

CONTROL No. 1

27 OCTUBRE 2024

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

HORA D'INICI: 10:00

LLOC: ONLINE

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.

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. A continuació es veurà una taula amb les tasques principals:

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	35%
5. Test	Test interns i externs (usuaris)	-	-	-	0%
6. Memòria i Presentació	Escriure la memòria i la presentació oral	-	-	-	0%
GRAU DE MADURESA:					25%

Es pot visualitzar que pel primer control s'ha arribat fins a les primeres fases de la implementació, que era realment el què es volia assolir.

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. La següent taula es mostrarà les subtasques del *Disseny de l'Aplicació i d'Usuari*:

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 classes	Diagrama mostrant les funcionalitats de l'aplicació i com es relacionen	Guillem	Guillem (50%), Martí, Joel, David (50% restant)	1	100%
3. Mock-Up	Disseny d'interfície d'usuari	Joel	Joel (80%) Martí, Guillem, David (20% restant)	1	100%

L'última taula de a continuació es veuen les tasques que s'ha de realitzar durant la *Implementació*:

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	80%
2. Crear Sol	Crear l'objecte Sol, amb les textures i il·luminació adequada	Martí	Martí (70%), Joel, David, Guillem (30% restant)	2	80%
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	30%
5. Crear altres planetes	Posar els altres planetes i fer-los orbitar	-	-	-	0%
6. Afegir constel·lacions	Crear les constel·lacions i afegir-les a les coordenades corresponents de l'espai	David	David (90%), Joel, Guillem, Martí (10% restant)	2-3	40%

Subtasca	Descripció	Responsable	Participants (%)	Durada (setmanes)	Grau Finalització
7. Visualització Terra	Afegir un mode de visualització des de la Terra veient el cel estrellat.	-	-	-	0%
8. Ombres/llums	Afegir i controlar ombres i llums als planetes i constel·lacions segons a on es troben	-	-	-	0%
9. Funcionalitats usuari	Afegir interaccions d'usuari com botons, zoom-in, zoom-out, entre altres.	Joel	Joel (80%), Guillem, David, Martí (20% restant)	4-5	20%
10. Geoguessr	Crear un mini-joc senzill de trobar el planeta/constel·lació triat aleatòriament	-	-	-	0%

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. A continuació es pot veure el *Diagrama de Classes* construït:

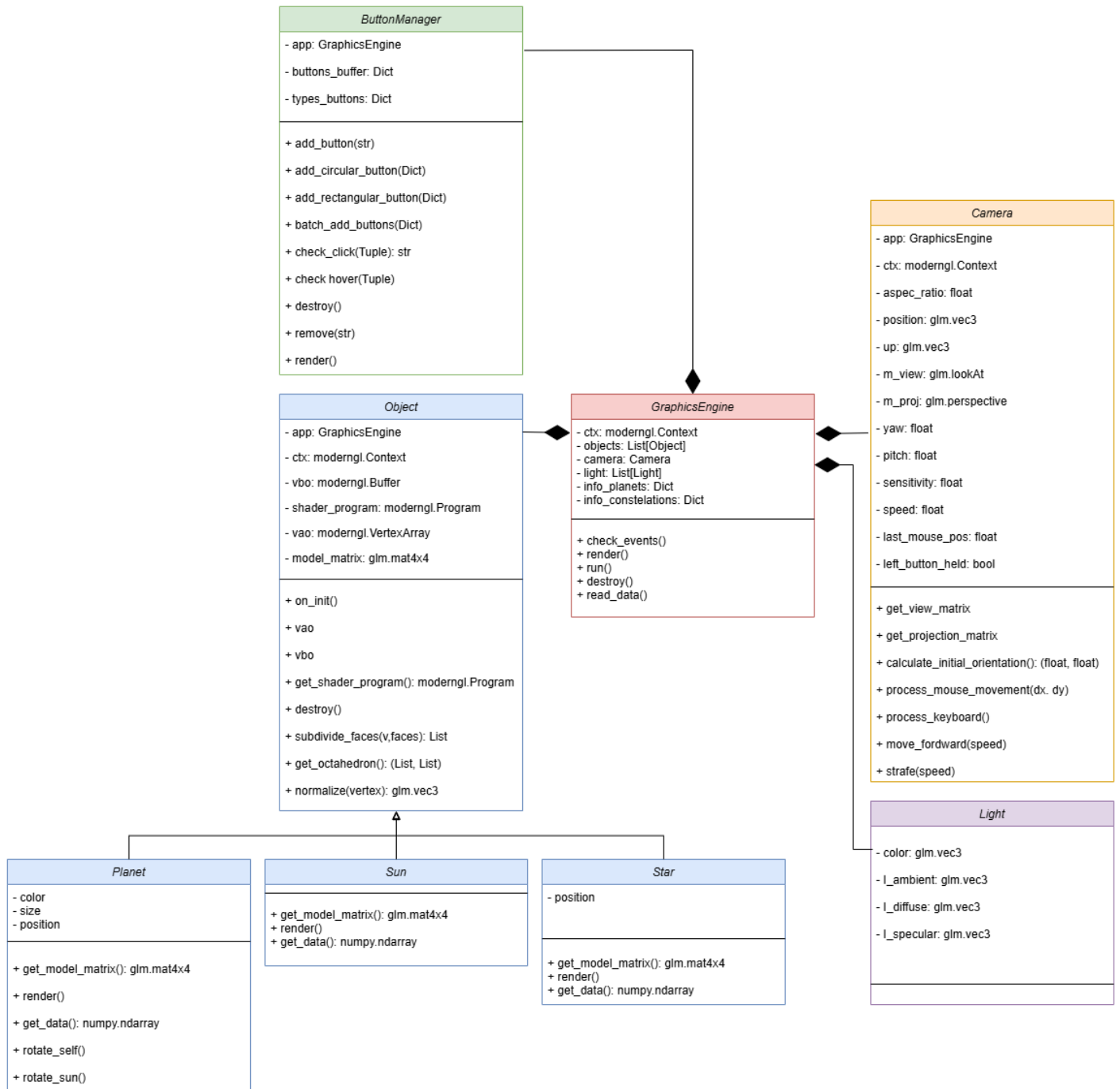


Figura 1: Diagrama de classes

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 el Control 1.

El Diagrama consta de cinc mòduls:

- **Object:** Serà la classe que s'utilitzarà per crear tot objecte de l'aplicació. Aquesta classe té més classes heretades, que es treballa específicament amb un tipus d'objecte. De moment, es té implementat els planetes, el Sol i les estrelles. Com es veu en el diagrama, actualment encara no està funcional les textures, i pot faltar altres grups d'objectes, com poden ser els satèl·lits.

- **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.

Alguns mòduls que es poden afegir en el futur són per la lectura i *parsing* de dades o bé també pel mode d'endevinar els planetes que es tenia pensat per fer.

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 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 un octaedre, normalitzar un vèrtex, crear una esfera per *stripes* o una esfera per subdivisió.

L'esfera per *stripes*, 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:

$$x = r * \sin(\theta) * \cos(\varphi)$$

$$y = r * \sin(\theta) * \sin(\varphi)$$

$$z = r * \cos(\theta)$$

Fórmula 1: Equacions de conversió a coordenades cartesianes

«on r és el radi de l'esfera que volem crear».

L'esfera per subdivisió, es basa primerament en crear un octaedre i anar afegint noves cares. L'algorisme consisteix en que per cada cara es construeix 4 noves a partir de calcular el els punts intermedis entre les seves arestes.

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

- **Classe Planet:** Cada planeta generarà una esfera per *stripes* o per subdivisió (depenent del que es passi com a paràmetre d'entrada), 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 una sol concepte 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 Star:** Les estrelles són en part més simples que els planetes i el Sol, perquè com a dades només necessitem saber les posicions d'aquestes i dibuixar els punts d'un mateix color.

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 polsar-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 és per 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

Com a resultats inicials, hem generat el planeta Terra i el Sol. El planeta es troba en moviment el·líptic, i també podem veure com el Sol va il·luminant el planeta. A demès, amb les *keybinds W, A, S i D* es pot moure la càmera i amb el ratolí apuntar la seva direcció.

