



**Escola  
d'Enginyeria  
UAB**

**UAB**  
Universitat Autònoma  
de Barcelona

# **VISUALITZACIÓ GRÀFICA INTERACTIVA**

**GRAU EN ENGINYERIA INFORMÀTICA  
APRENENTATGE BASAT EN PROJECTES**

## **PORTFOLI DE L'ALUMNE**

**SIMULADOR DEL SISTEMA SOLAR**

Guillem Garcia, Martí Llinés, David Montaña, Joel Tapia

Bellaterra, Desembre 2024

# ÍNDEX

<b>1. INTRODUCCIÓ I OBJECTUS.....</b>	<b>4</b>
<b>1.1 Planificació.....</b>	<b>4</b>
<b>1.2 Eines de desenvolupament.....</b>	<b>5</b>
<b>2. DESENVOLUPAMENT.....</b>	<b>6</b>
<b>2.1 Mòdul Objecte.....</b>	<b>6</b>
2.1.1. Submòdul Sol.....	6
2.1.2. Submòdul Planeta.....	7
2.1.3. Submòdul Satèl·lit.....	7
2.1.4. Submòdul Òrbita.....	7
2.1.5. Submòdul Asteroide.....	7
2.1.6. Submòdul Anell.....	8
2.1.7. Submòdul Estrella.....	8
<b>2.2. Mòdul Càmera.....</b>	<b>8</b>
2.2.1. Submòdul FollowCamera.....	8
<b>2.3. Mòdul Light.....</b>	<b>8</b>
<b>2.4. Mòdul GUIManager.....</b>	<b>9</b>
<b>2.5. Mòdul Reader.....</b>	<b>9</b>
<b>2.6. Mòdul GraphicsEngine.....</b>	<b>9</b>
<b>3. RESULTATS.....</b>	<b>9</b>
<b>3.1. Col·lisions.....</b>	<b>12</b>
<b>3.2. Problemes amb les textures.....</b>	<b>12</b>
<b>4. CONCLUSIONS.....</b>	<b>13</b>
<b>5. REFERÈNCIES ELECTRÒNIQUES I BIBLIOGRÀFIQUES.....</b>	<b>13</b>

<b>ANEXE I. ACTES DE LES REUNIONS.....</b>	<b>15</b>
<b>ANEXE II. DIAGRAMA DE MÒDULS.....</b>	<b>17</b>

# 1. INTRODUCCIÓ I OBJECTUS

Es requereix construir un simulador del Sistema Solar utilitzant els coneixements apresos en Entorns Gràfics i Realitat Augmentada. Aquest constarà dels seus planetes, satèl·lits i asteroides. També s'han de representar les estrelles visibles, així com algunes de les constel·lacions. A més, per donar certa llibertat als usuaris, es necessita afegir eines que el permetin desplaçar-se pel espai i botons que faciliti la seva experiència amb l'aplicació.

Aleshores, l'objectiu principal seria fer la implementació d'una aplicació gràfica sobre el sistema solar en *Python* amb l'ajut de les llibreries *ModernGL* i *Pygame*. A més a més, com a enginyers de dades tenim el repte de representar el Sistema Solar de la manera més fidel possible, el qual s'ha d'extreure i transformar dades de diferents bases de dades obtingudes per després visualitzar-les dintre de la nostra escena de l'aplicació.

Amb això, considerem que la nostra aplicació tindrà les següents funcionalitats:

- **Funcionalitat 1:** S'ha de tenir tant una càmera que sigui "lliure" com una per cada planeta que pugui observar el seu moviment. La càmera lliure ha de permetre un desplaçament lliure: anar cap endavant/enrere, esquerra/dreta i pujar/baixar. També es vol canviar la seva velocitat i direcció. Les càmeres que funcionen com si fossin satèl·lits, han de poder rotar el planeta de diferents formes possibles i seguir l'òrbita d'aquests.
- **Funcionalitat 2:** Es vol representar el sistema solar de la manera més realista possible. Així doncs, els planetes, satèl·lits i el Sol han d'estar representats de forma proporcional al que seria a la realitat (en termes de distàncies i radis). També, tant la forma d'orbitar com les seves textures han de ser el més realista possible. El Sol en especial ha de ser la nostra font d'il·luminació per la resta d'objectes.
- **Funcionalitat 3:** Hem de tenir tant events del teclat com opcions per facilitar l'experiència de l'usuari. Les opcions que ha de poder fer són: reiniciar la posició de la càmera a l'inici, poder canviar de càmera, realitzar els moviments de cada càmera, mostrar o ocultar objectes i finalment poder modificar paràmetres, com per exemple el temps al que transcorre la simulació.
- **Funcionalitat 4:** Hem de poder apreciar alguns detalls del Sistema Solar, com poden ser els anells de Saturn i Urà, o el cinturó d'asteroides. Per totes aquestes coses mencionades anteriorment, s'ha de tenir en compte que s'han de crear de forma eficient, ja que pot afectar al rendiment de l'aplicació si es tenen que crear molts objectes.

La metodologia de treball ha sigut a partir realitzar actes i informes sobre l'estat del treball setmanalment. No hi ha una sèrie de rols definits entre els participants del projecte, només es tenia una llista de tasques per a realitzar i s'assignaven entre els membres del grup durant les dotze setmanes disponibles. En el següent apartat s'explicarà amb detall la planificació del treball i com les eines i estat de l'art utilitzat.

