# Creazione delle geometrie del pannello

assi cartesiani in 3 dimensioni x,y,z

verso positivo asse x: est

verso positivo asse y: nord

verso positivo asse z: altezza rispetto al suolo (normale al suolo nel punto dell’osservatore)

verso negativo asse x: ovest

verso negativo asse y: sud

verso negativo asse z:

per tutte le geometrie studiate abbiamo discretizzato la superficie del pannello in triangoli (secondo il metodo agli elementi finiti) in modo da creare algoritmi che permettessero di calcolarne i tre vertici assicurandoci che si trovassero tutti su uno stesso piano (infatti sappiamo che in tre dimensioni per tre punti passa uno ed un solo piano).Abbiamo perciò di studiare alcune geometrie utili che potessero permetterci di discretizzare la maggior parte delle superfici reali esistenti, ovvero: ondulati, rami di parabola ,cilindri, piani, piramidi, coni e sfere.

Per quanto riguarda la modellizzazione delle superfici prese in esame abbiamo usato due metodi di base per discretizzarli in triangoli. I primi cinque modelli sopra elencati sono superfici che chiameremo “di traslazione” ovvero sono figure piane che poi vengono traslate nello spazio per fornirle di tridimensionalità. Le restanti invece sono superfici che definiamo “di rotazione” in quanto sono figure che discretizziamo fissando uno o due vertici e generiamo i triangoli dividendo la base.

**[COME LI CREIAMO]**

La creazione delle superfici “di traslazione” avviane seguendo tre passaggi:

Sul piano x,z studio la curva (per esempio del tipo z=a\*x) che mi definisce la sezione della supeficie

Sul piano y,z do la profondità della superficie (ovvero idealmente traslo la sezione calcolata sul piano x,z nello spazio fino ad ottenere una superficie tridimensionale)

Raggruppo i punti calcolati sequenzialmente in insiemi di tre in modo da avere dei triangoli. Quello che otteniamo sono due triangoli che creano un rettangolo che individuano uno stesso piano.

La creazione delle superfici “di rotazione” avviane seguendo tre passaggi:

Si fissa un punto che è il vertice della nostra figura( nel caso di una sfera ne fisso 2 antipodali)

Preso il vertice come punto comune a tutti i triangoli, calcolo gli altri due vertici dividendo la circonferenza di base in segmenti (nel caso della sfera prendo come circonferenza l’unica equidistante dai due punti scelti nel punto precedente, ogni segmento indica la base per 2 triangoli che hanno rispettivamente come terzo vertice i due punti di prima. Concettualmente è come se incollassimo 2 coni per le basi)

A questo punto ogni terna di punti sopra definiti trova un triangolo.

Il passo successivo alla creazione delle superfici in modo “comodo” rispetto agli assi cartesiani ruotiamo il pannello per intero rispetto all’angolo (inclinazione rispetto al piano del suolo, ovvero rotazione attorno all’asse y) e all’angolo (rotazione rispetto all’asse y, cioè rotazione attorno all’asse z).

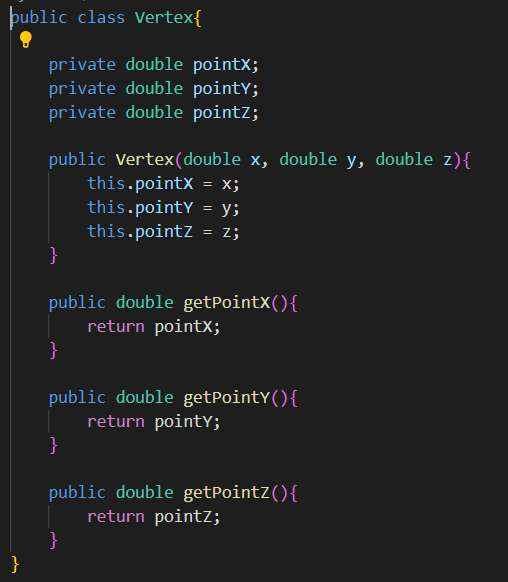
[PARTE DI ALE]

Per l’implementazione dei triangoli abbiamo deciso di utilizzare il linguaggio Java. È stato realizzato uno script capace di generare una lista di vertici appartenenti ai rispettivi triangoli a partire da input diversi per ogni tipo di figura geometrica.

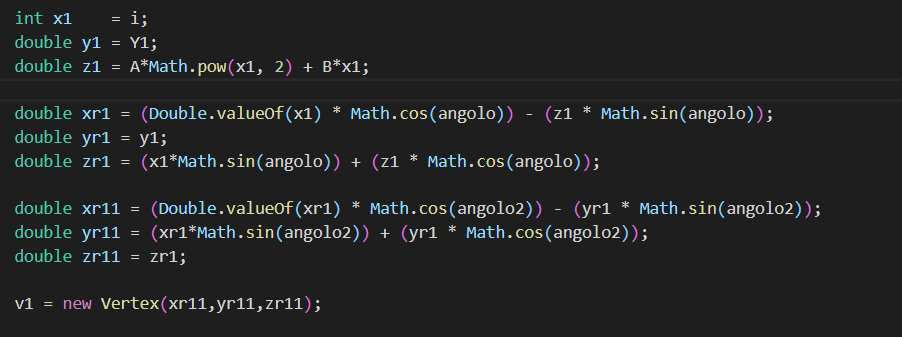
I vertici vengono scritti in un file con estensione csv in modo da poter essere letti facilmente dall’applicativo. Il file in questione è strutturato in questo modo:

