

CONTROL No. 2

21 NOVEMBRE 2024

ASSISTENTS: Guillem Garcia, Martí Llinés, David Montaña i Joel Tapia

HORA D'INICI: 12:30

LLOC: AULA Q5/1012

1. Objectius del projecte

La proposta i els objectius del projecte són els següents:

Es requereix construir un simulador del Sistema Solar i diverses constel·lacions utilitzant els coneixements apresos en Realitat Augmentada i Entorns Gràfics. La idea principal és poder representar el Sistema Solar al més semblant possible a la realitat, per la qual cosa es farà servir dades recopilades de diferents bases de dades.

Principalment, es vol construir dues vistes: una que representaria l'espai, on es podrien veure els planetes i satèl·lits en moviment, i una altra vista des del planeta Terra, que permeti a l'usuari controlar el temps i veure el cel estrellat quan sigui de nit.

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.

Si hi ha temps disponible, volem implementar un mode de joc que es basaria a endevinar on es troben els planetes, satèl·lits i constel·lacions. El joc tindria un temporitzador, i de forma aleatòria, seleccionaria un objecte a endevinar. L'usuari tindria un màxim de tres intents per cada objecte, i el joc acabaria o bé quan s'hagin intentat endevinar tots els objectes, o bé quan s'acabi el temps.

Les eines i llibreries que es faran servir per assolir els objectius són:

- **Python:** Llenguatge de programació a generar i executar el codi. La versió utilitzada és la 3.11.9.
- **Llibreria *pygame*:** Facilita la creació de videojocs i aplicacions multimèdia en Python. S'utilitzarà per construir l'aplicació interactiva 2D de forma senzilla.
- **Llibreria *moderngl*:** Llibreria per gràfics en 3D que proporciona una interfície d'alt nivell per *OpenGL* en Python. L'ús que es farà servir és per crear els objectes, i configurar els seus *shaders*, textures, il·luminació, etc.
- **Llibreria *glm*:** Llibreria de matemàtica freqüentment utilitzada per gràfics 3D. Es crearan a partir d'aquesta llibreria els vectors i matrius, i també per realitzar tota operació geomètrica que s'hagi de fer.
- **Llibreria *json*:** Permet treballar amb dades de format *JSON*. Sobretot es farà servir per a la interfície d'usuari.
- **Llibreria *sys*:** Ens permet interactuar amb el sistema operatiu i acabar l'execució de la *app*.
- **Llibreria *numpy*:** S'usarà per fer càlculs d'arrays de forma eficient.

- **Llibreria pandas:** Per tal de llegir les dades dels *datasets* i aplicar-los als objectes.

2. Planificació

La planificació l'hem dividit per tasques a realitzar. Per ser més curosos, hem volgut separar les tasques globals que s'han de fer, amb les subtasques específiques que comporta fer cada tasca.

Taula 1: Tasques globals del projecte

Tasca	Descripció	Responsable	Participants (%)	Durada (setmanes)	Grau Finalització
1. Estat de l'Art	Buscar productes/aplicacions similars al que es proposa, caracteritzar-los i comparar-los	Martí	Martí (80%), Guillem, Joel, David (20% restant)	1	100%
2. Disseny Aplicació	Disseny dels diferents components de l'aplicació i elecció d'eines de desenvolupament	Guillem	Guillem (80%), Martí, Joel, David (20% restant)	2	100%
3. Disseny Interfície d'Usuari	Disseny de la interfície d'usuari	Joel	Joel (80%) Martí, Guillem, David (20% restant)	2	100%
4. Implementació	Implementació de l'aplicació	David	David (35%) Martí, Guillem, Joel (65% restant)	8-10	70%
5. Test	Test interns i externs (usuaris)	Tots	Tots (25%)	2	0%
6. Memòria i Presentació	Escriure la memòria i la presentació oral	Tots	Tots (25%)	1	0%
GRAU DE MADURESA:					65%

En la *Taula 1* es pot mostrar les tasques principals a realitzar del projecte. Aquestes tasques estan molt generalitzades, pel que hem vist necessari desenvolupar sobretot les tasques que comporta fer tant el disseny d'aplicació i d'usuari, com també la implementació de l'aplicació.

Taula 2: Subtasques de disseny

Subtasca	Descripció	Responsable	Participants (%)	Durada (setmanes)	Grau Finalització
1. Mapa conceptual	Realitzar un esquema amb les possibles idees que podem implementar	Guillem	Guillem (90%), Martí, Joel, David (10% restant)	1	100%
2. Diagrama de	Diagrama	Guillem	Guillem (50%),	1	100%

Subtasca	Descripció	Responsable	Participants (%)	Durada (setmanes)	Grau Finalització
classes	mostrant les funcionalitats de l'aplicació i com es relacionen		Martí, Joel, David (50% restant)		
3. Mock-Up	Disseny d'interfície d'usuari	Joel	Joel (80%) Martí, Guillem, David (20% restant)	1	100%

Com es veu en la *Taula 2*, s'han desglossat les tasques de disseny. Principalment s'ha començat per una pluja d'idees, i seguidament s'ha realitzat un *Diagrama de Classes* i un *Mock-up* bàsic per tal de visualitzar de millor forma el que es vol fer.

