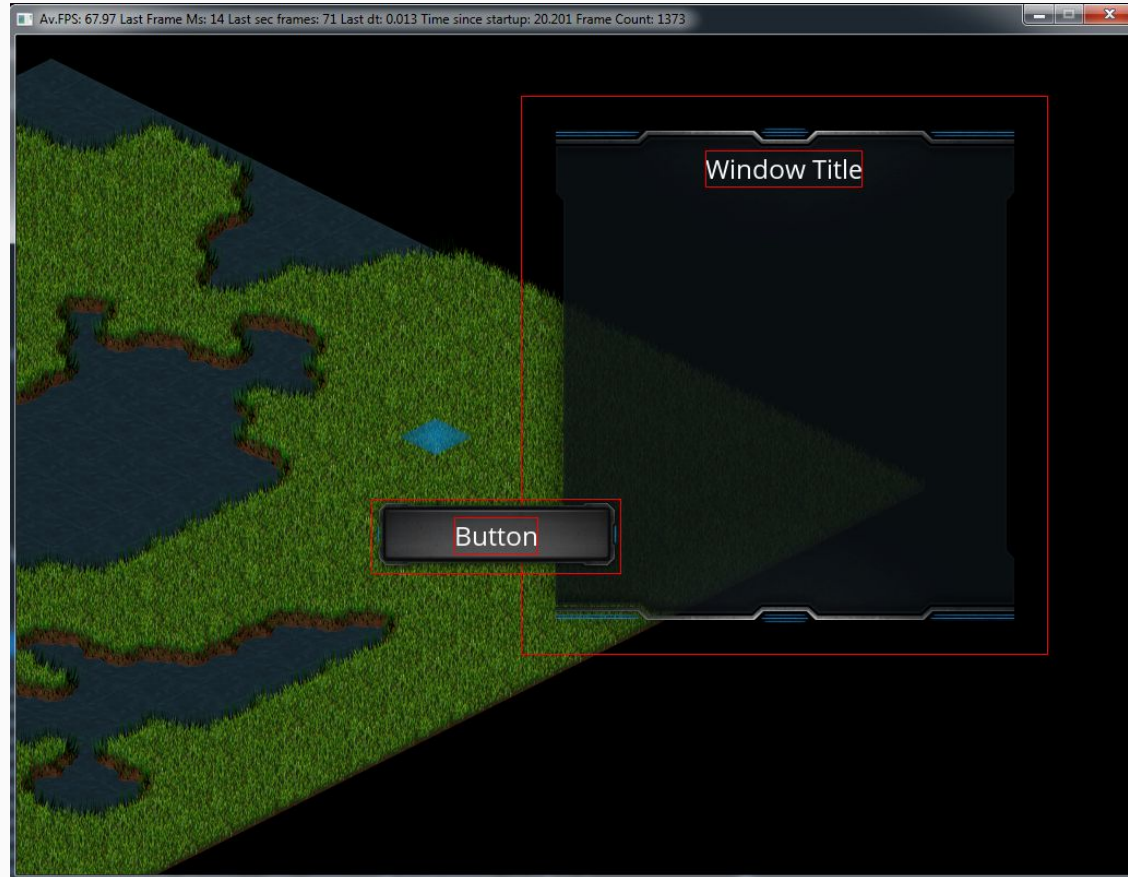


Game Development

UI Windows

Check solution.exe (press F8 for debug draw)



Work in groups

- What would you change for replicating this behavior ?
- Break down in groups and discuss:
 - What is the new functionality here ?
 - What is the **UML** structure to include this new functionality ?

Creating windows

- We have several challenges to overcome here:
 - How to nest UI
 - How to decide the order of UI from Input's POV
 - ... including "invisible" UI elements from Input's POV
 - Dragging elements with the mouse

Creating windows

- How to nest UI:
 - It's a **tree** :) ... but we will do a very simplistic implementation of a tree
- How to decide the order of UI from Input's POV
 - It is the inverse order of draw
- ... including “invisible” UI elements from Input's POV
 - All UI elements will have a “interactable” flag
- Dragging elements with the mouse
 - Flag “draggable” that just moves the rectangle with the mouse

TODO 1

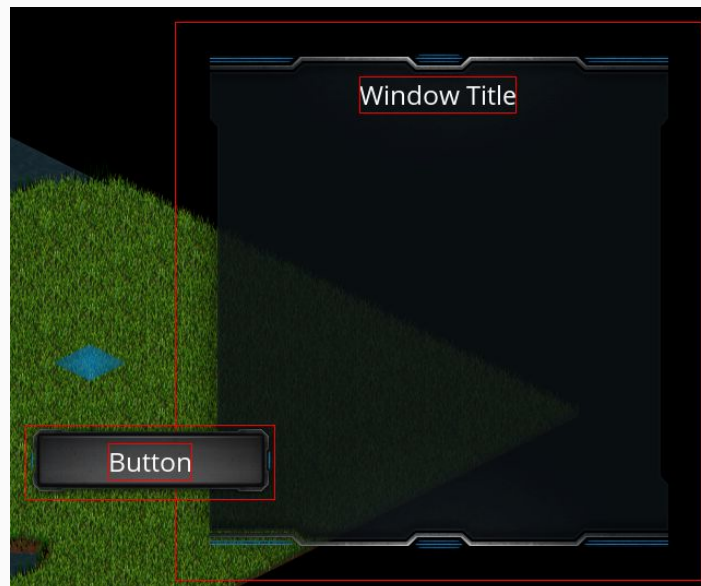
“To nest ui elements into each other simply add a parent”

- UI elements that have parent to **nullptr** mean they work with the screen
- Now each UI element has two set of coordinates: local and screen
- Actually, **Ui::rect** should be private!
- New methods: *GetScreenRect()* *GetLocalRect()* *GetScreenPos()*
GetLocalPos() *SetLocalPos()*
- Start with *GetScreenPos()*, then adapt the rest of the code

TODO 2

*“Make the **window** image be parent of the **button** image”*

- Then the button label have the button as parent
- And a new window title that has the window as parent



window sprite section in the atlas: {0, 512, 483, 512}

TODO 3

“Make any UI element draggable”

- Just add a flag when you are testing for events
- Then simply **add mouse motion** to UI position
- You should see that you can drag the window
- ... but what about the button ? they use the same screen area!

TODO 4

“Create a method that decides which element is under the mouse”

- It should just return the element under mouse cursor
- It can be only one!
- If mouse is under the button, only the button should be selected and not the window

TODO 5

“Now only one UI element may have the mouse hovering them at once”

- Replace your logic of “mouse inside UI rectangle”
- It should be “Am I the UI element that has the mouse ? ask UI Module”
- The result should be that you can drag things independently

TODO 6

“Add a flag to make each ui element interactive or not”

- The flag should make the UI element not be candidate for mouse hover
- ... and ignore all other Input
- When finished make the label of the button not interactable
- ... so you can now drag the button from the center

Homework - UPDATE YOUR UML!

- Add “masks” to UI elements:
 - No child element can be seen outside that window
 - The result would be that the button is not drawn outside the window
- Add focus system:
 - Only some UI can receive focus
 - Pressing TAB should cycle

