

VISUALITZACIÓ GRÀFICA INTERACTIVA

ENGINYERIA INFORMÀTICA

APRENENTATGE BASAT EN PROJECTES

PORTFOLI DEL GRUP 5

SIMULACIÓ D’ENLAIRAMENT D’UN COET

D. Herbón, F. Navarro, J. Luzón, N. Domene, J. Gonzalez, R. Díaz

Bellaterra, Gener 2018

**ÍNDEX**

[**1. INTRODUCCIÓ I OBJECTIUS**](#_dxd47jcwsueu) **4**

[**2. DESENVOLUPAMENT**](#_lnxbz9) **5**

[2.1. Trajectòria (físiques)](#_1ksv4uv) 6

[2.2. Textures](#_u80lfw1b0imh) 6

[2.3. Sistema de partícules](#_2jxsxqh) 6

[2.4. Opcions del menú](#_z337ya) 6

[2.5. Il·luminació](#_3j2qqm3) 7

[2.6. Fractals](#_1y810tw) 7

[2.7. Inserció d’objectes externs](#_4i7ojhp) 7

[2.8. Algoritme dels núvols](#_2xcytpi) 7

[2.9. Càmeres](#_1ci93xb) 7

[2.10. Coets i planetes extres](#_3whwml4) 7

[2.11. Telemetria](#_zfvjhyrob089) 7

[2.12. Cel i atmosfera](#_pqye4axsoeg) 8

[**3. PROBLEMES TROBATS I SOLUCIONS**](#_2bn6wsx) **8**

[3.1. Ús del Visual Studio](#_qsh70q) 8

[3.2. Organització del codi](#_qsh70q) 8

[3.3. Organització de les tasques](#_qsh70q) 8

[3.4. Organitzatció de l’equip](#_qsh70q) 9

[3.5. Creació de nous objectes complexos en 3D](#_qsh70q) 10

[3.6. Funció de rellotge (clock)](#_qsh70q) 11

[3.7. Implementació de textures](#_dbf0k756u5um) 11

[3.8. Trajectòria (físiques)](#_3as4poj) 12

[3.9. Sistema de partícules](#_76kdvq45g1b4) 12

[3.10. Il·luminació](#_1pxezwc) 12

[3.11. Càmeres](#_1pxezwc) 13

[3.12. Telemetría](#_qsh70q) 13

[3.13. Col·lisions](#_qsh70q) 13

[3.14. Atmosfera i cel](#_w71nnj6bi4y5) 13

[3.15. Múltiples textures](#_6nzlqhuxpqf7) 13

[**4. RESULTATS**](#_49x2ik5) **14**

[4.1. Trajectòria (físiques)](#_2p2csry) 14

[4.2. Il·luminació i ombres](#_j9g7g294ewfb) 15

[4.3. Textures](#_yaennuwtkkef) 16

[4.4. Coets i plataforma](#_3w45ajqmjvch) 18

[4.5. Planetes](#_zeowzam7et35) 19

[4.6. Ciutats](#_vhqqh2eitv8c) 19

[4.7. Núvols](#_3o7alnk) 19

[4.8. Sistema de partícules](#_23ckvvd) 20

[4.9. Capes de l’atmosfera](#_w8gojqz3q5xd) 20

[4.10. Càmeres](#_ihv636) 20

[**5. CONCLUSIONS**](#_1hmsyys) **22**

[**6. REFERÈNCIES ELECTRÒNIQUES I BIBLIOGRÀFIQUES**](#_41mghml) **23**

[6.1. Llibres](#_3f78g0lvwwe8) 24

[6.2. Pàgines web](#_utrc008qh736) 25

[6.3 Programari software](#_4i0pufjq8g0q) 25

[**ANNEXE I. ACTES DE LES REUNIONS**](#_2grqrue) **26**

[**ANNEXE II. CONTROLS**](#_6vexl4pizznc) **27**

[**ANNEXE III. CALENDARI D’ORGANITZACIÓ DELS SPRINTS**](#_2grqrue) **28**

[**ANNEXE IV. FORMULARI DE LES FÍSIQUES**](#_2grqrue) **29**

# 

# 1. INTRODUCCIÓ I OBJECTIUS

L’objectiu d’aquest projecte es implementar una aplicació gràfica en Visual C++ i OpenGL. Aquest projecte es de decisió lliure, per tant, a la primera reunió d’equip es va realitzar un brainstorming, per tal de tenir diverses idees de projectes a realitzar. Van sortir varies idees:

* Simulador d’avions
* Un pont llevadís
* Simulador d’aeroports
* Cursa de velers
* Simulador de coets

Després de discutir entre els diversos projectes, vam decidir de realitzar el simulador de coets. L’objectiu del projecte tracta de llançar un coet des de la Terra i posar-lo en òrbita. Per tant, es necessari poder representar un planeta i un coet realista, amb unes físiques que calculin la trajectòria mitjançant avança el coet.

Acte seguit, vam decidir que era important aprendre primer l’entorn OpenGL que se’ns donava i jugar amb ell per entendre’l. Un cop fet això vam determinar les funcions que volíem aplicar:

* Aplicar físiques realistes al funcionament del coet. Es farà ús de la gravetat i les formules necessàries.
* Es podrà escollir entre diferents tipus de coets.
* Es podrà escollir la quantitat de combustible. Aquest, també pot ser de diferents tipus i tindre diferents eficiències.
* S’introduiran animacions de foc, fum i núvols per a donar més realisme.
* La mida de la Terra també ha de ser modificable i s’ha de poder canviar entre diversos planetes.
* També es podrà escollir entre diverses càmeres.

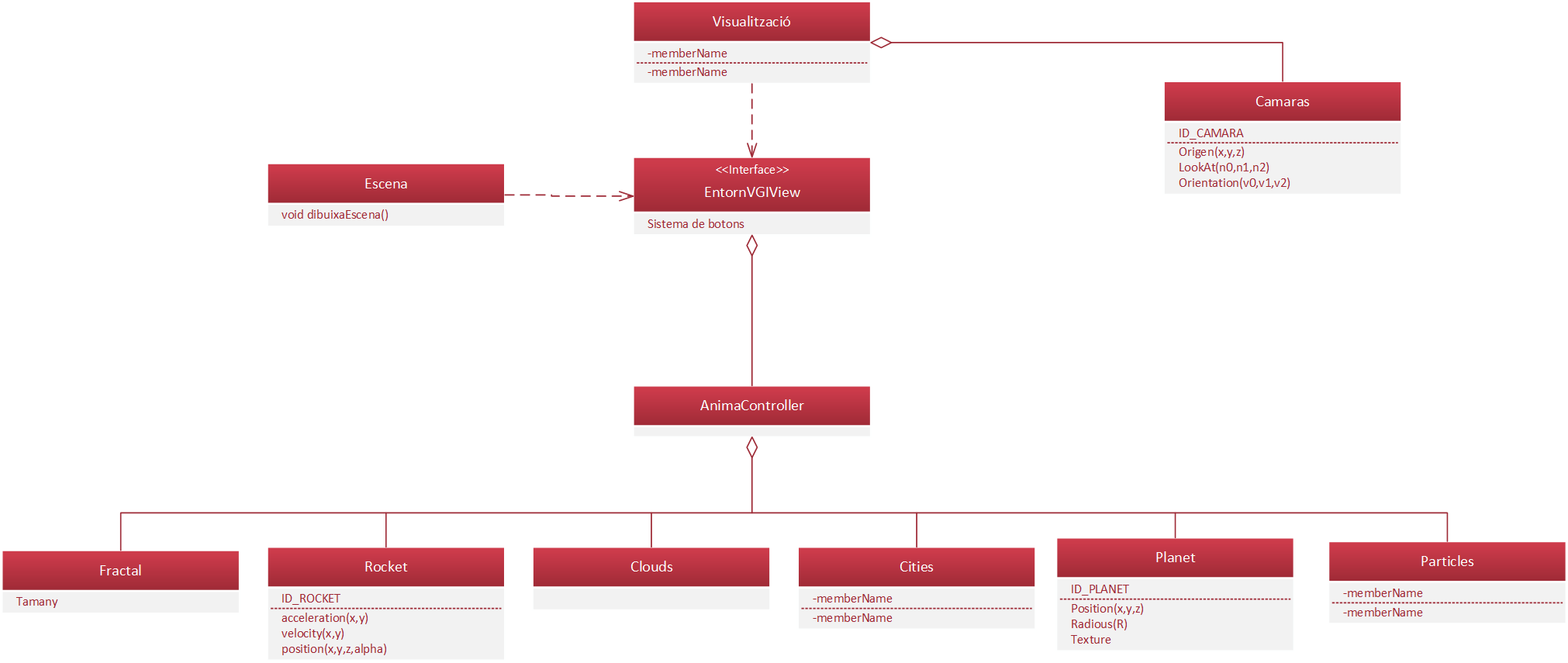
Aquest informe es troba dividit en capítols. Dintre de cada capítol es troben apartats i subapartats si escau. El capítol 2 tracta el desenvolupament de l’aplicació. S’explica mitjançant un diagrama la connexió entre els diferents mòduls de l’aplicació i el treball que realitza cada un. Al capítol 3 s’expliquen els diversos problemes que em tingut al llarg del desenvolupament i les solucions que hem trobat. Al capítol 4 es mostres resultats, de manera gràfica i acompanyats de les explicacions sobre cada resultat. En el capítol 5 posem en comú les conclusions a les que hem arribat realitzant el projecte, i parlem d’allò que hem fet i no em fet.

Com a afegit es troben els annexes, que hem inclòs quatre: els actes setmanals, els dos controls realitzats, el calendari d’organització dels sprints del projecte i el formulari per a les físiques.

# 2. DESENVOLUPAMENT

El diagrama de mòduls de la nostra aplicació és mostra en la figura 2.1 del qual a continuació explicarem els mòduls més importants de la nostra aplicació.

