

Pràctica 1

RayTracingToy: Enunciat general i planificació

GiVD - curs 2022-2023

Tema: RayTracing per visualitzar dades (RayTracingToy):

L'objectiu de la pràctica 1 és implementar un *RayTracing* que permet visualitzar escenes virtuals amb efectes realistes, simulant ombres, reflexions i transparències.

En aquest primer document trobaràs una explicació general de la pràctica, així com les fases i la planificació en el calendari de l'assignatura.

ÍNDEX:

1.	Software bàsic:	1
2.	Etaques del procés de visualització:	2
	ETAPA 1: Modelatge: Construcció de l'escena a partir de diferents fitxers	2
	1. Cas d'Escena Virtual: VIRTUALWORLD	2
	2. Cas d'escena procedent de dades geolocalitzades: REALDATA	2
	ETAPA 2: Visualització: Algorisme de RayTracing:	3
3.	Arquitectura de l'aplicació:	3
4.	Fases de desenvolupament:	4
	i. FASE 0: Introducció al codi i primeres visualitzacions	4
	ii. FASE 1: Nous tipus d'objectes i la seva visualització:	4
	iii. FASE 2: Il·luminacions amb efectes més realistes	5
	iv. FASE 3: Aplicació de textures i creació de noves escenes	5
5.	Planificació:	5

1. Software bàsic:

Disposes d'un software de partida que trobaràs a la tasca del classroom de github i que caldrà que canvieu i amplieu amb més funcionalitats:

<https://classroom.github.com/a/DAYlm59m>

El codi és un projecte basat en QtCreator per a desenvolupar el RayTracing (veure el fitxer [Main.cpp](#) del projecte [RayTracingToy](#)) que codifica una interfície gràfica basada en la llibreria Qt i disposa d'un conjunt de classes que cal completar per a calcular el color final de cada píxel a la imatge.

2. Etapes del procés de visualització:

A continuació es detallen les dues principals etapes del procés de *RayTracing* en la Figura 1: (1) l'etapa de modelatge, en la que es construeixen les escenes virtuals formades pels objectes i els seus corresponents materials i (2) l'etapa de visualització, en la que es fa el procés de *Raytracing* (o visualització de l'escena) amb una certa configuració (o *setup*). La sortida de la visualització es fa en una finestra de Qt i també és possible guardar els resultats en un fitxer png (o varis, si es fan animacions). A continuació s'expliquen les dues etapes del procés:

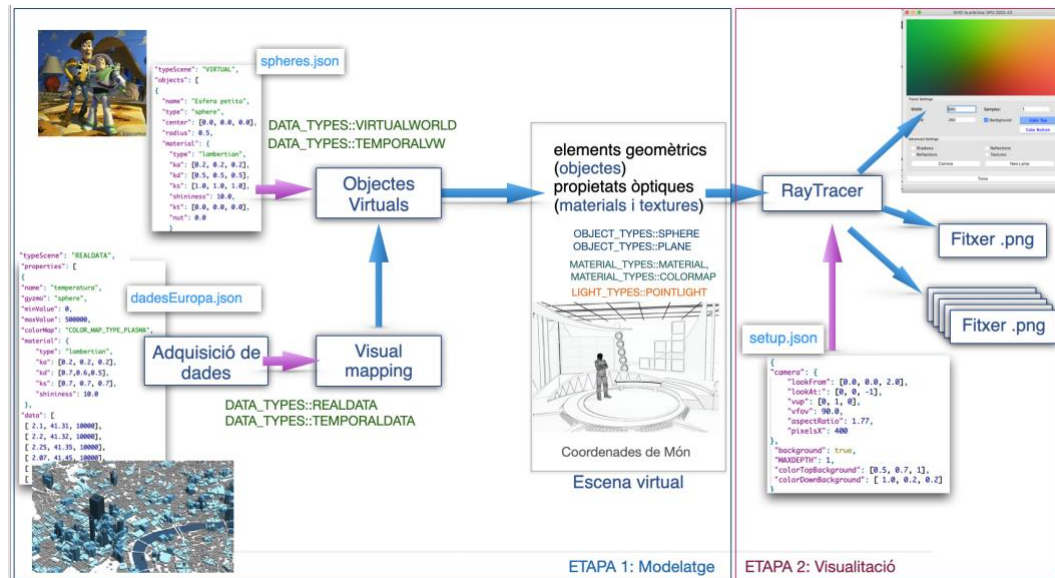


Figura 1: Etapes de la visualització en el codi de RayTracingToy de la pràctica.