## 1.1 Planificació

A continuació es presentarà una llista de les tasques principals que s'han desenvolupat durant les setmanes de treball.

Taula 1: Planificació i repartició de les tasques del treball

Tasca	Descripció	Responsable	Participants (%)	Durada (setmanes)	Grau Finalització
<b>1. Estat de l'Art</b>	Buscar productes/aplicacions similars al que es proposa, caracteritzar-los i comparar-los	Martí	Martí (80%), Guillem, Joel, David (20% restant)	1	100%
<b>2. Disseny Aplicació</b>	Disseny dels diferents components de l'aplicació i elecció d'eines de desenvolupament	Guillem	Guillem (80%), Martí, Joel, David (20% restant)	2	100%
<b>3. Disseny Interfície d'Usuari</b>	Disseny de la interfície d'usuari	Joel	Joel (80%), Martí, Guillem, David (20% restant)	2	100%
<b>4. Implementació</b>	Implementació de l'aplicació	David	David (35%), Martí, Guillem, Joel (65% restant)	10	100%
<b>5. Test</b>	Test interns i externs (usuaris)	Tots	Tots (25%)	2	100%
<b>6. Memòria i Presentació</b>	Escriure la memòria i la presentació oral	Tots	Tots (25%)	1	100%
GRAU DE MADURESA:					100%

Principalment s'ha hagut de realitzar sis tasques a nivell general durant el projecte. La primera era fer investigació sobre aplicacions que ja existeixen i agafar idees d'aquelles fonts. Seguidament es va plantejar un disseny de l'aplicació i d'interfície a nivell general, que es va fer principalment per establir una estructura de com serà l'aplicació. La tasca que ha tingut el procediment més llarg és la implementació del codi, i per últim tenim les creacions dels tests i la redacció dels informes i presentacions. En cas de voler consultar més detallament quines eren les tasques de disseny i implementació, es troba dintre del *Anexe I. Actes i Reunions*. La planificació s'ha complert en gran part, encara que hi han propostes que s'han quedat sense desenvolupar-se per la falta de temps.

## 1.2 Eines de desenvolupament

A part de les eines esmentades a l'apartat 1, s'ha utilitzat RStudio simplement pel motiu de la comoditat que ens comporta modificar i filtrar les dades dels datasets. Cal esmentar que s'han usat més llibreries de Python com a ajut per la implementació dels mòduls, que són les següents:

- **glm, math i numpy:** Aquestes tres llibreries han sigut d'ajuda sobretot per computar de manera més directa i eficient diferents càlculs necessaris per les nostres funcions.
- **json:** Permet treballar amb dades de format *JSON*. Sobretot es farà servir per a la interfície d'usuari.

- **sys i os:** Per tal de estar sempre en els directoris que pertocuen, utilitzem aquestes dues llibreries.
- **pandas:** Per tal de llegir les dades dels datasets i aplicar-los als objectes.

Part de les idees extretes s'esdevé d'alguns sistemes de visualització del Sistema Solar ja existents son els següents:

- **Stellarium** <sup>[1]</sup>: Aquesta pàgina ens permet veure el cel de nit des de la Terra, des de un punt de vista fixat. Es pot ajustar l'angulació de la càmera mitjançant el ratolí, i l'usuari també pot activar o desactivar algunes opcions per visualitzar el cel (constel·lacions, el terra, etc.). Si es fa clic a qualsevol objecte podem obtenir informació d'aquell.
- **Eyes of the Solar System (NASA)** <sup>[2]</sup>: Podem visualitzar el Sistema Solar. La càmera es pot ajustar de manera que podem allunyar-la fins veure tota la galàxia, i sempre mira cap el Sol. Dintre del Sistema Solar, es pot notar les òrbites dels planetes i altres cossos a temps real, però també es pot ajustar la velocitat. L'usuari pot habilitar o deshabilitar algunes opcions, com per exemple la visualització dels planetes i constel·lacions.

En els següents capítols s'explicarà amb detall tot el procediment del projecte. Es començarà pel disseny i desenvolupament dels mòduls de l'aplicació, i s'acabarà amb el raonament dels resultats i conclusions finals.

## 2. DESENVOLUPAMENT

El diagrama de classes que s'ha dissenyat correspon a la figura del *Anexe II: Diagrama de mòduls*. Consta de sis mòduls principals, on els hem dividit per seccions per explicar el què executa detalladament.

### 2.1 Mòdul Objecte

Serà la classe que s'utilitza per crear tot objecte en escena. Aquesta classe té més classes heretades, que es treballa específicament amb un tipus d'objecte. Amb aquest mòdul podem crear el Sol, els planetes, els satèl·lits, estrelles, constel·lacions, les òrbites, els asteroides i els anells dels planetes. El mòdul genèric conté les funcionalitats que comparteixen tot tipus d'objecte. Sempre els objectes tindran una textura, les dades dels punts en buffers, i finalment un shader.

El mètode més important del mòdul és la creació de les esferes *create\_sphere()*, ja que la majoria d'objectes mostrats en l'escena seràn esferes. En la funció generem els punts de l'esfera a partir de calcular les seves coordenades esfèriques, i també es computa les seves coordenades textura que s'emmagatzemen en un array.

En els pròxims sub-apartats s'explicarà com funcionen les classes que s'hereten del mòdul objecte, mostrant principalment les seves característiques úniques.