1. **EntornVGIview:** Part del codi base que se’ns va donar com a opció de partida i que serà el punt de connexió de tots els demés mòduls. Serà aquest mòdul el que contingui la instància d’AnimaController, que serà compartida amb Escena i contindrà la resta de mòduls que hem implementat.
2. **Escena:** Mòdul encarregat de determinar què es visualitza a cada instant. Rebrà a cada moment una instancia d’AnimaController, tal com hem comentat.
3. **AnimaController:** Es el mòdul principal que hem desenvolupat i s’encarrega de controlar les animacions i els principals ‘actors’ d’aquestes (coet, planeta, núvols, foc…). Determinarà alguns factors de les càmeres (si s’activa o no la vista múltiple, si la càmera principal ha de seguir al coet o no, el coet que està actiu en un moment determinat, si la càmera utilitza les coordenades esfèriques o no, etc…). A més a més, serà la manera de permetre que Escena i EntornVGIview comparteixin les instàncies del coet, planeta, fum i la resta.
4. **Rocket:** Conté tot allò necessari per a la execució de les trajectòries del coet. La posició actual del coet, la seva velocitat i acceleración (les components cartesianes d’aquest), l’angle de rotació… Aquesta clase no controlarà la visualització del coet (tipus de coet que es mostra), ja que d’això s’encarrega AnimaController. A més, la gravetat per la que es veu afectat el coet estarà determinat en aquesta classe i, si es vol canviar en un algun moment donat, s’ha de canviar en el módul Rocket i no en Planeta, ja que es Rocket el que controla tota la física d’animacióque afecta al coet.
5. **Clouds:** Controla la generació dels nùvols que es veuen en l’animació. Els diferents paràmetres que es poden configurar per a la einstància d’aquesta classe que hi haurà a AnimaController determinarà la quantitat de núvols que es generen, la grandària dels components de cada núvol, l’ample, la llargada, l’altura mitja i la distància entre elles, etc… La visualització dels núvols es realitza a la funció de Escena que rep com a paràmetre una instància d’aquesta classe i es aquí on es determina el color i la transparència dels núvols i la posició d’aquestes (a més del seu escalat o rotació en cas de ser necessari).
6. **Fractal:** Serà el mòdul encarregat de la generació del terreny muntanyenc on apareix el coet inicialment. En aquest mòdul es controlen els colors del fractal i alguns detalls com l’altura màxima i mínima del terreny (que realment es llegeixen d’un fitxer).
7. **Particles:** És la classe encarregada dels generadors de partícules que utilitzem per a generar el fum i el foc. Cada instància té les seves pròpies probabilitats de donar una velocitat o una altre a les noves partícules, així com la duració d’aquestes (vida). El color i la transparència es controlarà a Escena. Aquesta classe s’encarrega de controlar en tot moment les posicions de cada partícula i actualitzar-les segons les velocitats que tinguin. A cada actualització de la posició s’actualitza també l’edat de cada partícula, i quan una partícula arriba a la seva edat màxima es reinicia i la posició de memòria que ocupa en el vector de partícules és reutilitzada per a la nova, per tant en tot moment hi haurà el mateix número de partícules.



*Figura 2.1 Diagrama de mòduls de l’aplicació.(PROVISIONAL)*

## 2.1. Trajectòria (físiques)

@Dani

## 2.2. Textures

L’aplicació de textures als elements de la nostra simulació va suposar un repte en primer lloc degut a que no teníem que texturitzar el planeta i el coet. En el cas del planeta, la dificultat es trobava que es tractava d’una esfera i la solució va consistir en construir l’esfera manualment (en lloc d’utilitzar una funció d’OpenGL) definint els vèrtexs i punts de textures de forma manual. En el cas del fitxer .obj, la solució més senzilla que vam trobar va ser buscar objectes ja texturitzats o, en cas contrari, texturitzar-los nosaltres mateixos (en el cas dels coets dissenyats per nosaltres) i carregar-los amb la textura definida en un fitxer de textures.

## 2.3. Sistema de partícules

El funcionament del nostre sistema de partícules (per al fum i el foc, en el cas dels núvols es diferent) es com el d’una font de partícules. Donat un punt inicial en el que es generen un número determinat de partícules, generem una font, donant aleatòriament a cada una d’aquestes partícules una velocitat i una duració determinades. A cada moment es comprova la posició de totes les partícules i aquestes es dibuixen amb un color, transparència i mida diferent cada vegada (aleatori) per a donar més sensació de que es tracta realment de foc i fum. La velocitat d’una partícula es la mateixa des de que neix fins que es mor (quan la seva edat arriba al límit que se li va donat en el moment de creació). En aquest moment, la partícula torna a la posició inicial i rep uns nous paràmetres (com si es creés una nova partícula en el seu lloc, però reutilitzant la mateixa memòria).

## 2.4. Opcions del menú

Partint del menú que teníem en el codi base, hem hagut d’afegir algunes entrades per adaptar-ho a les necessitats de la simulació. Hem afegit dues entrades, una per a seleccionar el coet que s’utilitzarà al llançament i un altre per a determinar el planeta en el que es realitza. Respecte a la càmera, al menú hi donem l’opció d’utilitzar el mode multi-vista (amb vàries càmeres a la vegada) o utilitzar la càmera principal únicament. A més, donem l’opció de centrar la càmera en el planeta o el coet (quan es centra en el planeta es pot veure aquest desde la distància). La següent part a tindre en compte es el menú trajetòria, on es pot fer començar la trajectòria del coet amb l’opció “Launch”, parar-la amb “Stop” o reiniciar-la amb “Restart”.

## 2.5. Il·luminació

Bla bla bla malditos comunistas @PacoP

## 2.6. Fractals

Els fractals s’han insertat reutilitzant el codi de la Pràctica 3 de l’assignatura. Per tal de fer-ho més realista, s’ha modificat la paleta de colors i s’ha adabtat el tamany del fractal, la rotació i la traslació per tal de fer-lo coincidir a sobre de catalunya que és d’on farem enlairar el coet.

## 2.7. Inserció d’objectes externs

La inserció d’objectes externs s’ha realitzat reutilitzant el codi del propi entorn gràfic aportat per el professor. Hem utilitzat només objectes d’extensió .obj i hem combinat objectes trobats a internet amb altres dissenyats per nosaltres mateixos.

## 2.8. Algoritme dels núvols

@Nil

## 2.9. Càmeres

Distingim dos tipus de càmeres, les que utilitzen coordenades esfèriques y les que utilitzen la proporció cartesiana de l’observador i del punt al que mira. Hem desenvolupat una “vista múltiple” per al llançament, en el que es pot veure el coet des de diferents punts i utilitzant ambdós tipus de càmeres. Els efectes que hem pretès aportar són:

1. Càmera amb observador fixe que mira sempre al coet.
2. Càmera amb observador mòbil, que es mou igual que el coet mentre mira al coet.
3. Càmera amb coordenades esfèriques que gira al voltant del coet a llibertat de l’usuari o de forma automàtica si aquest no ho fa.
4. Càmera que es mou amb el coet i mira la terra com si estigués en el propi coet.

Aquestes càmeras s’han realitzat utilitzant les funcions d’OpenGl, que a partir de les coordenades(x,y,z) del coet ens permet fer tots els tipus de càmeres mencionades.

## 2.10. Coets i planetes extres

Hem desenvolupat diversos coets diferents (amb diferents .obj) per a la simulació, amb distintes formes i textures. De la mateixa manera donem l’opció a escollir entre diversos planetes que tenen mides i textures diferents (i que fan que el coet tingui una gravetat diferent).

## 2.11. Telemetria

Bla bla bla @Nil

## 2.12. Cel i atmosfera

Hem implementat un cel que consta d’un color de fons (que canvia segons l’altura a la que es trobi la càmera) i varies textures aplicades sobre esferes que recobreixen el planeta. Hi ha una esfera final que té una textura per a simular la galàxia (com estrelles) i varies esferes internes (es pot modificar el número d’esferes fàcilment per que es fan amb un bucle) que tenen textura de núvols i molt poca transparència per a donar efecte de ser capes atmosfèriques.

# 3. PROBLEMES TROBATS I SOLUCIONS

## 3.1. Ús del Visual Studio

El programa en el qual hem realitzat el projecte es el Visual Studio de Microsoft, ja que l’entorn gràfic que se’ns ha donat està realitzat amb aquest. L’entorn està preparat per funcionar amb la versió 2013. Com a estudiants, tenim accés al portal de software de Windows, però el Visual Studio 2013 ja no es trobava per descarregar.

Per solucionar això, ens vam posar d’acord en utilitzar tots la mateixa versió de Visual Studio i vam descarregar la versió Community 2017 ja que a més es gratuït. El projecte el varem actualitzar per a poder fer ús de la nova versió i des de llavors hem treballat amb aquesta.

## 3.2. Organització del codi

Des de la primera reunió vam haver de decidir una manera de compartir el codi sense trobar-nos amb problemes de sobre-escriptura ja que som molts integrants i podríem arribar a la situació de treballar a la vegada en el codi.

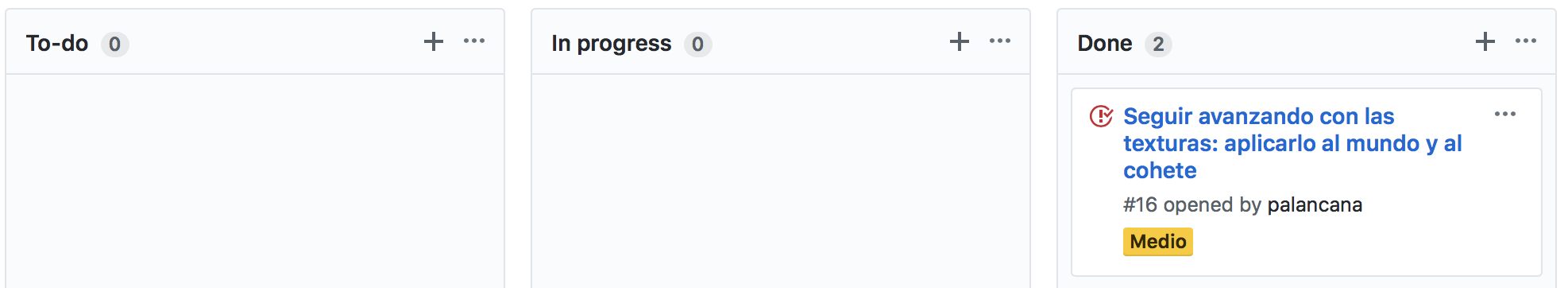
La solució és utilitzar un software de control de codi. En el nostre cas hem utilitzat GitHub ja que en Nil té un compte que permet la creació d’un repositori privat. De fet, fins i tot vam definir un patró per a fer els comentaris dels commits al GitHub, de tal manera que podíem saber de forma fàcil si era una modificació, un afegit o una eliminació.