ETAPA 1: Modelatge: Construcció de l'escena a partir de diferents fitxers

Les escenes virtuals es poden construir directament des de codi o bé es poden carregar des de fitxers de tipus json. Aquests poden contenir directament escenes virtuals o dades reals. A continuació es detallen ambdós casos:

1. Cas d'Escena Virtual: **VIRTUALWORLD**

En aquest cas, es podran llegir diferents tipus d'objectes superficials (esferes, malles, caps, cilindres, etc.). En la pràctica bàsica només es pot llegir un objecte de tipus esfera. Aquest fitxer s'haurà d'ampliar en el desenvolupament de la Fase 2 de la pràctica per a poder carregar diferents tipus d'objectes com cilindres, triangles i malles triangulars. Un exemple està a [oneSphere.json](#) en la carpeta de [resources](#).

2. Cas d'escena procedent de dades geolocalitzades: **REALDATA**

Les dades venen donades en un fitxer .json on es té una capçalera inicial i després a cada línia es tenen les coordenades en latitud i longitud i els valors mesurats en aquella coordenada. El format és el següent (mireu el contingut dels fitxers [dadesBCNOneValue.json](#) i [dadesBCN](#)).

En aquest cas, cal fer el procés de *Visual Mapping*, és a dir, fer la correspondència entre les dades i objectes virtuals, localitzant-los a l'espai, escalant-los segons el valor que representin i assignant-los colors/materials a partir dels valors les dades.

ETAPA 2: Visualització: Algorisme de RayTracing:

Per a visualitzar l'escena creada, s'utilitzarà l'algorisme de RayTracing, que és una de les diferents tècniques per visualització d'imatges per computador usades per mostrar escenes 3D. La idea bàsica és pensar que les imatges generades estan composades per llum (colors) i que aquesta llum prové de les fonts de llum i rebota per tota l'escena fins a arribar a l'ull (o a la càmera). En els 80, Turner Whitted proposa l'algorisme de RayTracing que simplifica el problema anterior tenint en compte només els rajos que arriben als ull després d'uns quants rebots, podent modelar reflexions directes, indirectes i transparències.

L'algorisme bàsic de RayTracing determina la visibilitat que l'observador té de l'escena a través de cada píxel. Per a cada píxel es calcula un raig de visió que interseca amb l'escena (Figura 2). Aquest raig s'anomena **raig primari**. El color a assignar a cada píxel és la intensitat calculada al punt d'intersecció visible, és a dir, la intensitat del punt intersecció amb l'escena més proper a l'observador. Podeu trobar més informació a la referència [1].

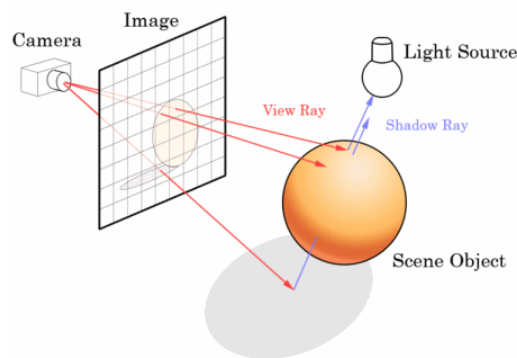


Figura 2: Tècnica de RayTracing: El raig de visió (View Ray) s'anomena **raig primari** i es calcula a partir de la posició de l'observador i el punt de l'àrea de projecció que es rasteritza en el píxel. El raig d'**ombra** (Shadow Ray) és el raig que va des del punt d'intersecció fins a la llum.

3. Arquitectura de l'aplicació:

A la Figura 3 es representa el model de classes de l'aplicació que es basa en una arquitectura Model-Vista-Controlador. En aquest cas, la Vista són les finestres de Qt que codifiquen la finestra principals amb els seus botons (veure classe [MainWindow](#)) i alguna finestra de diàleg secundària que permet canviar colors o la posició de la càmera.