Taula 3: Subtasques implementació

Subtasca	Descripció	Responsable	Participants (%)	Durada (setmanes)	Grau Finalització
1. Crear planeta Terra	Crear l'objecte Terra i afegir textures necessàries	Martí	Martí (70%), David (20%), Joel, Guillem (10% restant)	2	100%
2. Crear Sol	Crear l'objecte Sol, amb les textures i il·luminació adequada	Martí	Martí (70%), Joel, David, Guillem (30% restant)	2	100%
3. Moviment objectes	Fer orbitar els planetes sobre sí mateix i el Sol	Guillem	Guillem (70%), Joel, David, Martí (30%)	1	100%
4. Datasets	Llegir i afegir en els objectes les dades dels datasets	David	David (70%), Joel, Guillem, Martí (30% restant)	2	100%
5. Crear altres planetes	Posar els altres planetes i fer-los orbitar	Guillem	Guillem(80%), Joel, David, Martí (20% restant)	2	100%
6. Afegir estrelles	Crear les estrelles i afegir-les a les coordenades corresponents de l'espai	David	David (80%), Guillem, Martí, Joel (20% restant)	2-3	100%
7. Afegir satèl·lits naturals	Incorporació dels satèl·lits naturals als planetes	Guillem	Guillem (80%), Martí, Joel, David (20% restant)	1	100%
8. Punts de vista Sistema Solar	Afegir punts de vista de la càmera per tenir millor interactivitat	Martí	Martí (80%) Joel, David, Guillem (20% restant)	2	70%

Subtasca	Descripció	Responsable	Participants (%)	Durada (setmanes)	Grau Finalització
9. Detalls del Sistema Solar	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	50%
10. Constel·lacions	Implementar algunes de les constel·lacions més conegudes, connectant les estrelles que ja tenim	David	David (80%), Joel, Guillem, Martí (20% restant)	1	60%
11. Visualització Terra	Afegir un mode de visualització des de la Terra veient el cel estrellat.	-	-	-	0%
12. Ombres/llums	Afegir i controlar ombres i llums als planetes i constel·lacions segons a on es troben	-	-	-	0%
13. Funcionalitats usuari	Afegir interaccions d'usuari com botons, zoom-in, zoom-out, entre altres.	Joel	Joel (80%), Guillem, David, Martí (20% restant)	8-9	50%
14. Geoguessr	Crear un mini-joc senzill de trobar el planeta/constel·lació triat aleatòriament	-	-	-	0%

Per últim es té *Taula 3*, que es resum totes les tasques que s'ha tingut en compte dintre de la implementació. Es divideix bàsicament pels objectes que s'han de crear i les dades que s'han d'incloure dintre del codi.

3 Desenvolupament del projecte

A continuació, s'explicarà detalladament el procediment de cada tasca global, i quins resultats hem extret realitzant la tasques.

3.1. Estat de l'art

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

- **Stellarium** ^[1]: 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)** ^[2]: 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.

3.2. Disseny de l'aplicació

En qüestió del disseny de l'aplicació, s'ha optat a fer un *Diagrama de Classes* per tal d'entendre quins tipus de mòduls necessitem i les funcionalitats que ha de realitzar cada mòdul.