## 3.3. Organització de les tasques

Encara que ja a les primeres setmanes es va crear un calendari i definir uns sprints per tal de saber que tocava fer en cada moment i tindre una visió general del progrés (i saber si complíem els terminis o no), ja que desconeixíem l’entorn i l’OpenGL, no podíem crear tasques molt específiques. Tampoc teníem una eina amb la que poder posar-nos d’acord i no trepitjar-nos el treball l’un a l’altre.

*Figura 3.1. Visualització en GitHub d’una tasca concreta i els seus detalls.*

Per a solucionar això hem fet ús del mateix GitHub per a reduir l’ús d’eines a les bàsiques. GitHub té una opció per a crear “issues”. Aquestes es poden assignar als integrants de l’equip (tal com mostra la figura 3.1). El funcionament es similar a unes targes i vam crear diversos taulers per a saber si la tasca estava pendent, en curs o finalitzada, tal com es veu a la figura 3.2. A més, aquestes les vam separar en sprints i dificultat.

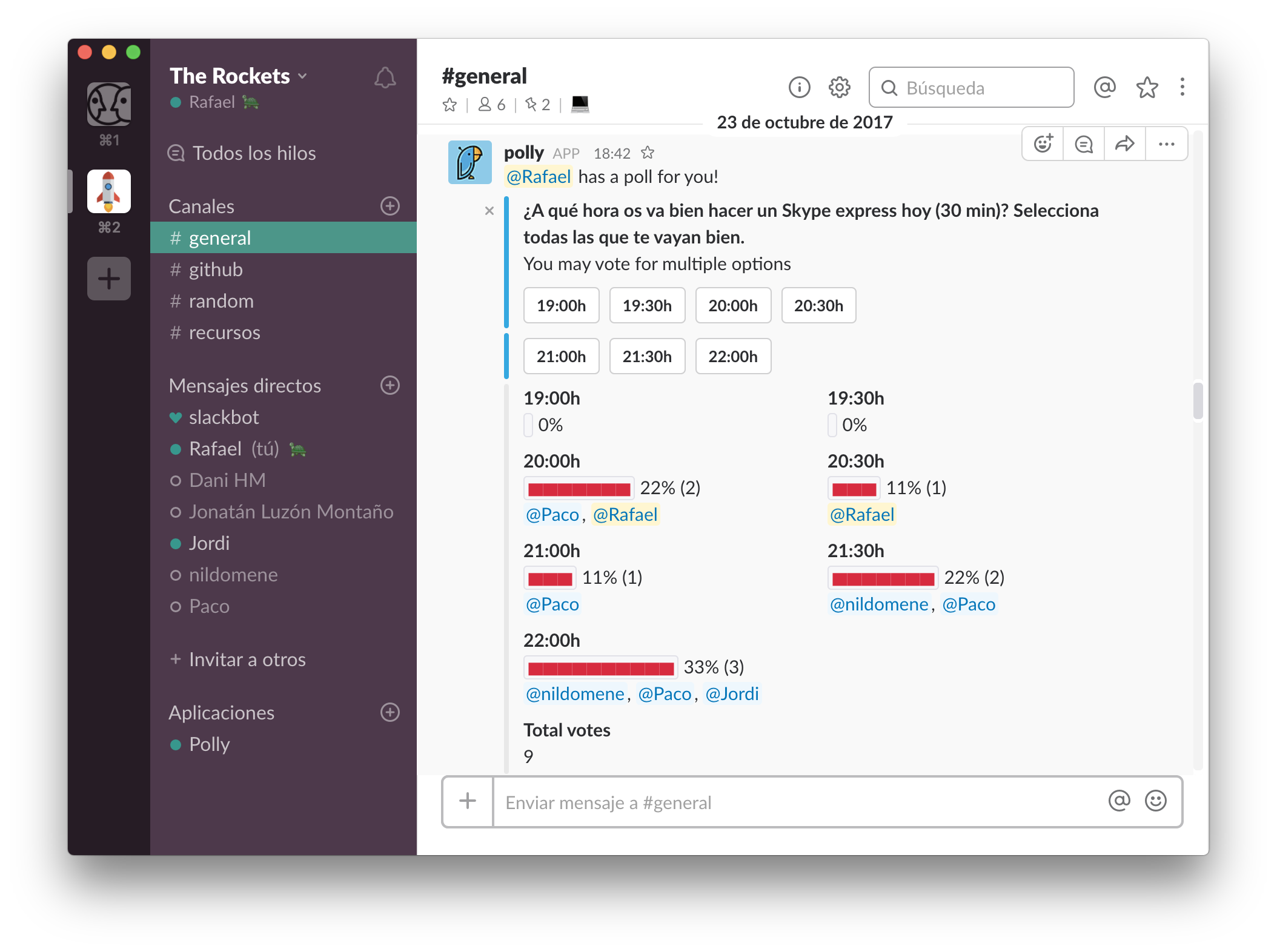


*Figura 3.2. Diferents estats d’una tasca concreta a l’sprint 3.*

## 3.4. Organitzatció de l’equip

Ésser un equip de sis persones pot ser difícil a l’hora de prendre decisions o organitzar-se per reunions fora de les programades en hora de classe.

Per a aquest motiu hem creat un equip a Slack, una aplicació de tipus xat en la que es poden organitzar les converses per fils (en el nostre cas vam crear un general, un de github on un bot feia avisos de les actualitzacions automàtiques en el repositori de GitHub, un altre de recursos per tindre un recopilatori de vincles interessants i un altre anomenat “random” per rauxa). Desde l’Slack també vam ser capaços de fer alguna enquesta per posar-nos d’acord i fer reunions. Aquest entorn es pot visualitzar a la figura 3.3.

*Figura 3.3. Visualització de l’entorn de l’Slack.*

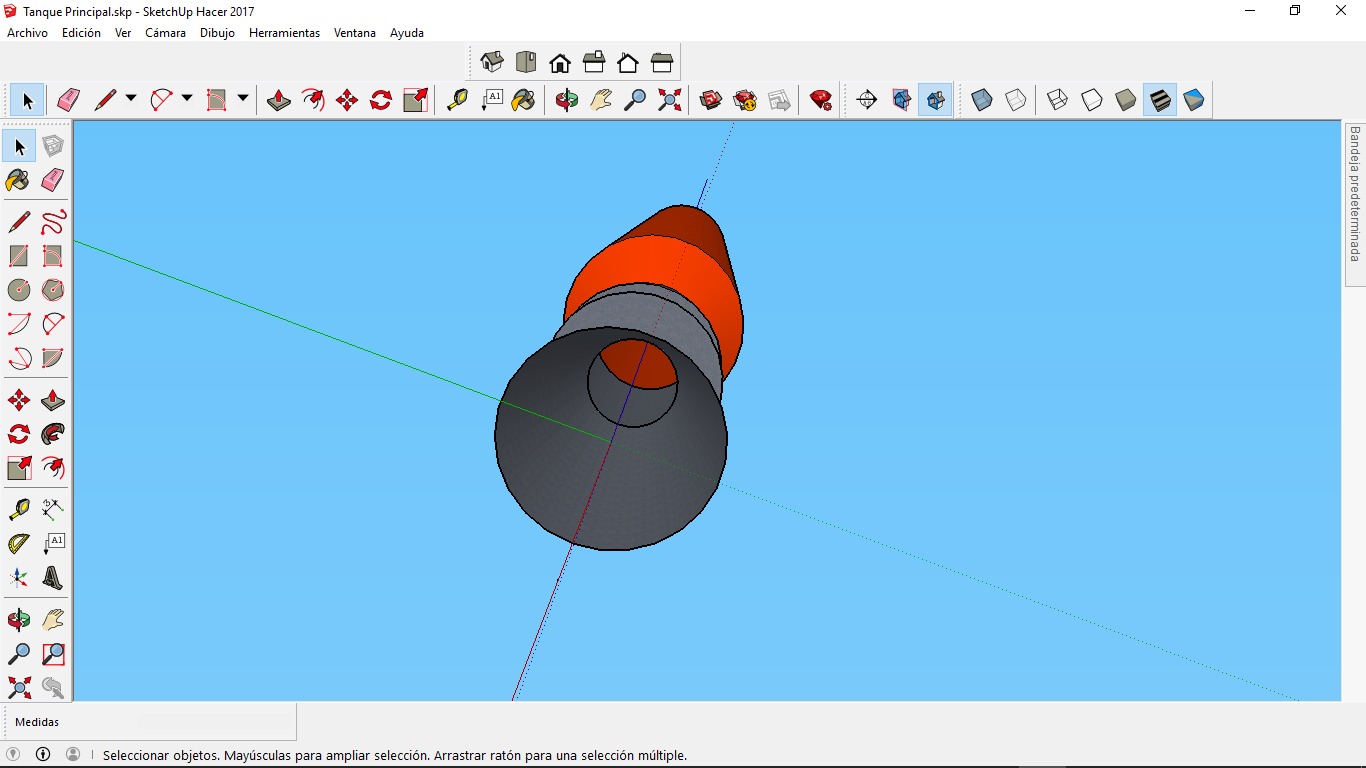
## 3.5. Creació de nous objectes complexos en 3D

Un cop vam veure la manera d’introduïr objectes externs a l’entorn gràfic, vam començar a buscar objectes en 3D en format .obj o .3ds que fossin els coets que volíem. Tret d’algun component, no vam trobar el que buscàvem.

