Mini-Sistem Solar în OpenGL

1. Prezentare generală
2. Implementare
3. Prezentare generală

Aplicația prezintă o simulare a unui Sistem Solar în miniatură ce expune patru planete: O stea centrală (Soare), două planete orbitale (Pământ și Venus) și o planetă de tip satelit (Luna) colorate cu un pattern corespunzător. Soarele execută o revoluție de 0.1˚ pe cadru, Pământ și Venus execută o rotație în jurul Soarelui de 0.1˚ respectiv -0.1˚ pe cadru, iar Luna execută o rotație în jurul Pământului de 0.4˚ pe cadru. Scena este interactibilă, permițând Observatorului să se deplaseze și orienteze în acest spațiu cu tastele W, A, S și D și, respectiv, mișcarea mouse-ului.

Convenții de notație interne: \*Pos – Punct în 3D; \*Dist – Vector în 3D; \*Dir – Vector normalizat în 3D.

1. Implementare

* Inițializarea scenei:

Programului îi este furnizat un fișier de tip „..obj” ce conține vârfurile și indicii pentru generarea unei sfere. Funcția „parseObj(const char\*, std::vector&, std::vector&)” preia calea spre fișierul menționat anterior și stochează în cei doi std::vectori vârfurile respectiv indicii pentru ordinea desenării.

Pentru generarea culorii planetelor se folosește funcția „generateRandomPlanetColoration(uint, Planet, std::vector&)” care furnizează în std::vectorul primit ca parametru culori generate aleator care imită pattern-ul din realitate al tipului de planetă primit ca parametru (de exemplu, pentru Soare se pune accent pe culorile galben și portocaliu, pentru Pământ, albastru și verde, pentru Venus, portocaliu și maro, iar pentru Lună, gri).

Pentru desen se declară un Vertex Array Object cu 6 Buffer-e: Un Buffer ce conține vârfurile sferei (asignat pe locația 0), patru Buffer-e ce conțin culorile pentru fiecare planetă (asignate pe locațiile 1, 2, 3 și 4, respectiv) și un Buffer ce conține indicii sferei pentru ordinea desenării.

În final, sunt inițializate unghiul de rotație al Soarelui, pozițiile planetelor în scenă, Matricea-Model (folosită în comun de toate obiectele din scenă), poziția, axa superioară și punctul de privire ale Observatorului precum și Matricea-Proiecție care setează un câmp de vedere de 60˚.

* Interactibilitatea scenei:

Pentru gestionarea input-ului de la utilizator, se asignează funcțiile KeyboardFunction(uchar, int, int) și PassiveMouseFunction(int, int) care modifică poziția în scenă și orientarea Observatorului corespunzător.

În funcția KeyboardFunction(uchar, int, int) se generează mai întâi sistemul de coordinate al Observatorului dându-se poziția, axa superioară și punctul de privire ale acestuia. Axa frontală este diferența dintre poziția și punctul de privire normalizată, axa superioară este dată deja, iar axa laterală este dată de produsul vectorial dintre axa superioară și axa frontală normalizat. Urmează ca poziția observatorului și punctul de privire să fie actualizate conform tastei apăsate la momentul dat.

În funcția PassiveMouseFunction(int, int) cursorul este automat deviat în mijlocul ecranului după fiecare cadru pentru comoditate și se consideră unghiul de rotație al orientării Observatorului diferența dintre poziția cursorului înainte de deviere și, respectiv după devierea în mijlocul ecranului. Înainte de a începe rotațiile, Observatorul este translatat în origine, păstrându-se aceeași distanță între poziția acestuia și punctul de privire. Pentru rotația laterală, este rotit punctul de privire în jurul axei superioare cu unghiul de rotație menționat anterior. Pentru rotația transversală, se generează întâi axa laterală (la fel ca în funcția KeyboardFunction(uchar, int, int)) apoi punctul de privire este rotit în jurul acesteia. În final, se readuce Observatorul în poziția inițială și, eventual, se regenerează axa superioară care acum este posibil să fie degenerată, datorită schimbării orientării Observatorului pe planul transversal (în aceast Demo s-a preferat dezactivarea acestei particularități pentru facilitarea orientării utilizatorului în scenă).

* Redarea scenei:

În ficare cadru al stagiului de redare, se fac următoarele actualizări în scenă:

* + Matricea-Vedere este actualizată cu parametrii curenți ai Observatorului (poziție, punct de privire și axă superioară) care este posibil să se fi actualizat în cadrul funcțiilor de interactibilitate;
  + Unghiul de rotație al Soarelui este incrementat cu 0.1˚;
  + Se leagă shader-ele pentru Soare; La locația 1 se regăsesc culorile; Matricea-Model pentru acestea este matricea de rotație în jurul axei superioare cu unghiul curent de rotație (sunRotAngle/100 – gradele sunt modulo 36000); Matricea Model-Vedere-Proiecție este transmisă shader-ului în locația variabilei uniforme „mvpMatrix”;
  + Se leagă shader-ele pentru Pământ; La locația 2 se regăsesc culorile; Noua poziție a planetei se obține rotind poziția inițială implicită (0, 0, -5) în jurul originii cu același unghi ca cel de rotație al Soarelui; Matricea-Model asignată acestei planete este produsul dintre matricea de translație la poziția calculată anterior și matricea de scalare cu un parametru arbitrar pentru sugerarea dimensiunii inferioare Soarelui; Matricea Model-Vedere-Proiecție este transmisă shader-ului în locația variabilei uniforme „mvpMatrix”;
  + Se leagă shader-ele pentru Lună; La locația 3 se regăsesc culorile; Noua poziție se obține aplicând următoarele operații asupra poziției implicite a Lunii (egală cu poziția planetei Pământ la momentul dat):
    - Translația cu 2\*, unde este vectorul dintre Pământ și Soare. Această translație captează intuiția că Luna se află întotdeauna la distanța 2 față de Pământ, în direcția dată de vectorul ;
    - Translația cu care duce axa de rotație a Lunii în origine (Soare);
    - Rotația Lunii în jurul originii cu unghiul de rotație al Soarelui înmulțit cu 4 (Luna încă se află la distanță 2 față de aceasta);
    - Translația cu care aduce axa de rotație a Lunii înapoi în mijlocul planetei Pământ.

Matricea-Model asignată acestei planete este produsul dintre matricea de translație la poziția calculată anterior și matricea de scalare cu un parametru arbitrar pentru sugerarea dimensiunii inferioare Pământului; Matricea Model-Vedere-Proiecție este transmisă shader-ului în locația variabilei uniforme „mvpMatrix”;

* + Se leagă shader-ele pentru Venus; La locația 4 se regăsesc culorile; Noua poziție a planetei se obține rotind poziția inițială implicită (0, 0, -9) în jurul originii cu unghiul opus ca semn ca cel de rotație al Soarelui; Matricea-Model asignată acestei planete este produsul dintre matricea de translație la poziția calculată anterior și matricea de scalare cu un parametru arbitrar pentru sugerarea dimensiunii inferioare Soarelui; Matricea Model-Vedere-Proiecție este transmisă shader-ului în locația variabilei uniforme „mvpMatrix”;