

Informatika Ingeniaritzako Gradua
Konputazioa

Gradu Amaierako Lana

**Sateliteen ibilbidearen kalkulua eta bistaraketa
Unity ingurunean**

Egilea

Iker Narbaiza Arrondo

2022

Informatika Ingeniaritzako Gradua
Konputazioa

Gradu Amaierako Lana

**Sateliteen ibilbidearen kalkulua eta bistaraketa
Unity ingurunean**

Egilea

Iker Narbaiza Arrondo

Zuzendaria

Joseba Makazaga

Laburpena

Gradu Amaierako Lan proiektu honen helburua satelite artifizialen ibilbidearen kalkulua eta bistaraketa egitea da. Era berean, Unity ingurunearen eta Oculus VR gailuen kalkulugaitasuna ere neurtu nahi da. Gailuak erabiltzeko bi era daude, alde batetik, kanpoko CPU baten bidez kalkulaturako ingurune birtuala erakusteko erabil daiteke, baina bestetik, gailuak berak kalkulaturako inguruneak erakusteko ere bailia daiteke. Azken erabilera hau askoz autonomoagoa da, baina gailuaren kalkulugaitasunaren menpe geratzen da erabiltzailea.

Sateliteen ibilbidearen kalkulua egiteko ekuazio diferentzial bat ebatzi behar da, eta horretarako zenbakizko metodoak erabiliko ditugu, beraz, zenbakizko metodoei deia egin beharko zaie eta ibilbidea kalkulatu ahala satelitea bere ibilbidearen kokapen berria bistaratu beharko da. Erabiliko den metodo nagusia Taylor-en garapenaren metodoa izango da, baina beste metodo gehiago ere ikasi eta inplementatuko dira. Kalkuluak egiteko hasierako kokapena eta abiadura behar dira, eta urratsez urrats, denboran aurrera eginaz, kokapen eta abiadura berriak kalkulatu dira. Horrela, hurrengo *frame*ean ze posizio berri eta zein direkzio bektore berri izango duen sateliteak lor ditzakegu.

Proiektu honetatik lortuko dugun aplikazioarekin hainbat sateliteren ibilbide desberdinak ikusiko ditugu gure konputagailuaren pantailatik edo gure Oculus VR gailuetatik. Sateliteak ikuspuntu ezberdinetatik ikusteko gai izango gara: kamera librea, finkoa, sateliteen ikuspuntua, edo sateliteen inguruko ikuspuntua...

Gaien aurkibidea

Laburpena	i
Gaien aurkibidea	iii
Irudien aurkibidea	vii
Taulen aurkibidea	ix
1 Sarrera	1
1.1 Motibazioa	1
1.2 Proiektuaren helburuak	2
1.3 Dokumentuaren antolaketa	3
2 Plangintza	5
2.1 Lanaren deskonposaketa egitura	5
2.1.1 Proiektuaren kudeaketa	5
2.1.2 Proiektuaren garapena	6
2.1.3 Dokumentazioa	8
2.1.4 Komunikazioa	8
2.2 Arriskuen analisisa	8
2.3 Gantt diagrama eta denboren taula	9

3	Oinarri teorikoak	13
3.1	Zenbakizko metodoak	13
3.1.1	Taylor-metodoa	14
3.1.2	Runge-Kutta metodoa (RK4)	17
3.1.3	Ralstonen metodoa	19
3.2	Satelitearentzako ekuazio diferentzialak	20
3.3	Satelitearen ekuazioak eta zenbakizko metodoen konbinaketa	23
3.3.1	Zenbakizko metodoen kudeaketa	24
4	Proiektuaren garapena	27
4.1	Unity ingurunea	27
4.2	Eszena	28
4.3	Implementazioa eta Scriptak	31
4.4	Errealitate birtualeko betaurrekoak: Oculus Quest 2	34
5	Emaitzen azterketa	37
5.1	Oculus gailuen gaitasunen azterketa	37
5.2	Zenbakizko metodoen erroreen analisia	39
5.2.1	Taylor-metodoa	40
5.2.2	RK4	42
5.2.3	Ralston-metodoa	43
5.2.4	Metodoen arteko alderaketa	44
5.3	Orbiten errorea	44
5.4	Sateliteen orbitak	46
6	Ondorioak	49

Eranskinak

A Programaren biltegia	53
B Aplikazioaren bideoak	55
Bibliografia	57

Irudien aurkibidea

2.1	LDE diagrama.	6
2.2	Gantt diagrama.	9
3.1	Lurraren txatatzea. Erradio polarra ekuatore-erradioa baino 21 km laburragoa da.	20
3.2	Lurraren zentrotik satelitera sortzen den erradioa: $Lurra = [0, 0, 0] \rightarrow r = \ q\ $	22
3.3	q_0 puntu bat izanda, q_1 puntua lor dezakegu. Puntu hau h urrats luzerarekiko menpekoa izango da.	23
4.1	Unity eszena. Sekzio desberdinetan banatuta azalpenerako.	28
4.2	Lehenetsita datorren Unity eszena.	29
4.3	Magellan_ex satelite objektua Unity eszenan.	30
4.4	Kameren errepresentazioa. 1 kamera librea, 2 kamera finkoa eta 3 sateliteen kamera.	31
4.5	Kubo bat Unityn.	32
4.6	Oculus Quest 2 gailuaren aginteak. Botoi bakoitza etiketa propioa du, programatzerakoan identifika ahal izateko. [Oculus, 2022]	34
4.7	Unity Hub-en <i>Android Build Studio</i> plataforma gehitzeko pantaila.	35
5.1	Ordenagailuen eta Oculus Quest 2 gailuen FPS alderaketa sateliteak gehitu ahala.	39

5.2	K pauso emandakoan, h handitu ahala jaso diren posizioaren errore erlatiboaren errepresentazioa grafiko bidez.	41
5.3	Orbitekiko posizio errorearen errepresentazioa grafiko bidez.	45
5.4	Orbitekiko abiadura errorearen errepresentazioa grafiko bidez.	45
5.5	(a) Lehenengo satelitearen orbita. (b) Bigarren satelitearen orbita (goitik ikusita).	47
5.6	(a) Hirugarren satelitearen orbita. (b) Laugarren satelitearen orbita.	47
5.7	Bosgarren satelitearen orbitaren bistaraketa.	48
5.8	Satelite guztien orbiten bistaraketa aldi berean.	48

Taulen aurkibidea

2.1	Estimatutako eta errealtateko denboren taula.	11
5.1	Ordenagailuan jasotako FPS eta ms emaitzak.	38
5.2	Oculus Quest 2 gailuetan jasotako FPS eta ms emaitzak.	38
5.3	Taylor-metodoaren posizioarekiko errore erlatiboaren taula.	40
5.4	Taylor-metodoaren abiadurarekiko errore erlatiboaren taula.	40
5.5	Taylor-metodoaren posizioarekiko errore absolutuaren taula.	41
5.6	Taylor-metodoaren abiadurarekiko errore absolutuaren taula.	41
5.7	RK4 metodoaren posizioarekiko errore erlatiboaren taula.	42
5.8	RK4 metodoaren abiadurarekiko errore erlatiboaren taula.	42
5.9	Ralston-metodoaren posizioarekiko errore erlatiboaren taula.	43
5.10	Ralston-metodoaren abiadurarekiko errore erlatiboaren taula.	43
5.11	Orbitekiko posizio errorearen taula.	44
5.12	Orbitekiko abiadura errorearen taula.	45
5.13	Satelitearen posizioa, abiadura eta Lurrarekiko distantzia orbitatzen joan ahala.	46

1. KAPITULUA

Sarrera

Azken mendean zehar milaka satelite artifizial jaurti dira orbitatzera, lehenengoa, 1957ko urriaren 4an jaurti zuen Sobietar Batasunak [P. Cracknell and A. Varotsos, 2007]. Aurten, 2022. urtean, *Union of Concerned Scientists* (UCS) erakundeak 4852 satelite artifizial orbitatzen daudela gordeta du bere datu baseetan [Concerned Scientists, 2022], datu hauek urte honetako lehenengo egunean eguneratu ziren, pasaden urtean, egun berean, 3372 satelite zituzten gordeta, honek esan nahi du 365 egunetan 1480 satelite gehiago kontatu direla (ia %44ko hazkundera) horregatik da hain garrantzitsua sateliteen gaineko kontrol zehatza izatea.

1.1 Motibazioa

Gratu Amaierako Lan (GrAL) hau tutoreak proposatutako ideia batetik dator. Nire intentzioa konputazioaren espezialitatean ikusitako ingurune birtualen bistaraketarekin erlazionatutako proiekturen bat egitea zen, graduan zehar ikusitako batekin edo beste berri batekin.

Unity ingurunean bistaraketa egiteko proposamenarekin, eta, gainera, errealitate birtualeko betaurrekoentzat bateragarria den aplikazio bat sortzeko aukerarekin, zaila zen niretzat proiektu hau ez aukeratzea. Graduan zehar ez dugu Unity ingurunea landu, beraz, motore grafiko hau hutsetik ikasteko aukera eman dit lan honek eta noski, VR tresnak erabili ahal izan ditut, gailu hauek guztiz berriak izan dira niretzat.

Honetaz gain, printzipio matematikoekin lan egiteak erosotasun nabarmen bat eman dit proiektu guztian zehar nire bizitzan inoiz ez baitut izan arazorik zenbakizko kalkuluekin. Hainbeste gauza berrirekin lan egin behar izan arren, eroso sentiarazten nauen esparru bat lantzeak GrAL honen garapena arindu du.

1.2 Proiektuaren helburuak

Helburu nagusia satelite artifizialen ibilbidea bistaritzen duen aplikazio bat sortzea da, Unity ingurunea erabiliz eta errealitate birtualeko betaurrekoekin bateragarria. Esan bezala, beharrezkoa da aplikazio honek ibilbidearen kalkulua egiteko ekuazio diferentzialak ebaztea eta kalkulua egin ahala bistaratzea.

Aplikazio honek erabiltzaileari hainbat ikuspegi kontrolatzea ahalbidetu behar dio, ikuspegi libre bat, orokor bat eta satelite bakoitzarekiko ikuspegi bat izateko. Era berean, Oculus gailu batean horrelako ikuspegiak erakutsiz erabiltzaileak murgiltze sentazioa lortu beharko du.

Honetaz gain, neurtu nahi da errealitate birtualeko betaurrekoek (Oculus tresnek) duten gaitasuna kalkuluak egin eta bistaratzeko, eta zenbat satelite/metodo batera kudea ditzakeen ere neurtu nahi da. Tresna horien kalkulu-gaitasuna ez balitz nahikoa, orduan gailua bistaraketa-tresna soil modura erabili beharko litzateke.

Helburu hauek betetzeko, betebeharrak hauek defini ditzakegu:

1. Unity ingurunean lan egiten ikasi beharko da, eta dispositibo ezberdinetarako exekutagarriak sortu beharko dira.
2. Unity ingurunetik kalkuluak egiteko kanpo funtzioei deiak nola egin eta kalkuluen emaitzen kudeaketa ere egin beharko da.
3. Sateliteen ibilbidea kalkulatzeko, ekuazio diferentzialen zenbakizko soluzioak kalkulatu beharko dira, beraz, ekuazio diferentzialen sistema horiek zein diren eta nola ebatzi behar diren ikasi beharko da. Eta beren arteko diferentziak ere erakutsi beharko dira.
4. Ibilbidea kalkulatu ahala ibilbidea grafikoki erakutsi beharko da, eta erakuste horretan ikuspuntu ezberdinak erabili ahal izango dira, beraz, kameraren kontrola ere egin beharko da.

1.3 Dokumentuaren antolaketa

Txosten hau kapitulutan banatuta dago, lehena hau izan da, sarrera.

Bigarren kapituluan proiektu honen plangintza ikusiko dugu.

Hirugarren kapituluan proiektua aurrera eramateko erabilitako oinarri teorikoak azalduko dira. Hemen zenbakizko metodoak eta sateliteen ekuazioak aurkituko ditugu.

Laugarren kapitulua proiektuaren garapenari dagokio. Kapitulu honetan Unity ingurunearen funtzionamenduaren azalpena, ingurune horretan garatutako aplikazioaren deskribapena eta aplikazio hori errealitate birtualeko betaurrekoetara nola moldatu den azalduko da.

Bosgarren kapitulu batean, aplikazioarekin egindako proben emaitzen azterketa aurkituko dugu. Oculus Quest 2 gailuen ahalmen konparaketa ordenagailuarekin, zenbakizko metodoen alderaketak eta horien urrats luzeraren menpeko erroreak aurkituko ditugu.

Azkenik, seigarren kapituluan, Gradu Amaierako Lan hau bukatutakoan atera ditzakegun ondorioak azalduko dira.

Memoria hau biribiltzeko, eranskinak eta bibliografia aurkituko ditugu amaieran. Eranskinetan, Unityn garatutako aplikazioa edozein ordenagailutan deskargatzeko DropBox-eko esteka aurkituko dugu, aplikazioaren Oculus gailuetan egindako exekuzioaren bideo batez gain.

2. KAPITULUA

Plangintza

Proiektuarekin hasi baino lehen, oso garrantzitsua da plangintza zuzen bat egitea. Plangintza on baten helburuetako bat erabakiak hobeto hartzea da, helburu jakin bat zehazteko helburuarekin. Beraz, plangintza-estrategia batek kontuan hartu behar ditu egungo egoera eta helburu hori lortzeko ondorioak eragin ditzaketen kanpoko eta berezko faktore guztiak.