La solució, doncs, va ser crear-los nosaltres. Això va implicar una búsqueda intensa dels programes d’edició en 3D existents. Això ens va deixar amb aquestos candidats:

* **Paint 3D:** Propietari de Windows. No existeix l’opció d’exportar al format desitjat.
* **SketchUp:** Opció de Google. Permet exportar al format desitjat i a més te una interfície amigable.
* **Blender:** Opció OpenSource. Permet exportar en el format desitjat però és difícil d’utilitzar i requireix molt temps d’aprenentatge.
* **TinkerCad:** Opció online. Les opcions poden ser massa senzilles.

Amb aquesta recerca vam decidir quedar-nos amb l’opció de l’SketchUp, que es el que vam utilitzar per crear els coets. A la figura 3.4 es pot veure l’entorn d’aquest programa.



*Figura 3.4. Visualització de l’entorn de SketchUp.*

## 3.6. Funció de rellotge (clock)

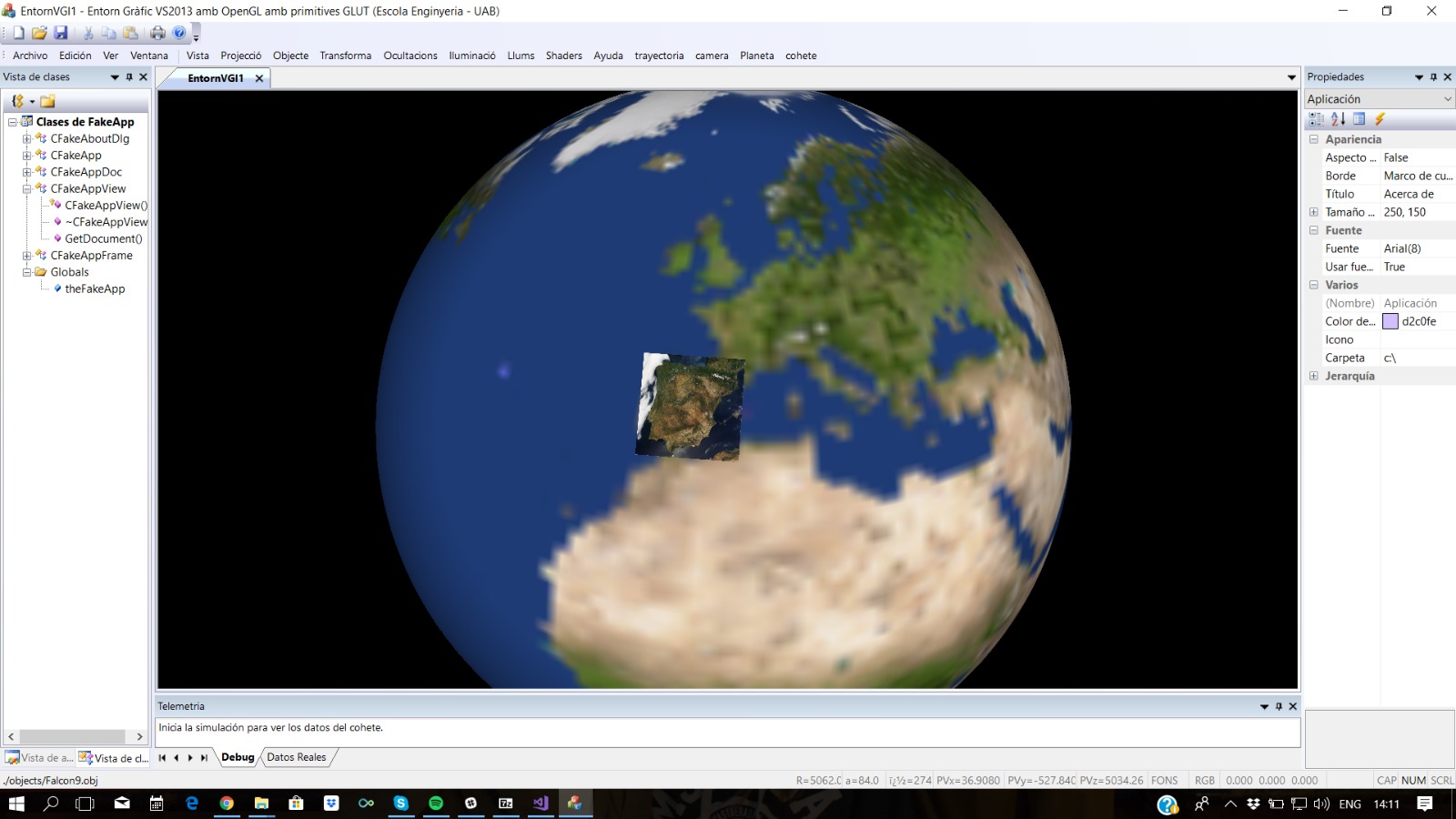
Vam tenir problemes per animar el coet a causa del fet que interpretàvem que el temps que passàvem al timer d’EntornVGI era en segons i en realitat era en mil·lisegons. Nosaltres li estàvem donant valors de l’ordre de 0,0001 mil·lisegons i l’animació no funcionava com esperàvem. Un cop entès el problema no hi ha va haver més complicacions.

## 3.7. Implementació de textures

Vam tenir molts problemes aplicant la textura de la terra a l’esfera que representa el planeta. Inicialment intentàvem generar les coordenades textura de forma automàtica, però no aconseguíem fer-ho adequadament. Finalment, vam decidir generar l’esfera de forma manual (en lloc d’utilitzar la funció GLUsphere). Amb això aconseguíem aplicar la textura però la qualitat era molt deficient, degut al número de vèrtexs que generàvem (vam posar un límit per evitar problemes).

Vam intentar augmentar el número de “talls” de l’esfera (afegir més vèrtexs) però el rendiment de la simulació empitjorava. Per això vam decidir a col·locar un pla a sobre de l’esfera amb una textura d’Espanya, amb l’idea de que aquella part es veiés millor, però el resultat (com es mostra a la figura 3.5) no ens va semblar bo, ja que no aconseguíem ajustar bé una textura amb l’altre per a que semblés la mateixa i vam decidir treure’l.

Finalment vam utilitzar una textura amb molta resolució i al aplicar alguns efectes (atmosfera i núvols i el fractal que es veu al fer l’enlairament del coet) aconseguíem dissimular la baixa qualitat de la imatge quan es feia molt zoom.



*Figura 3.5. Textura d’Espanya a sobre d’un altre. Opció que vam descartar.*

## 3.8. Trajectòria (físiques)

Texte d’explicació de los problemas de trajectoria bla bla @Dani

## 3.9. Sistema de partícules

El primer problema que vam trobar amb el sistema de partícules va ser una mala interpretació del sistema que havíem escollit per connectar EntornView i Escena, ja que Escena rebia una copia de la instància d’AnimaController que posseeix EntornView i nosaltres estàvem actualitzant aquesta còpia dintre del codi de les partícules però en la següent iteració les partícules seguien igual que al principi. La solució va consistir en passar la instància de AnimaController per referència.

El següent problema va ser aconseguir donar una trajectòria que fos mitjanament realista per a simular foc i fum. Vam intentar fer que la velocitat anés canviant amb el temps, però les trajectòries mai terminaven de ser les desitjades. Vam decidir, per solucionar-ho, de modificar aquesta part del codi per tal que les velocitats fossin fixes i evitar comportaments estranys.

## 3.10. Il·luminació

En un principi no vam terminar d’aconseguir adaptar les llums a les nostres necessitats ja que la llum que utilitzàvem inicialment (la del codi base) tenia tant la component de llum ambient com la component difusa. La solució que hem trobat ha sigut fer que dues fonts de llum amb colors e intensitats diferents i fer que una tingues la llum ambient i l’altra difusa. Després de fer diverses proves, vam aconseguir un efecte interessant.

El següent problema que vam trobar va ser que els sistemes de partícules apareixien negres o grisos (encara que li donàvem color blanc i vermell per a simular el foc i el fum) degut a la llum. Creem que això es deu a les normals dels vèrtexs generats en el sistema de partícules però no tenim molt clar com solucionar-ho.

## 3.11. Càmeres

Les càmeres ens va portar temps d’implementar per que no teníem clar el funcionament. Un cop entès, però, ens va resultar relativament senzill jugar amb elles per a generar els efectes desitjats. Vam fer que pugessin centrar les càmeres en el coet o el planeta a la nostra voluntat i que es pogués escollir si aquestes segueixen o no al coet.

## 3.12. Telemetría

Per tal de poder depurar les variables del coet a mesura que s’està executant la trajectoria s’ha creat una finestra on es van mostrant tots els valors útils per poder entendre que està passant en la simulació. En aquesta finestra es mostren les coordenades, la velicitat que porta en aquest moment, l’acceleració que porta, la quantitat de combustible que té i la distància del planeta que s’ha enlairat

## 3.13. Col·lisions

Más bla bla bla

## 3.14. Atmosfera i cel

En un principi només vam implementar el cel com una textura aplicada a una esfera molt gran que canviava a certa altura, però la transició entre la textura del cel (que es veia des de la superfície de la terra) i la textura de les estrelles (que es veien des de l’espai) era molt brusca.

La solució que vam trobar va ser aplicar transparències a ambdues textures i utilitzar un color de fons que anava variant amb l’altura del coet des d'un blau clar a negre perquè no fos un canvi brusc. A més, vam decidir de crear més esferes texturitzades intermèdies amb molta transparència i textures de núvols per a generar la sensació d’estar travessant una atmosfera.