#### 2.1.1 Submòdul Sol

El Sol realment és molt similar a com està definit en el mòdul principal, però l'esfera s'ha de generar-la de diferent forma que els demés objectes. El cas és que el Sol ha de ser la nostra font d'il·luminació usant *l'algorisme de Phong* <sup>[3]</sup>. Com aquesta font d'il·luminació està situada al centre del Sol, els vectors normals que calcula l'esfera hauràn d'estar invertits de signe. D'aquesta manera, l'objecte es veu il·luminat de manera uniforme i deixa passar la font de llum als altres objectes.

### 2.1.2 Submòdul Planeta

Cada planeta generarà una esfera per coordenades esfèriques, i les funcionalitats especials que té són les rotacions. Hem de tenir en compte que s'ha de rotar respecte el Sol, i també l'objecte ha de rotar sobre sí mateix. Aquestes funcions tracten de fer operacions de translació, rotació i escalat per tal de veure com el planeta va orbitant. En el cas específic de la rotació respecte el Sol, els objectes fan un moviment el·líptic, que s'ha realitzat a partir de calcular els semieixos (la posició en el pla XZ i l'excentricitat de l'el·lipse) i les següents equacions per saber la propera posició del planeta.

$$x = a * \cos(\theta)$$

$$z = b * \sin(\theta)$$

*Fórmula 1 i 2: Càlcul de les noves posicions dels planetes*

$a$  i  $b$  representen ser els semieixos de l'el·lipse que traça el planeta, mentre que  $\theta$  serà la distància que es desplaçarà l'objecte, calculat per la velocitat i el temps.

### 2.1.3 Submòdul Satèl·lit

Els satèl·lits tenen molta similitud amb els planetes, però s'ha de tenir més paràmetres en compte. S'ha de tenir una distància entre el planeta i el satèl·lit, a part de també passar-li la velocitat del propi satèl·lit. Això és perquè els satèl·lits a part de rotar sobre sí mateixos i respecte el Sol, també orbiten respecte el seu planeta.

### 2.1.4 Submòdul Òrbita

Aquesta classe permet traçar les òrbites dels planetes. Tenim aquest mòdul per tal d'observar amb més facilitat el seu moviment. També ens ha servit per acabar de comprovar que està ben implementada les funcions que computen les rotacions dels objectes.

### 2.1.5 Submòdul Asteroide

Crea el cinturó d'asteroides del Sistema Solar. S'utilitza *instancing* per tal de crear una gran quantitat d'asteroides sense afectar greument el rendiment de l'aplicació. Aquests asteroides són esferes amb un radi, distància i velocitat aleatòries dintre d'una distribució uniforme. Diferenciem els asteroides en dos tipus: els que es troba entre el planeta Mart i Júpiter, i els que es troben en l'òrbita de Júpiter, anomenats *Trojans*. Únicament estan programats per tal que rotin respecte el Sol, amb capacitat de col·lisionar amb el demès asteroides. En cas de succeir una col·lisió, s'aplica els següents canvis en els objectes implicats:

- Es fa un canvi d'angle i de velocitat de rotació dels asteroides. En cas de ser el asteroide que ha impactat a un altre (el que tenia més velocitat en el moment), aquests paràmetres es redueixen lleument, i s'augmenten a l'inrevés.
- Per no produir un canvi brusc en l'ubicació dels asteroides quan s'apliquin els canvis, s'ha volgut realitzar les modificacions progressivament durant una certa quantitat de *frames*.

Trobar les col·lisions realment és una tasca complexa, i s'ha fet de dues formes diferents, que en el final de la implementació ens hem quedat amb el més eficient.

- **Forma 1:** Buscar els veïns propers utilitzant l'algorisme de *Kd-trees* [7] i anant afegint a una llista si impacten el asteroide actual de l'iteració amb els seus veïns.

- **Forma 2:** Fer un producte de matrius, per tal de quedar-nos finalment amb una matriu amb valors booleans, sent True en cas de col·lisionar.

La manera de calcular si impacten o no, és la mateixa per les dues formes, que es fa a partir de la següent fórmula:

$$d^2 \leq (r_1 + r_2)^2$$

*Fórmula 3: Asteroides que col·lisionen*

En la *Fórmula 3* es veu la manera de trobar una col·lisió entre dues esferes, on la suma de radis al quadrat ha de ser menor a la distància al quadrat.

### 2.1.6 Submòdul Anell

Aquest submòdul crea el conjunt d'anells específicament pels planetes Saturn i Urà. També es fa amb instancing per crear-los, encara que tenen una implementació més simple. Els anells no són esferes, sinó discos (cercles). Per tant, els vèrtexs i les coordenades textures es calculen de diferent forma, només tenint en compte el eix X i Z per fer-ho.

### 2.1.7 Submòdul Estrella

Les estrelles són en part més simples que els altres tipus d'objectes, perquè com a dades només necessitem saber les posicions d'aquestes i dibuixar punts en l'espai d'un color donat per la magnitud de l'estrella. També es té incorporat en aquest mòdul les constel·lacions, que es basa en traçar línies entre estrelles.

## 2.2 Mòdul Càmera

La càmera serà el mòdul que projecta el que hi ha en escena. Tenim en el nostre cas dos tipus de càmera: la càmera lliure, que et permet desplaçar-te pel espai sense cap impediment, i també tenim la càmera en mode satèl·lit, que la seva funció principal és seguir els planetes i observar-los d'aprop. La classe principal s'encarrega de fer funcionar la càmera lliure, on les seves funcionalitats permet a la càmera moure's endavant, enrere, cap els costats, amunt i avall. En el següent apartat es té una classe heretada de Càmera que inclou la càmera com a satèl·lit.