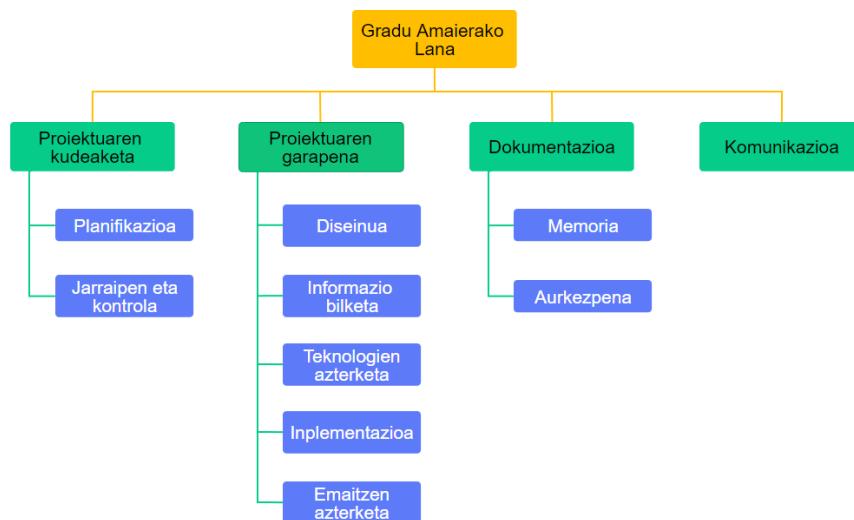
Hau kontuan izanik, 4 fasetan banatuko da proiektua: kudeaketa, garapena, dokumentazioa eta komunikazioa. Gainera fase bakoitzean aurki ditzakegun lan-paketeak definitu dira. Pakete hauek proiektuaren faseen dependentziak eta iraupena definituko dute. Hurrengo orrialdeetan pakete hauen azalpenetan sartuko gara.

2.1 Lanaren deskonposaketa egitura

Lan-paketeen eta proiektuaren arteko erlazioa erakusteko, LDE diagrama bat (Lanaren deskonposaketa egitura) aurkezten da [2.1](#) irudian. LDE erabiliz, GrALaren helburuak betetzeko egin behar den lan guztiaren deskonposizioa adierazten da.

2.1.1 Proiektuaren kudeaketa

Proiektua aurrera eraman ahal izateko kudeaketa zuzen bat izan behar du hasieratik bukaerara. Fase hau honako lan-paketeetan dago banatuta:



2.1 Irudia: LDE diagrama.

1. Planifikazioa (P): Proiektua hasi aurretik egingo da eta honen garapenaren bitartean eguneratuko da. Kapitulu hau adierazten duena da. Arriskuak, dudak eta desbiderapenak aurreikusiko ditu.
 - P.1: Eskakizunen identifikazioa, hasierako erabakiak hartzea, informazioaren analisia eta dudaren ebazpena.
 - P.2: Hasierako plangintza, garapenerako ingurunea prestatzeko.
 - P.3: Plangintzaren eguneraketa, beharrezkoa bada.
2. Jarraipen eta kontrola (JK): proiektuaren garapen egokia bermatuko duten atazak edukiko ditu eta konkretuki, proiektuaren epeen eta espezifikazioen betetzea.
 - JK.1: Proiektuaren garapenari buruzko informazio garrantzitsua jaso.
 - JK.2: Planarekin, jarraipeneko informazioaren konparaketa, desbideratze esanguratsuen eta sortzen diren arriskuen identifikazioa.
 - JK.3: Proiektuak arrakasta izan dezan, baldintzen bermatzea.

2.1.2 Proiektuaren garapena

Iraupen luzeena duen fasea da hau. Fase honetan aplikazioarekiko zerikusia duen guztia biltzen da, diseinutik, emaitzen azterketararte. Bost lan-paketetan dago banatuta:

1. Diseinua (D): Aplikazioarekin hasi aurretik egin beharko da. Aplikazioaren itxuraren eta implementazioaren diseinuari dagokio.
 - D.1: Aplikazioaren itxuraren diseinua.
 - D.2: Aplikazioaren implementazioaren diseinua.
2. Informazio bilketa (IB): Erabiliko diren kontzeptu matematikoen bilketa eta azterketa.
 - IB.1: Kontzeptu matematikoen bilketa eta azterketa.
3. Teknologien azterketa (TA): Unity softwarea eta Oculus VR gailu ezezagunen azterketan datza.
 - TA.1: Unity softwarearen erabilpenari buruzko informazio bilketa.
 - TA.2: Unity softwarearen instalazioa.
 - TA.3: Oculus VR gailuen erabilpenari buruzko informazio bilketa.
 - TA.4: Oculus VR gailuak erabiltzeko beharrezko implementazioa.
4. Implementazioa (I): Teknologia eta kontzeptu matematikoak erabat ezagunak izanda, aplikazioa sortzean datza.
 - I.1: Eszena sortzea.
 - I.2: Ibilbideen implementazioa.
 - I.3: Kameraren implementazioa.
 - I.4: VR gailuetara egokitzea. Ez da lan-pakete honekin hasiko aplikazioa ordenagailuan era egokian funtzionatu arte.
 - I.5: Erabiltzailearen interfazea sortzea.
 - I.6: Proiektua zuzenki komentatzea eta DropBox-era igotzea.
5. Emaitzen azterketa (EA): Aplikazioa erabat implementatua izanik, exekuzio desberdinak egitean eta aztertzean datza.
 - EA.1: Zenbakizko metodo desberdinekin lortutako emaitzen azterketa.
 - EA.2: Sateliteen ibilbidearen errore azterketa.
 - EA.3: VR gailuek kalkuluak egiteko duten gaitasunaren azterketa.

2.1.3 Dokumentazioa

Fase honetan memoriaren idazketa eta aurkezpenaren prestaketa egingo da. Memoria idazteko LaTeX editore bat erabili dut Overleaf softwarean.

1. Memoria (M): Dokumentu hau. Aplikazioa bukatzen joan ahala idazketa hasi behar-ko da.
 - M.1: Memoriaren eskema sortu.
 - M.2: Memoria idatzi.
2. Aurkezpena (A): Proiektu osoa bukatutakoan
 - A.1: Aurkezpena prestatu.
 - A.2: Gidoia idatzi.

2.1.4 Komunikazioa

Fase hau proiektua hasi baino lehen hasi zen, tutorearekin ideiak eztabaidatzean. Proiektu honen kasuan, komunikazioa asterokoa izan da, posta elektronikoz bidez. Dударik edo komentariarik izanda posta elektronikoz konpontzen saiatuko gara eta beharrezko kasuetan bulego bilerak egin dira.

Fase hau ez da banatuko lan-paketeetan.

2.2 Arriskuen analisisia

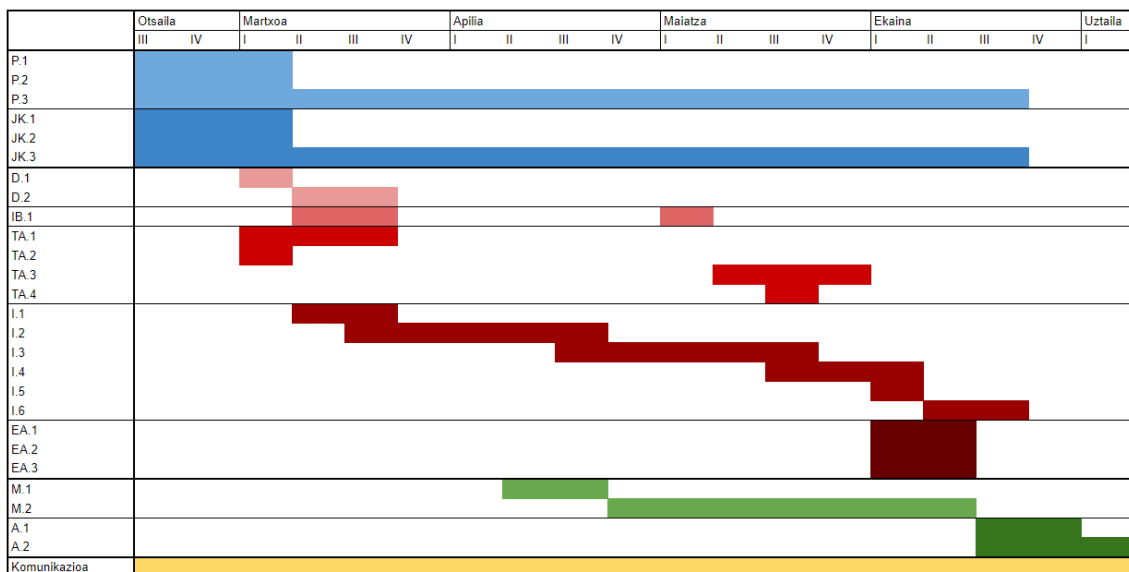
Proiektu honen iraupena kontuan izanda, beharrezkoa da garapenean zehar gerta daitezken ahalik eta arrisku gehien detektatzea proiektuaren helburuak lortzeko. Proiektu honen hasieratik ekaineko deialdian aurkezteko helburuarekin egin da eta aldi berean graduko beste 18 kreditutan matrikulaturik nagoela kontuan izanda, proiektuak lan banaketa egoki bat edukiko du, otsailaren erdialdetik lanean hasita eta ekainaren 20rako proiektua praktikoki bukatuta egoteko. Aldi berean kurtsatzen ari naizen ikasgairen baten baino gehiagotan arazorik izanda, proiektua iraileko deialdira atzeratzea kontuan har liteke beharrezkoa ikusiko balitz.

Proiektuaren garapenean egonda, aplikazioaren bertsio berriak sortzean, ZIP fitxategietan konprimituko dira sortzen diren bertsioak eta Google Drive tresnara igoko dira segurtasun kopiak sortzeko eta gailu desberdinetan lan egin ahal izateko. Tresna hau memoria mugatua duenez, espaziorik gabe ez geratzeko, bi astetan behin segurtasun kopiak disko gogor eramangarri baten gordeko dira.

2.3 Gantt diagrama eta denboren taula

Plangintza honekin bukatzeko, lan-paketeak osatzeko estimatutako denborak eta epeak irudikatzeko, Gantt diagrama eta denboren taula ditugu hemen.

Gantt diagraman (ikusi 2.2 irudia), otsailaren hirugarren astetik, proiektuaren defentsara arte (uztailaren lehengo bi asteetan gutxi-gorabehera) lan-pakete bakoitza osatzeko egongo naizen asteak irudikatzen dira koloreztatuta. Honela, epeak grafikoki ikus daitezke, jarraipena egiterakoan erraztasunak gehitzeko.



2.2 Irudia: Gantt diagrama.

Denboren taula (ikusi 2.1 taula) aurreikusitako denborak eta errealitatean gertatutakoak irudikatzen ditu. Proiektu hau hasterakoan, 311 orduko lana aurreikusten zen. Txostena-rekin amaitzerakoan, denbora errealak gehitu izan dira, eta, ikus daitekeenez, ez da desbiderapen nagusirik egon. Aurreikusitako denbora 311 ordukoa izan zen, errealitatean, txostena bukatzerakoan, 303 ordukoa izan da, oraindik aurkezpena prestatu behar izanik,

beraz, gehienezko 10 orduko desbiderapena aurki dezakegu, proiektu hain luze bat izanik, nahiko txikia.

	Estimazioa (ordu)	Errealia (ordu)
Kudeaketa	70	65
Planifikazioa	25	21
P.1	10	8
P.2	10	12
P.3	5	2
Jarraipen eta kontrola	45	42
JK.1	20	20
JK.2	20	20
JK.3	5	3
Garapena	146	153
Diseinua	15	17
D.1	5	5
D.2	10	12
Informazio bilketa	15	17
IB.1	15	17
Teknologien azterketa	46	42
TA.1	20	20
TA.2	3	2
TA.3	20	15
TA.4	3	5
Inplementazioa	58	63
I.1	5	4
I.2	20	22
I.3	20	24
I.4	8	8
I.5	2	2
I.6	3	3
Emaitzen azterketa	12	14
EA.1	4	6
EA.2	5	6
EA.3	3	2
Dokumentazioa	85	70
Memoria	65	70
M.1	5	5
M.2	60	65
Aurkezpena	20	-
A.1	10	-
A.2	10	-
Komunikazioa	10	15
GrAL	311	303

2.1 Taula: Estimaturako eta errealitateko denboren taula.

3. KAPITULUA

Oinarri teorikoak

Sortuko dugun aplikazioan bistaratuko diren sateliteen ibilbidea kalkulatzeko ekuazio diferentzialak erabiliko ditugu, eta ekuazio diferentzial hauek ebazteko, zenbakizko metodoei deia egin beharko diegu.

Kapitulu honetan zehar sateliteen ekuazio diferentzialak eta erabilitako zenbakizko metodoak azaldu dira.

3.1 Zenbakizko metodoak

Ekuazio diferentzialen soluzioa ezagutzen ez denez, soluzioaren hurbilketak lortzen dituzten metodoak erabiltzen dira problemaren soluzioak lortzeko. Jakina, hurbilketa hauek ez dira soluzio zehatzak, beti erroreren bat daukate, eta metodo desberdinek errore handiagoa edo txikiagoa lor dezakete, ebatzi nahi den problemaren eta erabilitako metodoaren arabera.

Metodo asko aurki ditzakegu gure problema ebazteko eta proiektu honen kasuan, Taylor-metodoa, Runge-Kutta 4 eta Ralston metodoak erabili dira.

3.1.1 Taylor-metodoa

Azalpena

Ekuazio Diferentzial Arruntak (Ordinary Differential Equation-ODE) ebazteko erabili dudan lehen metodoa Taylorren metodoa izan da. Funtzio batek t_0 une jakinean duen balioa ezagutzen badugu, hau da $y(t_0) = y_0$ bada, eta funtzioaren deribatuak ezagutu baditza-kegu, Taylorren hedapenak $t_1 = t_0 + h$ unean funtzioak hartuko duen balioa kalkulatzeko balio digu:

$$y(t_0 + h) = y(t_0) + h * y'(t_0) + \frac{1}{2} h^2 * y''(t_0) + \frac{1}{3!} h^3 * y^{(3)}(t_0) + \dots \quad (3.1)$$

ODEak ebazteko Taylor-en metodoa

Taylorren hedapena erabiliz, t_0 uneko funtzioaren balioa ezagutuz, aurreragoko t unean soluzioa hurbiltzea du helburu metodo honek. Hedapeneko nahi adina gai erabil daitezke, hurbilketa are eta hobeak lortzeko asmoz. Dena den, round-off errorea muga izango da beti. Daukagun arazoa da, ez ditugula ezagutzen $y(t)$ funtzioaren deribatuak, bakarra, lehenengoa, ezagutzen dugu, hau da, $y'(t)$ besterik ez dugu ezagutzen, beraz, deribatu horiek lortu behar ditugu.