## 3.15. Múltiples textures

Un cop teniem la textura de la Terra incertada, vam decidir crear diferents objectes amb funcions d’OpenGL i els propis .obj que els incertem amb les textures. El problema que ens va sorgir i que ens va costar molt solucionar-ho era que al tindre molts objectes amb les textures, les textures d’uns s’aplicaven a altres objectes.

Pero solucionar això vam veure que hi ha un buffer de textures que les textures que s’apliquen manualment has de saber a quina posició del buffer s’està guardant la referència ja que depenent de quants objectes .obj amb textures aquestes textures s’emmagatzemen just després d’aquestes.

# 4. RESULTATS

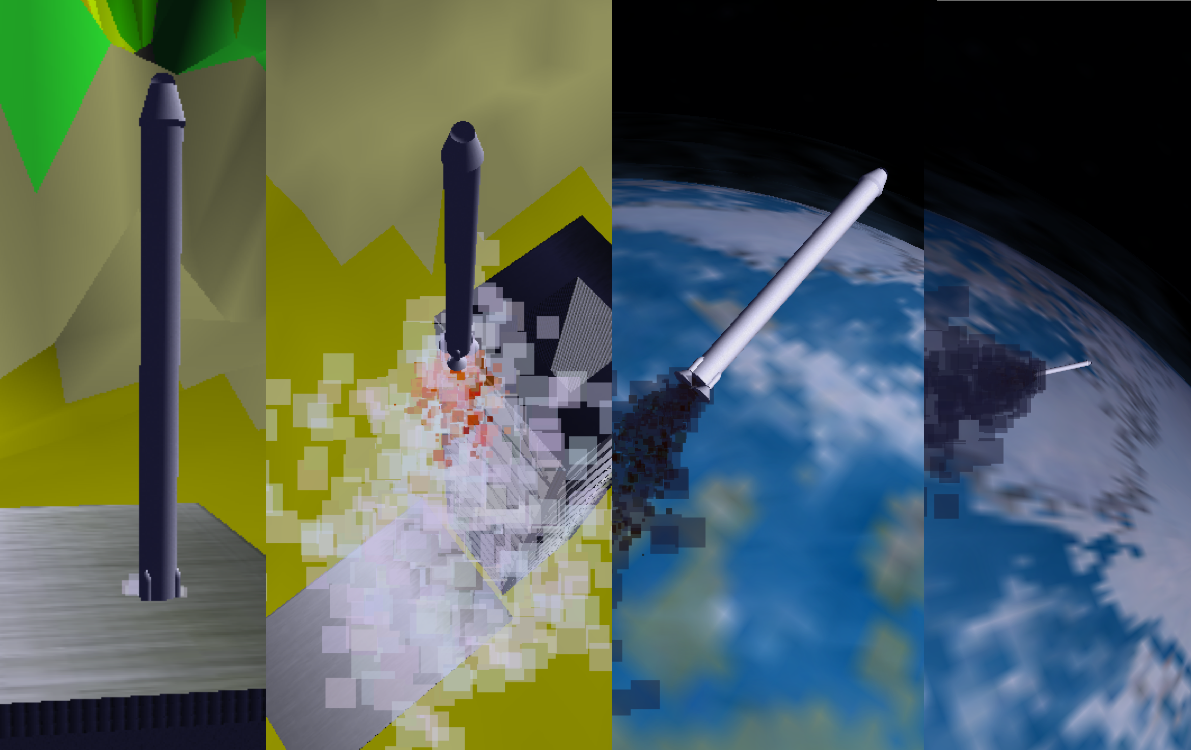
Els resultats del projecte són completament visuals, i per tant mostrem a continuació un recull d’imatges (figures) que mostren el resultat de les funcionalitats aplicades. A continuació enumerem les característiques que mostrarem en aquest capítol:

* Trajectòria (físiques)
* Il·luminació i ombres
* Textures
* Coets i plataforma
* Planetes
* Ciutats
* Núvols
* Sistema de partícules
* Capes de l’atmosfera
* Càmeres

## 4.1. Trajectòria (físiques)

En aquesta secció es pot observar com el coet canvia la inclinació cada cop que s’allunya més del planeta per tal d’aconseguir entrar en òrbita. A la figura 4.1 s’observen 4 etapes diferents. D’esquerra a dreta es pot veure com:

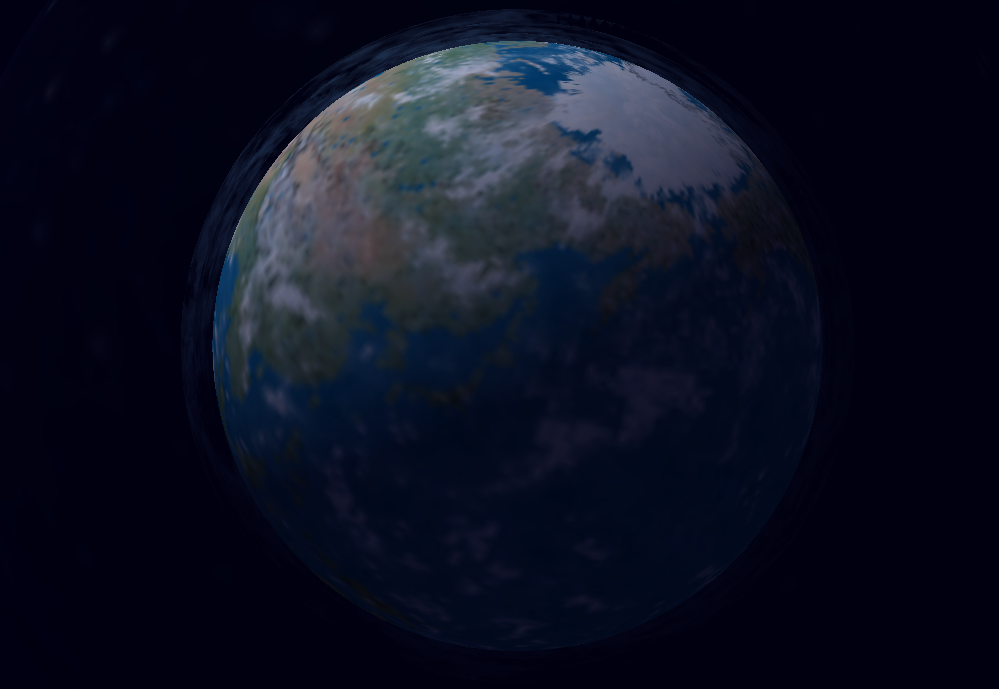
1. El coet es perpendicular al planeta i es troba a punt per ser llançat.
2. El coet comença a girar
3. El coet continua en el progrés d’entrar en òrbita
4. El coet ja ha entrat a òrbita



*Figura 4.1. Evolució de la trajectòria en un llançament.*

## 4.2. Il·luminació i ombres

En la figura 4.2 es pot observar el planeta Terra sencer. En aquesta figura es pot observar com incideix la il·luminació. A la figura 4.3 en canvi es pot veure un coet apropat. Si ens fixem no posseeix la mateixa il·luminació a tot l’objecte, el que provoca una sensació d’ombra, encara que l’ombra com a tal, en el sòl del planeta no s’ha implementat.



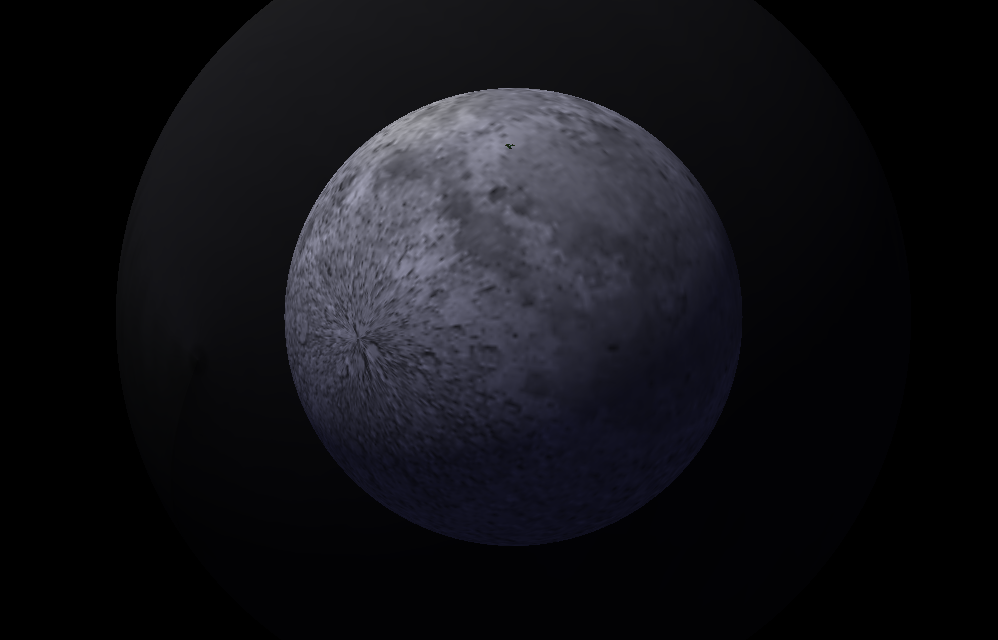
*Figura 4.2. Vista de tot el planeta Terra.*



