**1. Ce este un viewport?**

**Viewport-ul** este o regiune a ferestrei în care se realizează randarea grafică. În OpenGL, viewport-ul definește zona în pixeli din fereastra de ieșire unde vor fi afișate obiectele 3D după ce sunt procesate de pipeline-ul grafic. Setarea viewport-ului se face cu funcția GL.Viewport(x, y, width, height), unde (x, y) definesc colțul din stânga-jos al ferestrei, iar width și height sunt dimensiunile viewport-ului. Aceasta permite redimensionarea graficii astfel încât să se potrivească întotdeauna cu dimensiunile ferestrei.

**2. Ce reprezintă conceptul de frames per second (FPS) din punctul de vedere al bibliotecii OpenGL?**

**Frames per second (FPS)** reprezintă numărul de cadre randate de aplicație într-o secundă. În OpenGL, FPS-ul este o măsură a performanței și a eficienței aplicației de randare. Un FPS mai mare înseamnă o animație mai fluidă și o interacțiune mai rapidă. OpenGL nu gestionează în mod direct FPS-ul, ci depinde de cât de repede pot fi randate cadrele prin pipeline-ul grafic al GPU-ului și de sincronizarea aplicației (de exemplu, cu opțiunea VSync în OpenTK).

**3. Când este rulată metoda OnUpdateFrame()?**

Metoda OnUpdateFrame() este rulată înainte de OnRenderFrame() și este responsabilă de logica aplicației și de actualizarea stării. Aceasta este apelată o dată pentru fiecare cadru și este folosită pentru a actualiza pozițiile obiectelor, gestionarea intrărilor utilizatorului și alte aspecte ale logicii jocului sau aplicației. Frecvența apelării acestei metode depinde de valoarea specificată pentru **Update Rate** în apelul Run() din OpenTK.

**4. Ce este modul imediat de randare?**

**Modul imediat de randare** (immediate mode) este o metodă veche în OpenGL prin care se trimit direct comenzi individuale de desenare către pipeline-ul grafic. În acest mod, fiecare apel de funcție specifică un punct, o linie sau un triunghi separat, folosind comenzile GL.Begin() și GL.End(). De exemplu, în modul imediat, fiecare apel GL.Vertex3(x, y, z) definește un vârf. Deși este ușor de înțeles și implementat pentru aplicații simple, acest mod este ineficient pentru grafica complexă și a fost înlocuit cu moduri mai eficiente, cum ar fi **vertex buffer objects (VBO)** și **vertex array objects (VAO)**.

**5. Care este ultima versiune de OpenGL care acceptă modul imediat?**

Ultima versiune de OpenGL care acceptă oficial modul imediat este **OpenGL 3.0**. Începând cu OpenGL 3.1, modul imediat a fost depreciat și eliminat în versiunile de bază ale API-ului, fiind înlocuit cu tehnici mai moderne, cum ar fi VAO-uri și VBO-uri, care sunt mult mai eficiente pentru hardware-ul grafic modern.

**6. Când este rulată metoda OnRenderFrame()?**

Metoda OnRenderFrame() este apelată după OnUpdateFrame() și este responsabilă de randarea scenei 3D pe ecran. Aceasta este rulată o dată pentru fiecare cadru și este utilizată pentru a desena toate obiectele și pentru a actualiza imaginea afișată utilizatorului. În OpenTK, aceasta funcționează într-un mod "double-buffered", ceea ce înseamnă că o imagine este randată în fundal, iar când este completă, buffer-ul este schimbat cu cel afișat pentru o tranziție lină.

**7. De ce este nevoie ca metoda OnResize() să fie executată cel puțin o dată?**

Metoda OnResize() este folosită pentru a actualiza configurația viewport-ului și a matricii de proiecție în funcție de dimensiunea ferestrei. Aceasta este necesară pentru ca scena să fie scalată și poziționată corect în noua dimensiune a ferestrei, fără distorsiuni. La prima rulare, chiar dacă fereastra nu este redimensionată, această metodă este invocată pentru a stabili viewport-ul inițial și proiecția corectă.

**8. Ce reprezintă parametrii metodei CreatePerspectiveFieldOfView() și care este domeniul de valori pentru aceștia?**

Metoda CreatePerspectiveFieldOfView() creează o matrice de proiecție în perspectivă și are următorii parametri:

* **fovY (Field of View Y)**: Câmpul vizual vertical, specificat în radiani. Domeniul uzual este între aproximativ 45 de grade (0.785 radiani) și 90 de grade (1.57 radiani). Valori mai mici măresc efectul de zoom, în timp ce valori mai mari creează o vedere de ansamblu mai largă.
* **aspectRatio**: Raportul de aspect al ferestrei (lățime / înălțime). Asigură că scena nu este distorsionată și se adaptează la raportul de aspect al ferestrei, pentru ca obiectele să fie afișate proporțional.
* **nearPlane**: Planul apropiat de tăiere. Este distanța de la care începe să fie vizibilă scena. Valorile uzuale sunt în jur de 0.1 sau 1.0, dar trebuie să fie un număr pozitiv și mai mic decât farPlane.
* **farPlane**: Planul îndepărtat de tăiere. Este distanța maximă până la care sunt randate obiectele. Valorile uzuale sunt în jur de 100 sau mai mari, dar trebuie să fie mai mari decât nearPlane.