ERREKURTSIO METODOA:

Izan bedi honako ODEa:

$$\dot{y} = f(t, y) \quad (3.2)$$

eta izan bedi hasierako balioa $y(t_0) = y_0$. Ekuazio honen $y(t_0 + h)$ soluzioa serie moduan idatz dezakegu 3.1 ekuazioan agertzen den bezala.

Hasierako baldintza kontutan hartuz, 3.1 soluzioko lehenengo terminoa besterik ez dugu ezagutzen. Baina soluzioaren Taylor hedapeneko beste $y_{[k]}$ koefiziente guztiak falta zaizkigu. Koefizienteak lortzeko $f(t, y)$ erabil dezakegu.

Suposa dezagun t_0 -ren inguruan $f(t, y)$ -ren Taylorren hedapena ezaguna dela. Hedapen hori honela idatz dezakegu:

$$f(t, y(t)) = f_{[0]} + f_{[1]}(t - t_0) + f_{[2]}(t - t_0)^2 + \dots + f_{[k]}(t - t_0)^k + \dots \quad (3.3)$$

Hemen, $f_{[0]} = f(t_0, y(t_0))$, eta Taylorren $f_{[k]}$ koefizienteak t_0 uneko deribatuak izango dira, eta beraz,

$$f_{[k]} = \frac{1}{k!} \frac{d^k f}{dt^k}(t_0) \quad (3.4)$$

Baina $f(t, y(t))$ -ren Taylorren hedapena lortzeko $y(t)$ ere serie modura eduki behar dugu. Soluzioaren $y(t)$ serie hedapena lortzeko, 3.2 ekuazioa lagungarri dugu, izan ere, 3.1 seriea, ($t = t_0 + h$ izanik), t -rekiko deribatzen badugu, lortuko duguna da:

$$\frac{dy(t)}{dt} = y_{[1]} + 2y_{[2]}(t - t_0) + \dots + ky_{[k]}(t - t_0)^{k-1} + \dots \quad (3.5)$$

non

$$t - t_0 = h \quad (3.6)$$

den eta

$$y_{[k]} = \frac{1}{k!} \frac{d^k y}{dt^k} \quad (3.7)$$

den.

Eta 3.3 ekuazioa kontuan hartzen badugu, lortuko dugu:

$$y_{[1]} + 2y_{[2]}(t - t_0) + \dots + ky_{[k]}(t - t_0)^{k-1} + O(t^k) = f_{[0]} + f_{[1]}(t - t_0) + \dots + f_{[k-1]}(t - t_0)^{k-1} + O(t^k) \quad (3.8)$$

Ondorioz, t -ren potentziak berdinduz, $y_{[k]}$ eta $f_{[k-1]}$ balioak honela erlazionatzen dira:

$$y_{[k]} = \frac{f_{[k-1]}}{k} \quad k = 1, 2, 3, \dots \quad (3.9)$$

Hasieran $y(t_0 + h)$ -ren serie garapeneko osagai bakarra ezagutzen dugu, hau da, $y(t_0) = y_0$ denez, 3.1 ekuazioan $y(t_0 + h)$ seriearen hurbilketa moduan honakoa daukagu:

$$y(t_0 + h) \approx y^{[0]} = y_0 + O(h) \quad (3.10)$$

Hurbilketa horri dagokion $f(t, y)$ kalkulatu dezakegu eta horrela lortuko dugu 3.3 ekuazioa $f_{[0]}$ balioa. Balio horrekin, 3.9 ekuazioa erabiliz lortuko dugu $y_{[1]}$ balioa ezagutzeko, eta, ondorioz, $y(t_0 + h)$ serie garapeneren bigarren koefizientea ezagutu ahal izango dugu. Beraz,

$$y(t_0 + h) \approx y^{[1]} = y_{[0]} + y_{[1]} * h + O(h^2) \quad (3.11)$$

seriea lortuko dugu $y(t_0 + h)$ seriearen hurbilketa moduan, hau da, $y^{[1]}$ hurbilketak bi osagai edukiko ditu.

Prozesu hau behin eta berriz aplikatu daiteke, alegia, $y(t_0 + h) \approx y^{[k]}$ hurbilketa erabiliz, $f(t, y^{[k]})$ lortu, badakigu serie hori 3.3 moduko serie bat dela, eta goiko 3.9 ekuazioak y_{k+1} balioa lortzeko aukera emango digu, eta horrela $y^{[k+1]}$ hurbilketa lortuko dugu.

Konkretuki, prozesu errepikakor horren bigarren urratsean $y^{[1]}$ hurbilketatik abiatuz $y^{[2]}$ hurbilketa lortzeko prozesua honakoa litzateke:

Lehenengo $f(t, y^{[1]})$ kalkulatu behar dugu, horrek bi osagai ezagun edukiko ditu:

$$f(t, y^{[1]}) = f_{[0]} + f_{[1]}(t - t_0) + O(t^2) \quad (3.12)$$

Orain, 3.9 ekuazioa erabiliz $y(t_0 + h)$ -ren hirugarren koefizientea, alegia $y_{[2]}$, lortu ahal izango dugu. Osagai berri horrekin $y^{[2]}$ hurbilketa ezagutzen dugu, hau da:

$$y(t_0 + h) \approx y^{[2]} = y_{[0]} + y_{[1]} * h + y_{[2]} * h^2 + O(h^3) \quad (3.13)$$

Berriz ere prozesu bera aplikatzeko moduan geundeke eta $y^{[2]}$ tik abiatuz $y^{[3]}$ lortuko genuke. Hau da, $y^{[2]}$ erabiliz $f(t, y^{[2]})$ lortu, horrek hiru osagai dauzka:

$$f(t, y^{[2]}) = f_{[0]} + f_{[1]}(t - t_0) + f_{[2]}(t - t_0)^2 + O(t^3) \quad (3.14)$$

eta horrekin, 3.9 ekuazioan oinarrituz, eta $k = 3$ balioarekin $y(t)$ -ren garapenerako hurrengo osagaia, hau da, $y_{[3]}$ lortu ahal izango dugu:

$$y_{[3]} = \frac{f_{[2]}}{3} \quad (3.15)$$

Azkenean nahi adinako seriea edukiko dugu, eta serie horretan oinarrituz nahi dugun urratsa $(t - t_0)$ eman ahal izango dugu:

$$y(t_0 + h) = y_{[0]} + y_{[1]}h + y_{[2]}h^2 + \dots + y_{[k]}h^k + O(h^{k+1}) \quad (3.16)$$

Serie hau erabiliko dugu integrazioan urrats bat emateko. Hau da, denboran aurrera egin ahal izango dugu h aldagaiarentzat aukeratzen dugun balioa adina. Horrela $t_1 = t_0 + h$ unean aldagaien egoera berria ezagutuko dugu.

Eta orain prozesu bera egin dezakegu, alegia, t_1 unearen inguruan $y(t_1)$ egoera ezaguna denez, prozesua berriz errepikatuz, $y(t_1 + h)$ balioa kalkulatu dezakegu. Prozesu hau behin eta berriz errepikatuz kalkulatu nahi dugun balioa era zehatz batean lor dezakegu.

[[Taylor, 2022](#)]

3.1.2 Runge-Kutta metodoa (RK4)

Runge-kutta metodoek (RK) ekuazio diferentzial arrunten soluzioen hurbilketarako iteraziozko metodo (esplizitu eta inplizitu) multzoa osatzen dute, zehazki, hasierako balioeko problemaren ebazpenerako metodoak.

Izan bedi

$$y'(t) = f(t, y(t))$$

ODEa, $f : \Omega \subset \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}^n$, non Ω multzo irekia den, eta f ren hasierako balioak

$$(t_0, y_0) \in \Omega$$

betetzen du.

Kasu horretan (s etapako) RK metodoak, orokorrean, honako itxura du:

$$y_{n+1} = y_n + h \sum_{i=1}^s b_i k_i, \quad (3.17)$$

non h iterazio bakoitzeko urrats-luzera den. k_i koefizienteak tarteko hurbilketa puntuetan egindako f funtzioaren balioztatzeak dira.

$$k_i = f(t_n + hc_i, y_n + h \sum_{j=1}^s a_{ij} k_j) \quad i = 1, \dots, s. \quad (3.18)$$

Horrela, a_{ij}, b_i, c_i koefizienteek aukeratutako metodoa zehazten dute eta erabilitako zenbakizko integrazioaren araberako balioak dira.

Lau ordenako Runge-kutta metodoa (RK4)

Runge-Kutta metodoen arteko bat oso erabilia da, horregatik *Runge-Kutta* metodoa edo *RK4* izenek metodo horri egiten diote erreferentzia.

Hasierako balioaren problema bat defini dezagun:

$$y'(t) = f(t, y(t)), \quad y(t_0) = y_0$$

Orduan problema honentzat RK4 metodoa hurrengo ekuazioak adierazten du:

$$y_{i+1} = y_i + \frac{1}{6}h(k_1 + 2k_2 + 2k_3 + k_4) \quad (3.19)$$

Non,

$$k_1 = f(t_i, y_i) \quad (3.20)$$

$$k_2 = f(t_i + \frac{1}{2}h, y_i + \frac{1}{2}k_1h) \quad (3.21)$$

$$k_3 = f(t_i + \frac{1}{2}h, y_i + \frac{1}{2}k_2h) \quad (3.22)$$

$$k_4 = f(t_i + h, y_i + k_3h) \quad (3.23)$$

Honela, hurrengo y_{n+1} balioa une honetako y_n balioak eta h urrats-luzera eta malda estimatuaren arteko biderketak zehazten dute. Malda hainbat malden batezbesteko haztatua da.

k_1 hasierako uneko malda da.

k_2 tartearen erdiko puntuko malda da. Tartearen erdian, $t_n + \frac{h}{2}$ unean, y balioa lortzeko k_1 malda erabili behar da Eulerren metodoaren aplikazioan.

k_3 berriz ere tartearen erdiko puntuko malda da, baina oraingoan Eulerren metodoaren aplikazioan k_2 malda erabili da y kalkulatzeko.

k_4 tartearen bukaerako malda da, eta y balioa lortzeko k_3 malda erabili da.

Lau maldak haztatzean tarteko puntuetako maldei pisu handiagoa ematen zaie:

$$malda = k_1 + 2k_2 + 2k_3 + k_4$$

Runge-kutta metodoaren forma hau lau ordenako metodoa da, horrek esan nahi du urrats bakoitzean egindako errorea $O(h^5)$ neurrikoa dela eta prozesu osoan metatutako errorea $O(h^4)$ ordenakoa dela.

[[Runge-Kutta, 2022](#)]

3.1.3 Ralstonen metodoa

Ralstonen metodoa Anthony Ralstonek proposatutako Runge-Kutta metodo bat da. Metodo hau, bi ordenako metodo bat da.

Izan bedi,

$$y' = f(x, y), \quad y(x_0) = y_0$$

Ralstonen bi ordenako metodoak esaten duen bezala:

$$y_{i+1} = y_i + \frac{h}{3}(k_1 + k_2), \quad x_i = x_0 + ih, \quad (3.24)$$

non,

$$k_1 = f(x_i, y_i) \quad (3.25)$$

$$k_2 = f\left(x_i + \frac{3}{4}h, y_i + \frac{3}{4}hk_1\right). \quad (3.26)$$

Ralstonen bigarren ordenako hurbilketa honela idatz daiteke:

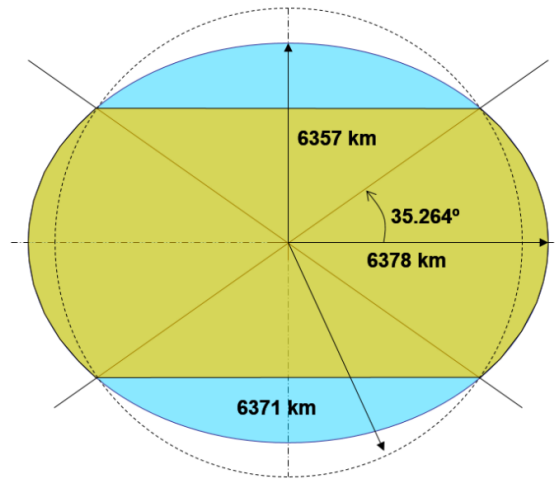
$$y_{n+1} = y_n + \frac{h}{3}f(x_n, y_n) + \frac{2h}{3}f\left(x_n + \frac{3h}{4}, y_n + \frac{3h}{4}f(x_n, y_n)\right), \quad n = 0, 1, 2, \dots \quad (3.27)$$

[Dobrushkin, 2022]

3.2 Satelitearentzako ekuazio diferentzialak

Zenbakizko metodoak erabat definituta ditugunean satelitearentzako ekuazio diferentzialak definitu behar ditugu.

Ekuazio hauen bitartez, hasierako posizio bat eta abiadura bektore bat jasoz, hauen deribatua lortuko ditugu, abiadura eta azelerazioa, hurrenez hurren. Satelite hauen ibilbidearen kalkulua egiteko, Newtonen legeen indarrak eta Lurraren deformazioaren eragina kontuan izango ditugu.



3.1 Irudia: Lurraren txatatzea. Erradio polarra ekuatore-erradioa baino 21 km laburragoa da.

Ekuazio hauei esker, goian ikusitako zenbakizko metodoak aplikatu ahalko ditugu.

Hasteko, finakoak izango diren balio batzuk definitu behar ditugu. Definitu behar dugun lehen balioa Lurraren erradioa da, 6378.135km direla, μ -k grabitazio-parametro estandarri egiten dio erreferentzia eta J_2 balioa Lurraren orbitaren deformazioaren balioari dagokio, balio hau, bigarren harmoniko zonala bezala ere ezagutzen da eta ez da unibertsala, planeta bakoitzak bere J_2 balio propioa du [Schaub and Alfriend, 2001].

