

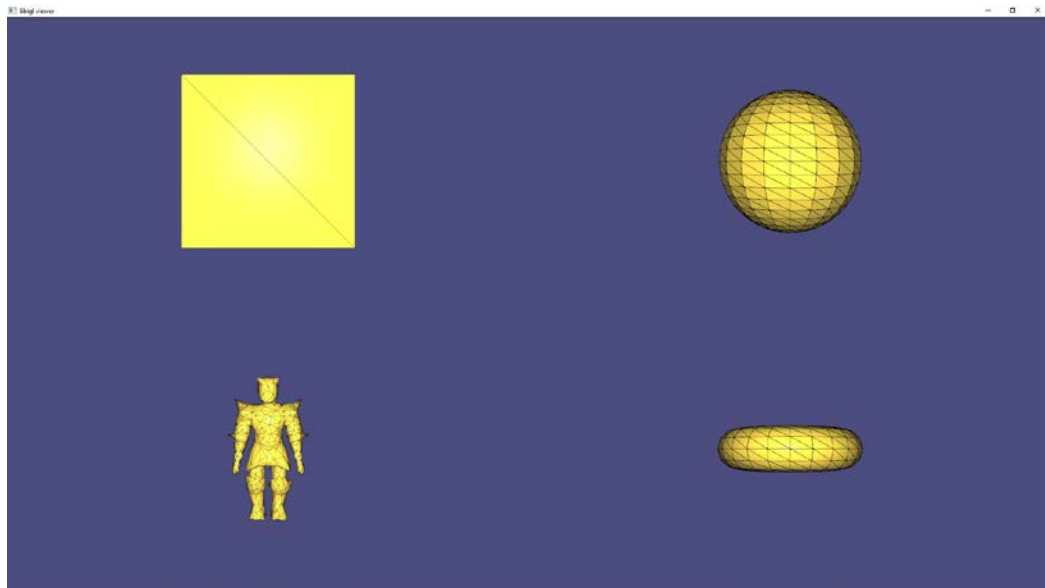
Εργασία 4

στο μάθημα "Γραφικά με Υπολογιστές"

Καταληκτική ημερομηνία παράδοσης : **Τετάρτη 9 Ιουνίου 2021, ώρα 23:55**

Να δημιουργηθεί αρχείο κώδικα (.cpp) το οποίο θα παράγει ένα παράθυρο διαπροσωπείας χρήστη, όπως παράγει ο viewer της LibIgl, για το οποίο θα ικανοποιούνται τα εξής:

- 1) Τέσσερα (4) διαφορετικά viewports (α) επάνω-αριστερά, β) επάνω-δεξιά, γ) κάτω-αριστερά και δ) κάτω-δεξιά, τα οποία να έχουν χωρική ανάλυση ίση με 640x400. Όπως φαίνεται στο Σχήμα 1, σε κάθε viewport θα είναι δυνατόν να φορτώνεται ένα διαφορετικό 3D αντικείμενο της επιλογής σας (.off αρχείο) και θα υπάρχει η δυνατότητα να φορτωθούν νέα αντικείμενα σε κάθε viewport. Κάθε φορά θα πρέπει να υπάρχει μόνο ένα αντικείμενο σε κάθε viewport.
- 2) Με αριστερό κλικ του ποντικιού επάνω σε ένα από τα αντικείμενα από τα διαφορετικά viewports (βλ. συνάρτηση `igl::unproject_onto_mesh` - παράδειγμα 206) να επιλέγεται ένα πρόσωπο `fid` και η κοντινότερη κορυφή `vid` από την θέση του ποντικιού. Για την επιλεγμένη κορυφή να εντοπίζεται η γειτονιά `1-ring()` και `2-ring()` των γειτονικών της προσώπων της οποίας τα πρόσωπα θα ζωγραφίζονται με χρώμα κόκκινο και πράσινο, αντίστοιχα. Επίσης, να εμφανίζονται τα κανονικά διανύσματα των προσώπων της `1-ring` και `2-ring` γειτονιάς (`igl::per_face_normals`) με χρώμα κίτρινο (1,1,0) ξεκινώντας από το βαρύκεντρο του κάθε προσώπου (`igl::barycenter`). Τέλος, είτε στο command prompt είτε σε παραγόμενο dialog box να παρουσιάζεται η μέτρηση της επιφάνειας του `fid`, της `1-ring` και της `2-ring` γειτονιάς σε σχέση με την συνολική επιφάνεια του αντικειμένου (`igl::doublearea`).