L'aplicació està gestionada per un Controlador (classe [Controller](#)) que és l'encarregat de servir les peticions de la MainWindow. Des del [Controller](#), per a poder construir diferents escenes procedents de diferents tipus de fitxers (mons virtuals – [VIRTUALWORLD](#) - o dades – [REALDATA](#)).

La part del Model conté tant les parts de [Modelatge](#) dels elements que formen part de l'escena virtual (objectes, materials, llums, transformacions geomètriques) i les parts de [Rendering](#), i.e. Visualització (Camera, Raig, Shadings, algorisme de RayTrace). En el model també es disposen de parts que tracten amb la part de persistència, és a dir, amb la lectura i escriptura de dades i les seves transformacions visuals ([DataInOut](#)). S'inclouen també Factories per a cada jerarquia de classes que es necessita. Per exemple com tenim objectes de diferents tipus representats en una herència de la classe [Object](#) (esferes, caps, malles, etc.), es té la classe [ObjectFactory](#) que donarà la utilitat de crear-los.

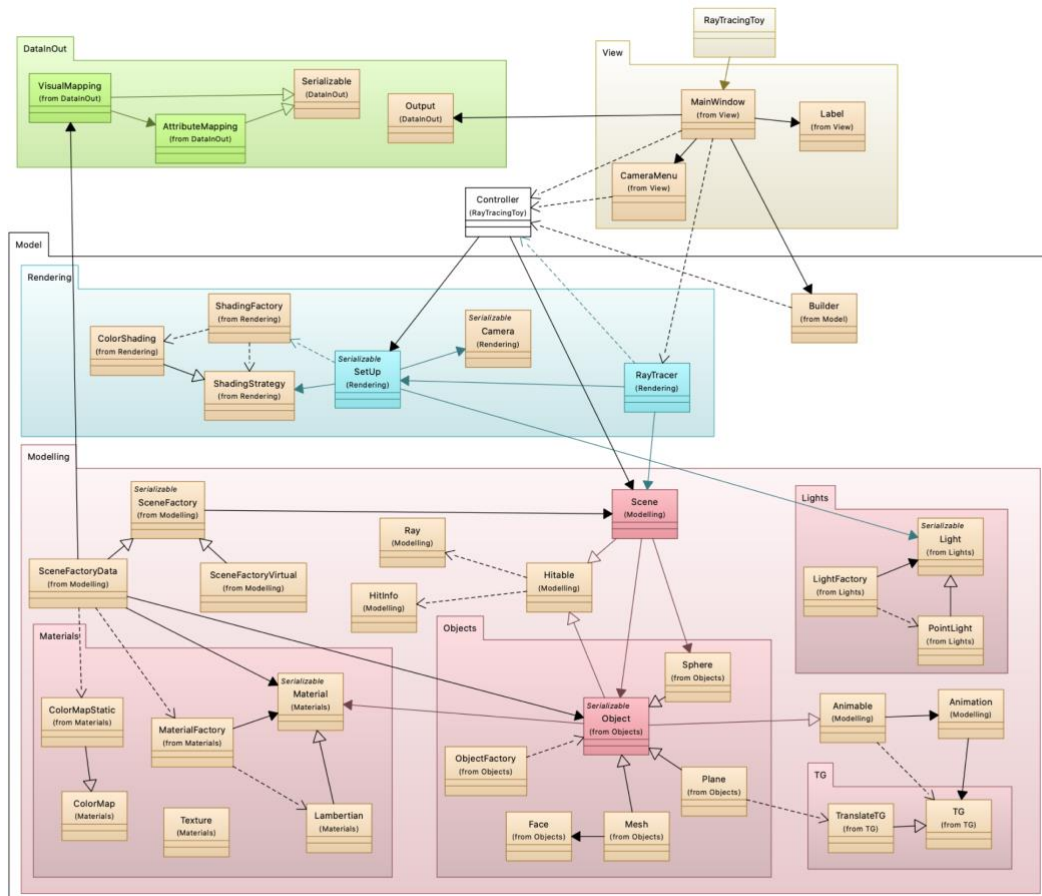


Figura 3: Arquitectura de classes de l'aplicació RayTracingToy.