μ balioa honela kalkula dezakegu:

Mekanika astronomikoetan, gorputz astronomiko baten grabitazio-parametro estandarra (μ) grabitazio konstantearen (G) eta gorputzen masaren (M) arteko produktuaren emaitza da. Bi gorputzentzako, parametro hau $G(m_1 + m_2)$ edo GM bezala defini daiteke, bi gorputzetatik bat bestea baino askoz handiagoa bada. Gure kasuan azkeneko hau erabiliko dugu Lurraren masa sateliteena baino askoz handiagoa baita. [[Gravitational Parameter, 2022](#)]

$$\mu = GM \quad (3.28)$$

Gure kasuan, G eta M balio ezagunak dira, Lurrarekiko konstanteak direlako:

[[Gravitational, 2022](#)] [[R. Williams, 2021](#)]

$$G = 6.674 \times 10^{-11} m^3 kg^{-1} s^{-2} \quad M = 5.9722 \times 10^{24} kg \quad (3.29)$$

beraz,

$$\mu = GM = 6.674 \times 10^{-11} m^3 kg^{-1} s^{-2} * 5.9722 \times 10^{24} kg \approx 398600 km^3 / s^2 \quad (3.30)$$

Honela geratu lirateke balio finko hauek:

$$\text{Lurraren erradioa} : R_e = 6378.135 km, \quad \mu = 398600 km^3 / s^2, \quad J_2 = 0.0010826157 \quad (3.31)$$

Balio hauek definitutakoan, hasierako q_0 posizio puntua eta v_0 noranzko bektorea definituko ditugu:

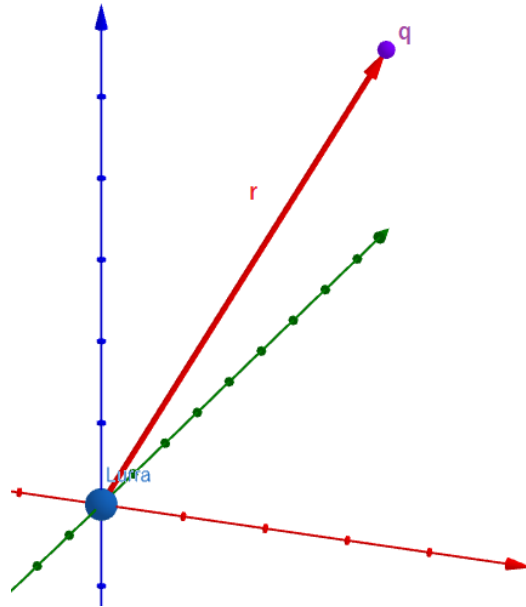
$$q_0 = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix}, \quad \vec{v}_0 = \begin{bmatrix} \dot{x}_0 \\ \dot{y}_0 \\ \dot{z}_0 \end{bmatrix}$$

Hau jakinda, beti suposatuko dugu Lurraren zentroa $(0, 0, 0)$ puntuan egongo dela, beraz zentrotik q_0 puntuarekiko erradioa erraz kalkula dezakegu: (ikusi [3.2](#) irudia)

$$\text{Lurraren zentroa} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

izanda,

$$r = \|q\| \quad (3.32)$$



3.2 Irudia: Lurraren zentrotik satelitera sortzen den erradioa: $Lurra = [0, 0, 0] \rightarrow r = \|q\|$

Orain, $V(q)$ funtzioa definituko dugu:

$$V(q) = \frac{C}{2r^3}(3\sin^2\theta - 1) \quad (3.33)$$

non,

$$\sin\theta = \frac{q_z}{r}, \quad C = J_2\mu R_e^2 \quad (3.34)$$

Datu hauek izanda, q eta v -ren deribatuak lortzeko hau izan behar dugu kontuan:

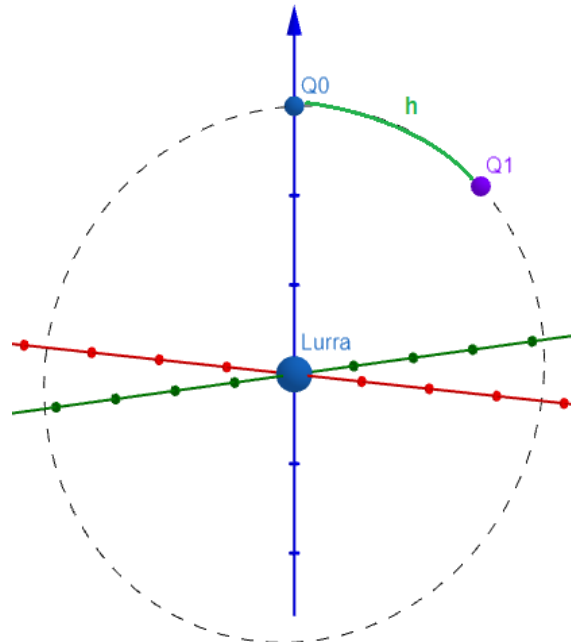
$$\begin{aligned} \frac{d}{dt}q &= rv, & q(0) &= q_0, \\ \frac{d}{dt}v &= -\frac{\mu}{r^2}q - r\nabla V(q), & v(0) &= v_0, \end{aligned}$$

Ekuazio diferentzial hauek definituta, edozein puntu eta abiadura jasota, hauen deribatuak lor ditzakegu.

3.3 Satelitearen ekuazioak eta zenbakizko metodoen konbinaketa

Zenbakizko metodoak definituta izanda eta sateliteen posizioaren eta noranzko bektoreen nahi adina deribatu lortzeko metodoa definituta, dena bateratu dezakegu denboran zehar sateliteen posizioak lortzeko. Hau egiteko, h urrats luzera kontuan izan behar dugu, h geroz eta txikiagoa izanda, errorea murriztu egingo da, zenbakizko metodoetan ikusi dugun bezala, Taylor-metodoa erabiliz, errore globala $O(h^d)$ izango da, d ordena izanik, RK4 erabiliz, $O(h^4)$, eta Ralston erabiliz, $O(h^2)$, alegia.

Azaldu ditugun hiru metodoetako edozein aplikatu dezakegu (eta azaldu ez ditugunak ere bai) denboran zehar posizio eta abiadura bat emanik, hurrengo posizio eta abiadura lortzeko (ikusi 3.3 irudia).



3.3 Irudia: q_0 puntu bat izanda, q_1 puntua lor dezakegu. Puntu hau h urrats luzerarekiko menpekoa izango da.

3.3.1 Zenbakizko metodoen kudeaketa

Zenbakizko metodoak kudeatzeko 3 funtzio nagusi programatu dira, bakoitza dagokion zenbakizko metodoa inplementatzeko. RK4 eta Ralston metodoek honako hau jasotzen dute: `double[]` hiru elementuko lista bi, q eta v bektoreak kudeatzeko, eta h `double` parametro bat urrats-luzerarentzako. Funtzio hauek 3.1.2 eta 3.1.3 ataletan ikusi bezala ekuazio diferentzialen kalkulua egiten duen funtzio bati egiten diote dei, funtzio honek q eta v bektoreak jasotzen ditu eta 3.2 atalean ikusitako kalkuluak egiten ditu. Kodea sinpleago izateko, `double[]` bektoreen arteko eragiketak egiten dituzten funtzio laguntzaileak sortu dira, funtzio hauek, bektoreetako elementuen arteko batuketak eta kenketak egiten dituzte eta konstante bategatik biderkatu edo zatitzen dituzte bektoreko elementuek.

Bestalde, Taylor-metodoaren kudeaketa era ezberdin batean egin behar da. Taylor-metodoan denbora guztian serieekiko kalkuluak egin behar dira. 3.1.1 atalean ikusi dugun bezala, Taylorren hedapena kalkulatzeko ordena + 1 elementuko polinomio bat kalkulatu dugu. Kalkuluak era egokian egiteko, q eta v -ren elementu bakoitzeko polinomio bat kudeatuko dugu, hau da, q_x kalkulatzeko, `double[]` lista bat izango dugu itxura hau izango duena hasieran: $q_x = [q_{0x}, 0h, 0h^2, \dots, 0h^d]$. Beraz, horrelako 6 aldagai erabiliko ditugu: q_x , q_y , q_z , v_x , v_y eta v_z .

Hau definiturik, ekuazio diferentzialen kalkulua egiten duen funtzioari dei egin behar zaio, baina ez da beste zenbakizko metodoek erabiltzen duten funtzio berdina, funtzio honek 6 `double[]` lista jasoko baititu, lista bat q eta v -ren elementu bakoitzeko eta kalkulatu behar duen ordena kudeatuko duen integer bat. ODEaren balioztapenak eta bertan ageri den $V(q)$ funtzioaren gradientearen kalkuluak sarrera gisa serieak jasotzen dituzte, eta 3.2 atalean ikusitako kalkuluak egin behar dituzte sarrerako serie horiekin, hau da, kalkulu horiek polinomio edo serie arteko eragiketak dira. Beraz, serieen edo polinomioen arteko eragiketak egiten dituzten funtzio laguntzaileak definitu behar izan dira:

Polinomioen arteko batuketa:

a eta b batzen diren polinomioak izanik,

$$c_k = a_k + b_k \quad (3.35)$$

non, $k = 0, 1, 2, \dots, d$

Polinomioen arteko kenketa:

a eta b kentzen diren polinomioak izanik,

$$c_k = a_k - b_k \quad (3.36)$$

non, $k = 0, 1, 2, \dots, d$

Polinomioen arteko biderketa:

a eta b bidertzen diren polinomioak izanik,

$$c_k = \sum_{j=0}^k a_j * b_{k-j} \quad (3.37)$$

non, $k = 0, 1, 2, \dots, d$

Polinomioen arteko zatiketa:

a eta b zatitzen diren polinomioak izanik,

$$c_k = \frac{1}{b_0} \left(a_k - \sum_{j=0}^{k-1} c_j b_{k-j} \right) \quad (3.38)$$

non, $k = 0, 1, 2, \dots, d$

Polinomio baten berreketa:

a polinomioa izanik eta b berretzailea,

$$c_0 = a_0^b \quad c_k = \frac{\sum_{j=0}^{k-1} ((b * (k - j)) - j) * a_{k-j} * c_j}{k * a_0} \quad (3.39)$$

non, $k = 1, 2, \dots, d$

Polinomio baten biderketa konstante bategatik:

a polinomioa izanik eta b konstantea,

$$c_k = a_k * b \quad (3.40)$$

non, $k = 0, 1, 2, \dots, d$

4. KAPITULUA

Proiektuaren garapena

Eduki matematikoa teorikoki azaldu ondoren, kapitulu honetan aplikazioa bera garatzeko prozesua ikusiko da: diseinua, inplementazioa, probak eta baldintzak.

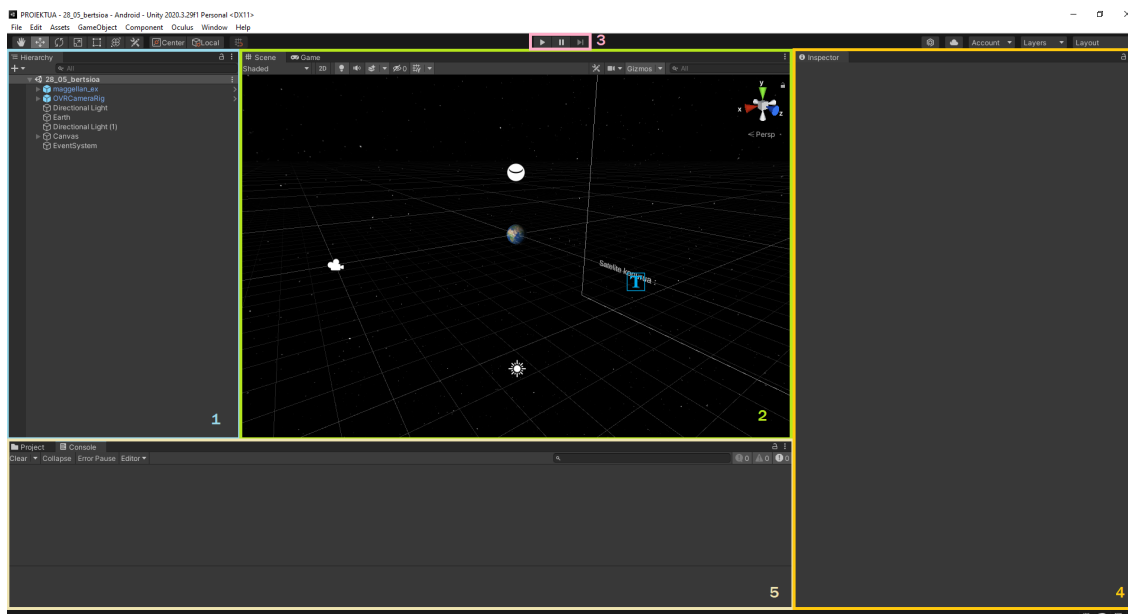
4.1 Unity ingurunea

Unity erabiltzeko egin beharreko lehen gauza, hau instalatzea izan da. Horretarako bere webgunera jo dezakegu.

<https://unity.com/es/download>

Web orrialde horretatik *Unity Hub* aplikazioa deskargatuko dugu, aplikazio honek proiektuak kudeatzeko erosotasunak eskaintzen dizkigu eta nahi dugun Unity editorearen bertsioa deskarga dezakegu bertatik. Proiektu hau 2020.3.29f editore bertsioan garatuta dago. Softwarea prestatzeko azken gauza lizentzia aukeratzea da, nire kasuan erabilpen libreko lizentzia aukeratu dut.

Unity softwarea prest dugunean erabiltzeko, proiektu berri bat sortzeko gai gara.