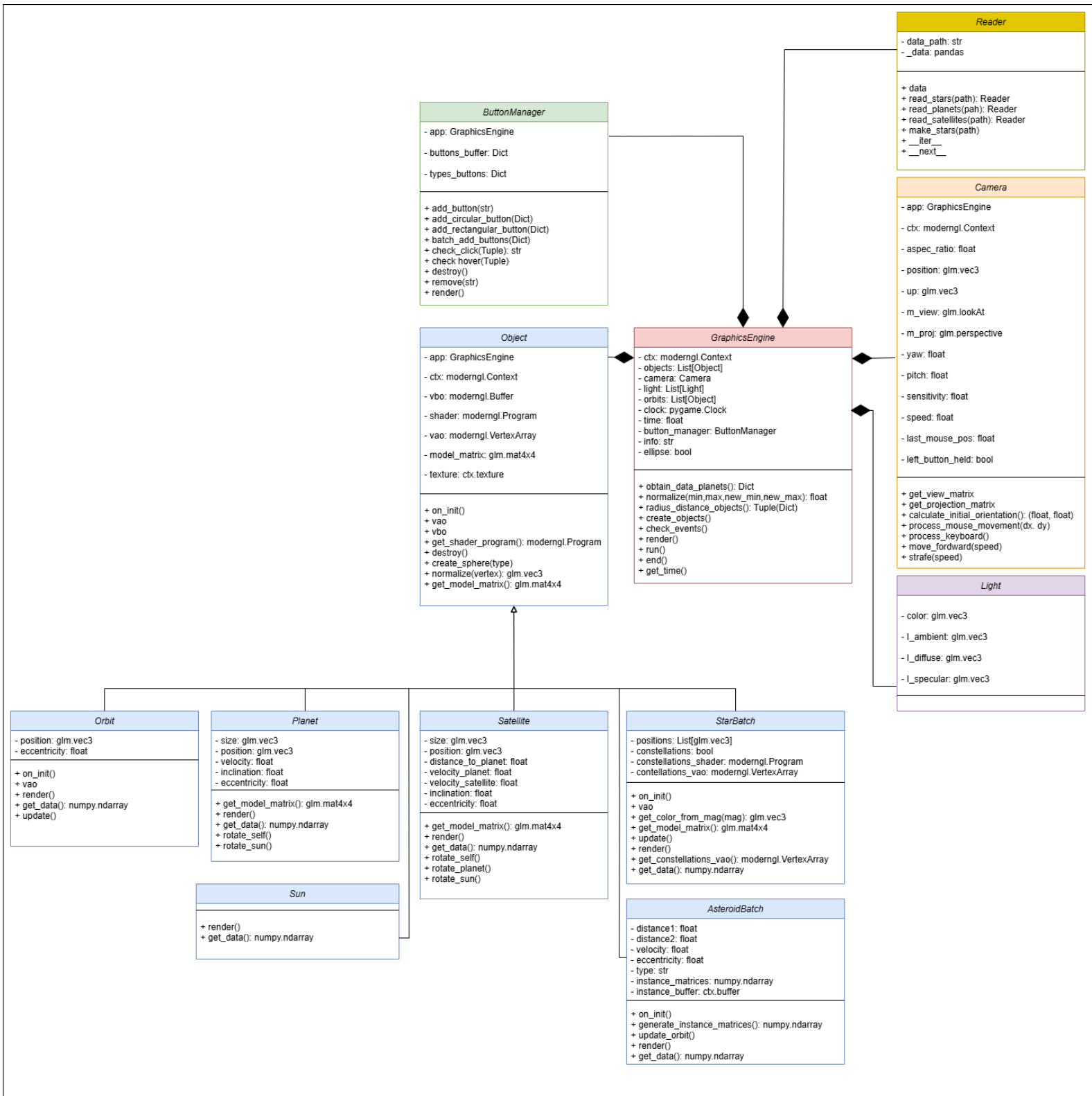


Figura 1: Diagrama de classes

En la *Figura 1* es pot veure el Diagrama de Classes construït fins al moment. Cal aclarir que aquest no és el disseny definitiu. És una representació dels mòduls que són més prioritaris i els que hem començat a desenvolupar fins al Control 2.

El Diagrama consta de sis mòduls:

- **Object:** Serà la classe que s'utilitza per crear tot objecte de l'aplicació. Aquesta classe té més classes heretades, que es treballa específicament amb un tipus d'objecte. Amb aquest mòdul podem crear de moment el Sol, els planetes, els satèl·lits, estrelles, constel·lacions, les òrbites i els asteroides.
- **Camera:** Classe que controla la càmera (el què veiem des de el *View Port*). Les funcionalitats de la classe permet inicialitzar la càmera i desplaçar-la per l'espai.
- **Light:** Classe que serveix per establir els paràmetres d'il·luminació dels objectes.
- **ButtonManager:** Estableix els botons de l'aplicació. Aquesta classe controlarà tot el que té a veure amb la interfície d'usuari.
- **GraphicsEngine:** Motor de l'aplicació. Aquesta classe sincronitza els altres mòduls i els executa.
- **Reader:** Llegeix les dades de diferents datasets. Els datasets són dades rellevants dels objectes del Sistema Solar, específicament es té bases de dades d'estrelles, planetes i satèl·lits.

3.3. Disseny de la interfície d'usuari

Hem optat a que la interfície d'usuari sigui el més senzill possible per tal de que sigui intuïtiu l'ús de l'aplicació. Es té pensat que les interaccions dels usuaris puguin ser tant a través de *keybinds* del teclat, com bé amb diferents botons que s'incorporarà a l'aplicació i s'encarregarà el mòdul *ButtonManager*.

Les opcions que podrà realitzar l'usuari són els de la següent llista:

- Moure la càmera espacialment
- *Zoom-in, Zoom-out*
- Habilitar/Deshabilitar mostrar un tipus d'objecte
- En cas de estar en la Terra veient el cel, controlar el temps per veure'l de dia i de nit.
- Parar/Continuar òrbita dels planetes.

3.4. Implementació

En aquest punt es profunditzarà com estan construïts els mòduls i els submòduls del nostre disseny de l'aplicació esmentat anteriorment.

3.4.1. Classe Object

Tot objecte creat dintre de l'escena compartirà una sèrie de característiques, que totes estan incorporades dintre d'aquesta classe. El que ha de tenir tot objecte principalment és un *vao*, un *vbo* i un *shader* per tal de poder-se renderitzar. Si ens fixem en la *Figura 1*, també s'observa que conté altres tipus de funcionalitats que són genèriques, com per exemple crear una esfera o també carregar la textura de l'objecte.

Les esferes que construïm estaran feta per *stripes*, que consisteix bàsicament en crear l'esfera a partir de coordenades esfèriques. Es tenen els paràmetres φ i θ que tenen un rang de $[0, \pi]$ i $[0, 2\pi]$ respectivament, i a partir de fer segmentacions es crea l'esfera convertint les coordenades esfèriques a cartesianes amb les següents equacions:

$$\begin{aligned}x &= r * \sin(\theta) * \cos(\varphi) \\ y &= r * \sin(\theta) * \sin(\varphi) \\ z &= r * \cos(\theta)\end{aligned}$$

Fórmula 1: Equacions de conversió a coordenades cartesianes

«on r és el radi de l'esfera que volem crear». En la *Fórmula 1* es veu com es calcularà les coordenades cartesianes x , y i z tenint els paràmetres r , φ i θ .

També tenim les classes heretades, on cadascuna té les seves característiques pròpies.

