Pojednostavljena interaktivna simulacija zračnog prometa primjenom alata Unity

Seminarski rad iz kolegija "Interaktivni simulacijski sustavi"

Iva Goluža, Petra Petrović, Jurica Žirović, Dean Trkulja, Petar Marković

20. Siječnja 2023.

Djelovođa: Siniša Popović

Seminarski rad opisuje postupak razvoja, upute korištenja i glavne značajke jednog primjera interaktivne simulacije. Ostvarili smo pojednostavljenu interaktivnu simulaciju zračnog prometa primjenom alata Unity u 2D. Zrakoplovi slijede plan leta kroz zadane navigacijske točke. Simulacija je interaktivna na način da korisnik može upravljati smjerom kretanja zrakoplova i tako izbjegavati njihove sudare. Simulacija uključuje i nekontrolirane zrakoplove koji slobodno lete. Na kraju simulacije u datoteci korisnik može pogledati mjere performanci protekle simulacije.

1 Uvod

Računalna simulacija je izvođenje modela na računalu, što obuhvaća određivane izlaznih vrijednosti modela za zadane početne i/ili ulazne vrijednosti. Naša simulacija je interaktivna simulacija koja ima mogućnost interakcije s korisnikom, jer istodobno omogućuje prikaz rezultata simulacije korisniku i akcije korisnika koje utječu na daljnji tijek simulacije. Interakciju simulacije i korisnika smo ostvarili kroz mogućnost upravljanja smjerom leta aviona.

2 Uloge pojedinih članova tima

U sveukupnim poslovima na izradi seminarskog rada te pisanju ovog izvješća, članovi tima sudjelovali su na sljedeći način:

- Iva Goluža – Ostvarenje rotacije zrakoplova na korisnikovo upravljanje. Ispis određenih performanci u datoteku. Upravljanje krajem simulacije. Dokumentacija.
- Petra Petrović – Responzivna dropdown lista ovisno o korisnikovom odabiru brojeva. Ostvarenje rotacije zrakoplova na korisnikovo upravljanje. Ispis performanci u datoteku. Prikaz zadnjih pozicija kretanja zrakoplova.
- Dean Trkulja – Kreiranje Git repozitorija. Rad s datotekama (ostvarenje mogućnosti čitanja i pisanja u konfiguracijske datoteke tijekom simulacije. Izgled i funkcionalnost upravljačkog dropdowna. Kretanje zrakoplova po unaprijed zadanim navigacijskim točkama.
- Jurica Žirović – Ostvarenje izgleda kontroliranih i nekontroliranih zrakoplova i pozadine. Početna scena. Detekcija sudara zrakoplova međusobno i s navigacijskim točkama. Upravljanje krajem simulacije i timer. Ostvarenje mogućnosti korisnikovog odabira broja kontroliranih i nekontroliranih aviona u simulaciji te implementacija njihovog ponašanja u simulaciji.
- Petar Marković – Ostvarenje rotacije zrakoplova na korisnikovo upravljanje. Pomoć pri radu iznad navedenih zadataka. Prezentacija.

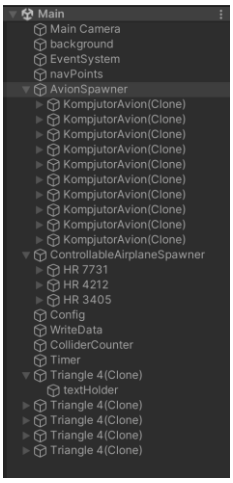
3 Interaktivna simulacija zračnog prometa

Interaktivne simulacije imaju brojne primjene. Koriste se za treniranje kognitivno-emocionalnih sposobnosti, evaluaciju i selekciju osoba za visoko stresne zadatke, visoku razinu multitaskinga, proučavanje degradiranih sposobnosti operatora...

