

Game Dev: UI Input & Events

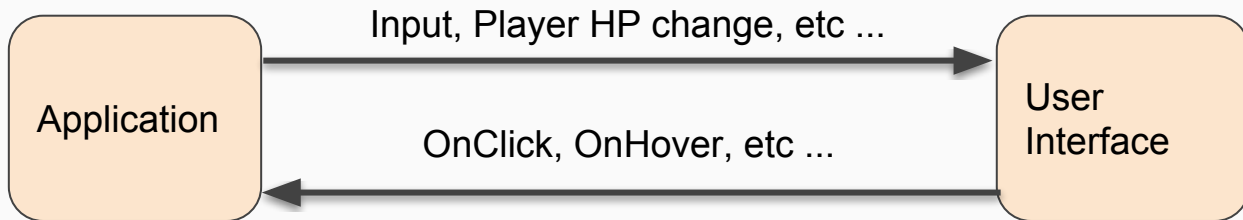
Ricard Pillosu - UPC



Input & UI

Some basics:

- Have debug functionality that draw all rectangles that the UI uses (use F8)
- Remember: there must be communication between UI and other modules



Input & UI

- We want our UI to react to Input Events in different ways:
 - When mouse clicks on the rectangle
 - When mouse enters/leaves the rectangle (“hover”)
 - What else would you add here ?
- How do we react to those events ?
 - Change the image of the UI (make it “shine”)
 - Execute some logic
 - What else ?

TODO 1

“Create an enum with all the possible events”

- Do not create events that happen every frame
- So MouseEnter and MouseLeave are good, MouseHover is bad

TODO 2

“Create a method to receive gui events”

- It is a method in the base Module class
- Should receive the ui element that triggered and the event type

TODO 3

“Create a listener variable on your basic UI class”

- We will use it to track which Module is listening to events
- Should we have one or a list of modules ?

TODO 4

“Check for mouse hover”

- First run a test every frame to see if the mouse is in our rectangle
- If it is, send the mouse_enter event
- Track that the mouse is still there, so you know when to send the mouse_leave event
- Test the functionality printing with LOG

TODO 5

“Now check for mouse clicks”

- We should only test for them if mouse is hovering our rectangle
- For now only test against left and right click
- LOG the event

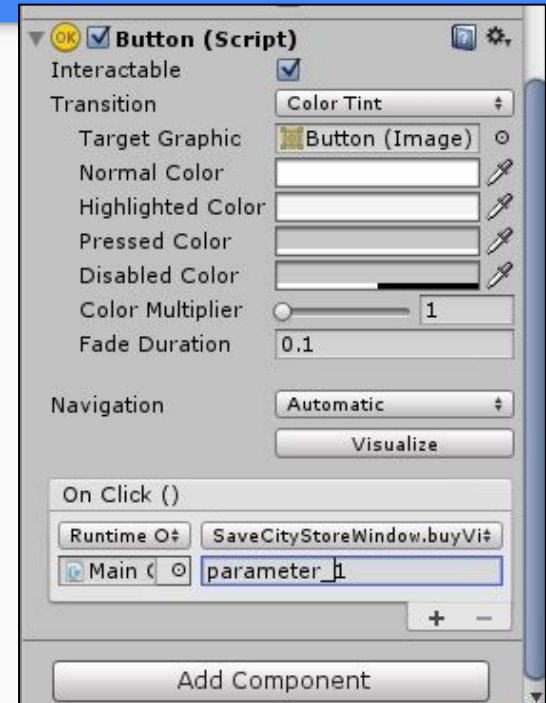
TODO 6

“Change the label text based on the event. Also change the button based on the hover / clicked state”

- Check solution.exe
- Normal Button coords are {0,113,229,69} - hover state
- Bright Button coords are {411,169,229,69} - click state
- Dark Button coords are {642,169,229,69} - normal state

Unity Button Widget

- Check Unity's way to create [UI buttons](#)
- It's simple and allows variety
- What can you replicate ?
- How will you notify other modules that a button has been pressed ?



Homework

- **UPDATE YOUR UML** with any structural change that you might have done
- You should be able to implement full buttons easily now
 - Try having the buttons with today's functionality embedded (3 sprites)
 - And capabilities to "colorize" any UI (check how rectangles are drawn to see color options)
- Enable that pressing TAB we visit each UI element and give the focus
- Add a new event to receive and lose focus