

Universitatea Politehnica Bucuresti, Romania
Facultatea de Automatica si Calculatoare



Editor de scene 3D

~ Specificatiile sistemului ~

Membrii echipei:
Hotea Cristian
Ionescu Eugen
Tamirjean Bogdan

Echipa: CEB Graphics
Indrumator: Florentina Oprea



Cuprins

1. Introducere.....	3
1.1. Scopul acestui document.....	3
1.2. Definirea termenilor.....	3
2. Descrierea generala.....	4
2.1. Scopul produsului.....	4
2.2. Functionalitatile produsului.....	4
2.3. Contextul produsului.....	5
2.4. Beneficii.....	5
3. Cerinte functionale.....	6
3.1. Actori.....	6
3.1.1. Modelator / Artist.....	6
3.1.2. Game Developer.....	6
3.1.3. Gamer.....	7
3.2. Utilizarea programului.....	8
3.2.1. Informatii generale.....	8
3.2.2. Resource Manager.....	11
3.2.3. Scene Manager.....	11
3.2.4. Camera.....	12
3.2.5. Resursele programului.....	12
4. Cerinte nefunctionale.....	14
4.1. Cerinte pentru interfata utilizator.....	14
4.1.1. Interfata grafica.....	14
4.1.2. Utilizarea Interfetei.....	14
4.1.3. Fisierul RM.....	15
4.1.4. Fisierul SM.....	15



1. Introducere

1.1 Scopul acestui document

Scopul acestui document este de a prezenta in detaliu caracteristicile generale ale aplicatiei, precum si modalitatile de interactiune cu aceasta. Documentul este adresat tuturor pasionatilor de grafica, care vor dezvolta in continuare acest produs. In acest document sunt oferite specificatii si pentru utilizatorii obisnuiti care vor folosi produsul.

1.2 Definitia termenilor

Sistem – programul actual

Motor 3D - Aplicatia care va ingloba toate functionalitatile programului

GPU – Unitatea de procesara grafica.

Model – Un obiect poate fi reprezentat, aproximativ, utilizand triunghiuri ca si primitive, un model consta in multimea tuturor varfurilor acestor triunghiuri precum si toate attributele acestor varfuri

Textura - Reprezinta o imagine care va fi atribuita unui model

Obiect - Entitate ce defineste un model, texturile asociate acestuia precum si alte informatii definitorii.

Shader - Program care ruleaza pe GPU si care ne ajuta sa obtinem efecte de mare calitate.

Banda grafica - Intregul proces de transformare a unor primitive intr-o imagine 2D afisata pe ecran.

Actor - Entitatea care joaca un rol in relatia cu sistemul.

SM - Scene Manager

RM - Resource Manager

SkyBox – Textura Cubica

VBO – Vertex Buffer Object

IBO – Index Buffer Object

Camera – punctual si directia din care privim scena



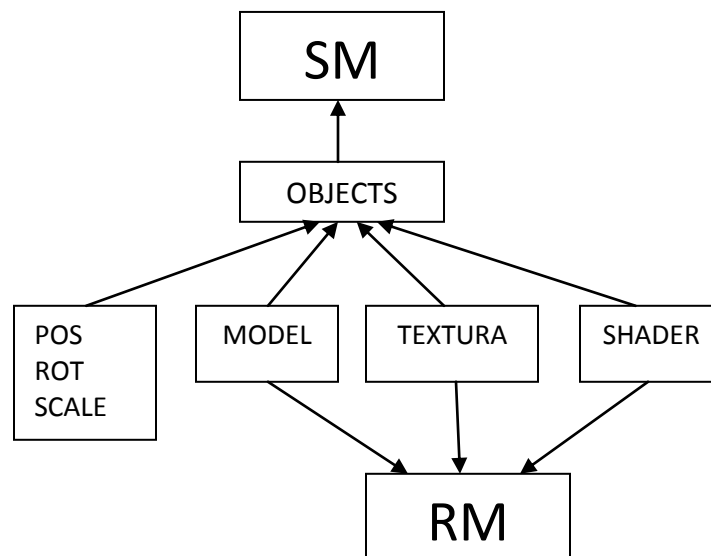
2. Descrierea generala a produsului

2.1 Scopul produsului

Scopul produsului este de a oferi o imagine 2D a unei lumi 3D specificata de catre utilizator. Mai mult de atat, reprezinta partea de baza cu ajutorul careia se poate produce un joc 3D sau 2D. Orice joc trebuie sa aiba la baza lui un engine care sa creeze lumea dorita si sa o afiseze pe ecran. Produsul poate fi vandut pe piata catre game developeri care doresc sa realizeze un joc cat mai repede si cu efort minim.

