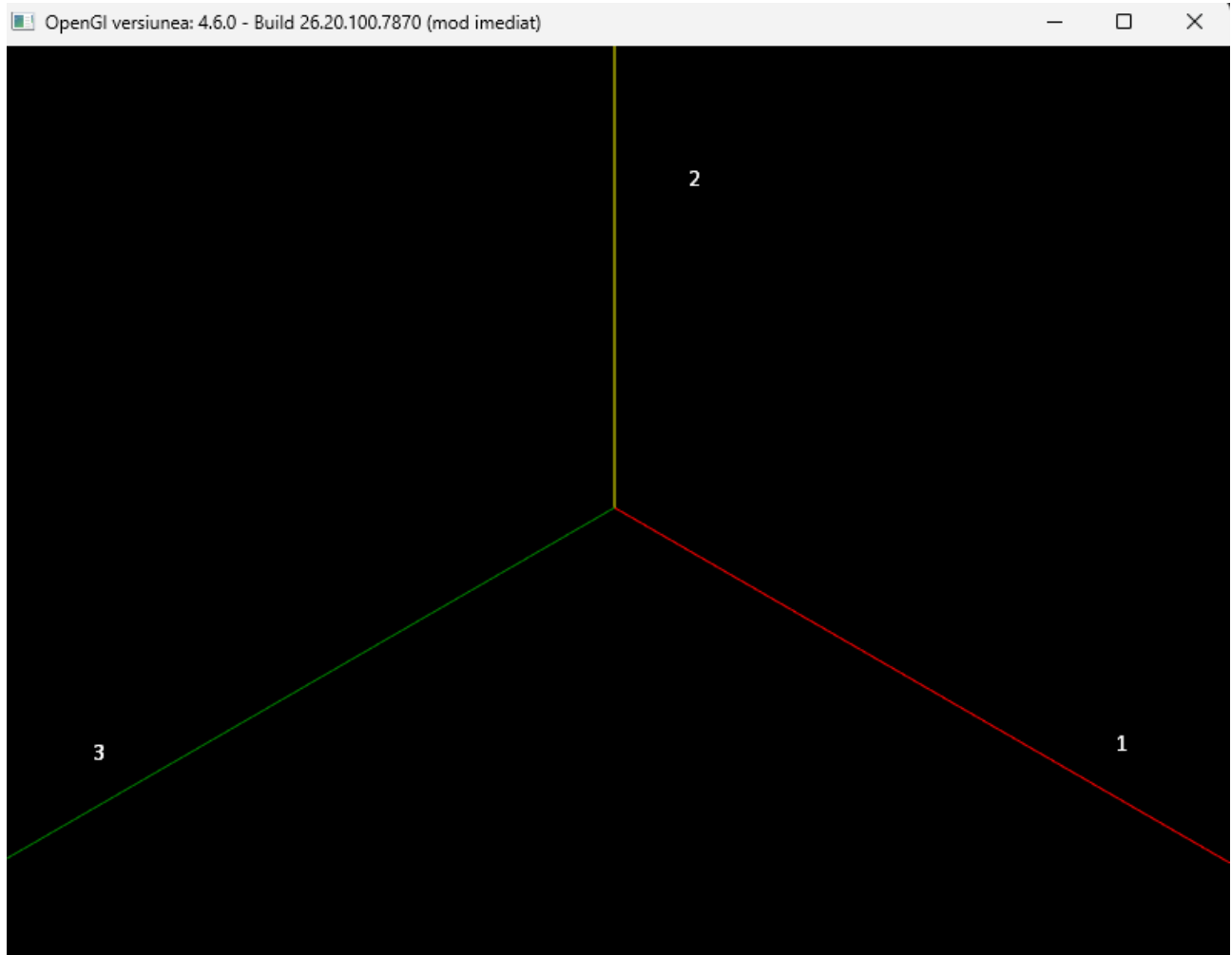


# Laborator 3 EGC

1. Ordinea de desenare a vertexurilor din aplicatia template este in anti-orar.



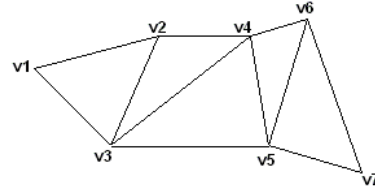
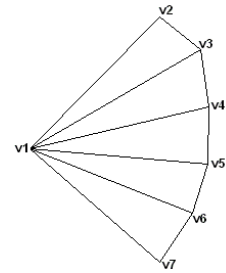
2. **Aliasingul** în imagini poate fi descris ca margini zimțate care se găsesc adesea în afișaje de rezoluție inferioară. Aceste margini zimțate sunt văzute deoarece monitorul sau alt dispozitiv de ieșire nu utilizează o rezoluție suficient de mare pentru a arăta linia netedă. **Anti-aliasing**, deci, este o tehnologie care încearcă să rezolve aliasing-ul găsit în imagine.

3. `GL.PointSize(float)` → seteaza marimea unui punct

`GL.LineWidth(float)` → seteaza grosimea unei linii

Aceste functii nu functioneaza in interiorul unei zone `GL.Begin` (insa nu va exista eroare de compilare), deoarece sunt functii care afecteaza starea globala (vor afecta toate obiectele de dupa apelarea lor).

4. a) Directiva LineLoop va conecta segmentele de dreapta “in mod continuu” incepand cu primul vertex si pana la ultimul, ultimul fiind conectat cu primul.
- b) Directiva LineStrip va crea un poligon “deschis” format din drepte introduse in ordinea lor.
- c) Directiva TriangleFan va crea o serie de triunghiuri cu un punct comun (primul introdus) ca in imagine:
- d) Directiva TriangleStrip creeaza o serie de triunghiuri asemanator ca in exemplul precedent, doar ca acestea sunt formate in ordinea introducerii punctelor.



6. Utilizarea de culori diferite este importanta pentru a face diferentierea intre vertex-uri (daca toate ar avea aceeasi culoare ar parea un singur vertex). Avantajul ar fi identificarea rapida a fiecarui vertex in cod (“search color”).
7. Gradientul de culoare reprezinta tranzitia de la o culoare la alta. In OpenGL gradientul se obtine setand diferite culori pentru punctele care formeaza un vertex.
8. Canalul de transparenta are rolul de a seta intensitatea unei culori respective.
10. Utilizarea unei culori diferite pentru fiecare vertex duce la crearea de gradient.