### 2.2.1 Submòdul FollowCamera

En el mode satèl·lit la càmera té diferents funcionalitats. S'ha de tenir en compte primerament quin planeta s'està seleccionant per mostrar. Seguidament, la càmera s'ubica en una posició que es pugui veure el planeta escollit i els seus satèl·lits. Aquest submòdul també admet a la càmera desplaçar-se mentre segueix el moviment dels planetes, on els podem rotar-la de diferents formes. Com que a l'espai no té sentit definir un pla de referència, l'usuari pot desorientar-se amb els moviments *WASD*. És per això que aquesta càmera pot rotar la visió en tres plans, a diferència de la càmera lliure on només es pot rotar en dos.

## 2.3 Mòdul Light

En la classe *Light* només s'ajusten els paràmetres d'il·luminació que es necessiten pel algorisme de *Phong*. Aquests paràmetres són la intensitat ambient, la intensitat difusa i la intensitat especular. Les variables es fan ús directament en el vertex shader i fragment shader.



## 2.4 Mòdul GUIManager

El mòdul *GUIManager* crearà tota l'interfície necessària per millorar l'experiència dels usuaris dintre de l'aplicació. Totes les funcionalitats que s'han afegit a partir d'aquesta classe són les de a continuació, que cadascuna correspon a un submòdul de la classe:

- **Botons:** Aquests poden ser de diferents formes (circulars o rectangulars).
- **Text:** Implementació de text dintre de l'aplicació. Bàsicament tots els botons tenen incorporat un text nomenant la seva funcionalitat.
- **Toggle:** Fer aparèixer o desaparèixer botons a partir d'un toggle. Bàsicament és perquè quedi més simple i no hi hagi sobreexcés d'informació.
- **Slider:** Barra que es pot desplaçar-se. En el nostre cas, en servirà per canviar de forma ràpida les velocitats dels planetes.

## 2.5 Mòdul Reader

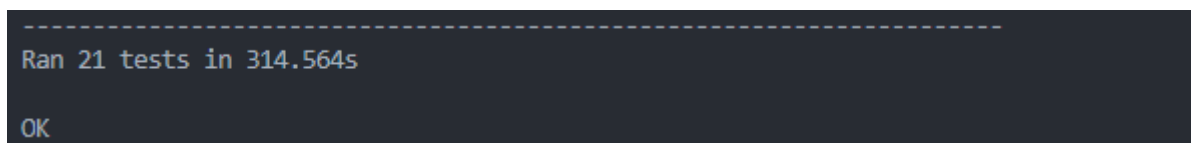
Llegeix les dades de diferents datasets. Els datasets són dades rellevants dels objectes del Sistema Solar, concretament sobre estrelles, planetes i satèl·lits. Totes aquestes dades al final es passaràn en els seus mòduls per crear els objectes, amb un valor que són realistes.

## 2.6 Mòdul GraphicsEngine

Crearà les instàncies dels altres mòduls i executarà l'aplicació *frame per frame*. Específicament, cada segon com a màxim hem establert que es pugui executar 120 *frames*. Dit d'una altra forma, aquest mòdul és el cicle de vida de l'aplicació: calcula el temps, que és un paràmetre imprescindible sobretot pel moviment dels objectes; s'encarrega d'emmagatzemar els objectes de l'escena i de cridar els càlculs de posició corresponents; i conté la interfície d'usuari, així com el control d'events que ofereixi l'usuari.

## 3. RESULTATS

Abans de mostrar els resultats, cal ressaltar que s'ha fet una sèrie de *unittesting* per tal de comprovar si les funcionalitats individualment funcionaven com s'esperava.

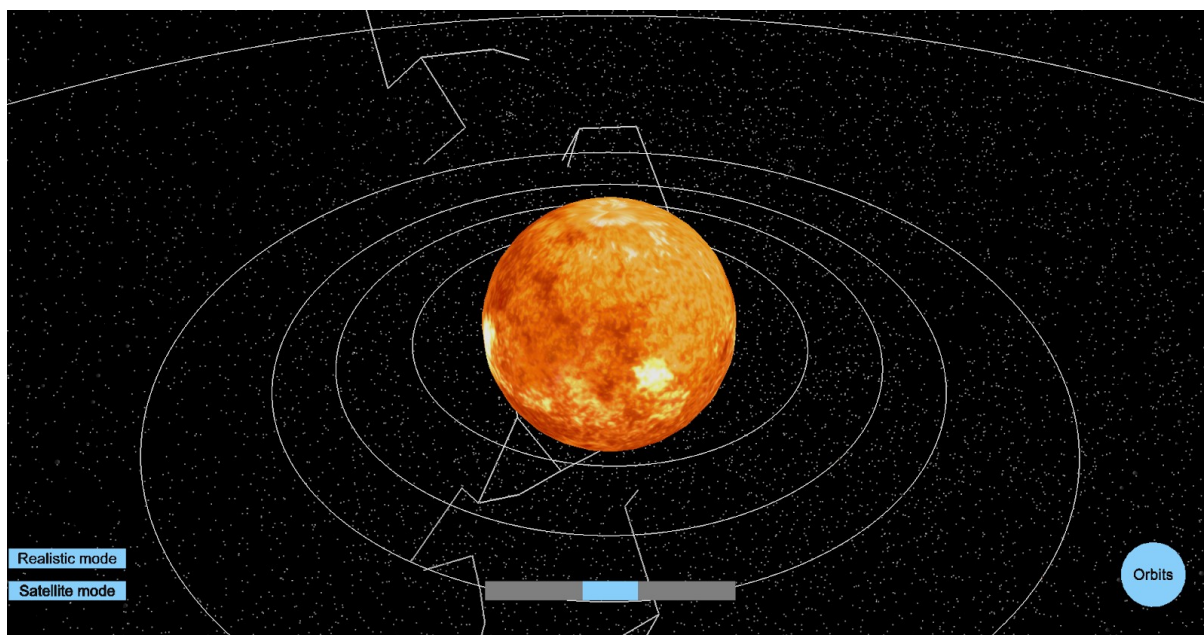


```
-----  
Ran 21 tests in 314.564s  
  
OK
```