2.2 Functionalitatile produsului

Principala functionalitate a produsului este redarea unei scene 3D specificata de catre utilizator. Sistemul are la baza 2 entitati Manager confor figurii de mai jos:



Mai exact, utilizatorul va avea acces la aceste 2 entitati Manager. Ele vor fi reprezentate prin simple fisiere de configurare. RM va contine asa cum arata si figura modele, texturile, shaderele pe care le avem la dispozitie pentru a crea



scena 3D. SM va contine entitatile obiect, acestea fiind clase definite la nivelul programatorului care inglobeaza toate elementele necesare constituirii obiectului. Asa cum arata si figura un obiect va avea un model, textura, shadere (acestea vor fi preluate din RM) precum si attribute legate de pozitia, rotatia si scalarea obiectului respectiv.

2.3 Contextul produsului

Produsul face parte din aplicatiile prezente in industria grafica. Este folosit local, nu are functionalitate de comunicare in retea cu alte entitati.

2.4 Beneficii

Pe langa functionalitatile de baza produsul poate fi privit ca fiind un generator de imagini cu ajutorul caruia se pot realiza prezentari ale unor obiecte/proiecte de mare anvergura. Mai mult de atat, se pot realiza mici filmulete in care se pot modela diverse scene care pot fi introduse in industria de televiziune si in industria filmelor de cinema. Dezvoltand acest produs putem realiza secvente de calitate superioara in care putem modela diverse fenomene reale cum ar fi focul, ceata, cerul-orizontul, miscarea apei, etc.



3.Cerinte functionale

3.1 Actori

3.1.1 Modelator / Artist

In cazul in care un modelator sau un artist va folosi programul acesta va putea sa isi testeze munca si sa vada rezultatele. Un modelator va putea incarca modelul creat de el in Motorul 3D pentru a-l vedea din diferite unghiuri si pentru a-i putea aplica diferite efecte. Artistul este cel care va desena texturile, mai exact imaginile care vor fi atasate obiectelor din scena. Astfel acesta va putea testa functionalitatea unei texturi, mai exact, corelarea dintre modelul primit de la modelator avnd anumite specificatii cu formatul texturii create.

3.1.2 Game Developer

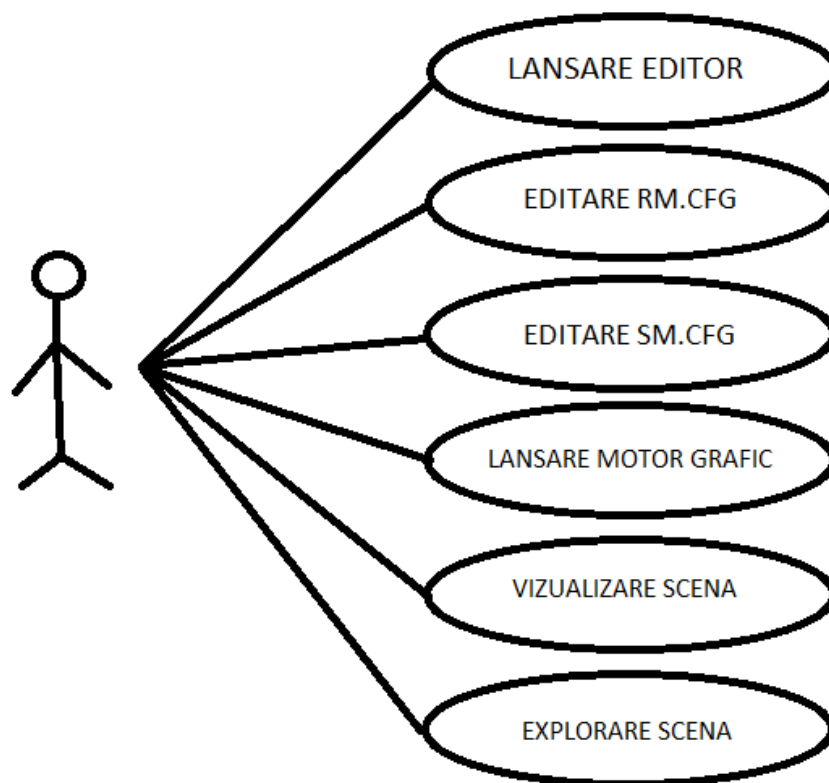
Un game developer va putea sa foloseasca programul pentru a dezvolta in continuare un joc. Acesta va trebui sa implementeze toate conceptele jocului, acestea fiind camera, elemente de fizica, precum si miscarea obiectelor in scena,etc. Pe langa facilitatile oferite de acest produs, dezvoltatorul va putea crea si integra cu usurinta alte modele,texturi si shadere, deci va putea crea obiecte cu un grad ridicat de detaliu. Proiectul va contine cateva exemple de modelare a fenomenelor reale cum ar fi ceata, focul, reflexia luminii, miscarea apei, cerul,etc ,aceste obiecte fiind diferite fata de obiectele simple din scene(de exemplu o cladire). Modelarea acestor fenomene presupune obtinerea unor efecte cat mai reale astfel ca apa trebuie sa contina valuri, reflexia suprafetei sale, refractia fundului sau,etc.