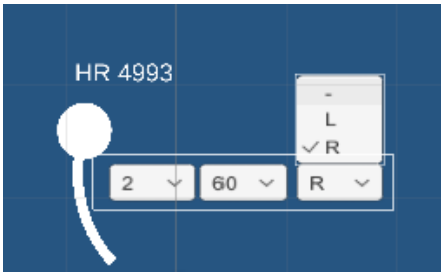
Kao motivacija i uvod za naš projektni zadatak poslužio nam je upravo simulator ovakve primjene. Radi se o simulaciji korištenoj za testiranje kandidata kontrolora zrakoplovnog prometa, osmišljenoj tako da su zrakoplovi postavljeni na kurseve sudara tijekom izvršenja zadatka, što će povećati iskustvo stresa kod kandidata. Više o ovoj temi [1]. Još jedan odličan članak [2] nam je pružio dodatna znanja o ovoj temi i predstavio nam LABY, računalno zasnovano dinamičko okruženje "čovjeka u petlji" (engl. man-in-the-loop). Kontrolor koji ga koristi mora izdati naredbe za usmjeravanje kako bi vodio zrakoplov po unaprijed određenoj ruti, dok izbjegava potencijalne sukobe i istodobno se brine o drugim pristiglim informacijama. Više o arhitekturi ATC simulatora i njihovoj upotrebi saznali smo iz [3]. Simulacije su predstavljene kao ključni dio ATC-a. Služe u najmanje tri svrhe; kao obuka učenika kontrolora, procjena novih protokola ili manevara, testiranje novih sučelja ili softvera. Tri ljudska aktera izvode simulaciju: kontrolori, pseudo-piloti i piloti u simulatoru leta. Arhitektura ATC simulacije koju smo proučavali prikazana je na slici 13. Kontrolori, pilot i pseudo-pilot međusobno djeluju putem simulatora prometa i simuliranog radija temeljenog na Voice over IP-u (VoIP). Komunikacija između kontrolora i pilota ili pseudo-pilota je u potpunosti glasovna. Signal je kodiran u 8 Khz i 8 bita PCM korištenjem μ-zakona optimizacije.

3.1) Elementi i funkcionalnosti interaktivne simulacije zračnog prometa

Scena naše pojednostavljene interaktivne simulacije zračnog prometa sastoji se od 3 glavna elementa: navigacijskih točaka, kontroliranih i nekontroliranih zrakoplova. Kontrolirani zrakoplovi su bijeli dok su nekontrolirani crvene boje. Tijelo zrakoplova prikazano je kružićem kojeg prati tanka linija duljine 1 cm koja prikazuje zadnje položaje tog zrakoplova. Radijus kružića je parametar svakog objekta zrakoplova zasebno, no svi su postavljeni na istu vrijednost. Navigacijske točke prikazane su kao jednakostranični trokuti žute boje s veličinom polumjera trokutu opisane kružnice jednakom duploj veličini radijusa kružića koji predstavlja tijelo aviona zbog bolje vidljivosti. Svaka navigacijska točka ima radijus kao svoj parametar ali su postavljene na istu vrijednost. Kretanje zrakoplova je jednostavno i linearno, jednolikom brzinom tijekom cijele simulacije. Zrakoplovi kao parametar imaju i svoju brzinu kretanja. Kontroliranim zrakoplovima brzina kretanja je svima postavljena na istu vrijednost dok se nekontrolirani zrakoplovi kreću različitim, nasumično odabranim brzinama. Interaktivnost simulacije ostvarena je kroz mogućnost korisničkog upravljanja smjerom kretanja zrakoplova.



Slika 1. Prikaz svih objekata scene (Unity)



Slika 2. Dropdown izbornik upravljačkog zrakoplova HR 4993.

Klikom miša na tijelo određenog zrakoplova korisnik odabire taj zrakoplov za ručno upravljanje njegovog novog smjera. U blizini zrakoplova otvara se dropdown izbornik koji koristimo kao upravljačku ploču tog zrakoplova. Izbornikom možemo odrediti kut zakretanja zrakoplova i stranu u koju će se zakrenuti. Prvo odabiremo znamenku stotica kuta, tako da izbornik ponudi znamenke 0, 1, 2 i 3.