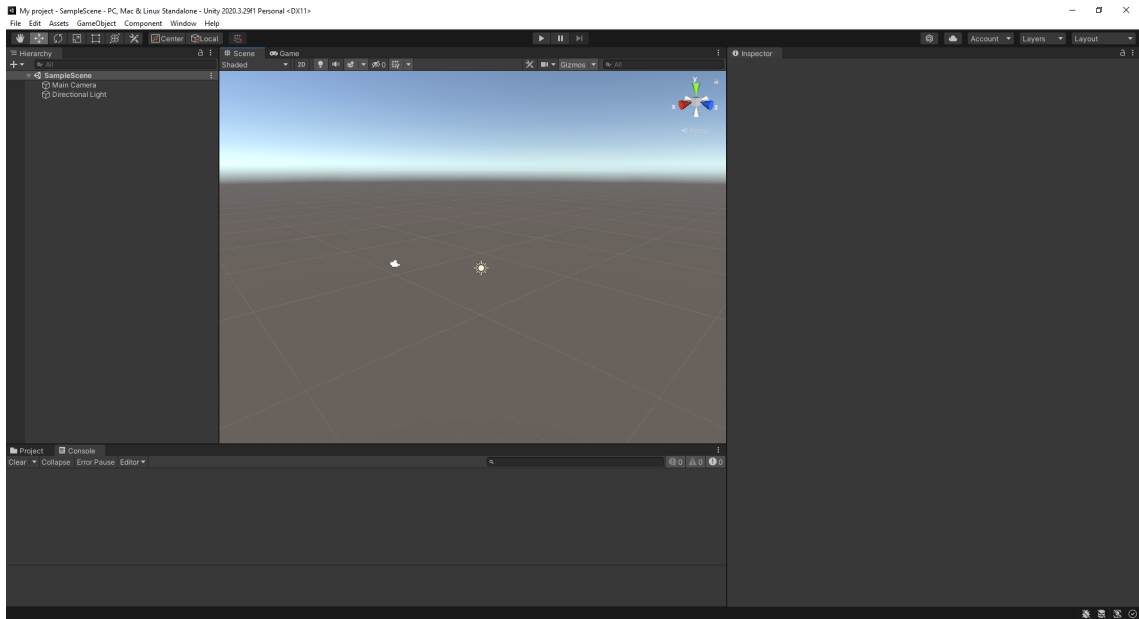
4.1 Irudia: Unity eszena. Sekzio desberdinetan banatuta azalpenerako.

Honela funtzionatzen du Unity editorearen interfazeak: (ikusi 4.1 irudia)

1. puntuan, ezker aldean, eszenan aurki ditzakegun objektuak, argiak, kamerak eta horrelakoak daude kokatuta.
2. puntuan, eszenaren bistareketa daukagu. Objektuak aukeratu eta mugitu ditzakegu hemendik. Play botoia sakatzean *Game* pantailara aldatuko da eta kamerek erakusten dutena bistaratuko da hemen.
3. puntuan, Play eta Pause botoiak ditugu, bistaraketa hasi, gelditu eta amaitzeko.
4. puntuan, pantailaren eskuinaldean, aukeratutako objektuaren informazioa aurki dezakegu. Hemen posizioa, eskala, biraketa, erantsitako scriptak, etab. ikusi eta aldatu ahalko ditugu.
5. puntuan, azpialdean, proiektuaren direktorioetan mugitzeko eta exekuzio kontsola aurki ditzakegu.

4.2 Eszena

4.2 irudian ikus daitekeen bezala, eszena honetatik hasi behar gara proiektua sortzen. Eszena honetan 5 elementu nagusitan banatu dezakegu: atzealdea, Lurra, satelitea, argiz-



4.2 Irudia: Lehenetsita datorren Unity eszena.

tapena eta kamerak.

Lurra eta satelliteak kargatzeko, Unityren *Asset Store*a erabiliko dugu. Unityk, web orrialde honetatik, beste erabiltzaileek igotako objektuak edo/eta eszenak inporta ditzakegu.

<https://assetstore.unity.com/>

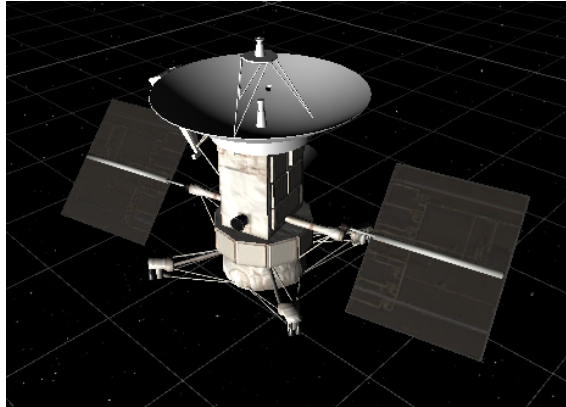
Proiektu honen kasuan, doako *asset*ak erabiliko ditugu. Lurraren *asset*aren kasuan atzealdearentzat ongi zetozen material eta testurak kargatu ditugu, beraz, deskarga honekin atzealdea eta Lurra sortu ditugu. *Asset Store*tik kargatutako eszena guztia ez nuenez erabili nahi, Lurra sortzeko esfera motatako objektu bat sortu dut eta honi inportatutako objektuaren materiala eta normalen mapa erantsi dizkiot.

Atzealdea sortzeko, *SkyBox Material* materiala eszenara arrastatu dut, gero material honi *StarTexture 2D* testurak erantsi behar izan dizkiot ardatz guztietan, honela, HDR zuzen bat sortu dut eta eszena espazioan kokatzea lortu dut.

Satelitearen kasuan, objektu bezala kargatu dut eszenan. Satellite objektu hau, *Magellan_ex* izena duena objektu ume askok osatzen dute. (Ikusi 4.3 irudia)

Eszena argiztatzeko bi argi direkzional kokatu ditut, lehenengoa eguzkiaren argiztatzea simulatzeko eta bigarrena kontrako noranzkoan Lurraren atzeko aldea guztiz ilunduta ez egoteko.

Erabiltzailearen ikuspuntua sortzeko, VR betaurrekoak onartzen dituen kamera bat kokatu



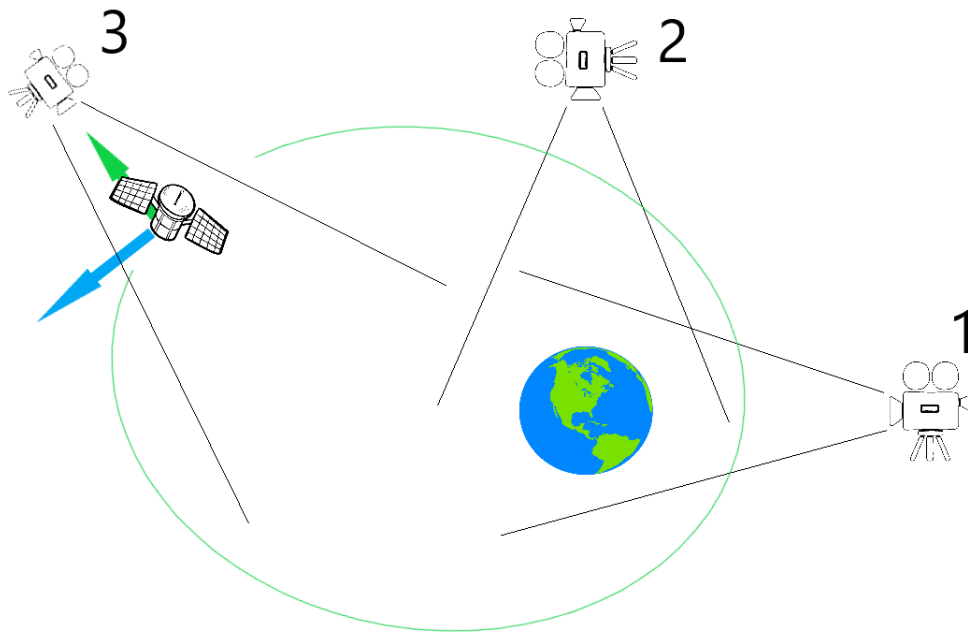
4.3 Irudia: Magellan_ex satelite objektua Unity eszenan.

dut. Kamera hau honela funtzionatuko du: (ikusi [4.4](#) irudia)

1. Lehenetsitako egoeran, kamera libre bat bezala funtzionatzen du, saguarekin biratuko da eta *WASD* teklekin edo Oculusen ezkerreko aginte-palankarekin mugitu daiteke.
2. Bigarren egoera kamera finko bat bezala tratatzen da, egoera honetan gure kamera Lurraren Ipar Poloaren gainean egongo da eta ezin izango da mugitu ezta biratu ere, baina VR betaurrekoekin buruaren norabidea jarraituko du oraindik.
3. Azkenik, satelitei jarraitzeko egoera dugu. Egoera honetan kamera satelitearen gainean kokatuta egongo da, satelitearen *Y* ardatz lokalean, alegia. Defektuz, sateliteari begira dagoen ikuspuntua izango dugu, Oculus gailuekin buruaren mugimendua jarraituko du gure kamerak baina ordenagailuan exekutatzekoan kameraren mugimendua aktibatzeko *X* tekla sakatu beharko dugu eta saguarekin biratu egingo da kamera. Kamera sateliteruntz begira finkatzeko *X* tekla sakatu beharko dugu berriz.

Kameraren egoeren artean aldatzeko, *K* tekla edo Oculusen agintearen *Y* tekla sakatu beharko ditugu.

Azkenik, erabiltzaileak aplikazioa exekutatzeko duenerako interfaze bat sortu dut. Interfaze hau, *Canvas* motatako objektu batekin kudeatzen da, objektu honek testu etiketak gordetzen ditu. Testu etiketa hauen bidez, erabiltzaileak, satelite kopurua, aplikazioa exekutatzeko daraman denbora eta FPS kopurua ikusi ditzake denbora guztian.



4.4 Irudia: Kameren errepresentazioa. 1 kamera librea, 2 kamera finkoa eta 3 sateliteen kamera.

4.3 Inplementazioa eta Scriptak

Objektuak, kamerak eta argiak eszenan kokaturik izanda funtzionamenduaren inplementazioa falta zaigu egitea. Unityn, programatzeko C# hizkuntzako scriptak erabiltzen dira. Scriptak objektuen osagai bezala gehitzen dira [Technologies, 2022].

Aplikazio honetan, Unityren objektuen mugimendu eta biraketen gainean lan egin behar dugu batez ere, honetarako *Unity Engine* liburutegia erabiliko dugu. Hauek dira liburutegi honetatik erabiliko ditugun klase nagusiak:

1. **Vector3:** 3 floatekin osatutako array baten funtzioa betetzen du. Hurrengo puntuko *Transform* klaseko ezaugarriak aldatzeko erabiliko dugu.
2. **Transform:** Objektuen kontrola egiteko. Umeen kontrola, posizioa, eskala eta biraketa aldatzea, etab. ahalbidetzen digu.
3. **GameObject:** Objektuei buruzko informazioa ematen du. Gure proiektuan objektuak aurkitzeko erabiliko dugu.

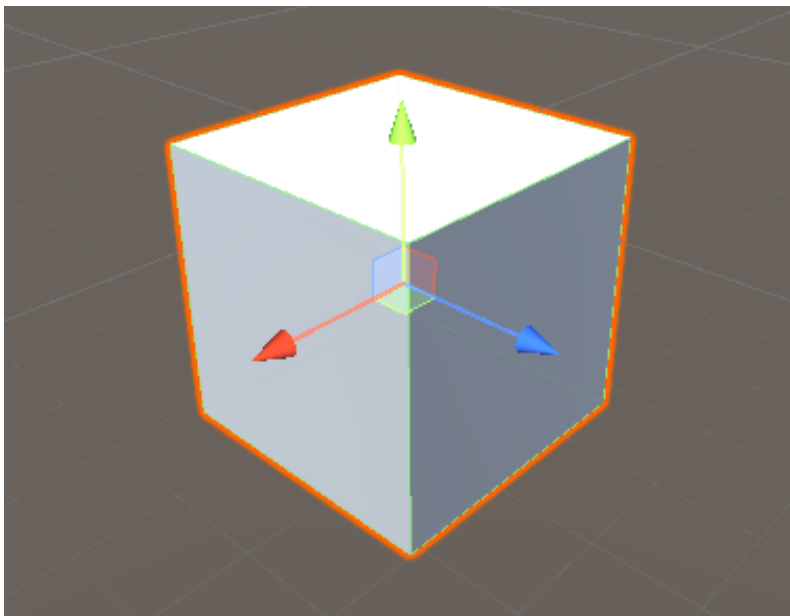
4. **Quaternion:** Biraketa errepresentatzeko erabiltzen da. Gure sateliteak v bektorearen noranzkoa izateko erabiliko dugu.

Objektuak Unityn

Sateliteak transformatzeko, Unityk objektuak nola tratatzen dituen ulertu behar dugu. Objektu bakoitzak hiru ardatz lokal ditu: X , Y eta Z . *Transform* liburutegiak ardatz hauek right, up eta forward bezala ezagutzen ditu eta Unityk kolore ezberdinekin irudikatzen ditu, gorria, berdea eta urdina, hurrenez hurren. (Ikusi 4.5 irudia)

Lehen esan bezala, objektuak transformatzeko *transform* liburutegia erabiliko dugu, posizioa aldatzeko *transform.position* atributua aldatuko dugu, *Vector3* parametro bat sartuz.

Objektuak biratzeko, aldiz, bi funtzio eta parametro bat erabiliko ditugu. *transform.LookAt()* funtzioaren bitartez, sartzen diogun *Vector3* puntuari begira jarriko dugu objektua. *transform.Rotate()* funtzioari zein ardatzetan eta zein abiaduratan biratzeko eman behar dizkiogu. Objektua biratzeko azken metodoa noranzko bektore bat emanik da, horretarako, *transform.rotation = Quaternion.LookRotation()* kodea erabiliko dugu. Bi metodo hauek right bektore bat erabili behar dugu ere up bektorearen kalkulu zuzena egiteko.



4.5 Irudia: Kubo bat Unityn.

Scriptak

Guztira 7 script sortu ditut aplikazioa kudeatzeko:

Biraketa.cs: Lurra biratzeko *transform.Rotate()* funtzioa erabiliz.

Sateliteak.cs: Lurrari erantsiko diogu script hau. q eta v parametroak ditu, hasieran parametro hauek gure lehen satelitearen posizioa eta abiadura izango dira. Zenbakizko teklatuko + tekla edo Oculusen agintearen X teklak sakatzean, satelite berri bat instantziatuko duen funtzio bat exekutatzeko da. Funtzio horrek, switch baten bidez hurrengo satelitearen posizio eta abiadura aldatuko ditu satelite desberdinak jaurtitzeko eta boolean baten bidez infinitu satelite jaurti ahal ditugu, hau bakarrik neurketa probetarako erabiliko da. Script honek exekuzioan dauden satelite kopurua ere gordeko du eta canvasean bistaratuko du.