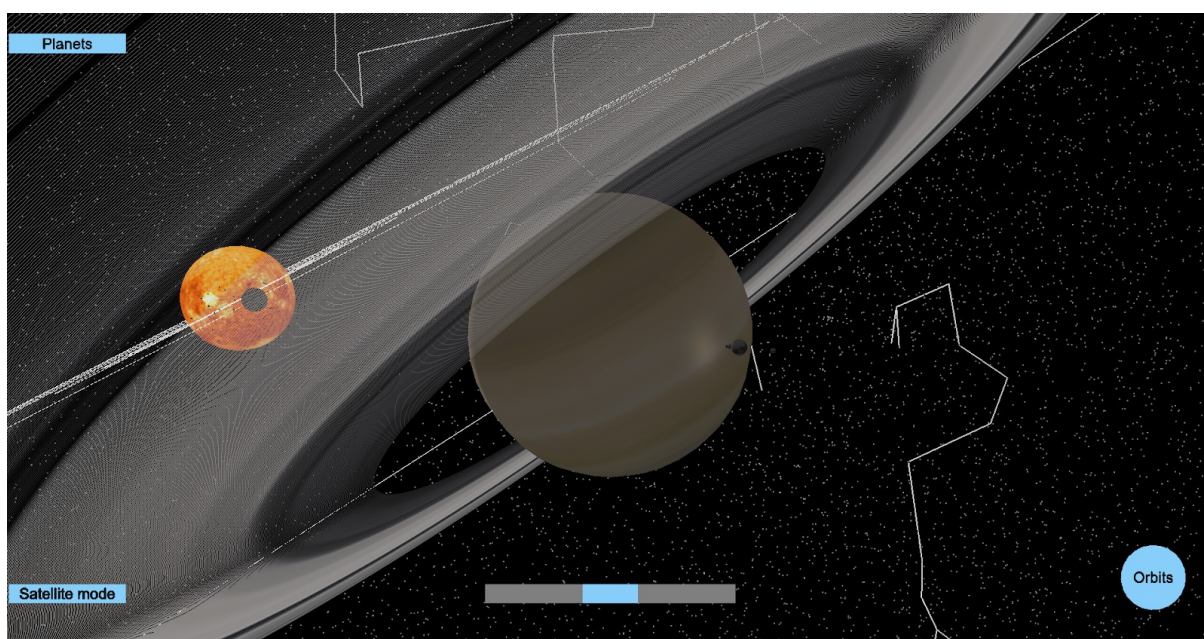
Figura 1: Execució dels *unittest*

En la Figura 1 es poden veure el nombre de test realitzats en total i el temps d'execució. Realment no s'ha comprovat totes les funcionalitats ja que és costós a nivell de temps. Encara que, hem anat comprovant amb l'execució si moltes d'aquelles funcionaven, com per exemple la rotació i translació dels objectes.

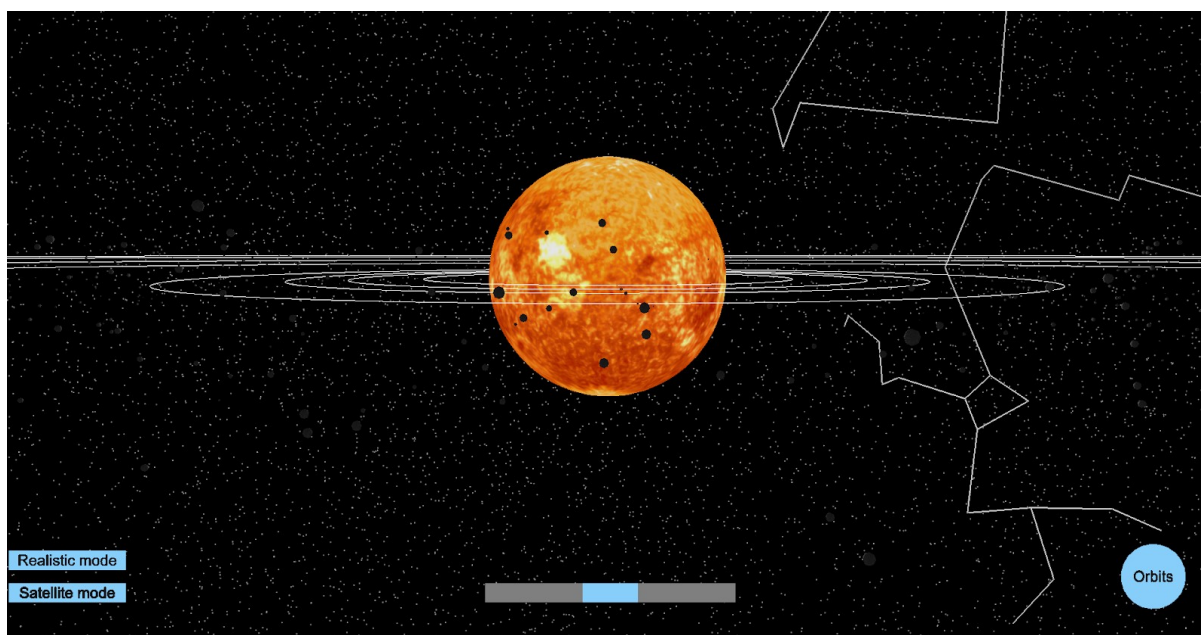
Dintre de l'escena de l'aplicació, podem visualitzar tots els objectes implementats. En les següents figures es mostraran exemples d'aquests.