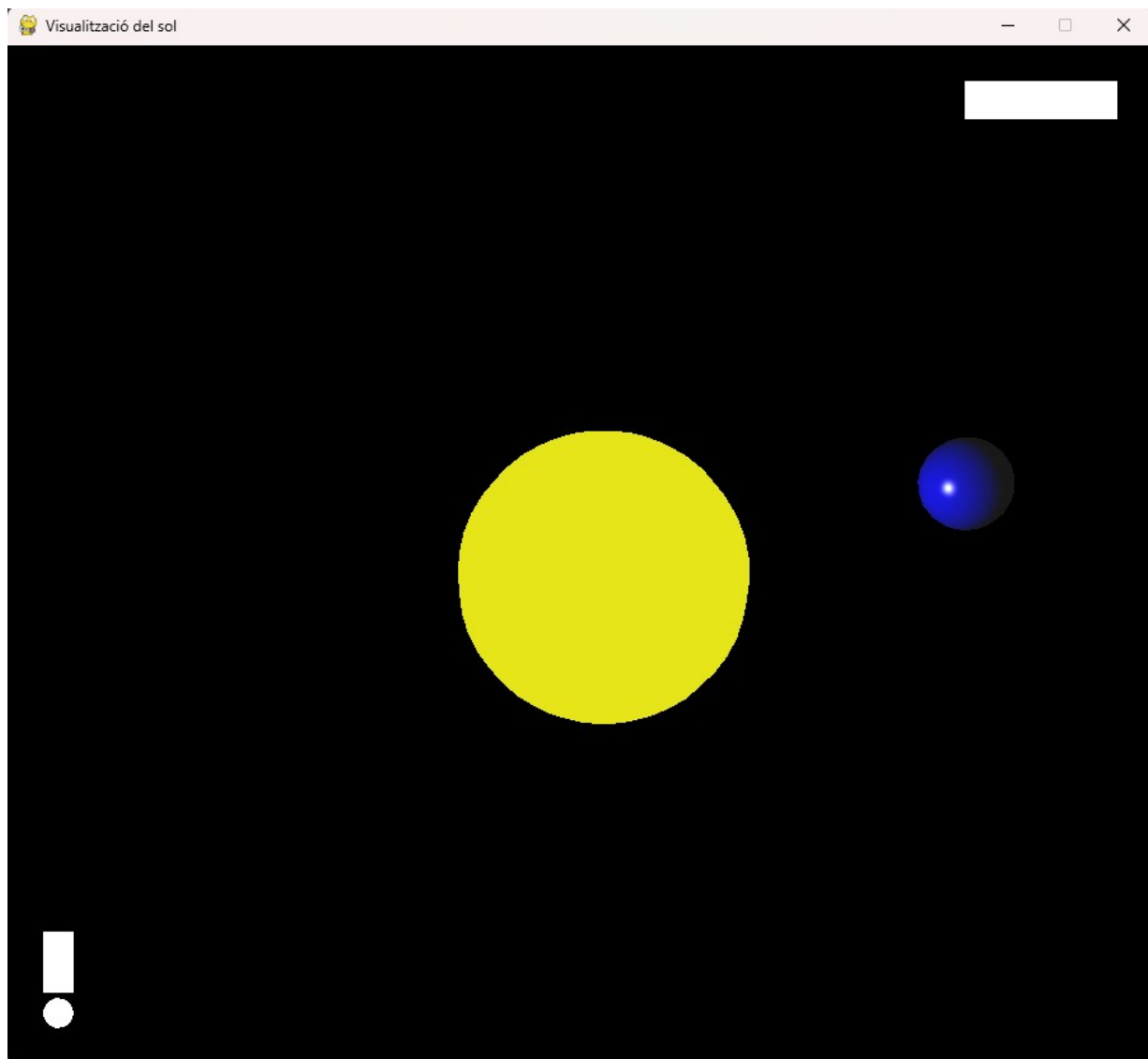


Figura 2: Resultats inicials

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