widget && mouse_went_down)

    value = get_widget_value(widget_type, hot_widget, active_widget, this_widget, clamp(j_mouse -
j_min + one(j_min), zero(j_min), j_max - j_min + one(j_min)), last_value)

    active_widget = try_reset_active_widget(hot_widget, active_widget, null_widget, this_widget,
mouse_went_up)

    hot_widget = try_reset_hot_widget(hot_widget, active_widget, null_widget, this_widget,
!mouse_over_widget)

    return hot_widget, active_widget, null_widget, value
end

```

```

function do_widget!(widget_type::TextBox, hot_widget, active_widget, null_widget, this_widget,
i_min, j_min, i_max, j_max, i_mouse, j_mouse, ended_down, num_transitions, text, characters)

    mouse_over_widget = (i_min <= i_mouse <= i_max) && (j_min <= j_mouse <= j_max)
    mouse_went_down = went_down(ended_down, num_transitions)
    mouse_went_up = went_up(ended_down, num_transitions)

    hot_widget = try_set_hot_widget(hot_widget, active_widget, null_widget, this_widget,
mouse_over_widget)

    active_widget = try_set_active_widget(hot_widget, active_widget, null_widget, this_widget,
mouse_over_widget && mouse_went_up)

    value = get_widget_value!(widget_type, hot_widget, active_widget, this_widget, text,
characters)

    active_widget = try_reset_active_widget(hot_widget, active_widget, null_widget, this_widget,
!mouse_over_widget && mouse_went_up)

    hot_widget = try_reset_hot_widget(hot_widget, active_widget, null_widget, this_widget,
!mouse_over_widget)

    return hot_widget, active_widget, null_widget, value
end

```

```

function do_widget!(widget_type::TextBox, hot_widget, active_widget, null_widget, this_widget,
i_min, j_min, i_max, j_max, i_mouse, j_mouse, ended_down, num_transitions, text, characters)

    mouse_over_widget = (i_min <= i_mouse <= i_max) && (j_min <= j_mouse <= j_max)
    mouse_went_down = went_down(ended_down, num_transitions)
    mouse_went_up = went_up(ended_down, num_transitions)

    hot_widget = try_set_hot_widget(hot_widget, active_widget, null_widget, this_widget,
mouse_over_widget)

    active_widget = try_set_active_widget(hot_widget, active_widget, null_widget, this_widget,
mouse_over_widget && mouse_went_up)

    value = get_widget_value!(widget_type, hot_widget, active_widget, this_widget, text,
characters)

    active_widget = try_reset_active_widget(hot_widget, active_widget, null_widget, this_widget,
!mouse_over_widget && mouse_went_up)

    hot_widget = try_reset_hot_widget(hot_widget, active_widget, null_widget, this_widget,
!mouse_over_widget)

    return hot_widget, active_widget, null_widget, value
end

```

Thank You!

Siddharth Bhatia (GitHub: @Sid-Bhatia-0)

<https://github.com/Sid-Bhatia-0/JuliaCon2022Talk>

<https://github.com/Sid-Bhatia-0/SimpleIMGUI.jl>