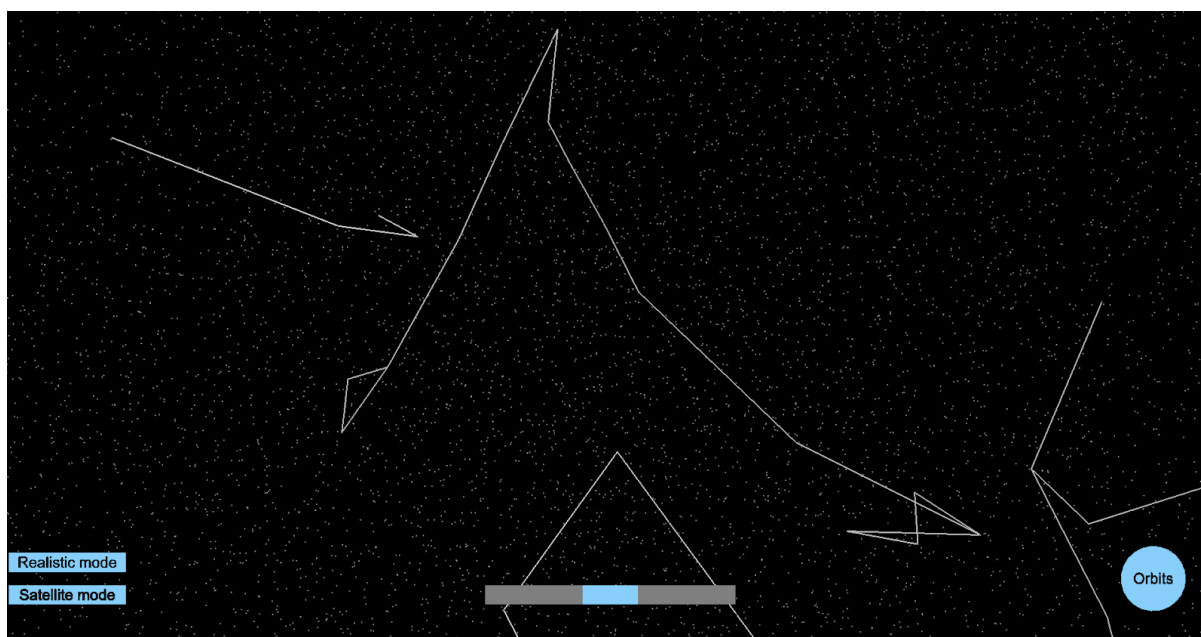
*Figura 2: Entrada de l'aplicació: Sol i òrbites*



*Figura 3: Visualització dels planetes, anells i satèl·lits*



*Figura 4: Visualització dels asteroides*



*Figura 5: Visualització de les estrelles i constel·lacions*

En la *Figura 2* és el que es mostra al iniciar l'aplicació, on es pot veure clarament el Sol i les el·lipses que traçen els planetes. Si obrim el mode satèl·lit, podrem veure per exemple la *Figura 3*, que com es pot veure en aquesta es veu el planeta en qüestió, i els seus satèl·lits i anells si en té. Si ubiquem la càmera entre Mart i Júpiter ens podrem trobar el cinturó d'asteroides semblant a la *Figura 4*. Per últim es pot observar les estrelles i constel·lacions com en la *Figura 5*. Estan els dotze signes del zodíac i les constel·lacions Pegasus i Cassiopeia implementades.

En els següents sub-apartats també es detallaran alguns paràmetres que hem vist necessari mostrar, i a part els problemes que podem tenir a nivell de textura i/o il·luminació.

### 3.1 Col·lisions

Fent referència a l'explicació dels asteroides (capítol 2.1.5), s'ha comparat quin era el millor mètode per trobar-los. A nivell lògic, la *Forma 1: Kd\_tree* té una complexitat quadràtica  $O(n^2)$ , ja que primerament es busquen els seus veïns, i novament s'iteren els veïns per comprovar si col·lisiona amb cert asteroide. En canvi, la *Forma 2: Numpy Product* només serà  $O(n)$ , ja que és un producte d'arrays i s'iteren solament un únic cop per trobar els índexs que es col·lisionen. Llavors hem utilitzat la *Forma 2* per detectar col·lisions. També s'ha demostrat això empíricament, veient quin és el *speedup* resultant al aplicar les dues tècniques.