* ci sono tre colonne, ogni colonna contiene il vertice del triangolo, dunque le posizioni x y z
* ci sono n righe corrispondenti ai triangoli

Abbiamo definito una classe Vertex relativa al vertice dei triangoli che ha la seguente struttura:



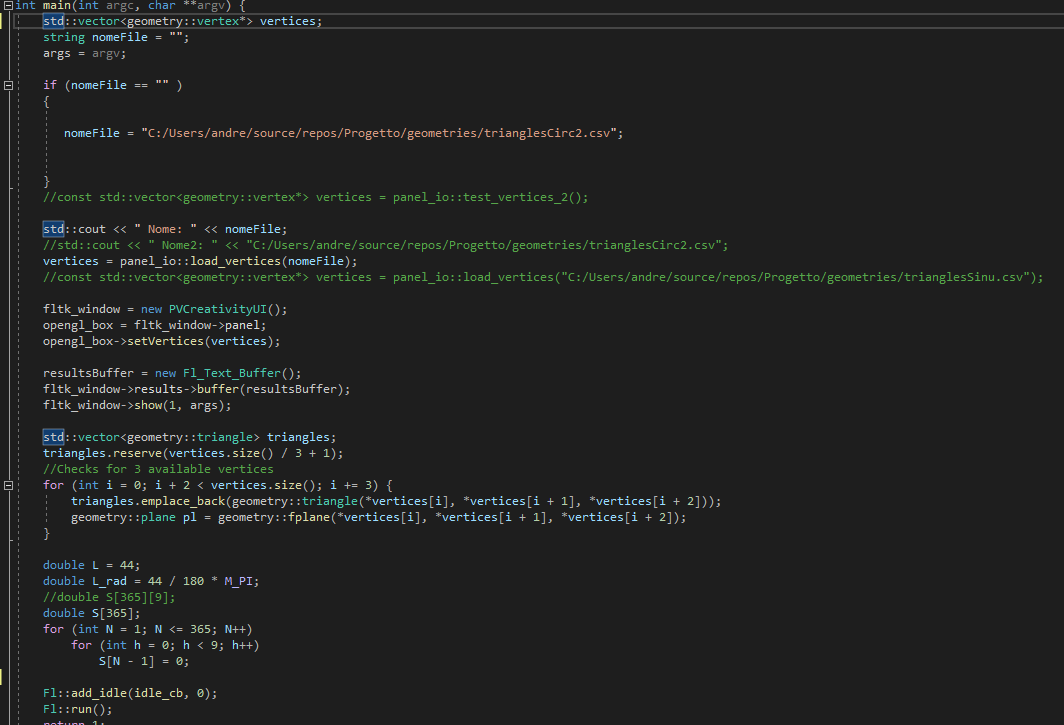
Lo script è composto da diversi metodi che corrispondono alle diverse figure geometriche, ogni metodo prende in input diversi valori. A seconda delle figure gli input cambiano. All’interno di un ciclo for il cui numero di cicli viene definito in input, vengono creati quattro vertici, a partire da questi quattro vertici vengono scritti nel file CSV due triangoli, dunque ogni ciclo genera due triangoli. Di seguito un esempio di creazione di vertice relativo alla parabola:



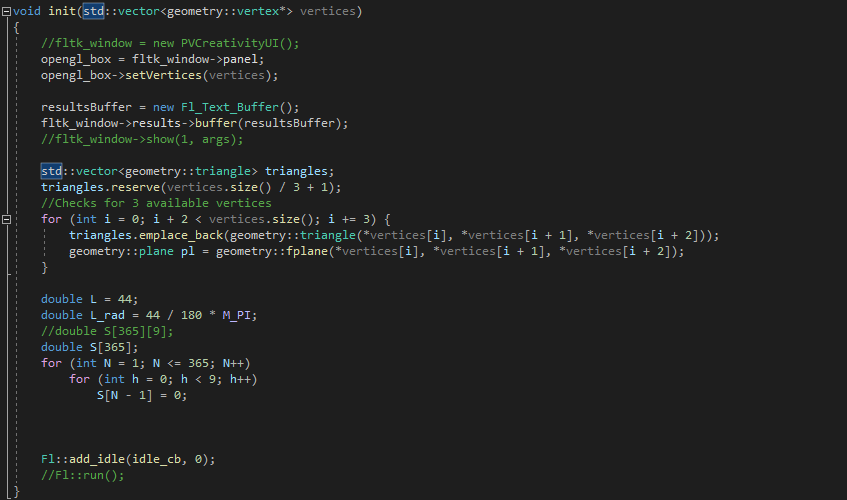
[Parte andrea]

Caricamento - Aggiornamento

Dopo aver definito alcune variabili globali in un header comune, si va ad assegnare dinamicamente il file pre-selezionato e si conclude con lo istanziare gli elementi atti a far visualizzare la forma.



Quando si va a selezionare il file il metodo init(vertices) viene chiamato dal main ma senza fare il Fl::run(). Basta solo aggiornare lo idle state e chiamare il metodo draw() del panel. Facendo così si puó far caricare al programma qualunque tipo di file senza interruzio



Per cercare di far vedere come la luce illumina i vari pannelli sono stati usati varie funzioni di GLUT. Dopo aver definito la posizione della luce bianca con l’esecuzione del programma si puó notare come la luce “casca” sulla superficie dei pannelli.