Orbita.cs: Script honetan 3. kapituluan ikusi ditugun zenbakizko metodoak kudeatzen dira sateliteen orbita bistartzeko. Script hau, sateliteari erantsita dago, satelitea instantziatzen denean, lehen frameean, *Sateliteak.cs* scriptetik q eta v parametroak jasotzen ditu satelitearen hasierako posizioa eta abiadura hasieratzeko. Balio hauek izanda, frame bakoitzean, aukeratu dugun zenbakizko metodoa kalkulatu eta posizioa eta noranzkoa aldatzen dio sateliteari. Azkenik, 4.5 irudian ikus dezakegun 3. kamera mota kudeatzen du, lehen esan bezala, egoera honetan kamera dagokion satelitearen gainean kokatzen du eta X tekla sakatzean kamera saguarekin biratzea ahalbidetzen du.

KamerenKontrola.cs: Script honetan, lehen erreferentziatu dugun 4.5 irudian ikus ditzakegun 1. eta 2. kamera motak kudeatzen dira. Kamera egoeren artean mugitzean, kamera mota horietako baten txanda iristen denean dagokion posizio eta noranzkoan kokatzen du kamera. Zenbat satelite dauden jakiteko, *Sateliteak.cs* scriptetik lortzen du informazioa.

KamerenMugimendua.cs: Script hau kamerari dago erantsita eta egiten duena hau da: ordenagailuan exekutatu ari bagara aplikazioa, *WASD* tekletan mugitzea eta saguarekin biratzea ahalbidetzen du. Oculus gailuekin, ezker aginte-palankarekin mugituko da.

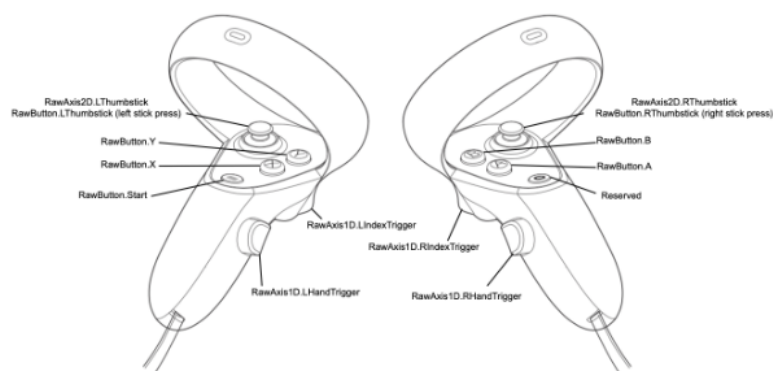
KamerenHelburua.cs: Script hau 3. kamera mota satelitei begira egoteko erabiltzen da. Ordenagailuan X tekla sakatzean, saguarekin biratzeko egoera aktibatzen da.

FPSkalkulua.cs: Script honen bidez fotograma bakoitzaren egungo erreproduktzio-tarteak kalkulatu eta aplikazioa exekutatu denbora canvasean bistaratu dira. [programadorclic.com, 2021]

4.4 Errealitate birtualeko betaurrekoak: Oculus Quest 2

Dokumentuan zehar esan dugun bezala, bistaraketa ordenagailuan edo errealitate birtualeko betaurrekoetan egingo da. Proiektu honen kasuan, unibertsitateak utzitako Oculus Quest 2 gailuetan egin izan dira proba guztiak.

Betaurrekoetatik gain, gailu hauek esku bakoitzarentzat aginte bat dute. Aginte bakoitzak hiru botoi, bi trigger eta palanka bat du (Ikusi 4.6 irudia).



4.6 Irudia: Oculus Quest 2 gailuaren aginteak. Botoi bakoitza etiketa propioa du, programatzera-koan identifika ahal izateko. [Oculus, 2022]

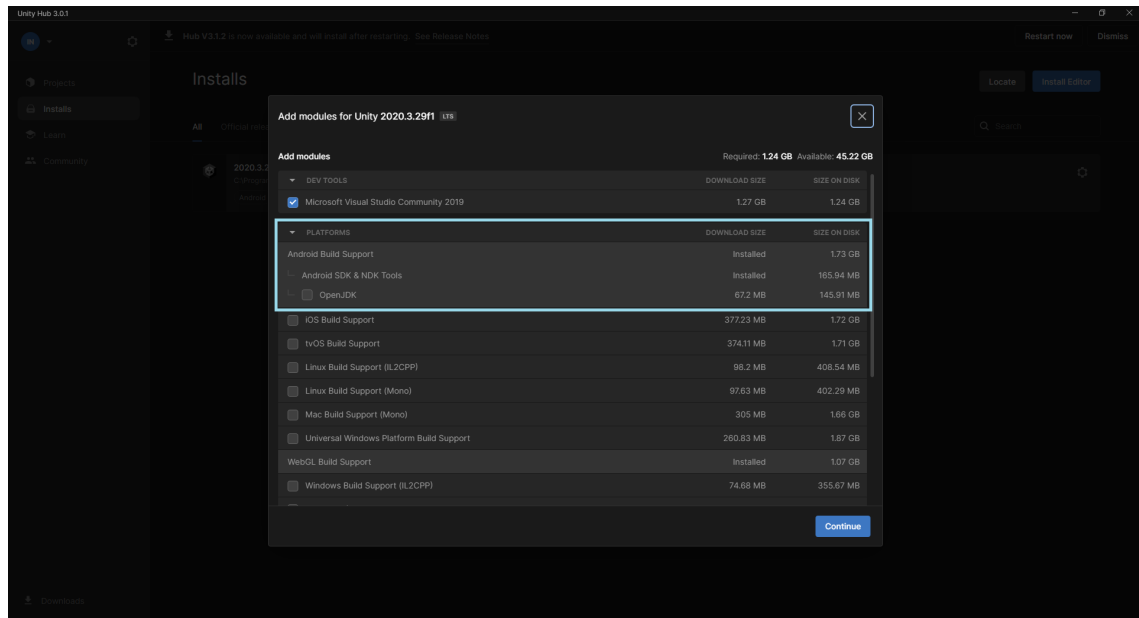
Plangintzan esan dugun bezala, ez ditugu betaurrekoak erabiliko aplikazioa ordenagailuan ondo funtzionatu arte, une horretara heldutakoan hasi gara aplikazioa betaurrekoetan erabiltzeko egokitzen. Hori lortzeko, Unityk eskaintzen dizkigun 3 pakete inportatu behar izan dira. Hona hemen pakete horien izenak: *Oculus Integration*, *Oculus XR Plugin* eta *XR Plugin Management*.

Pakete hauek inportatuta izanda, proiektuari hainbat direktorio gehitu zaizkio, honetako batek *Oculus* izena du. Direktorio honetan, VR betaurrekoak erabiltzeko behar ditugun kamerak aurki ditzakegu (*Oculus>VR>Prefabs*). Kamera honek *OVRCameraRig* izena du eta gure kamera arruntarekin aldatuko dugu, ondoren, kamerak dakarren "esku" bakoitzean *OVRControllerPrefab* bat gehituko diogu.

Momentu honetan, egin beharreko gauza bakarra, Oculusetan exekutatzeko ingurunea konfiguratzea da, horretarako, Unity Hub bidez, gure editore bertsioan (2020.3.29f) *Android Build Studio* plataforma gehitu behar dugu (ikusi 4.7). Ondoren, proiektuaren konfigurazioan, *XR Plugin Management* atalean, Oculus Plugin-a gehitu behar dugu.

Hau guztia eginik, Oculus Quest 2 gailua USB-3 bitartez ordenagailura konektatzeko eta

erabiltzeko prest dugu. Aplikazioa exekutatzeko, Build and Run sakatu behar dugu gure Oculus gailua aukeraturik daukagula.



4.7 Irudia: Unity Hub-en *Android Build Studio* plataforma gehitzeko pantaila.

5. KAPITULUA

Emaitzen azterketa

Proiektua bukatutzat emateko, hainbat proba egingo dira gailuen eta metodoen arteko alderaketak egiteko. Bi proba mota egingo dira: Oculus gailuen gaitasunen azterketa eta erabilitako zenbakizko metodoek duten erroreen azterketa. Azken honekin, metodoen arteko alderaketa egiteko gai izango gara.

Honetaz gain, sortu ditugun 5 satellite desberdinen orbiten bistaraketa Unity ingurunetik erakutsiko ditugu kapitulu honetan.

5.1 Oculus gailuen gaitasunen azterketa

Proba honetan, Oculus gailuen gaitasuna neurtzeko, satellite kopuru erraldoi bat abiatuko dugu. Hori lortzeko kodea apur bat aldatuko dugu eta ordenagailuan proba berak egingo ditugu alderaketa egin ahal izateko. Zenbat frame sor ditzakeen segundo bakoitzean (FPS) eta frame bat sortzeko behar den denbora, kalkulu gaitasunaren parekoa da datu hau, (milisegundotan) neurtuko dugu, Oculus gailuen kasuan, monitorea beharrean be-turrekoen pantailak izango lirateke. Satellite batekin, 5ekin, 10ekin, 50ekin, 100ekin eta 200ekin egingo dira probak.

Ordenagailuan:

Satelite kopurua	FPS	ms
1	321	3.1
5	275	3.5
10	260	3.9
50	175	5.7
100	90	10.5
200	45	23.3

5.1 Taula: Ordenagailuan jasotako FPS eta ms emaitzak.

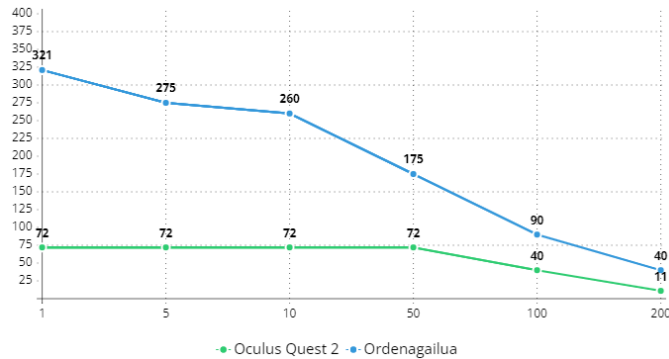
Oculus Quest 2 gailuetan:

Satelite kopurua	FPS	ms
1	72	13.9
5	72	13.9
10	72	13.9
50	72	13.9
100	40	25.3
200	11	88.8

5.2 Taula: Oculus Quest 2 gailuetan jasotako FPS eta ms emaitzak.

5.2 taulan ikus daitekenez, gailuen gaitasuna 72 FPS eta 13.9ms-tara mugatuta dago, alegia, ezin du hori baino sarriago freskatu pantaila, beraz, nahiz eta kalkuluak azkarrago egin ahal izan ez ditu aldaketak pantailaratzen. Horregatik kalkuluak denbora hori baino gutxiago behar duen bitartean 72 FPS mantentzea lortzen du. Baina kalkuluek denbora gehiago behar duten momentutik aurrera frame kopurua murrizten hasten da, beraz, hortik aurrerako datuei begiratu behar zaie. Ordenagailuan, aldiz, freskatze gaitasuna askoz hobe da, eta 321 FPS ere lortzen ditu. Kalkulu gaitasuna ere askoz hobe da, eta 100 sateliterekin oraindik, betaurrekoen hasierako gaitasuna baino ahalmen gehiago du. 200 sateliterekin kalkuluak egiteko behar den denbora Oculusetan gutxi gorabehera 3 aldiz handiagoa dela ikus daiteke (ikusi 5.1 taula).

Informazio hau, kamera geldirik izanda eta objektu denak enfokatzerakoan hartu dira, objektu denak enfokatuko ez balira, segunduroko frame kopurua eta frame bat sortzeko behar den denbora handituko lirateke, baina kamera mugitzen saiatuko bagina, datu hauek txikituko lirateke.



5.1 Irudia: Ordenagailuen eta Oculus Quest 2 gailuen FPS alderaketa satelliteak gehitu ahala.

5.2 Zenbakizko metodoen erroreen analisia

2. kapituluaren ikus daitekeen bezala, proiektu honetan 3 zenbakizko metodo inplementatu dira satelliteen ekuazio diferentzial arruntaren ebazpena egiteko. Metodo horiek hurrengokoak dira: Taylor-metodoa, lau ordenako Runge-Kutta metodoa (RK4) eta Ralston-metodoa (Runge-Kutta metodoen familiakoa).

Dakigunez, metodo hauek ez dira zehatzak eta bakoitzak errore global bat du metodoaren ordenaren eta urrats-luzeraren arabera. Atal honetan, h urrats-luzera berdinarekin abiatuz, urrats kopuru berdina kalkulatzeko h handituz metodo bakoitzak sortzen duen errore erlatiboaren kalkulua egingo da, honela, metodo desberdinetan errorea zein abiaduratan handitzen den konparatu ahal izango dugu.

Aukeratutako hasierako h -a 0.005ekoa izango da, eta $32 \cdot K$ urrats egiterakoan, $K = 100$ izanda (gutxi-gorabehera 18 orbita Lurretik gertuen dagoen satellitearentzat), lortzen ditugun q eta v bektoreak kontuan hartuz hasiko gara errorea kalkulatzeko. Gero h bikoiztuko dugu, satelliteak posizio "berdinean" bukatzeko, urrats kopurua $16 \cdot K$ -ra murriztuko dugu, izan ere, urrats luzera bikoiztu badugu, denboran $2 \cdot h \cdot 16 \cdot K = 32 \cdot K$ unera iritsiko da. Ondoren, $4h$ -ekin eta $8 \cdot K$ urratsekin egingo dugu, etab.

Kontuan izango dugun satellitea Lurretik gertuen orbitatzen dagoen satellitea izango da, satellite honek sortuko baitu errorerik handiena, Lurretik geroz eta gertuago egonda abiadura handiagoa delako.

Ikusiko ditugun tauletan agertzen den informazioa errore erlatiboa da. Errore hau kalkulatzeko h urrats-luzerarekin ateratako posizio eta abiadurak, q_h eta v_h alegia, h handituz joan ahala lortutako posizio eta abiadurarekiko diferentzia kalkulatu dugu. Balio hauek kilometro eta kilometro/segundo unitatetan egongo dira, baina errore erlatiboa lortzeko,

balio horiek posizio edo abiaduraren normarengatik zatitu beharko ditugu, hau da:

$$\text{posizioaren errore erlatiboa} = \frac{\|q_h - q\|}{\|q_h\|}$$

$$\text{abiaduraren errore erlatiboa} = \frac{\|v_h - v\|}{\|v_h\|}$$

5.2.1 Taylor-metodoa

Taylor-metodoa aztertzeko, $h = 0.005$ izateaz gain, d ordena aukeratu behar dugu, kasu honetan, $d = 10$ izatea aukeratu da.