3.1.3 Gamer

Programul poate fi dezvoltat in continuare astfel obtinandu-se un joc. Jocurile pe calculator sunt foarte populare in toata lumea, tinerii fiind principalii cotizanti in industria de jocuri pe calculator. Un astfel de proiect poate sta la baza unui joc de care multi gameri sa se bucure, promovandu-l in acelasi timp.

In figura de mai jos este reprezentata diagrama de cazuri de utilizare.



3.2 Utilizarea programului

3.2.1 Informatii generale

Utilizatorul programului va avea acces la cele 2 fisiere de configurare. Pe langa Motorul 3D implementat, exista un editor/manipulator simplu cu interfata al fisierelor de configurare. Cele 2 fisiere de configurare definesc cele 2 entitati de tip Manager din program – Scene Manager (Managerul scenei) si Resource Manager (Managerul resursei).

Fisierul de configurare specific RM contine informatiile necesare despre resursele pe care Motorul poate sa le foloseasca. Mai exact modelele, texturile si shaderele. Fiecare din acestea va contine o cale relative sau absoluta catre fisierul specific din folderul de resurse, precum si alte informatii referitoare, toate putant fi configurate de catre utilizator.

Astfel, motorul nostru este unul generic care stie sa acceseze si sa retina toate modelele, texturile, shaderele din folderul cu resursele si sa le manipuleze corespunzator fisierului de configurare SM. Acest lucru ne da libertatea sa adaugam fisiere noi in folderul de resurse fara a mai fi nevoie sa modificam programul. Acesta va fi capabil sa acceseze si noile resurse introduse, singurul lucru pe care trebuie sa il facem e sa le mentionam in fisierul de configurare specific RM.



Fisierul de configurare specific SM contine informatiile necesare despre obiectele definite in scena noastra. Fiecare obiect definit aici trebuie sa aiba un model asociat. Pe langa model, acesta poate avea una sau mai multe texturi care vor da culoare obiectului. Fiecare obiect va avea asociat un shader care va fi programul pe GPU care va manipula modelul si texturile asociate obiectului astfel incat obiectul final sa fie cel dorit in scena. Astfel, in acest fisier vom putea defini oricate obiecte pe care Motorul 3D le va randa si le va afisa in fereastra specifica.

