**VILNIAUS UNIVERSITETAS**

**MATEMATIKOS IR INFORMATIKOS FAKULTETAS**

**PROGRAMŲ SISTEMŲ KATEDRA**

**LĖKTUVO TŪPIMO IR KILIMO TRAJEKTORIJŲ TRIMATIS VIZUALIZAVIMAS**

**3D VISUALISATION OF PLANE LANDING AND TAKE-OFF TRAJECTORIES**

**Bakalaurinis darbas**

Martynas Žuravliovas (parašas)

Darbo vadovas:

doc. Kristina Lapin (parašas)

Vilnius, *2016*

Turinys

[Įvadas 3](#_Toc477018664)

[Motyvacija 3](#_Toc477018665)

[Tikslas 3](#_Toc477018666)

[Uždaviniai: 3](#_Toc477018667)

[1. Pirmas skyrius 5](#_Toc477018668)

[2. Trajektorijos rizikos 5](#_Toc477018669)

[3. Nukrypimo nuo idealios trajektorijos atvaizdavimas 5](#_Toc477018670)

[4. Kameros 6](#_Toc477018671)

[Šaltiniai 7](#_Toc477018672)

[Išvada 8](#_Toc477018673)

# Įvadas

## Motyvacija

## Tikslas

Sukurti programos prototipą, kuri leistų skrydžių vadovui ekrane stebėti lėktuvo nusileidimo ir pakilimo trajektorijas.

## Uždaviniai:

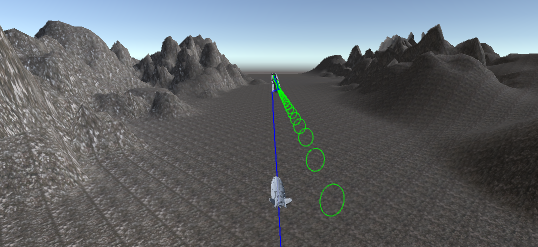
* Teisingos trajektorijos (toliau darbe – ideali trajektorija) analizė.
* Teisingos trajektorijos atvaizdavimas.
* Atsižvelgiant į galimas paklaidas, nustatyti kokiu skersmeniu turės būti piešiamas idealios trajektorijos tunelis.
* Išanalizuoti būdus, kaip pažymėti nukrypimą nuo trajektorijos ir parodyti skrydžio vadovui atsiradusius netikslumus (Informacinės lentelės, su aiškiu būsenos atvaizdavimu, taip pat galimybė išjungti vieną ar kitą vizualų komponentą).
* Kameros buvimo vietos analizė bei galimybė persijungti į kitą perspektyvą.

Pirminis darbų eigos planas:

1. Pirmiausia reikėtų išanalizuoti idealios trajektorijos duomenų lentelę. Iš pateiktų duomenų išgauti trajektorijos koordinates (Sugalvoti kaip tai padaryti, turima informaciją reikia paversti koordinatėmis, o galbūt tinkamas variantas leisti naudotojui nuskaityti idealios trajektorijos parametrus ir atitinkamai pagal juos suformuoti objektų pozicijas, pvz: pakilimo taką padėti per centrą, paskutiniajai idealios trajektorijos koordinatei).
2. Išgavus koordinates, bus nupiešiamas tunelis (aptvarkyti nusileidimo taką, pagal idealią trajektoriją jeigu jo pozicija nebus dinaminė trajektorijos atžvilgiu).
3. Tuomet, nustatyti maksimaliai leidžiamo nukrypimo nuo trajektorijos atstumą (ieškoti informacijos), kuris bus mūsų piešiamo tunelio spindulys R. Šis dydis, nebūtinai turi būti konstanta, galima padaryti reguliuojamą tunelio spindulį, jeigu nėra žinoma, koks jis turi būti iš tikrųjų.
4. Nupiešus teisingą trajektoriją, sugalvoti, kokiu būdu bus nustatomas nukrypimas nuo trajektorijos (pirma mintis: kiekviename koordinatės taške tikrinti su trajektorijos tunelio rėžiais, pvz: idealios trajektorijos taškui pridedame spindulį ir lyginame ar tai nėra daugiau už realios trajektorijos tašką. Ir taip tikriname į visas.
5. Gavus požymį, ar lėktuvas skrenda teisingoje trajektorijoje pradėti analizuoti nukrypimo atvaizdavimą, pvz: papildomas skydelis ekrane, su spalvinėmis žymomis (išsiaiškinti, ką Unity siūlo naudoti 3D scenose).
6. Taip pat, įgalinti kitos kameros perspektyvą. Esamame skydelyje, paprastas mygtukas kameros vaizdui keisti.

# Idealios trajektorijos integracija

Idealios trajektorijos nuskaitymui buvo panaudotas tas pats kodas, kuris nuskaito realios trajektorijos koordinates. Sukurtas atskiras idealios trajektorijos koordinačių masyvas. Pakoregavus esamą „PathFollower“ skriptą taip, kad tunelis būtų piešiamas naujai nuskaitytai idealiai trajektorijai. Taip gaunamas pirminis, realios skrydžio trajektorijos palyginimo su idealia, vaizdas. Pritaikyti terrain realų google maps reljefą.