Σχήμα 1: Ενδεικτική διαπροσωπεία χρήστη με 4 viewports

Υποσημείωση 1: Για την εύρεση των γειτονικών κορυφών μίας επιλεγμένης κορυφής να χρησιμοποιηθεί η συνάρτηση `igl::adjacency_list` που έχετε διδαχθεί στο μάθημα, και για την

εύρεση των γειτονικών προσώπων μίας κορυφής να χρησιμοποιηθεί η συνάρτηση `igl::vertex_triangle_adjacency(V.rows(), F, VF, VI)` η οποία επιστρέφει ένα διάνυσμα VF με τα ids των γειτονικών προσώπων κάθε κορυφής (Το VI πρέπει να δοθεί σαν είσοδος ως έναν διάνυσμα διανυσμάτων με int αλλά δεν χρησιμεύει η έξοδος του).

Υποσημείωση 2: Με την εντολή `viewer.select_hovered_core()` ενημερώνεται το `viewer.selected_core_index` με το id του viewport που βρίσκεται το ποντίκι. Στη συνέχεια με την `viewer.core_list[viewer.selected_core_index]` λαμβάνεται το id του πυρήνα.

- 3) Με την βοήθεια της ImGui να δημιουργηθεί ένα μενού το οποίο θα περιέχει τα default εργαλεία (βλέπε παράδειγμα 106). Σε αυτό το μενού θα πρέπει να προστεθούν
- (α) ένα `ImGui::InputInt` με όνομα 'selectedViewport' το οποίο θα λαμβάνει τιμές {0,1,2,3} ως αναφορά στα αντικείμενα που βρίσκονται στα viewports α,β,γ,δ, αντίστοιχα,
 - (β) δύο `ImGui::Button` με ονόματα 'decimate' και 'subdivide' τα οποία θα εφαρμόζουν τα παραδείγματα 703 και 711 στο mesh που βρίσκεται στο 'selectedViewport',
 - (γ) δύο `ImGui::Checkbox` με ονόματα 'Gaussian Curvature' και 'Curvature Directions' τα οποία όταν είναι επιλεγμένα (μόνο ένα τη φορά) θα εφαρμόζουν πάνω στο επιλεγμένο αντικείμενο ('selectedViewport') τα παραδείγματα 202 και 203,
 - (δ) ένα `ImGui::Checkbox` με όνομα 'Surface classification' με την επιλογή του οποίου να υπολογίζεται το γινόμενο των ελάχιστων και μέγιστων καμπυλοτήτων $k_1 * k_2$ για κάθε κορυφή (παράδειγμα 203) και στη συνέχεια να αποδίδονται τρία διαφορετικά χρώματα της επιλογής σας σε κάθε κορυφή ανάλογα με την κατηγορία που εντάσσεται βάσει των κανόνων: για $k_1, k_2 > 0$ (ελλειπτικό), $k_1, k_2 < 0$ (υπερβολικό) και $k_1 = 0$ ή $k_2 = 0$ (παραβολικό).

Το παραδοτέο αρχείο θα πρέπει να έχει όνομα E4_AM.cpp όπου AM ο αριθμός μητρώου και να συμπιεσθεί σε αρχείο (.rar, .zip) για να υποβληθεί στο e-class.