Game dev: UI Input Box

Ricard Pillosu - UPC

Work in groups

- Identify the new functionality
- Decide what needs to be changed in the UML

New functionality

1. New UI: Input Box
2. Concept of **focus**: even if the mouse if away, once clicked it keeps writing
3. Detecting special characters with the keyboard (altgr + 2 = @)
4. Handle of the text cursor: arrow keys, backspace, delete

New functionality

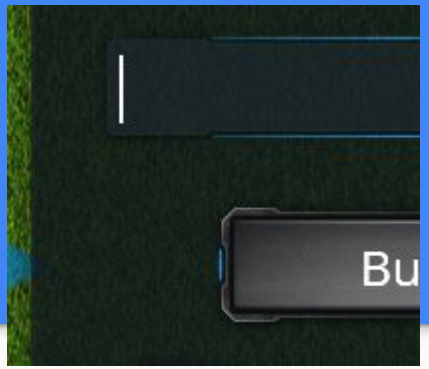
1. Input Box
    - New UI element with a image and a label
2. Concept of **focus**
    - Add new events and receive focus when clicked
3. Special characters
    - SDL can do this for us: TextInput API
4. Text cursor
    - We manually detect all those events and move the cursor or change the string

TODO 1

“Create a new InputText UI element with it's own label and image and draw it”

- The label and the image simply are childs of our new ui element
- But we should manually call their draw
- Try having this on the screen before moving forward

TODO 2



“Draw cursor when focus is received use a simple DrawQuad. For the size and position use App->font->CalcSize”

- When the *InputText* receives focus remove the default text, then draw the cursor
- Use *App->font->CalcSize* to calculate the height of the font and the position in X of the cursor. Remember that each letter can have different width!

TODO 3

“Add method to start / stop / get text from SDL_TextInput”

- In **ModuleInput** we should have method to enable / disable TextInput
- Read the SDL help page carefully
- You can ignore **SDL_SetTextInputRect** for now
- Now make sure to enable / disable SDL TextInput when UI element “*InputText*” receives focus

TODO 4

“Capture SDL_TEXTINPUT event. You can ignore SDL_TEXTEDITING for now read and store what you receive so you can return it”

- Check carefully the help for the event
- My recommendation is to ignore *SDL_TEXTEDITING* event, it seems to give inconsistent results (but happy to be proven wrong!)
- Now you should be able to receive simple text

TODO 5

“Calculate where the cursor has to be placed and update your label in the InputText ui element”

- All this code makes sense in ui inputtext Update()
- Remember that letters can have different width!
- Get from the **ModuleInput** the current text and draw it in your label
- Be nice and send an event every time the content changes

Homework

- Add code in Module Input to enable functionality for:
  - Backspace
  - Delete (Supr)
  - Array keys
  - Home
  - End