Posizioarekiko errorea (q):

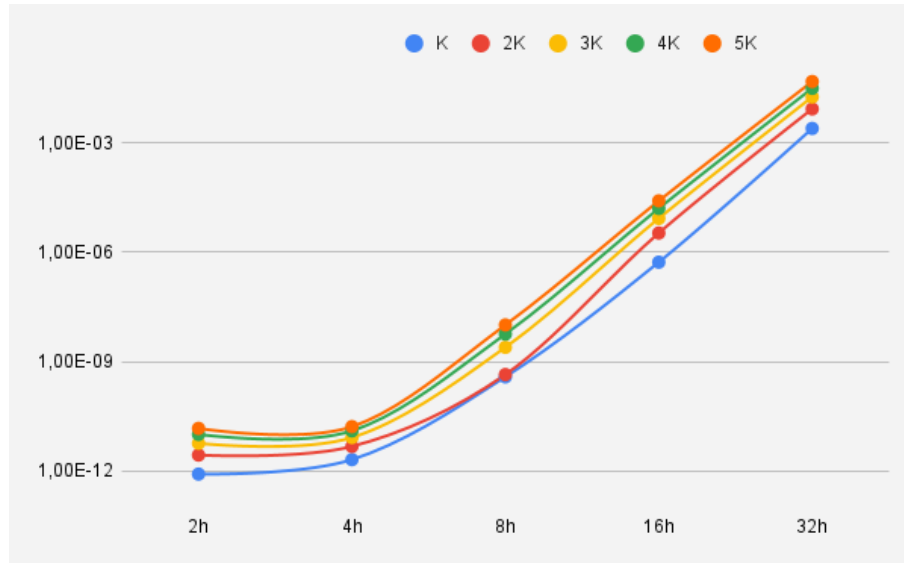
	1K	2K	3K	4K	5K
2h	$8.28e-13$	$2.75e-12$	$5.78e-12$	$1.01e-11$	$1.48e-11$
4h	$2.07e-12$	$4.75e-12$	$8.33e-12$	$1.27e-11$	$1.66e-11$
8h	$3.84e-10$	$4.44e-10$	$2.48e-09$	$5.72e-09$	$1.02e-08$
16h	$5.34e-07$	$3.37e-06$	$8.49e-06$	$1.59e-05$	$2.56e-05$
32h	0.00248	0.00852	0.018117	0.031263	0.047957

5.3 Taula: Taylor-metodoaren posizioarekiko errore erlatiboaren taula.

Abiadurarekiko errorea (v):

	1K	2K	3K	4K	5K
2h	$8.27e-13$	$2.75e-12$	$5.78e-12$	$1.02e-11$	$1.48e-11$
4h	$2.06e-12$	$4.74e-12$	$8.32e-12$	$1.27e-11$	$1.65e-11$
8h	$3.79e-10$	$4.53e-10$	$2.49e-09$	$5.74e-09$	$1.02e-08$
16h	$5.37e-07$	$3.37e-06$	$8.50e-06$	$1.59e-05$	$2.57e-05$
32h	0.00248	0.00852	0.018105	0.031263	0.047902

5.4 Taula: Taylor-metodoaren abiadurarekiko errore erlatiboaren taula.



5.2 Irudia: K pauso emandakoan, h handitu ahala jaso diren posizioaren errore erlatiboaren errepresentazioa grafiko bidez.

Ikus dezakegunez, bi tauletan lortutako balioak oso antzekoak dira, honi buruz informazio gehiago lortzeko errore absolutuen balioak ikusiko ditugu orain:

Posizioarekiko errorea kilometrotan (q):

	1K	2K	3K	4K	5K
2h	$5.947e-09$	$1.977e-08$	$4.151e-08$	$7.301e-08$	$1.064e-07$
4h	$1.485e-08$	$3.41e-08$	$5.971e-08$	$9.11e-08$	$1.189e-07$
8h	$2.756e-06$	$3.185e-06$	$1.778e-05$	$4.104e-05$	$7.297e-05$
16h	0.0038363	0.024145	0.0609463	0.114223	0.183967
32h	17.826793	61.114394	129.909525	224.164809	343.845497

5.5 Taula: Taylor-metodoaren posizioarekiko errore absolutuaren taula.

Abiadurarekiko errorea kilometro/segundotan (v):

	1K	2K	3K	4K	5K
2h	$6.17e-12$	$2.053e-11$	$4.312e-11$	$7.582e-11$	$1.107e-10$
4h	$1.539e-11$	$3.538e-11$	$6.202e-11$	$9.447e-11$	$1.234e-10$
8h	$2.828e-09$	$3.376e-09$	$1.859e-08$	$4.28e-08$	$7.601e-08$
16h	$4.006e-06$	$2.516e-05$	$6.345e-05$	0.000119	0.000191
32h	0.018512	0.063539	0.135036	0.232939	0.357177

5.6 Taula: Taylor-metodoaren abiadurarekiko errore absolutuaren taula.

Ikus dezakegunez, errore erlatiboak oso antzekoak izan arren, errore absolutuak desberdinak dira. Honela ikus dezakegu ze zehaztasunez egin diren orbitak abiaduraren eta posizioaren errore erlatiboak ia berdinak direla.

Taylor-metodoarekin lortu ditugun emaitzak oso positiboak direla ondoriozta dezakegu, errore erlatiboa ez da handia $32h$ urrats-luzera izan arte, 0.16 -ko urrats-luzera, alegia, urrats-luzera handiegi bat kalkulu zehatzak egiteko. h horrek $5K$ urrats egitean sortzen duen errorea posizioarekiko 343.845km -koa da, eta datuetan ikus daitekeen tendentzia kontuan izanda errore hau handiagotuko dela ikus dezakegu, beraz, $32h$ ez da urrats-luzera zuzen bat kalkulu zehatzak egiteko.

5.2.2 RK4

RK4 metodoa 4 ordenako metodo bat da, beraz, espero ditugun emaitzak, Taylor-metodoarekin lortu ditugun emaitzak baino txarragoak izango dira.

Urrats-luzera berdina erabiliz metodo honen errorea hain handia izan da $16h$ erabiltzean ezin izan direla datu horiek lortu, satelitearen erorketa $(0,0,0)$ puntura azkarregia delako, puntu horretara heltzean, sateliteak errore ematen du eta aplikaziotik desagertu egiten da, beraz, honekin bakarrik baliozta dezakegu metodo hau Taylor-metodoa bezain ona ez dela. Hona hemen jasotako erroreak:

Posizioarekiko errorea (q):

	1K	2K	3K	4K	5K
2h	0.000565	0.001763	0.003795	0.006593	0.010157
4h	0.014535	0.054269	0.119174	0.209076	0.323508
8h	0.426176	1.479137	1.84343	0.580565	1.420557

5.7 Taula: RK4 metodoaren posizioarekiko errore erlatiboaren taula.

Abiadurarekiko errorea (v):

	1K	2K	3K	4K	5K
2h	0.000498	0.001764	0.003795	0.006592	0.010153
4h	0.014537	0.054307	0.119292	0.209337	0.323992
8h	0.430592	1.509349	1.901965	0.604431	1.499405

5.8 Taula: RK4 metodoaren abiadurarekiko errore erlatiboaren taula.

Ikus daitekenez, RK4 metodoarekin, errore-erlatibo handia lortu dugu erabili dugun h urrats-luzerarekin. Metodo hau erabili nahi izango bagenu, h murriztu beharko genuke.

5.2.3 Ralston-metodoa

Ralston-metodoa 2 ordenako metodoa da, beraz, RK4 metodoarekin gertatu zaigun bezala, aurreko bi metodoek baino emaitza txarragoak lortuko ditu metodo honek.

RK4-rekin gertatu zaigun bezala, Ralston-metodoak errore handi bat du h handitzean. Hala ere, RK4-rekin ez bezala, metodo honekin lortutako errorea orbita handitzea eragiten du, horregatik, datu gehiago azter ditzakegu. $32h$ -rekin ordea, hain da handia orbita errorearengatik Unityren eszenak duen limiteetatik ateratzen dela satelitea 1K urrats egin baino lehen.

Posizioarekiko errorea (q):

	1K	2K	3K	4K	5K
2h	0.60557	1.81409	1.46702	1.62121	0.28719
4h	1.71342	2.27148	0.42497	1.01705	1.71095
8h	1.93113	2.44294	4.06722	3.56461	5.19553
16h	13.6011	17.7976	10963.094	144431244000	$1.8928E + 18$

5.9 Taula: Ralston-metodoaren posizioarekiko errore erlatiboaren taula.

Abiadurarekiko errorea (v):

	1K	2K	3K	4K	5K
2h	0.59554	1.75699	1.409904	1.53579	0.26493
4h	1.50981	1.84604	0.21329	0.68633	1.15992
8h	1.19841	1.25756	1.42396	0.89368	1.29407
16h	1.00911	0.68673	0.86818	0.931208	1.04399

5.10 Taula: Ralston-metodoaren abiadurarekiko errore erlatiboaren taula.

Ikus dezakegun bezala, metodo honen errorea handiegia da urrats-luzera bikoiztuz bakarrik, datuak testuinguratzeko, $16h$ -rekin $5K$ urrats egitean satelitearen posizioa ($1.0929639411342e + 22, -1.21666711993782e + 21, 8.28664210372422e + 21$) da, hasierako posizioa $(7170.822, 0, 0)$ izanik. Noizbait metodo hau erabili nahiko bagenu h oso txiki bat erabili beharko genuke.

5.2.4 Metodoen arteko alderaketa

Metodoek sortzen duten errorea jakinda, euren arteko alderaketa egitea oso erraza da. Erraz ikus daiteke Taylor-metodoa hiruretatik hobereena dela askogatik. Metodo hau $h = 0.005$ -etik hasita, $32h$ -rekin kalkulua egin ahal izan duen bakarra izan da, gainera, urrats-luzera horrekin beste metodoek $2h$ -rekin lortzen duten baino emaitza ia hobeagoak lortzen dituela esan genezake.

5.3 Orbiten errorea

Taylor-metodoa metodorik hobereena dela ikusita, hainbat orbita egin eta gero metodo honek sortzen duen errorea aztertuko dugu. Horretarako lehen erabilitako $h = 0.005$ urrats-luzera eta $d = 10$ ordena erabiliko ditugu, baina errore erlatiboa kalkulatu beharrean, errore bakoitza bere unitatean kalkulatu dugu, hau da, q posizioa kilometrotan eta v abiadura kilometro/segundotan.

100 orbita egin ditu gure sateliteak eta errorea kalkulatzeko hasierako posizioa erabili da (ikusi 5.13 taularen lehen ilara). Hauek izan dira lortutako emaitzak:

Posizioarekiko errorea (q):

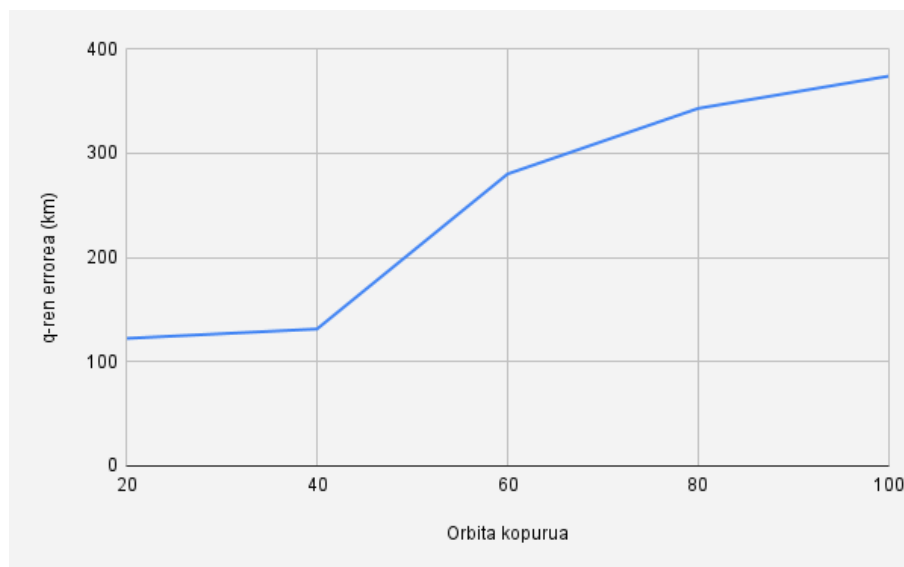
Orbita kopurua	Errorea (km)
20	122.261
40	131.281
60	279.955
80	342.830
100	373.786

5.11 Taula: Orbitekiko posizio erroreen taula.

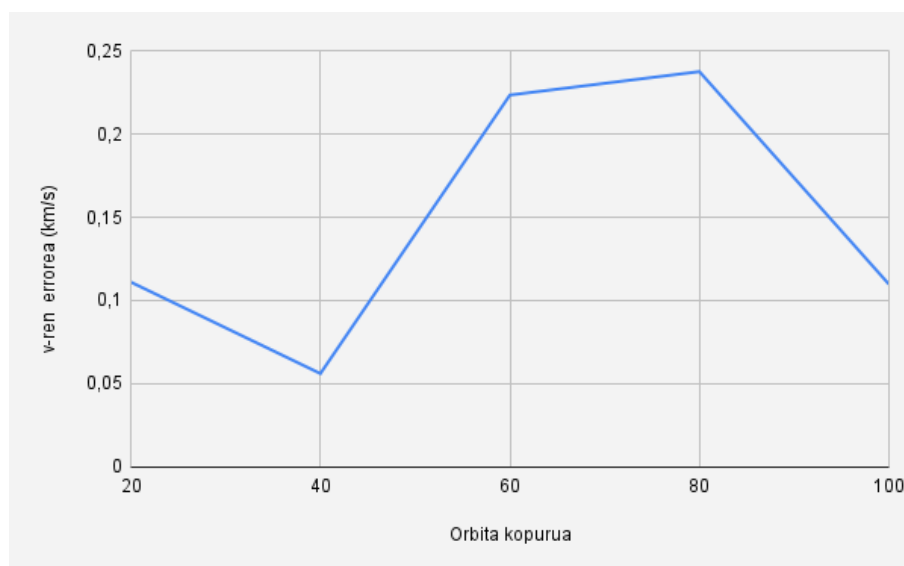
Abiadurarekiko errorea (v):

