Graphic User Interface (GUI) : technical implementation

Romain PEREIRA

29th October 2017

This draft on the vast topic of graphical interfaces is only meant to document my VoxelEngine. I do not pretend this to be a formal document! Still, I will try to stick to a standard scientific publication format, to make readable and interesting.
Thank you for your attention, have a nice reading, and feel free to report me any mistakes I could have written.

Summary

Abstract

• 'states the question investigated and the principal objectives of the investigation,

• When developping my game engine, I faced the problem of implementing graphic user interfaces (GUI) I wanted to implement a robust but still flexible GUI system, so that I (or anyone else), can re-use it and customize it easily on any other OpenGL projects.

• So far, I have implemented a this GUI library: it handles input events and the entire rendering pipeline, in modern OpenGL (4.0)

• ' summarizes your results, and states your major conclusions.

1   Introduction

In this document, I will explain how I've implemented the graphic user interface system in my VoxelEngine. There is not many prerequisites to understand this paper, only a basic knowledge of computer programming and modern graphic rendering should be enough.

2 Materials and methods

2.1 Components (attributes, rendering...)

2.1.1 Abstract GUI component definition

2.1.2 Colored rectangle

2.1.3 Texture

2.1.4 Text (or Label)

2.1.5 Button

2.1.6 More complex components...

2.2 Inputs

2.2.1 Component states

2.2.2 Listeners and Events

2.2.3 Custom controllers

2.3 GUI manager

2.3.1 Controler

2.3.2 Renderer

3   General conclusion

References

[1] Wikipédia, *Détermination des surfaces cachées*,
https://fr.wikipedia.org/wiki/Détermination_des_surfaces_cachées,
Date d'accès: 22 Janvier 2017.