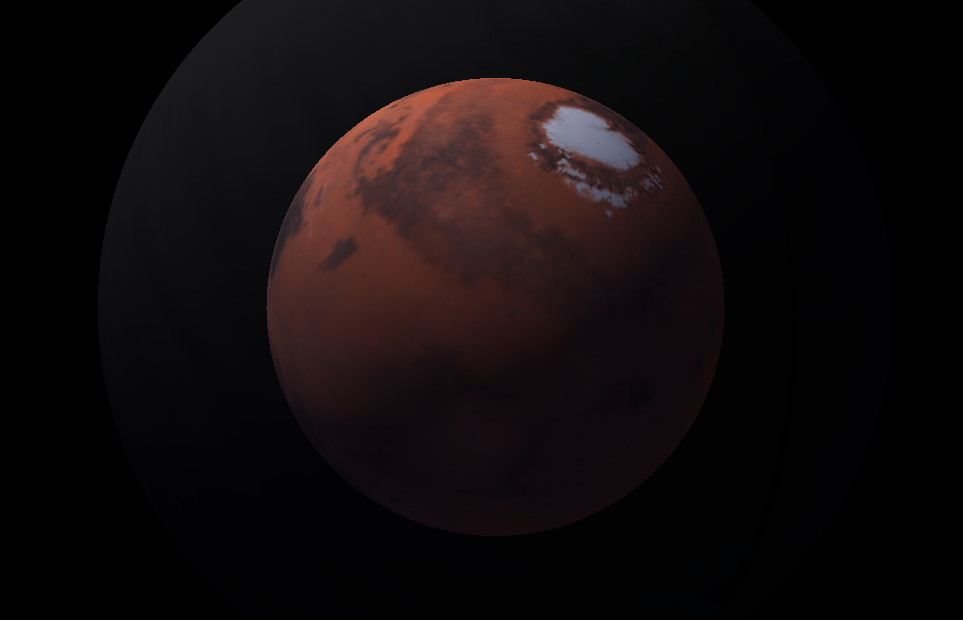
*Figura 4.3. Vista d’un objecte coet.*

## 4.3. Textures

A la figura 4.2 ja es possible apreciar la textura aplicada a un planeta, el planeta Terra. També es pot apreciar a la figura 4.4 i 4.5 amb textures d’altres planetes (la lluna i Mart respectivament). Els tres planetes contenen una única textura aplicada al llarg de tota la superfície, tal com s’explica a l’apartat 2.2.

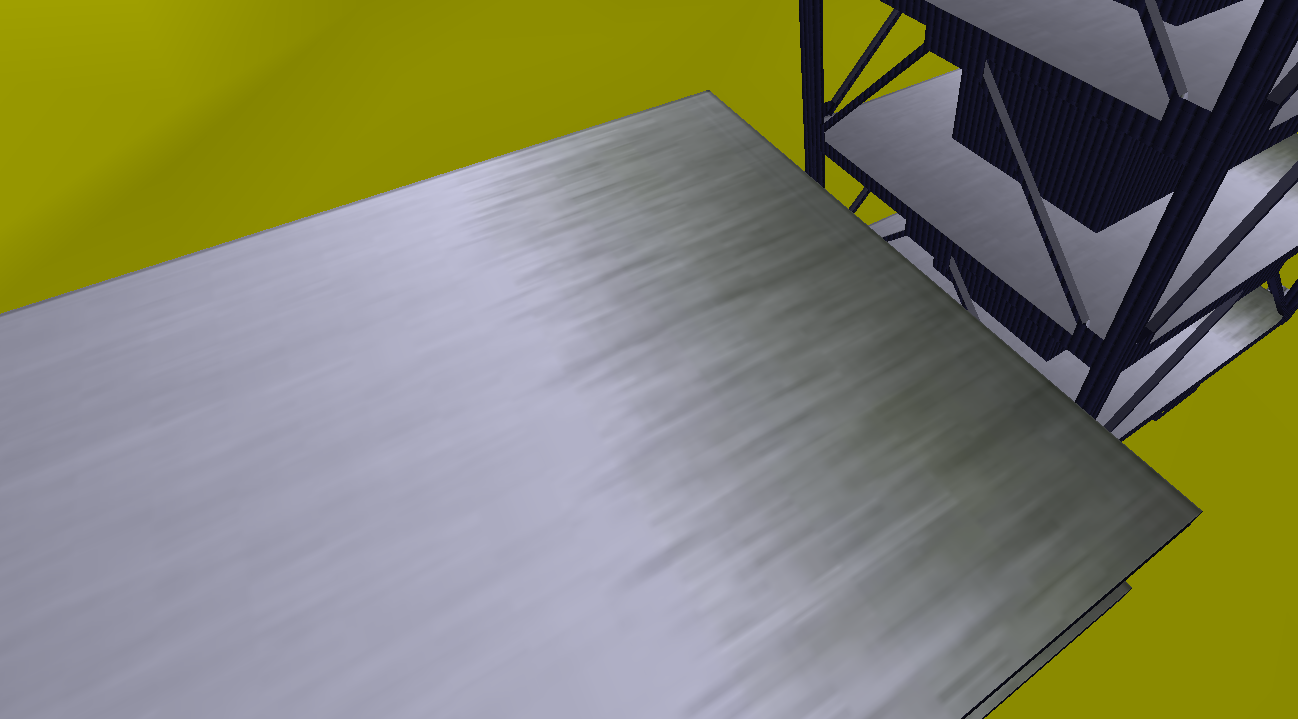


*Figura 4.4. Vista del planeta 2 (la lluna).*



*Figura 4.5. Vista del planeta 3 (Mart)*

A la figura 4.6 es pot apreciar la textura aplicada al suport del coet, que venia junt amb un objecte que vam aconseguir online i per tant, està importat com a objecte extern. A la figura 4.7 es pot apreciar un coet amb unes textures creades per nosaltres en un programa extern (SketchUp).



*Figura 4.6. Vista de les textures de la plataforma.*



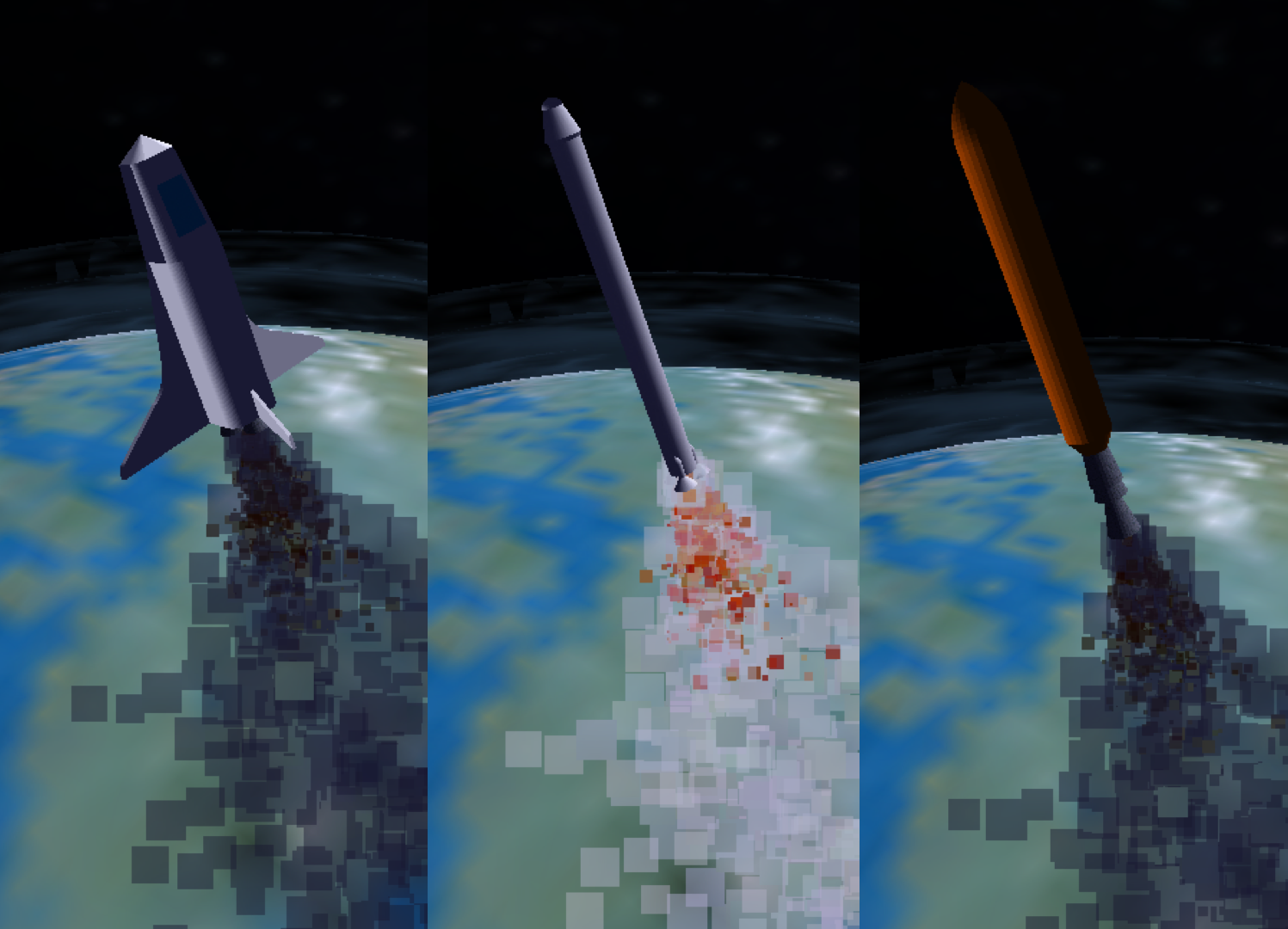
*Figura 4.7. Vista de les textures d’un coet.*

## 

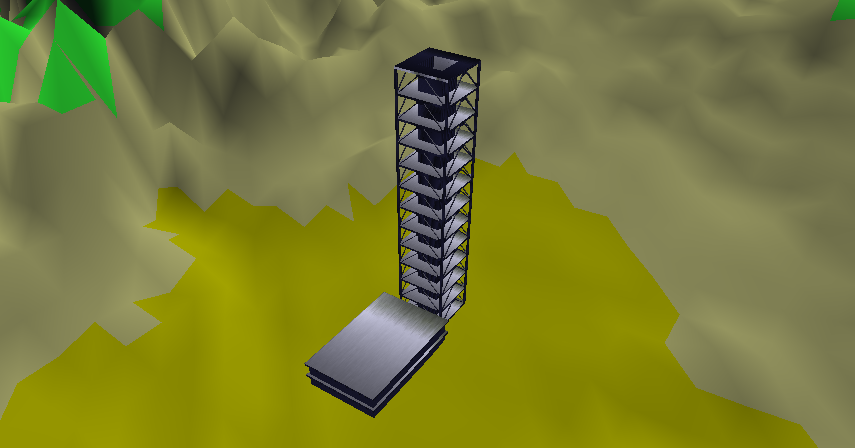
## 

## 4.4. Coets i plataforma

S’han creat diversos coets, que es poden apreciar en conjunt a la figura 4.8. També s’ha creat una plataforma des de la que els coets es llancen que es pot veure a la figura 4.9.