Orbita kopurua	Errorea (km/s)
20	0.111063
40	0.05616
60	0.223377
80	0.237441
100	0.109795

5.12 Taula: Orbitekiko abiadura erroreen taula.



5.3 Irudia: Orbitekiko posizio erroreak errepresentazioa grafiko bidez.



5.4 Irudia: Orbitekiko abiadura erroreak errepresentazioa grafiko bidez.

Ikus dezakegunez, errorea handia da, horretarako testuinguratuko ditugu datu hauek:

Orbita kopurua	Posizioa (q)	Abiadura (v)	Lurrarekiko distantzia (km)
0	(7170.822, 0, 0)	(0, -1.111576, 7.376071)	7170.822
20	(7169.779, 75.0156, -96.536)	(0.111057, -1.110575, 7.375384)	7170.821
40	(7169.626, 110.181, 71.382)	(-0.056149, -1.112624, 7.375691)	7170.829
60	(7165.377, 145.256, 239.261)	(-0.223273, -1.116946, 7.371847)	7170.842
80	(7162.638, 287.198, -187.037)	(0.237284, -1.1033315, 7.373473)	7170.833
100	(7161.121, 370.259, -50.313)	(0.109716, -1.107409, 7.375838)	7170.863

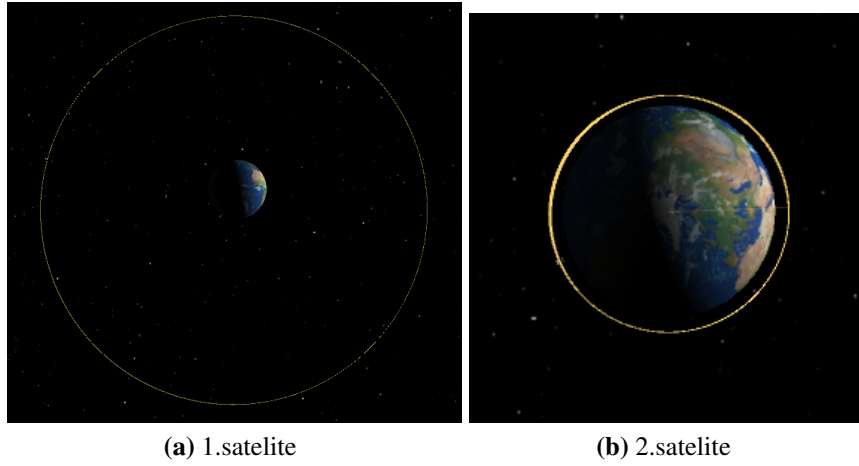
5.13 Taula: Satelitearen posizioa, abiadura eta Lurrarekiko distantzia orbitatzen joan ahala.

Ikus dezakegunez [5.13](#) taulan, satelitea orbitatzen doan heinean Lurrarekiko distantzia era oso murriztuan aldatzen da, baina posizioa eta abiadura errealak hasierakoetatik aldentzen dira.

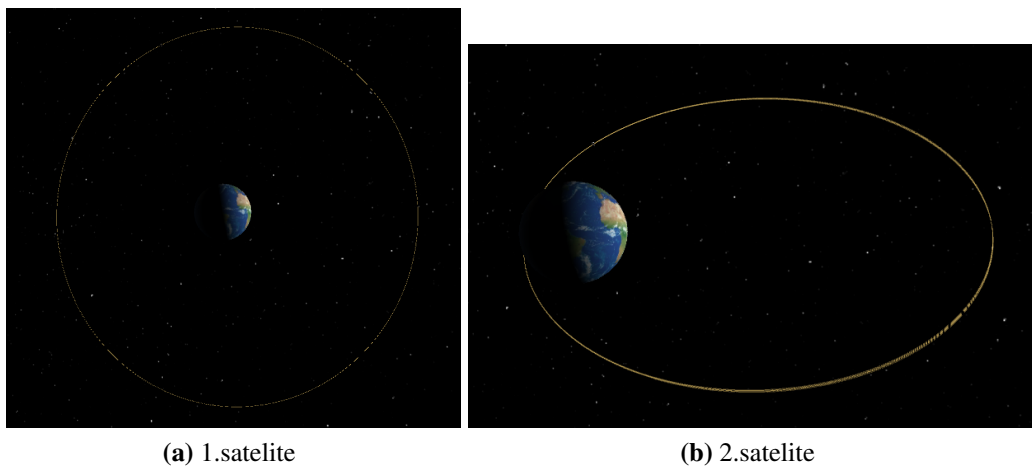
5.4 Sateliteen orbitak

Kapitulu honekin bukatzeko, programatuta dauden 5 satelite desberdinen orbiten bistaraketak ikusiko ditugu irudietan.

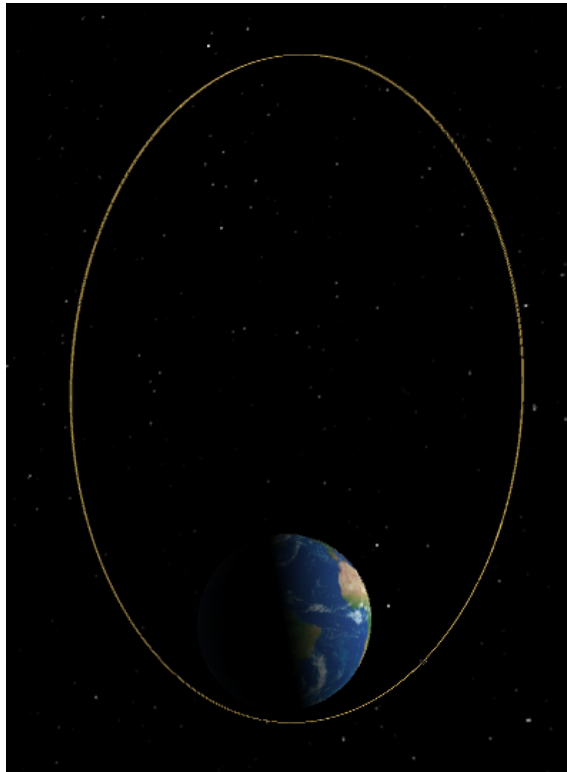
1. Lehenengo satelitea: Orbita zabala eta zirkularra, Lurra beti erdigunean du. (Ikusi [5.5a](#) irudia)
2. Bigarren satelitea: Lurretik oso gertu. Orbita zirkularra. (Ikusi [5.5b](#) irudia)
3. Hirugarren satelitea: Lehenengo orbitaren antzekoa baina hasiera-puntu eta noranzko ezberdinekin. (Ikusi [5.6a](#) irudia)
4. Laugarren satelitea: Orbita oso eliptikoa, Lurra elipsearen alde batean kokatuta du. (Ikusi [5.6b](#) irudia)
5. Bosgarren satelitea: Laugarren satelitea bezala baina kokapen eta biraketa desberdinekin. (Ikusi [5.7](#) irudia)



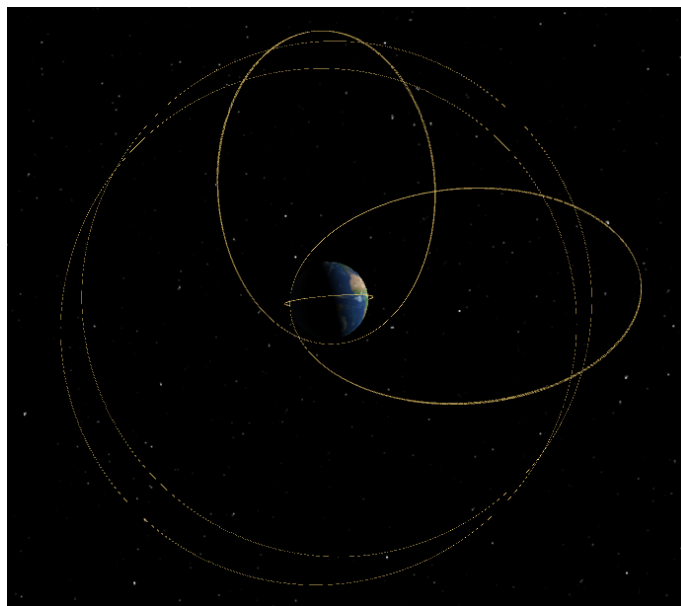
5.5 Irudia: (a) Lehenengo satelitearen orbita. (b) Bigarren satelitearen orbita (goitik ikusita).



5.6 Irudia: (a) Hirugarren satelitearen orbita. (b) Laugarren satelitearen orbita.



5.7 Irudia: Bosgarren satelitearen orbitaren bistaraketa.



5.8 Irudia: Satelite guztien orbiten bistaraketa aldi berean.

6. KAPITULUA

Ondorioak

Lan hau ixteko, kapitulu honetan proiektuaren garapenean zehar ateratako ondorioak bilduko dira. Honetaz gain, gertatutako desbiderapenak eta arazoak deskribatuko dira. Hau guztia kontuan izanda, etorkizunerako hobekuntza batzuk ere proposatuko dira.

Gradu Amaierako Lan honetarako diseinatu ziren helburu denak era arrakastatsuan bete dira eta dokumentu honetan azaldu dira helburu horiek betetzeko jarraitu diren pausuak. Sateliteen ibilbidearen kalkulua eta bistaraketa egiten duen aplikazioa sortu da. Horretarako, Unity ingurunea erabiltzen ikasi behar izan da, niretzat ingurune guztiz berria izanik. Aplikazioa sortzeko, sateliteen ibilbidearen kalkulua egiten duten ekuazio diferentzialak ikasi eta ebazten ikasi behar izan dira, era arrakastatsuan hori ere.

Aplikazioaren laguntzaz, 5. kapitulan jaso diren emaitzak lortu dira. Emaitza hauekin hau ondoriozta dezakegu: Oculus Quest 2 betaurrekoek kalkuluak egiteko eta bistaratzeko gaitasun oso ona daukate. Bistaraketa Oculus gailuekin egiterakoan, erabiltzailearen begientzat ez da nabari izan segundo bakoitzeko frame kopuruaren jaitsiera satelite kopurua oso handia izan arte.

5.2 puntuan erabili dugun urrats-luzera [5.5b](#) irudian ikus dezakegun satelitearen errorea handiegia izan ez dadin aukeratu dugu eta satelite denak urrats-luzera horrekin ari dira kalkuluak egiten. Errealitatea da, beste satelite guztiek ez dutela urrats-luzera hain txikia behar. Beste satelite guztiek behar baino lan gehiago egiten ari dira gure aplikazioan, hau konpontzeko, etorkizunerako hobekuntza bezala, satelite bakoitzari errore minimoa lortzen duen urrats-luzera batekin kalkuluak egiten duen programa bat implementa ahal zaio. Hau aplikazioaren optimizazioari buruzko hobekuntza bat izango litzake.

Bistaraketari dagokionez, aldiz, Taylor-metodoa erabiliz, nahi dugun edozein urrats-luzera erabil dezakegu kalkulua egiteko, alegia, $h = 0.0001$ izanez, sateliteak, bisataraketa egiteko abiadura egoki batean doaz eta kalkuluen errorea minimoa izango da. Urrats-luzera hau handituz gero, errorea oraindik oso txikia izango da baina bistaraketarako, sateliteak oso azkar mugitzen ikusiko genituzke.

Eranskinak

A. ERANSKINA

Programaren biltegia

Nahi izanez gero, hemen erantsita geratzen da programa deskargatu ahal izateko DropBox-eko esteka:

https://www.dropbox.com/s/gflylgfs1dou672/GrAL2022_IkerNarbaizaArrondo.zip?dl=0

B. ERANSKINA

Aplikazioaren bideoak

Oculus gailurik izan ezean eta programaren funtzionamendua era erraz batean ikusteko helburuarekin, YouTube plataformara igo da errealitate birtualeko betaurrekoekin grabatutako bideo bat. Hona hemen esteka:

<https://youtu.be/DntrYPZuVuw>

Bibliografia

- [Concerned Scientists, 2022] Concerned Scientists, U. o. (2022). Ucs satellite database. <https://www.ucsusa.org/resources/satellite-database>.
- [Dobrushkin, 2022] Dobrushkin, V. (2022). Mathematica tutorial under the terms of the gnu general public license (gpl) for the first course. part ii: Autonomous equations.
- [Gravitational, 2022] Gravitational, C. (2022). Gravitational constant. https://en.wikipedia.org/wiki/Gravitational_constant.
- [Gravitational Parameter, 2022] Gravitational Parameter, S. (2022). Standard gravitational parameter. https://en.wikipedia.org/wiki/Standard_gravitational_parameter.
- [Oculus, 2022] Oculus, F. D. (2022). Map controllers: Oculus developers. https://developer.oculus.com/documentation/unity/unity-ovrinput/?locale=es_ES.
- [P. Cracknell and A. Varotsos, 2007] P. Cracknell, A. and A. Varotsos, C. (2007). Editorial and cover: Fifty years after the first artificial satellite: From sputnik 1 to envisat. <https://www.tandfonline.com/doi/abs/10.1080/01431160701347147?journalCode=tres20>.
- [programadorclic.com, 2021] programadorclic.com, C. (2021). <https://programmerclick.com/article/51581941309/>.
- [R. Williams, 2021] R. Williams, D. (2021). Earth fact sheet. <https://nssdc.gsfc.nasa.gov/planetary/factsheet/earthfact.html>.
- [Runge-Kutta, 2022] Runge-Kutta, . (2022). Runge–kutta methods.

[Schaub and Alfriend, 2001] Schaub, H. and Alfriend, K. T. (2001). J2 invariant relative orbits for spacecraft formations - celestial mechanics and dynamical astronomy. <https://link.springer.com/article/10.1023/A:1011161811472>.

[Taylor, 2022] Taylor, S. (2022). Taylor series. https://en.wikipedia.org/wiki/Taylor_series.

[Technologies, 2022] Technologies, U. (2022). Unity scripting reference. <https://docs.unity3d.com/ScriptReference/index.html>.