MainMenu

- startBounds : Vec4
- startTexture : GLuint
- quitBounds : Vec4[*]
- quitTexture : GLuint
- optionsBounds : Vec4
- optionsTexture : GLuint
- creditsBounds : Vec4
- creditsTexture : GLuint

- generateTexture(bounds : Vec4, texture : GLuint) : void
- renderTexture(bounds : Vec4, texture : GLuint) : void
+ getUIElementName(X : int, Y : int) : String

clickInfo

- location : Vec3
- objName : String

+ clickInfo() : void
+ getLocation() : Vec3
+ getObjName() : String

Camera

- position : Vec3
- front : Vec3
- up : Vec3
- target : Vec3
- rotation : Vec3
- direction : Vec3
- projection : Matrix4

+ move(target : vec3) : void
+ rotate(angle : float) : void
+ lookAt(target : Vec3,) : vec3
+ getposition() : Vec3
+ getRotation() : Vec3
+ getDirection() : Vec3
+ Update(deltatime : float) : void
+ getView() : Matrix4

UIEventHandler

+ clickWhat(mouseX : Int, mouseY : Int) : clickInfo

Scene

- renderablePLayers : list
- renderableObjects : list
- renderableTerrain : list

+ getClickOn(mouseX : int, mouseY : int) : String
- rayTrace(mouseX : Int, mouseY : Int, Camera Cam) : Vec3
- renderPLayers(players : list} : void
- renderObjects(objects : list) : void
- renderTerrain(terrain : list) : void

Scene Class Design and implementation will be later defined. This is a placeholder to demonstrate the relationship with the event handler.

GameLoopEventHandler

The Eventhandler will check input and on mouse click, it will take the mouse's coordinated and compare it with the UI elements. It will then return The UI element's name, or raytrace for an object in the worldspace and return the name of that object. The game loop can use this name (or ID) to determine what action to take.