- **Classe Planet:** Cada planeta generarà una esfera per *stripes*, i les funcionalitats especials que té és la rotació respecte el Sol i respecte a 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 planetes 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 2: Càlcul de les noves posicions dels planetes

«on a és el semieix major, b és el semieix menor, i θ és la velocitat en què volem orbitar»

- **Classe Sun:** Pel Sol només tenim un mètode que és diferent a la classe principal. Com aquest serà el nostre objecte que il·luminarà els demès, aquest al calcular l'esfera necessitarà que els vectors normals estiguin canviats de signe.
- **Classe Satellite:** Aquest mòdul permet incorporar els satèl·lits naturals que tenen cada planeta. La diferència respecte els planetes, principalment és que es té una distància respecte el planeta, i també han de rotar respecte aquest.
- **Classe Orbit:** Aquesta permet traçar les òrbites dels planetes. Tenim aquest mòdul per tal d'observar amb més facilitat el seu moviment.
- **Classe AsteroidBatch:** Crea el cinturó d'asteroides del Sistema Solar. La diferència principal amb la resta de classes és que s'utilitza *instancing* per tal de crear una gran quantitat d'asteroides sense afectar greument el rendiment de l'aplicació.
- **Classe StarBatch:** 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 els punts 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.

3.4.2. Classe Light

En la classe Light només s'ajusten els paràmetres d'il·luminació que es necessiten pel algorisme de *Phong* [3]. Aquests paràmetres són la intensitat ambient, la intensitat difusa i la intensitat especular.

3.4.3. Classe Camera

La classe càmera conté funcionalitats per tal de obtenir les matrius de projecció i de *view*, i a part per moure-la espacialment. *move_forward()* i *strafe()* s'utilitzen per moure la càmera endavant, enrere, dreta i esquerra. Hem de tenir en compte que quan es mou la càmera, també s'han d'actualitzar les matrius *view* dels objectes. I *process_mouse_movement()* serveix per establir la direcció de la càmera (que en el nostre cas es farà amb el ratolí).

3.4.4. Classe ButtonManager

Les funcionalitats d'aquesta classe ens permetrà posar en escena els botons per a que l'usuari pugui interactuar més apropiadament amb l'aplicació. De moment, només es poden afegir botons circulars i rectangulars, i al pulsar-los encara no fa cap interacció.

3.4.5. Classe GraphicsEngine

Crearà les instàncies dels altres mòduls i executarà l'aplicació *frame per frame*. Les funcions que té es basen controlar tota interacció que fa l'usuari *check_events()*, actualitzar el temporitzador *get_time()* (utilitzat pel moviment dels planetes), renderitzar totes les instàncies *render()*, i per últim començar o finalitzar el programa.

3.5. Resultats

Fins el primer Control, els resultats que es van arribar van ser els següents:

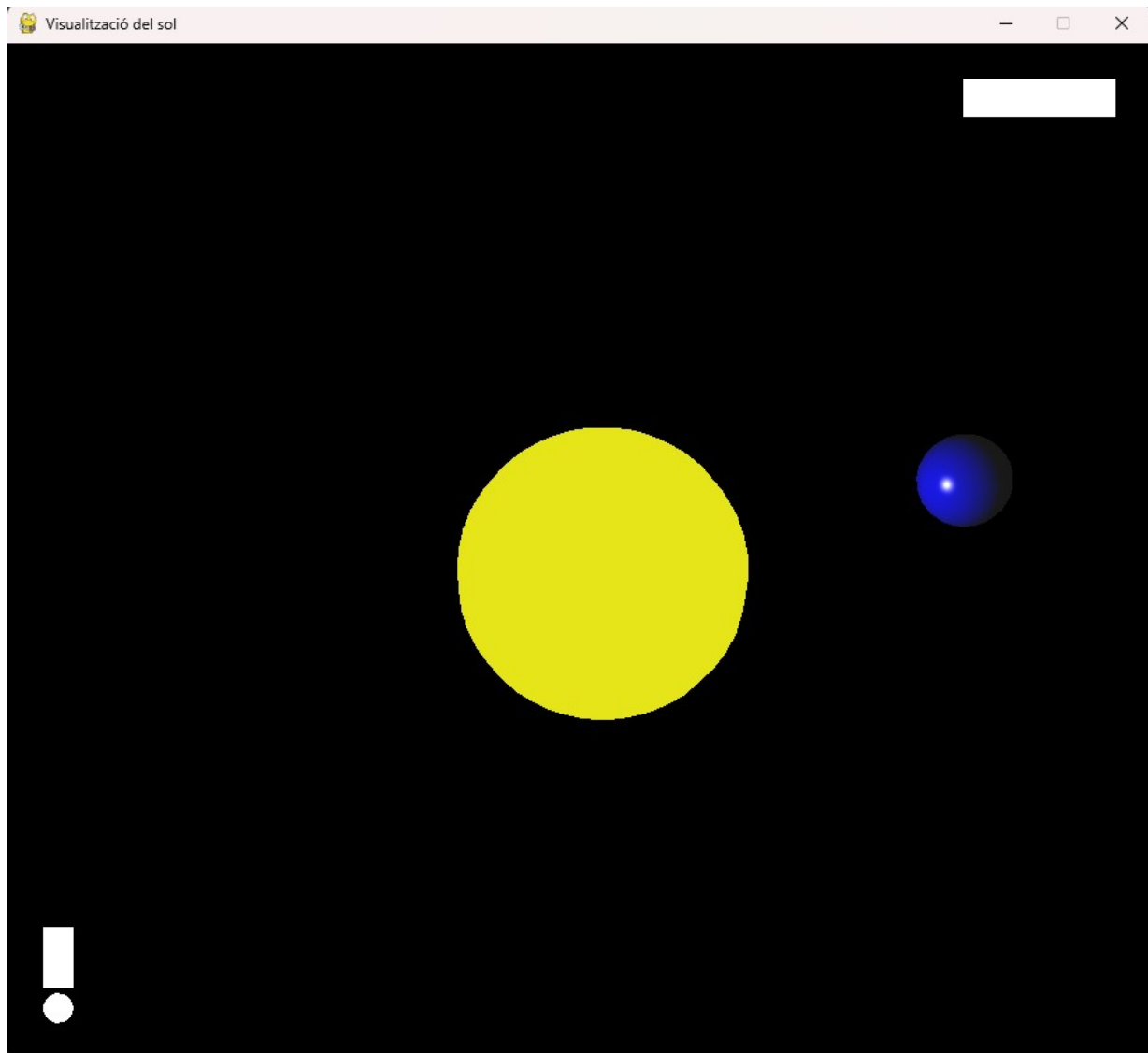


Figura 2: Resultats Control 1

En la *Figura 2* es poden veure aquests resultats, on només es té creats el Sol i la Terra i que aquesta vagi orbitant. També es tenia les primeres implementacions amb la càmera, aconseguint moure-la amb les tecles *W,A,S,D*.

Com a segon control hem arribat als resultats que es mostraran a continuació, explicant alguns exemples de les visualitzacions que hem obtingut.

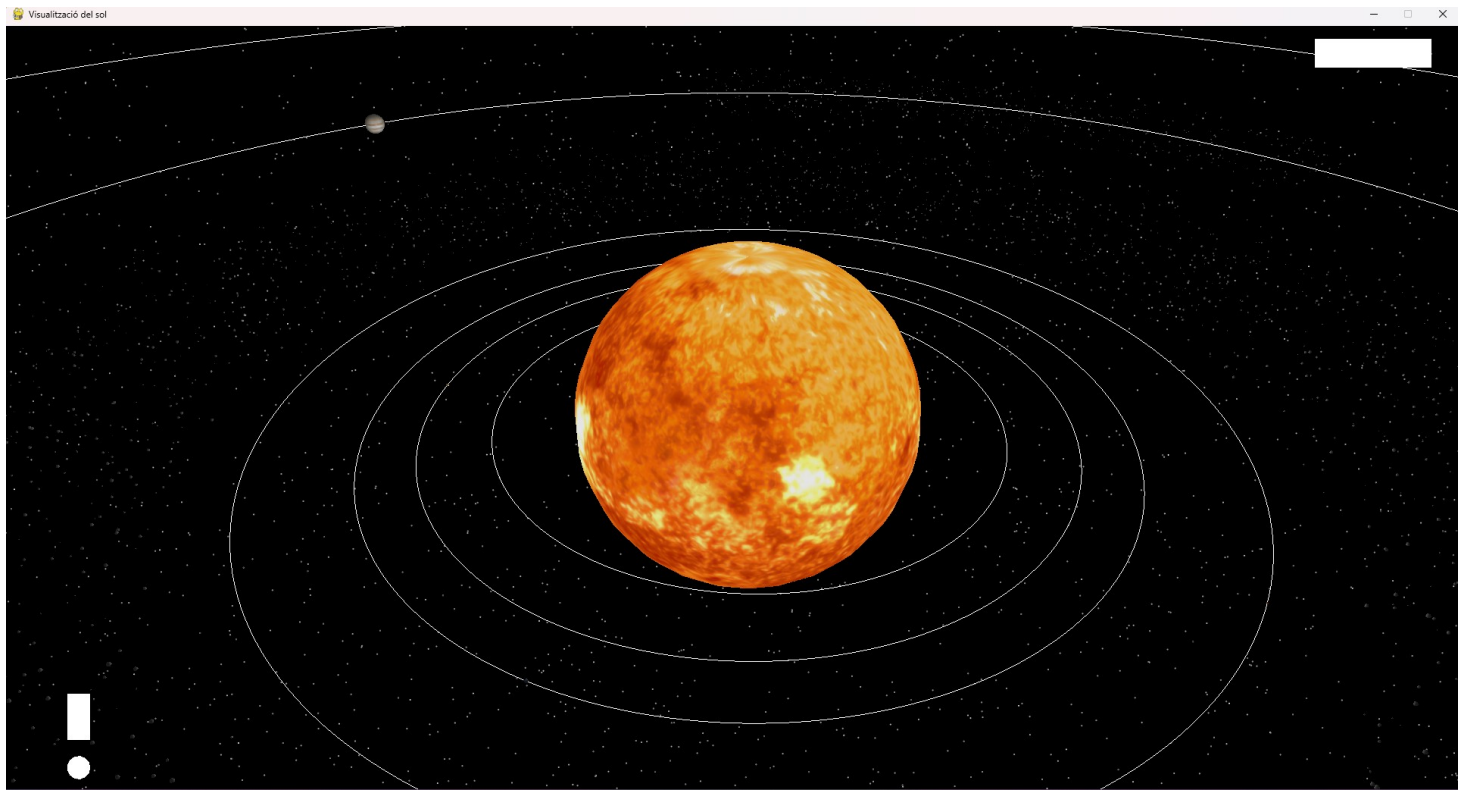


Figura 3: Visualització del Sistema Solar

En la *Figura 3* es pot veure que s'ha anat seguint implementant durant les sessions. Es pot observar que els objectes ara tenen una textura. També, es pot observar que tots els planetes es troben orbitant seguint les el·lipses blanques traçades, o bé que s'han afegit correctament les estrelles.