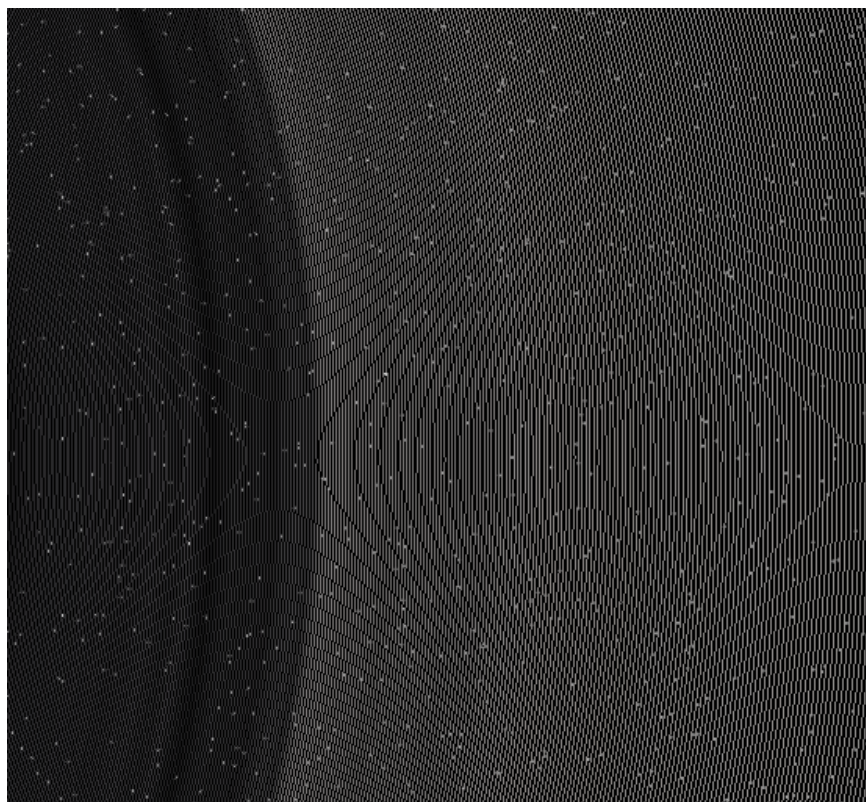
*Taula 2: Comparacions de temps en trobar col·lisions*

#Asteroides	Temps Kd_tree	Temps Numpy product	Speedup
250	0.0275 segons	0.001 segons	27x
500	0.094 segons	0.0035 segons	26.85x
1000	0.3455 segons	0.0161 segons	21.59x

En la *Taula 2* es pot veure la clara millora que hi ha en utilitzar el segon mètode, el que ens va permetre utilitzar un nombre d'asteroides més alt pel cinturó, i fer que aquest estigui més ple.

### 3.2 Problemes amb les textures

A nivell general no s'ha tingut molts problemes amb les textures, però hi ha diferents objectes que es veu l'*efecte de Moiré* [8], en concret quan creem els anells de Saturn i de Urà.



*Figura 6: Efecte de Moiré en els anells del planeta Saturn*



En la Figura 6 es pot veure clarament aquest efecte si la càmera es troba a prop dels anells. Encara que s'hagi intentat fer un mipmap per solucionar aquest efecte, amb els anells no ha estat funcional. Hem interpretat que aquestes línies comporten un patró molt repetitiu, que acaba generant aquest efecte.

## 4. CONCLUSIONS

En resum, s'ha arribat a desenvolupar una aplicació en *Python* amb les llibreries de *Moderngl* i *Pygame* un simulador del Sistema Solar intentant ser el més fidelment possible a la realitat, i també aconseguir veure les estrelles i constel·lacions del nostre voltant.

En qüestió dels objectius creiem que hem pogut arribar a assolir la gran majoria del que es va pensar en el començament del projecte, amb millors resultats que es tenien com a expectativa. Les funcionalitats principals que es van definir s'han complert.

En part, creiem que implementar el projecte en *Python* únicament amb les llibreries *pygame* i *opengl* porta avantatges com també desavantatges. No tenir un entorn ens limita bastant sobretot en la part d'interfície d'usuari.

Dintre del que hem desenvolupat, hi han certes coses millorables, com poden ser les textures i evitar els efectes que es veuen en l'apartat 3.2. També es pot continuar el treball per algunes idees que vam proposar l'inici del projecte que al final no s'han desenvolupat, com per exemple la visualització de l'espai des de un planeta, o bé la implementació d'un minijoc per endevinar els objectes que estan en escena, per promoure un aprenentatge en cas que l'usuari ho desitgi.

## 5. REFERÈNCIES ELECTRÒNIQUES I BIBLIOGRÀFIQUES

- [1. Stellarium] <https://stellarium-web.org>, versió en línia de l'aplicació Stellarium per a l'exploració de cels estrellats i constel·lacions en temps real (Data darrer accés: octubre 2024).
- [2. NASA Eyes] <https://eyes.nasa.gov/apps/solar-system/>, aplicació oficial de la NASA per a visualitzar el sistema solar en 3D i explorar l'òrbita de planetes i altres cossos celestes (Data darrer accés: octubre 2024).
- [3. Ombrage de Phong] [https://fr.wikipedia.org/wiki/Ombrage\\_de\\_Phong](https://fr.wikipedia.org/wiki/Ombrage_de_Phong), article de Wikipedia sobre l'ombratge de Phong, una tècnica de modelatge d'il·luminació de gràfics 3D (Data darrer accés: octubre 2024)
- [4. Kaggle - Planet Dataset] <https://www.kaggle.com/datasets/iamsouravbanerjee/planet-dataset>, base de dades pública amb informació sobre planetes i satèl·lits, disponible a Kaggle per a anàlisi de dades (Data darrer accés: octubre 2024).
- [5. HYG Database] <https://github.com/astronexus/HYG-Database/>, base de dades pública HYG que inclou informació sobre estrelles per a projectes astronòmics (Data darrer accés: octubre 2024).
- [6. VizieR] <https://vizier.cds.unistra.fr/viz-bin/VizieR?-source=V%2F53A>, base de dades de VizieR per a la consulta de catàlegs astronòmics (Data darrer accés: octubre 2024).