4. Fases de desenvolupament:

La pràctica 1 es divideix en diferents fases i cadascuna d'elles té una fitxa diferent amb una sèrie de passos a desenvolupar:

i. FASE 0: Introducció al codi i primeres visualitzacions

En aquesta fase coneixeràs el codi, com canviar de *background*, veuràs unes primeres escenes i jugaràs amb alguns *shadings* (o mètodes d'il·luminació locals bàsics). Al final d'aquesta fase, obtindreu la visualització de tres esferes de diferents colors, posicions i mides a l'escena, segons el fitxer [spheres.json](#). També provaràs a carregar un primer fitxer de dades [data0.json](#) per veure les dades reals que es mapejarà a un objecte o *gizmo* de tipus esfera.

ii. FASE 1: Nous tipus d'objectes i la seva visualització:

Per **escenes virtuals**, heu de suportar diferents primitives (esferes, triangles, cilindres i malles triangulars - fitxers .obj -). Per **escenes amb dades geolocalitzades** creareu l'escena virtual usant *visual mapping*: Heu fer la correspondència de les dades amb objectes geomètrics usant diferents primitives, posicionant-los a l'escena i escalant-los segons el valor que representen, usant **transformacions geomètriques** de translació i escalat que es necessitin i per a aplicar-los el color bàsic definit en una **paleta**.

Heu d'implementar un RayTracing que permeti visualitzar els objectes de tipus esfera, pla, caps, triangles i objectes .obj, utilitzant color bàsic (difús).

iii. **FASE 2: Il·luminacions amb efectes més realistes**

Creareu diferents materials per a poder tenir efectes de reflexions difuses, especulars i transparències. Podreu llegir tota la informació a partir dels fitxers .json. Creareu també llums i diferents estratègies de visualització (Blinn, Blinn-Phong, etc.).

En relació a les **dades geolocalitzades**, en el procés de **visual mapping**, fareu la correspondència dels **valors** de les dades amb la paleta i amb els **materials** associats a cada objecte. En aquest mapeig també s'afegirà com a context de les dades un mapa de referència (encara sense textures).

A la visualització final es veuran ombres, reflexions i transparències, definint els materials a partir de la paleta de colors i els valors associats a cada objecte.

iv. **FASE 3: Aplicació de textures i creació de noves escenes**

En aquesta fase, es mapejarà una textura sobre el mapa de referència. També es mapejaran textures en els objectes virtuals.

A més, s'han de preparar altres conjunts de dades per a fer la demostració de tota la feina realitzada. Opcionalment, es poden crear animacions, bé de l'escena virtual, bé de sèries temporals dels objectes que representen les dades.

Adicionalment, al final de la pràctica, cal que lliureu un informe en el README del projecte github, on detal·leu les proves realitzades, expliqueu les tècniques que heu desenvolupat i les noves escenes que heu definit donant captures de pantalla i les animacions dels resultats finals que aconsegiu, inclouen els fitxers de configuració utilitzats. Es aconsellable anar portant un diari de les diferents imatges que aneu generant durant el desenvolupament. En aquest README també **cal que detal·leu la feina que ha fet cada component del grup**.

5. **Planificació:**

Febrer	20	21	22	23	24	Pràctica 1: FASE 0 Aprentent l'entorn, Càlcul de les interseccions amb escena
	27	28	1	2	3	Pràctica 1: FASE 1 Càlculs d'interseccions i situació dels gizmos de les dades a l'escena virtual.
Març	6	7	8	9	10	Pràctica 1: FASE 2 Il·luminació local i ombres
	13	14	15	16	17	Pràctica 1: FASE 2 Reflexions i transparències
	20	21	22	23	23	Pràctica 1: FASE 3 - Aplicacions de textures en el pla i obtenció de dades noves. Parts opcionals
	27	28	29	30	31	Dia 8: Lliurament al campus i al github. Entrevista P1 (en horari de l'examen parcial)

Material de referència: [1] [Conceptes bàsics de Raytracing](#)