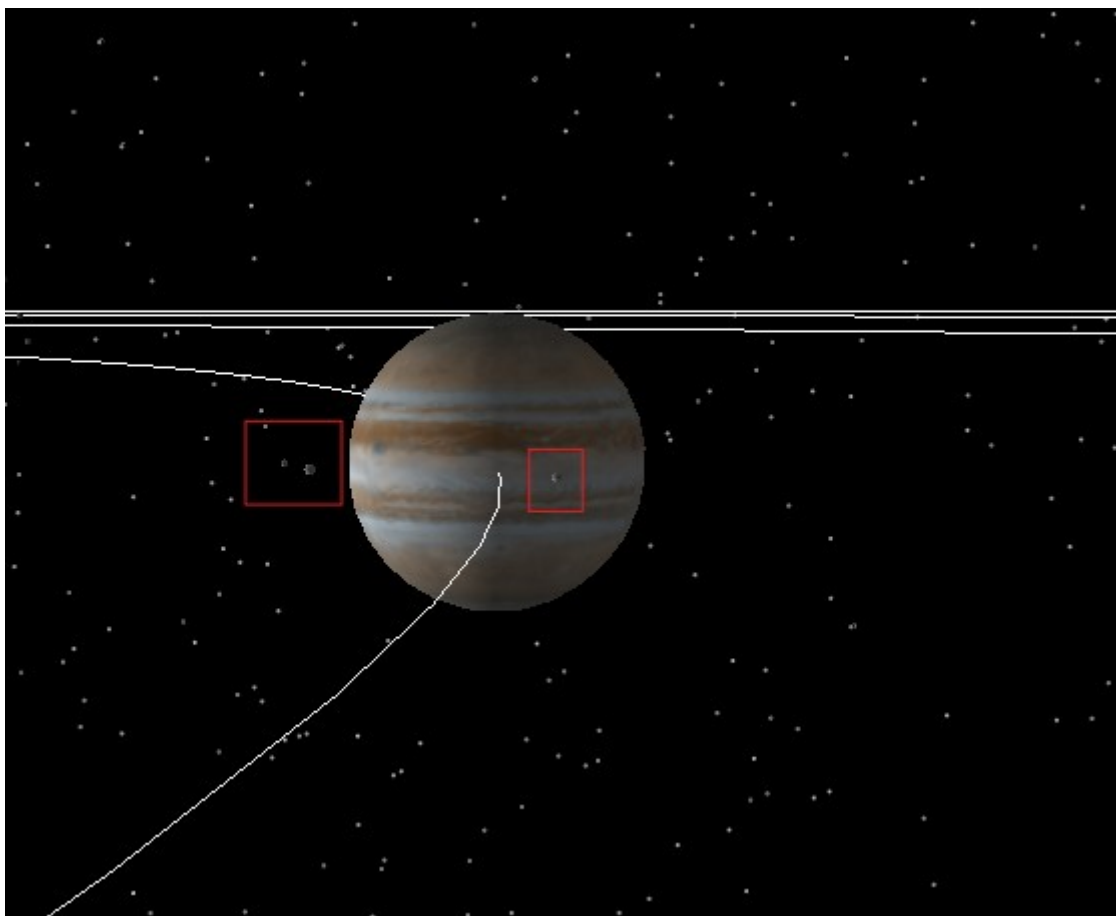


Figura 4: Visualització dels satèl·lits

En la *Figura 4* podem apreciar millor els satèl·lits que s'han creat en aquest cas en Júpiter.