- [7. KDTree] <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KDTree.html>, documentació oficial sobre l'ús de KDTree per a consultes ràpides en dades espacials dins de la llibreria Scikit-Learn (Data darrer accés: decembre 2024).
- [8. Wikipedia - Moiré Pattern] [https://en.wikipedia.org/wiki/Moir%C3%A9\\_pattern](https://en.wikipedia.org/wiki/Moir%C3%A9_pattern), article de Wikipedia que explica el fenomen de patrons Moiré, les seves aplicacions i causes en diferents camps (Data darrer accés: decembre 2024).

## ANEXE I. ACTES DE LES REUNIONS

Creiem que és necessari sobretot profunditzar l'evolució de les tasques 2, 3 i 4, ja que en ambdós impliquen fer diverses subtasques a la vegada.

*Taula 3: Subtasques per realitzar en el disseny*

Subtasca	Descripció	Responsable	Participants (%)	Durada (setmanes)	Grau Finalització
<b>1. Mapa conceptual</b>	Realitzar un esquema amb les possibles idees que podem implementar	Guillem	Guillem (90%), Martí, Joel, David (10% restant)	1	100%
<b>2. Diagrama de classes</b>	Diagrama mostrant les funcionalitats de l'aplicació i com es relacionen	Guillem	Guillem (50%), Martí, Joel, David (50% restant)	1	100%
<b>3. Mock-Up</b>	Disseny d'interfície d'usuari	Joel	Joel (80%) Martí, Guillem, David (20% restant)	1	100%

En la *Taula 3* es pot observar quines han són les tasques de disseny abans de començar amb la implementació de codi. Bàsicament consta d'un mapa d'idees, un diagrama de classes per estructurar els mòduls, i un Mock-Up per com serà visualment l'interfície.

*Taula 4: Subtasques per realitzar la implementació de l'aplicació*

Subtasca	Descripció	Responsable	Participants (%)	Durada (setmanes)	Grau Finalització
<b>1. Crear planeta Terra</b>	Crear l'objecte Terra i afegir textures necessàries	Martí	Martí (70%), David (20%), Joel, Guillem (10% restant)	2	100%
<b>2. Crear Sol</b>	Crear l'objecte Sol, amb les textures i il·luminació adequada	Martí	Martí (70%), Joel, David, Guillem (30% restant)	2	100%
<b>3. Moviment objectes</b>	Fer orbitar els planetes sobre sí mateix i el Sol	Guillem	Guillem (70%), Joel, David, Martí (30%)	1	100%
<b>4. Datasets</b>	Llegir i afegir en els objectes les dades dels datasets	David	David (70%), Joel, Guillem, Martí (30% restant)	2	100%

Subtasca	Descripció	Responsable	Participants (%)	Durada (setmanes)	Grau Finalització
<b>5. Crear altres planetes</b>	Posar els altres planetes i fer-los orbitar	Guillem	Guillem(80%), Joel, David, Martí (20% restant)	2	100%
<b>6. Afegir estrelles</b>	Crear les estrelles i afegir-les a les coordenades corresponents de l'espai	David	David (80%), Guillem, Martí, Joel (20% restant)	2-3	100%
<b>7. Afegir satèl·lits naturals</b>	Incorporació dels satèl·lits naturals als planetes	Guillem	Guillem (80%), Martí, Joel, David (20% restant)	1	100%
<b>8. Punts de vista Sistema Solar</b>	Afegir punts de vista de la càmera per tenir millor interactivitat	Martí	Martí (80%) Joel, David, Guillem (20% restant)	2	100%
<b>9. Detalls del Sistema Solar</b>	Afegir els anells a Saturn, cinturó d'asteroides entre Mart i Júpiter, col·lisionament entre asteroides, etc.	Guillem	Guillem (80%), Joel, David, Martí (20% restant)	2	100%
<b>10. Constel·lacions</b>	Implementar algunes de les constel·lacions més conegudes, connectant les estrelles que ja tenim	David	David (80%), Joel, Guillem, Martí (20% restant)	2	100%
<b>12. Perfeccionar implementacions</b>	Afegir o millorar els objectes que tenim implementats en el Sistema per tal de perfeccionar aquest	Tots	Tots (25%)	2	100%
<b>13. Funcionalitats usuari</b>	Afegir interaccions d'usuari com botons, zoom-in, zoom-out, entre altres.	Joel	Joel (70%), Martí (30%)	10	100%

Per últim tenim la *Taula 4*, amb el repartiment de les tasques d'implementació que s'han realitzat durant el projecte. Les tasques principalment es basa en crear objectes o afegir detalls dintre de l'escena.



## ANEXE II. DIAGRAMA DE MÒDULS