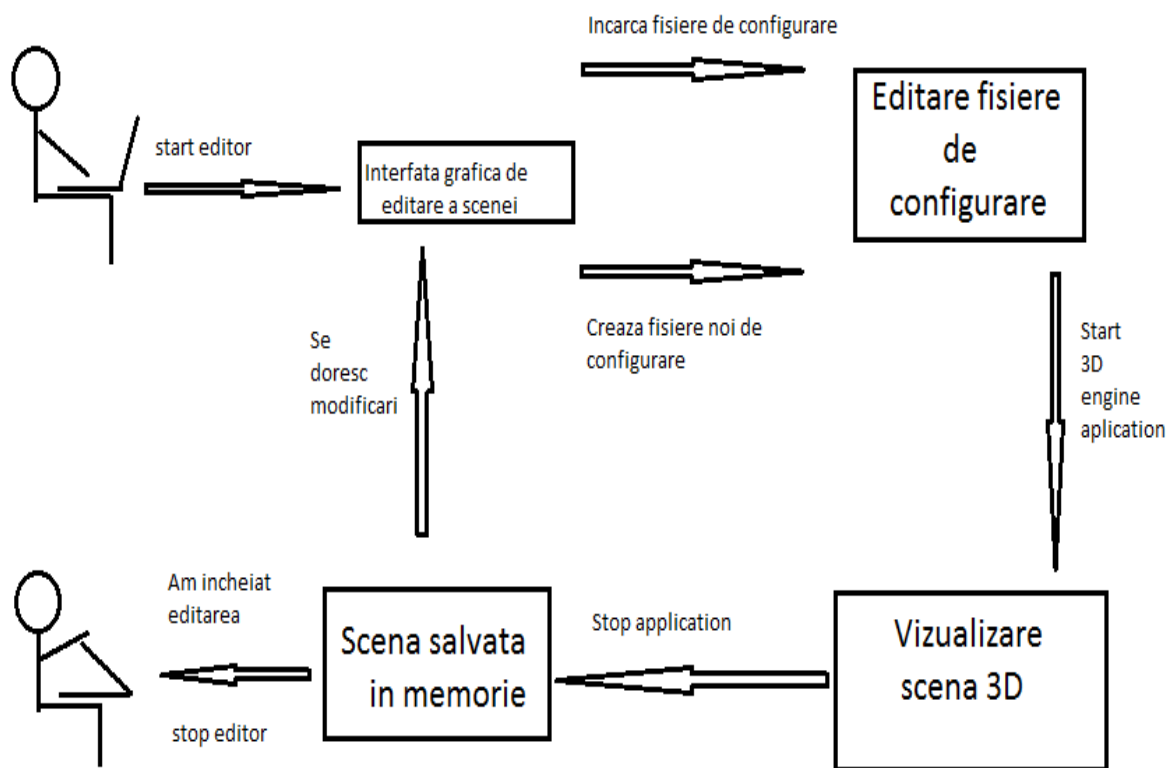
Toate informatiile referitoare la un obiect vor fi mai exact legaturi catre informatiile din RM, mai exact, in fisierul-RM vom specifica existenta modelului X avand un indice interior Y, iar in fisierul-SM vom specifica pentru un anumit obiect doar indicele Y, Motorul stiind sa faca legatura intre indicele Y si entitatea resursa din RM.

Pe langa aceste informatii importante, trebuie sa specificam care este relatia dintre obiect si lumea scenei. Mai exact, unde vrem sa fie positionat obiectul. Pentru aceasta va trebui sa specificam care este pozitia obiectului in lume, care este rotatia sa initiala, precum si scalarea sa fata de origine.

Pe langa obiectele cerute in scena, in fisierul-SM vom putea specifica si pozitia si rotatia camerei pentru a da imaginea de inceput asupra lumii. Camera va fi positionata initial intr-un punct din lume, specificat in acest fisier. O alta entitate care trebuie specificata in fisierul de configurare specific SM este sursa de lumina. Sursa de lumina defineste un punct din spatial scenei, punct fata de care vom calcula culorile obiectelor din scena, mai exact, in functie de distant pana la acest punct. Pe langa pozitie, lumina trebuie sa aiba si o componenta de culoare si una de miscare. Lumina poate sa fie statica sau sa se miste pe o anumita traiectorie.



In figura de mai jos este reprezentata diagrama de stari-tranzitii.



3.2.2 Resource Manager (Managerul de resurse)

Managerul de resurse inglobeaza totalitatea resurselor disponibile programului. Acestea se afla intr-un folder care poate fi actualizat oricand, programul nefiind afectat. In design-ul programului acesta va fi instantiat o singura data de catre Motorul 3D.

In momentul instantierii, Motorul va parcurge fisierul de configurare specific RM si va salva toate datele disponibile in acesta, mai exact dupa aceasta faza va putea manipula toate modelele, texturile, shaderele specificate. Dupa aceasta faza se va instantia Scene Managerul.

3.2.3 Scene Manager (Managerul scenei)

Managerul de resurse inglobeaza totalitatea obiectelor specificate in scena respectiva. In design-ul programului, acesta va fi instantiat o singura data de catre Motorul 3D.