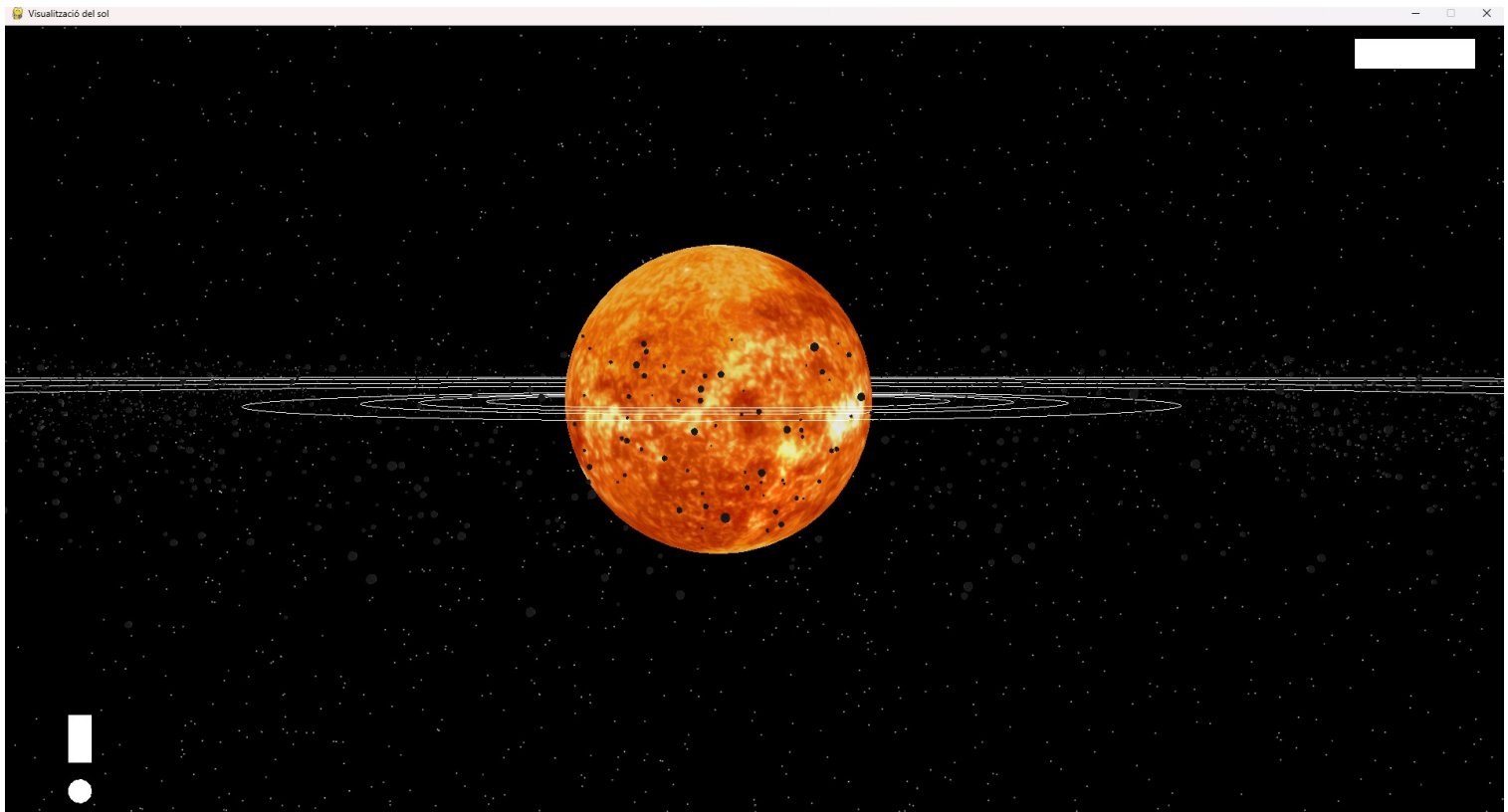


Figura 5: Visualització dels asteroides

Com es veu en la *Figura 5*, es pot veure entre Mart i Júpiter el cinturó d'asteroides que s'ha creat, on cada asteroide tenen una posició, mida i velocitat diferent, que va orbitant respecte el Sol.

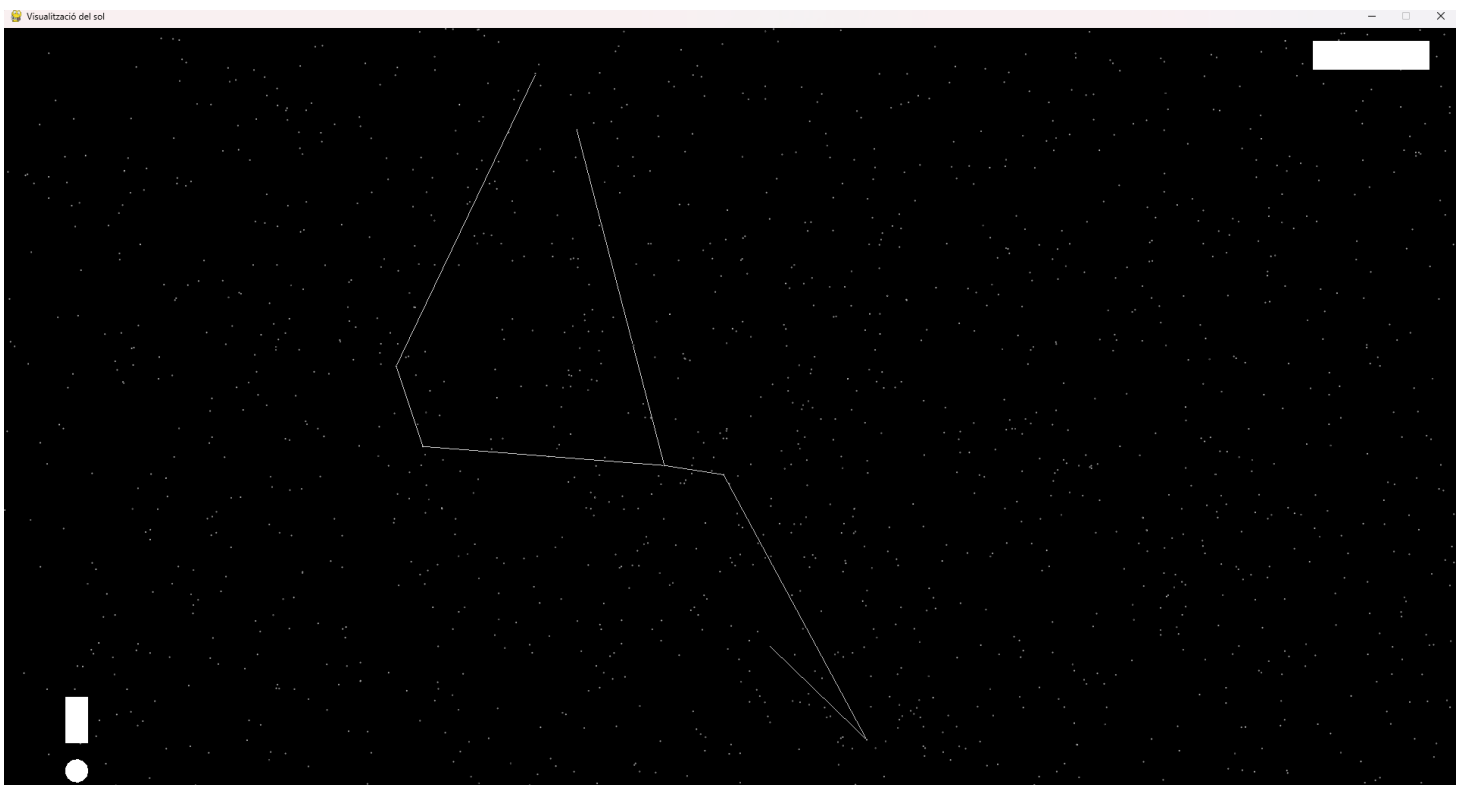


Figura 6: Visualització de les constel·lacions

Per últim podem veure la *Figura 6*, que s'ensenya una de les constel·lacions que hem traçat. Aquesta constel·lació s'anomena *Pegasus*.