*Figura 4.8. Diferents coets creats.*



*Figura 4.9. Plataforma de llançament.*

## 4.5. Planetes

Els diversos planetes creats es poden observar a continuació. La figura 4.2 mostra el planeta Terra, i la figura 4.4 i 4.5 mostren els altres planetes.

## 4.6. Ciutats

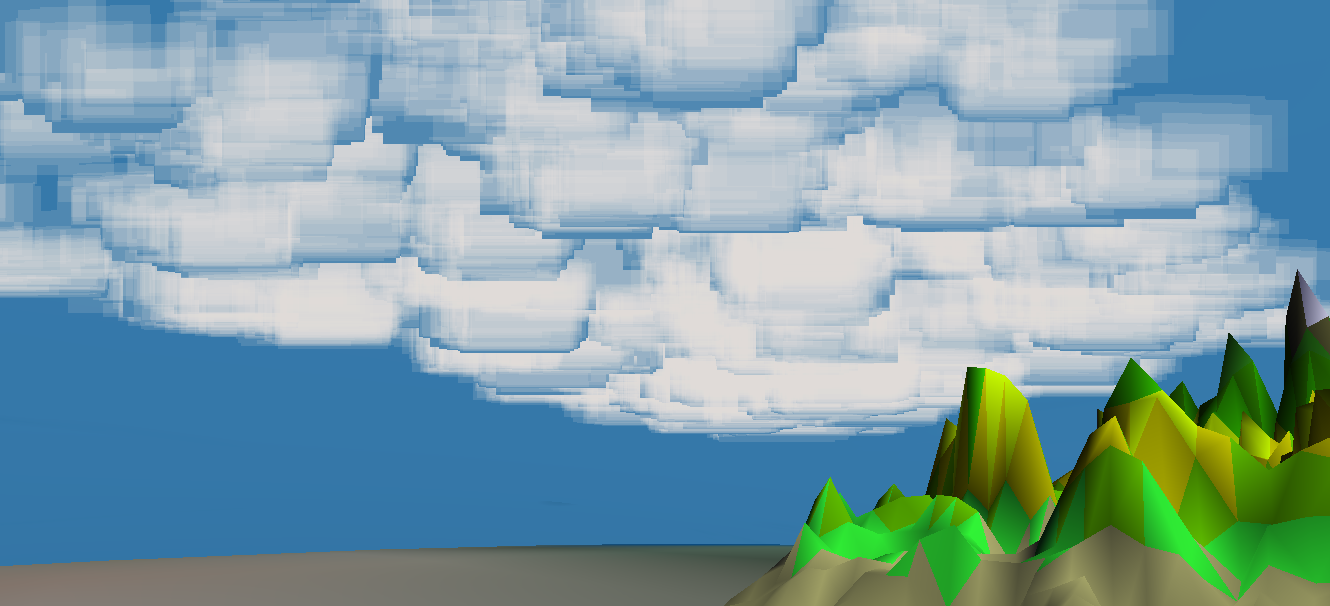
Com es pot veure a la figura 4.10, les ciutats donen més realisme al coet, fent que sembli que es llença des d’una població civilitzada.

IMATGE PENDENT

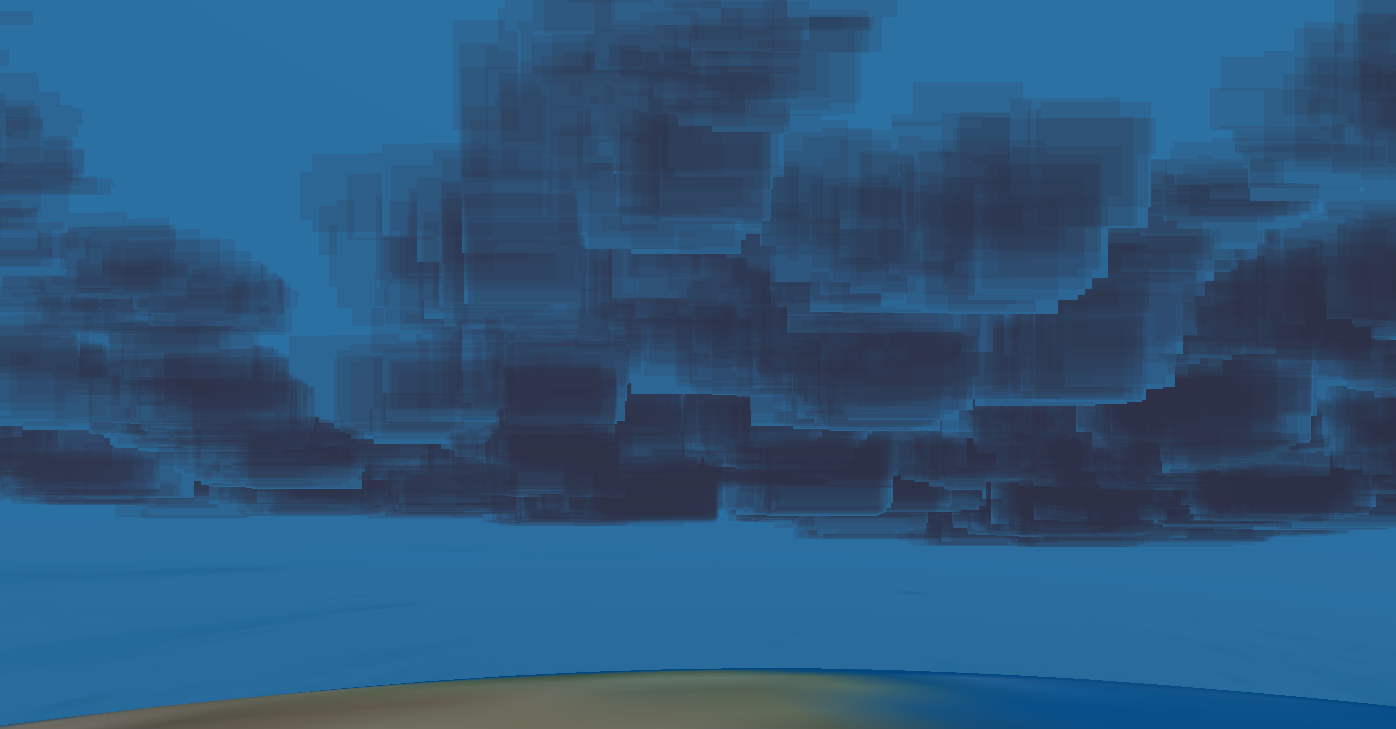
*Figura 4.10. Ciutats implementades al planeta Terra.*

## 4.7. Núvols

A la figura 4.11 s’aprecien els núvols de forma correcte. Per algun motiu que no hem arribat a descobrir, encara que sospitem que te a veuré amb la il·luminació, per què els núvols es veuen obscurs si es visualitzen des d’algun angle concret, tal com mostra la figura 4.12. La solució temporal que hem trobat és desactivar la llum quan es visualitzen els núvols.



*Figura 4.11. Visualització correcte dels núvols.*



*Figura 4.12. Visualització incorrecte dels núvols.*

## 4.8. Sistema de partícules

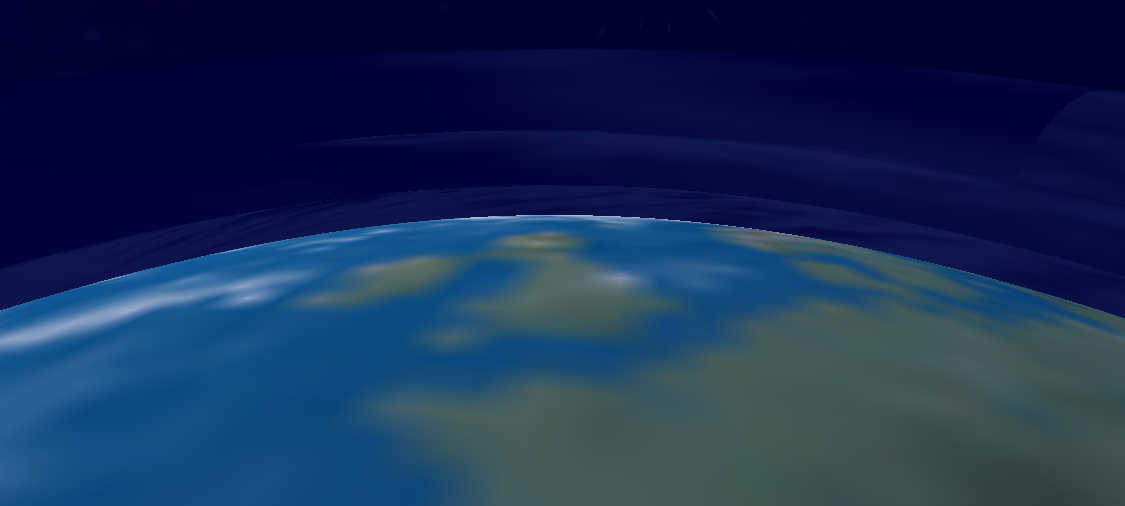
A tots els coets s’aplica el mateix sistema de partícules que simula el foc i el fum. Tal i com es veu a la figura 4.8. A la figura 4.13 es veu aquest sistema de més a prop.