In momentul instantierii, Motorul va parcurge fisierul de configurare specific SM si va salva toate datele disponibile in acesta, mai exact dupa aceasta faza va putea manipula toate obiectele specificate impreuna cu informatiile lor. In aceasta faza Motorul va prelua datele salvate in etapa specifica RM si le va manipula constituind obiectul final descris de utilizator. Dupa aceasta faza, Motorul va avea toate datele necesare crearii lumii 3D pe care o va afisa frame cu frame pe ecran.



3.2.4 Camera

Camera reprezinta pozitia vizualizatorului fata de lume. Aceasta reprezinta un alt element al comunicarii utilizatorului cu programul. Camera va putea fi manipulate utilizand tastatura. Mai exact, pentru translatii (miscari inainte-inapoi,dreapta-stanga,sus-jos) vor fi folosite tastele W-S,D-A,R-F – oarecum intuitiv.Pentru a roti camera(implicit si unghiul de privire asupra lumii) vom folosi tastele UP-DOWN,LEFT-RIGHT pentru rotirea in jurul lui OX, respectiv OY.

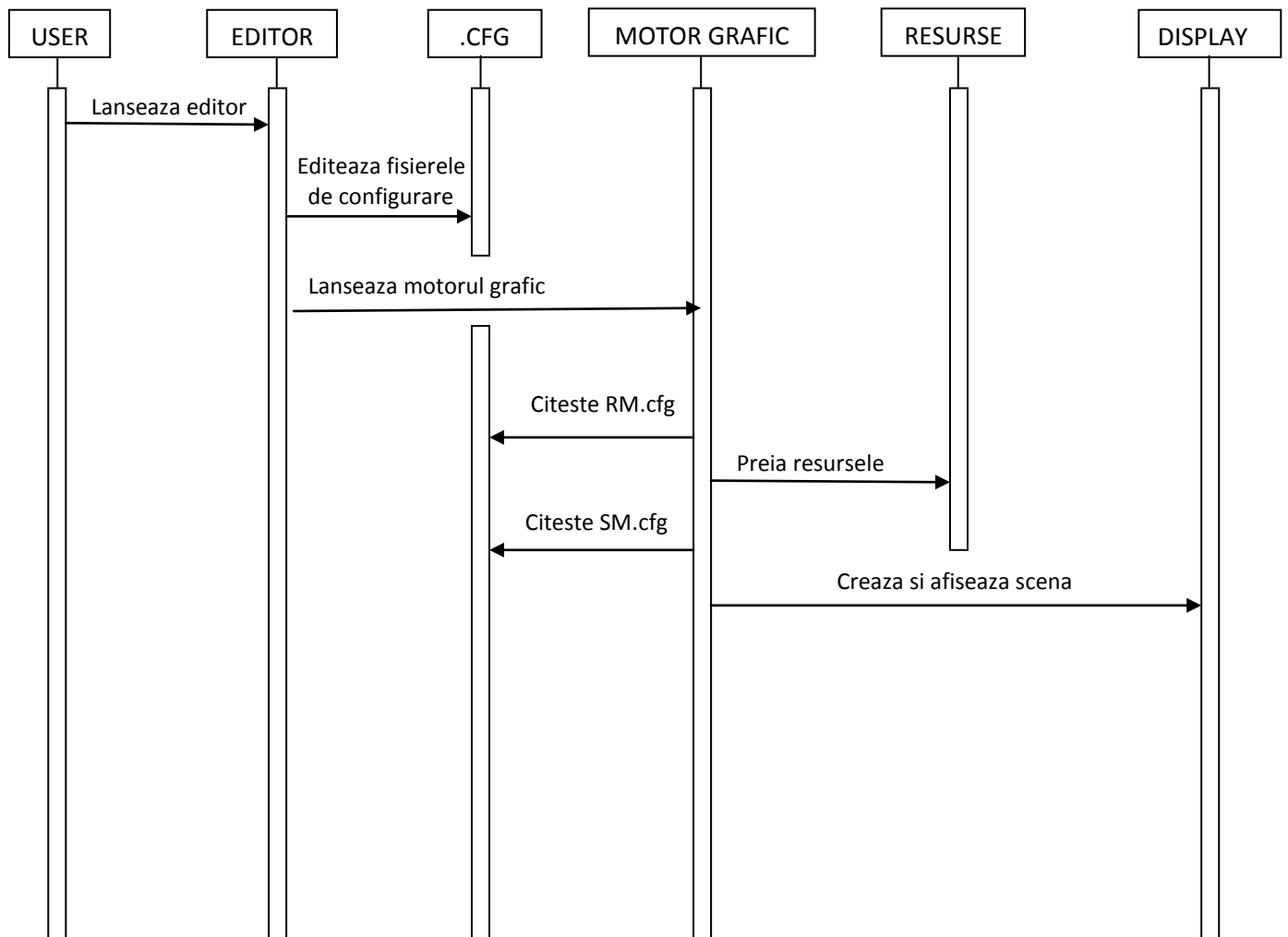
Camera este foarte importanta pentru program deoarece da posibilitatea utilizatorului sa se plimbe prin lumea create de el. Astfel poate sa se apropie de obiecte, pentru a le inspecta nivelul de detaliu, poate sa depisteze anumite buguri in crearea modelului respective, sau poate sa determine simtul estetic pe care texturile asociate obiectelor il dau intregii scene.