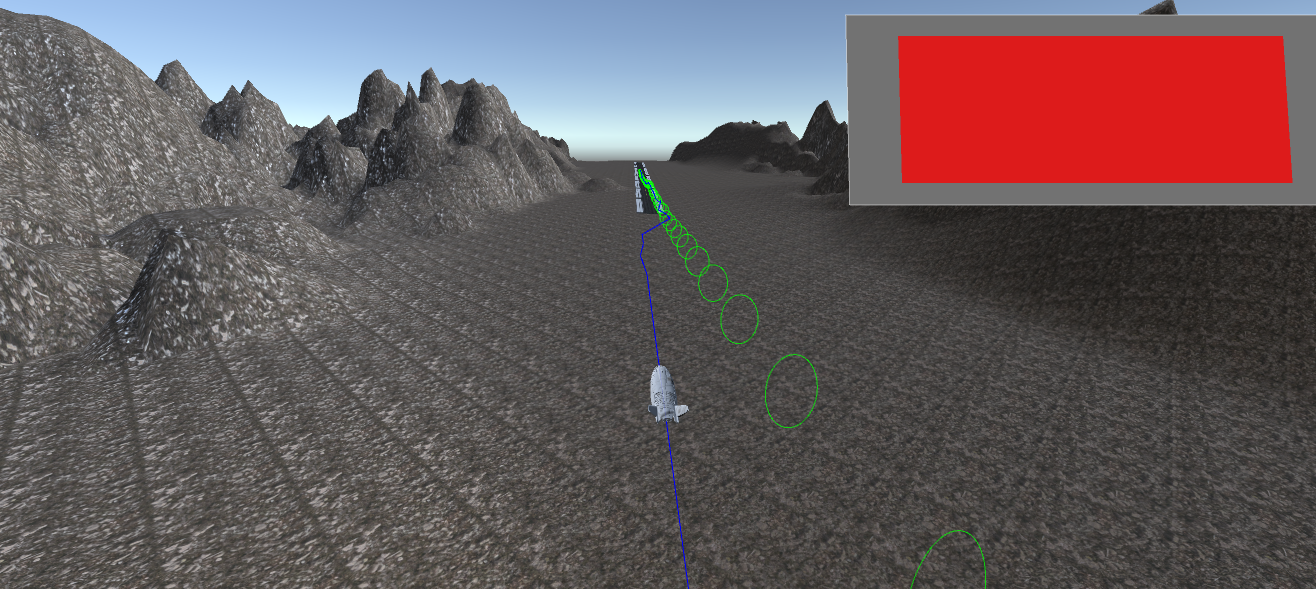


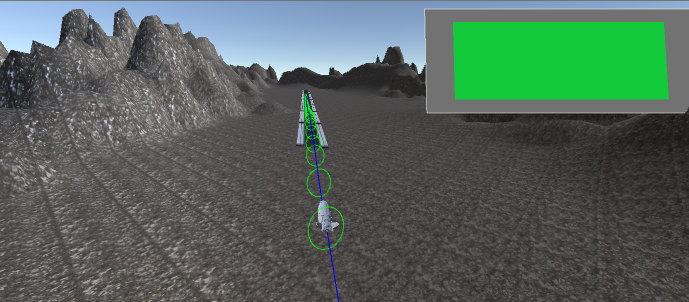
# Trajektorijos rizikos valdymas

Trajektorijų rizikos valdymui buvo sukonstruotas koordinačių paklaidos matavimo metodas. Šiam metodui paduodama dinaminė reikšmė – koordinatės paklaida. Šią paklaidą pridėjus bei atėmus iš idealios trajektorijos koordinatės gauname rėžius, kuriuose privalo atsidurti realios trajektorijos koordinatė konkrečiu skrydžio momentu. Iteruojant per realaus skrydžio koordinates, kiekvienoje iteracijoje iškviečiamas šis metodas, kuris atlieka paprastą patikrinimą ar reali koordinatė priklauso šiems rėžiams ar ne. Metodas grąžina reikšmę, kuri padeda nuspręsti ar lėktuvo skrydis vykdomas kaip numatyta.

# Nukrypimo nuo idealios trajektorijos atvaizdavimas

Nukrypimo atvaizdavimui sukuriamas UI dialogas (performuluoti) su spalviniu indikatoriumi (kodėl būtent toks sprendimas). Pasinaudojant anksčiau minėtu metodu, kuris grąžina požymį apie lėktuvo nukrypimą nuo trajektorijos, keičiame šio indikatoriaus spalvą. Esant nukrypimui užsidega raudona spalva, priešingu atveju dega žalia. Taip skrydžio vadovui nereikia didelių pastangų, kad suprasti ar skrydis vykdomas teisingai.





# Kameros

Pagrindiniam lėktuvo objektui sukuriamos papildomos, kito vaizdo perspektyvos kameros, kurias būtų galima keisti skrydžio metu norint matyti kitas skrydžio detales. Kamerų keitimo funkcionalumas pasiekiamas paspaudus klaviatūros klavišą „C“ (Nes didžioji dauguma žaidimų ir kitų simuliatorių naudoja būtent šį klavišą kameros keitimui), kuris leidžia paprastu būdų pakeisti kamerą. Paspaudus klavišą, iškviečiamas kameros keitimo metodas, kuris pagal eilę, deaktyvuoja esamą kamerą bei aktyvuoja sekančią. Kamera galėtų pasikeisti pagal konkrečią koordinatę, kuri indikuoja riziką.

# Šaltiniai

# Išvada