Cal ressaltar que el Sistema Solar principal no és a escala real, sinó que estan els radis i distàncies normalitzats. És a dir, proporcionalment entre els radis i les distàncies són les mateixes, però els hi establim un rang per tal de que la visualització final sigui més compacte.

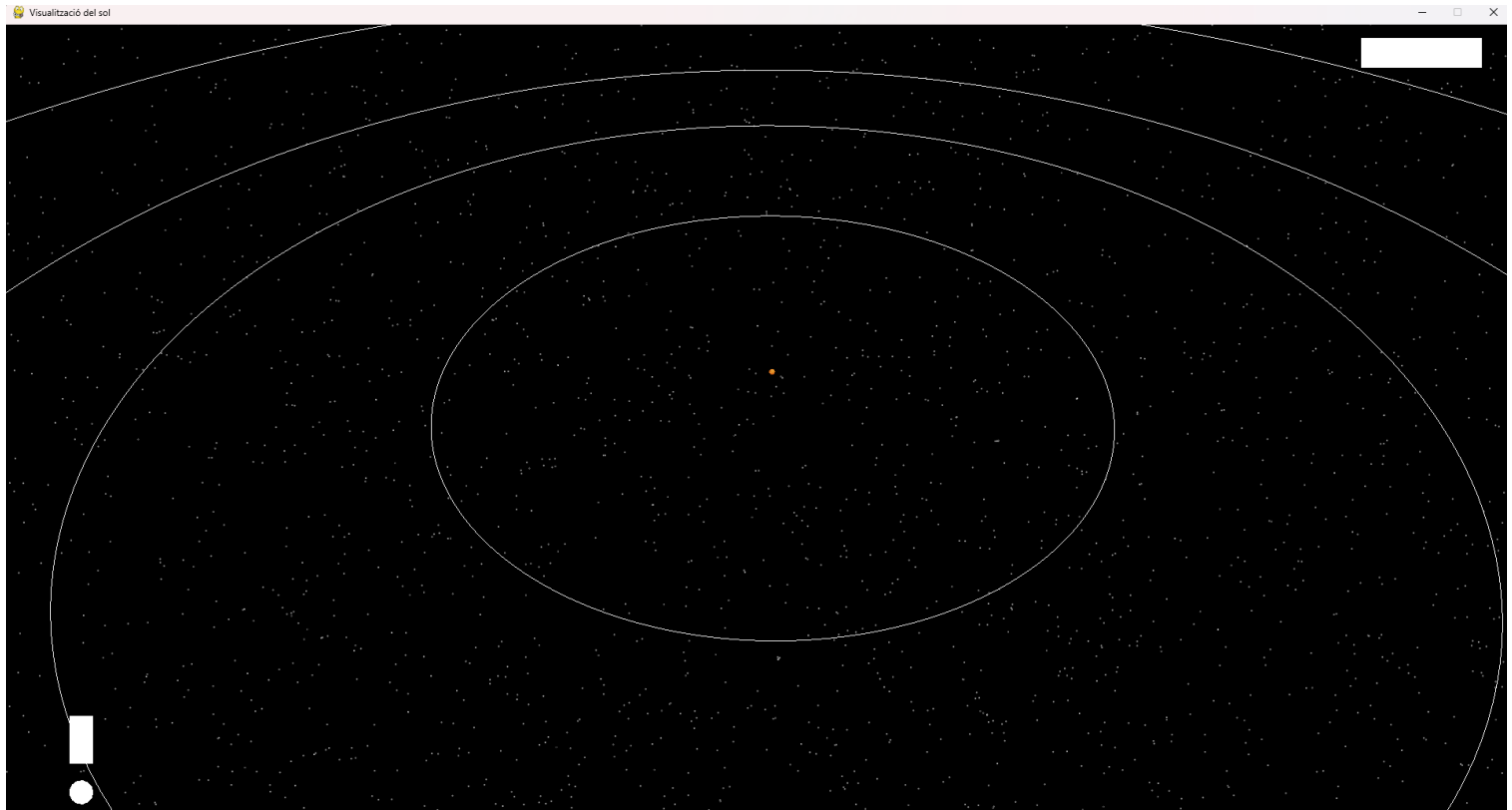


Figura 7: Visualització del mode realista

El que veiem en la *Figura 7* és el que passa si mantenim tot amb les mateixes unitats però sense normalitzar-ho. El que ocorreix és que les distàncies són molt més grans que els radis dels objectes respectivament, el que dona una visualització on és bastant costós veure els objectes.

5. Bibliografia i referències

- [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, 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, 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).

Signatures:

Guillem Garcia Dausà

Martí Llinés Viñals

David Montaña Tseitlin

Joel Tapia Salvador