3.2.5 Resursele programului

Resursele programului se afla intr-un folder prestabilit care este accesibil de catre utilizator.Utilizatorul va putea adauga noi resurse sau sterge inregistrati actuale in functie de preferinte.De fiecare data cand vom rula programul acesta va prelua informatiile din fisierele aflate in acest folder in conformitate cu fisierul de configurare specific Managerului de resurse.



In figura de mai jos este reprezentata diagrama de secventa.



4.Cerinte nefunctionale

4.1Cerinte pentru interfata utilizator

Dupa cum a fost precizat mai sus , utilizarea programului , si manipularea scenei 3D se va efectua cu ajutorul a 2 fisiere de configurare , Scene Manager si Resource Manager. Pentru a simplifica editarea acestor fisiere utilizatorul va avea la dispozitie o interfata grafica .

4.1.1 Interfata grafica

Interactiunea utilizatorului cu fisierele de management SM si RM se va realiza prin interfata grafica pusa la dispozitie. Aceasta va permite editarea scenei intr-un mediu simplu si intuitiv. Astfel , un utilizator nefamiliar cu formatul fisierelor de intrare poate invata rapid metode de a interactiona cu obiectele din scena 3D.

4.1.2 Utilizarea Interfetei

Pentru a incepe editarea unei scene se va crea o noua pereche de fisiere de configurare (SM si RM) , sau se va incarca o pereche existenta. Utilizatorul va putea apoi , prin intermediul unor butoane , slider-e ,meniuri etc. ,sa modifice valori precum numarul de obiecte din scena , texturile aplicate acestora , pozitiile si modul de iluminare al surselor de lumina.

Printr-o simpla re-incarcare a motorului grafic , utilizatorul va putea vedea efectul modificarilor asupra scenei 3D , fara a fi nevoie de folosirea unui editor de text obisnuit si de cunostinte avansate asupra formatului datelor de intrare.



4.1.3 Fisierul RM

Fisierul Resource Manager (RM pe scurt) va contine date referitoare la resursele folosite de catre motorul grafic . Prin intermediul interfetei , utilizatorul poate adauga modele , texturi si shadere scenei . Calea catre aceste resurse va fi scrisa in fisierul RM (in formatul specific acestuia) pentru a fi folosite de catre aplicatie.

Selectarea unui fisier este astfel mult mai intuitiva decat inscrierea direct in fisier a unei noi intrari

Add new Texture file :

NewResourcesPacket/Textures/Terrain_blend_map.tga

Select file

4.1.4 Fisierul SM

Fisierul Scene Manager (SM pe scurt) va contine date referitoare la manipularea resurselor din fisierul RM. In cadrul interfetei grafice , utilizator poate modifica aspecte ale scenei prin simpla apasare a unui buton . Acesta va avea acces la caracteristicile fiecarui obiect precum si la caracteristici generale ale scenei .

In cazul in care va fi incarcat un fisier existent , aplicatia va citi valorile din acesta si ii va permite utilizatorului sa efectueze ajustari fine asupra valorilor .

Object Id :

1

Position :

X = 100

Y = 0

Z = 100



Light Type :

AMBIENTAL

Color:

R = 1.0

G = 0.0

B = 0.0