*Figura 4.13. Sistema de partícules de prop.*

## 4.9. Capes de l’atmosfera

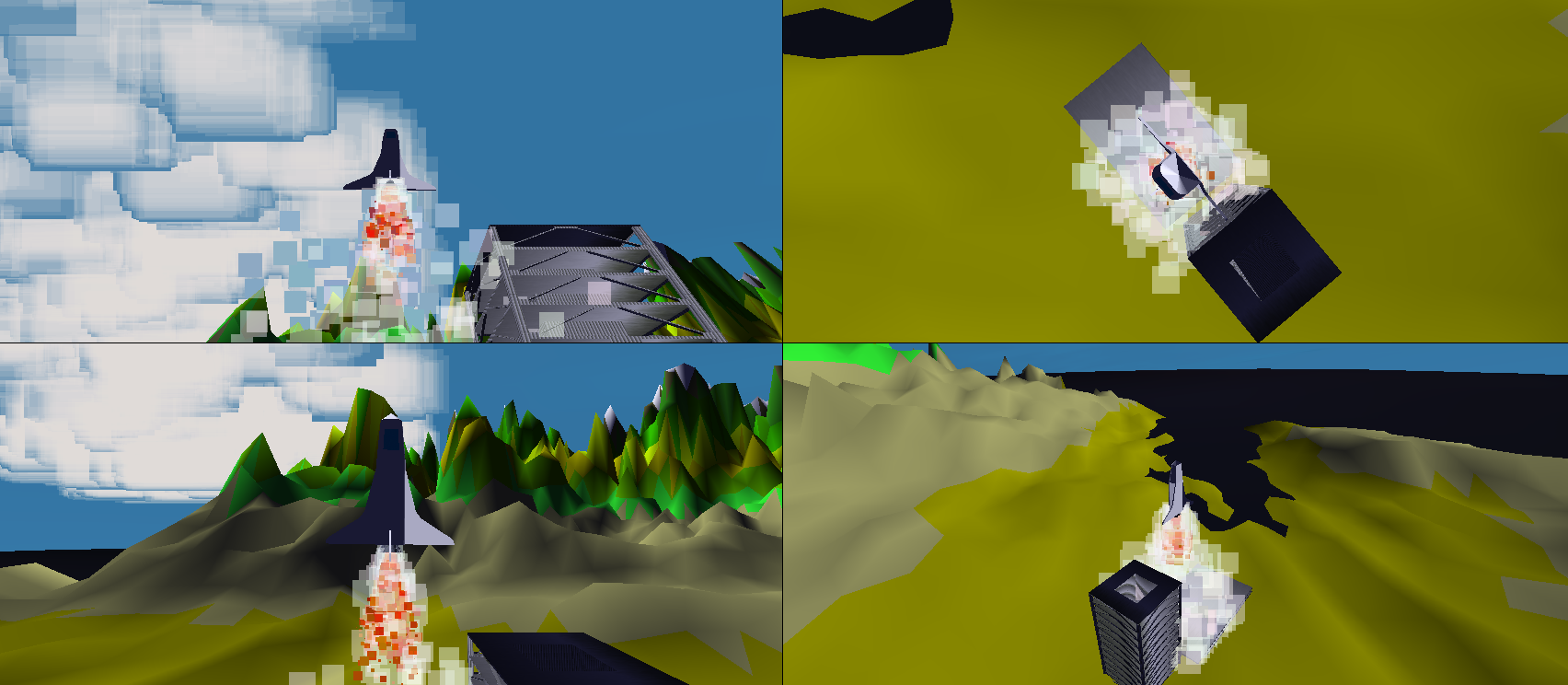
Hi han diverses capes que donen més realisme a la Terra. En la figura 4.2 es pot veure el planeta rodejat d’aquestes capes. A la següent figura, la 4.14 es pot veure amb més detall.



*Figura 4.14. Detall de les capes de l’atmosfera de la Terra.*

## 4.10. Càmeres

Hi han dues visualitzacions implementades, una d’elles conté 4 càmeres diferents. La visualització lliure té fixe la càmera en el coet i es l’utilitzada en les figures anteriors. La figura 4.15 mostra en canvi la visualització amb quatre càmeres diferents a la vegada. Aquestes es troben definides a la secció 2.9 de l’informe.



*Figura 4.15. Visualització multi-vista.*

# 

# 

# 5. CONCLUSIONS

* S’ha aconseguit desenvolupar una aplicació en Visual C++ i OpenGL per a la visualització de la simulació d’un llançament d’un coet.
* S’ha pogut aplicar unes físiques mitjanament realistes que simulen el comportament d’un coet real.
* El sistema de partícules per al fum, el foc i els núvols aconsegueixen reproduir d’una manera senzilla aquestos components, però manquen realisme.
* Les textures s’han pogut aplicar amb èxit.
* Hi han hagut opcions extres, com múltiples planetes, múltiples coets i afegir ciutats que ha donat temps a implementar.
* La telemetria permet seguir la informació de les dades del coet en temps real.

Com principals millores de l‘aplicació destacaríem:

* Afegir millores en el sistema de partícules, aplicant shaders. Hem intentat aplicar-les però per temps i dificultat del tema no hem sigut capaços.
* A l’inici del projecte volíem afegir més opcions de personalització, com que l’usuari pugues modificar la quantitat de combustible o que la mida del planeta fos variable. Per la dificultat que donava la relació entre les fórmules físiques i l’escalat d’objectes no ha sigut possible.
* Optimitzar els recursos d’ordinador necessaris per a executar el programa, ja que si es un ordinador amb poca potència pot veure’s massa lent.
* Comprendre amb més exactitud l’ús de la il·luminació ja que ens ha provocat molts problemes.

Com a equip, a més, ens agradaria afegir que la experiència del desenvolupament en equip en el format ABP ha sigut positiva. A més de conèixer altres persones, aplicar tècniques de treball en equip ens ha semblat molt interessant ja que es simula més el treball que podríem fer en el futur. En quant a l’ús d’OpenGL ens ha agradat conèixer un llenguatge d’ús global que no havíem vist abans i veure tant la complexitat com la facilitat que això porta.

# 

# 

# 6. REFERÈNCIES ELECTRÒNIQUES I BIBLIOGRÀFIQUES

EN CURS...

Inserir les referències (llibres, articles en revistes, adreces web, etc.) que heu utilitzat per a realitzar el projecte. Procureu ordenar alfabèticament les referències per autor.

IMPORTANT: Totes les referències que apareguin en aquest capitol s’han de referenciar en el texte del portfoli.

El format és el següent:

LLiBRE:

[Autors-any] Autors, *Títol*, Editorial, Any.

REVISTA:

[Autors-any] Autors, Títol, *Revista*, volum(número):pàgina inicial-pàgina final, any.

REVISTA:

[nom] Adreça web, explicació de què es tracta.

Apellido, A. A. (Fecha). Título de la página. Lugar de publicación: *Nombre de la página web*. dirección de donde se extrajo el documento (URL).

A continuación un ejemplo ilustrativo:

Argosy Medical Animation. (2007-2009). Visible body: Discover human anatomy. New York, EU.: Argosy Publishing. Recuperado de http://www.visiblebody.com

Hay que poner…

El TFG, Libro de las trayectorias del nil, página de donde sacamos la plataforma, los apuntes del profe, lo del aleatorio grande del nil.

Usando www.citationmachine.net

## 6.1. Llibres

[MaP-05] E. Martí, E. Pujadas, *Un sistema per a la implementació de colisions de forma jeràrquica sobre arbres octals*, IEEE Transactions on Pattern Recognition and Image Análisis, 23(5):125-135, 2005.

[Mar-05] E. Martí, *Anàlisi de documents gràfics*, Springer Verlag, 2005.

[MPO-05] E. Martí, E. Pujadas, I. Ortuño, Un sistema per a la implementació de colisions de forma jeràrquica sobre arbres octals, *IEEE Transactions on Pattern Recognition and Image Analysis*, 23(5):125-135, 2005.

## 6.2. Pàgines web

Ideastudio. Destination Moon Rocket - 3d model- .3ds, .obj, .max. <https://free3d.com/3d-model/destination-moon-rocket-11814.html>

Sketchup. Video Tutorials: Getting Started. <https://www.sketchup.com/es/learn/videos/826?playlist=58>

24 de Septembre de 2012. YouTube. SketchUp - Creación de Esferas. <https://www.youtube.com/watch?v=5dKkvFujO5E>

01 de Novembre de 2013. YouTube. Como diseñar objetos curvos con Sketchup. <https://www.youtube.com/watch?v=yR_EkjKpjKQ>

[OpenGL] [www.opengl.org](http://www.opengl.org), web oficiosa d’OpenGL amb informació sobre extensions de la llibreria i FAQ’s.

[3dgep] <https://www.3dgep.com/texturing-and-lighting-in-opengl/#Putting_it_All_Together> web amb informació sobre llums i textures en openGL

[glprogramming] <http://www.glprogramming.com/red/chapter05.html> web amb tutorials d’OpenGL. Destaquem especialment aquest tutorial sobre llums.

[openglbook] <http://openglbook.com>, pàgina en forma de llibre creat per Eddy Luten.

## 

## 6.3 Programari software

## 

# ANNEXE I. ACTES DE LES REUNIONS

En les següents pàgines s’adjunten les actes de les reunions. Aquestes es realitzaven de forma setmanal.

# ANNEXE II. CONTROLS

En les següents pàgines s’adjunten els controls realitzats de forma periòdica. En aquest cas hi han dos, que es van entregar els dies diumenge, 29 octubre i diumenge, 26 novembre de 2017.

# ANNEXE III. CALENDARI D’ORGANITZACIÓ DELS SPRINTS

A l’inici del projecte es va crear un calendari que hem anat seguint per a poder tindre un control de tot allò que havíem d’entregar o fer. El calendari es mostra a continuació.

# ANNEXE IV. FORMULARI DE LES FÍSIQUES

Per a la realització de les físiques es va realitzar tot un formulari que ens va servir a l’hora de programar. A continuació es mostra.