Zatim za odabir znamenke desetica i jedinica kuta izbornik nudi brojeve 00, 10, 20, 30, 40, 50, 60, 70, 80 i 90. Ako korisnik odabere 0, 1 ili 2 kao znamenku stotica, za znamenku desetica i jedinica bit će ponuđene sve moguće opcije. Ako korisnik za znamenku stotica odabere 3, ponuđene opcije za znamenku desetica i jedinica su 00, 10, 20, 30, 40, 50 i 60. Na taj način ograničavamo korisnika na unos kuta do 360 stupnjeva. Nakon odabira kuta korisnik treba odabrati i smjer zakretanja aviona za zadani kut. Ponuđene opcije su L za lijevo i R za desno. Kada korisnik odabere i smjer zakretanja, avion se rotira u zadanom smjeru i postupno mijenjamo vektor navigacije aviona u novi vektor. Zakretanje je ostvareno linearnom interpolacijom između trenutnog i zadanog vektora smjera.

Koristili smo odličan alat za ovaj projektni zadatak, Unity. Rotaciju smo ostvarili korištenjem postupnog mijenjanja vektora smjera i korištenjem formule kutne brzine koja nam za fiksnu brzinu rotacije i deltaTime da stupnjeve za koje će se zrakoplov zaokrenuti u toj iteraciji ukupne rotacije, tako da složeniji matematički modeli nisu bili potrebni.

Dok se izvršava ta radnja postupne rotacije za određeni broj stupnjeva, ti “obrađeni” stupnjevi će se ubrajati u ukupni kut zakretanja, anglesTravelled (line 165, 173) i rotacija je u potpunosti izvršena kada taj kut koji inkrementalno povećavamo bude jednak kutu zakretanja koji je korisnik zadao (line 189). Tada resetiramo upravljački izbornik i ostale varijable koje koristimo za ostvarenje funkcionalnosti rotacije.

```
161         if (dropdown.getChanged())
162         {
163             if (dropdown.getTurn() == "Right")
164             {
165                 anglesTravelled += Time.deltaTime * turnSpeed;
166                 myBody.transform.localEulerAngles += new Vector3(0f, 0f, -Time.deltaTime * turnSpeed);
167                 myBody.transform.Translate(transform.up * speed * Time.deltaTime);
168                 canvas.transform.rotation = Quaternion.identity;
169                 nameHolder.transform.rotation = Quaternion.identity;
170             }
171             else
172             {
173                 anglesTravelled += Time.deltaTime * turnSpeed;
174                 myBody.transform.localEulerAngles -= new Vector3(0f, 0f, -Time.deltaTime * turnSpeed);
175                 myBody.transform.Translate(transform.up * speed * Time.deltaTime);
176                 canvas.transform.rotation = Quaternion.identity;
177                 nameHolder.transform.rotation = Quaternion.identity;
178             }
179             myObject.writeLog(string.Format("[{0}] Airplane {1} has new rotation angle {2}, {3}",
180                Timer.dateTime,airplaneName, dropdown.getNum(), dropdown.getTurn()));
181         }
182         else
183         {
184             myBody.transform.Translate(Vector3.up * speed * Time.deltaTime);
185             canvas.transform.rotation = Quaternion.identity;
186             nameHolder.transform.rotation = Quaternion.identity;
187         }
188
189         if (Mathf.Abs(anglesTravelled) >= dropdown.getNum())
190         {
191             travel = false;
192             anglesTravelled = 0f;
193             myBody.transform.Translate(transform.up * speed * Time.deltaTime);
194             dropdown.setChanged(false);
195             dropdown.setTurn("-");
196             canvas.transform.rotation = Quaternion.identity;
197             nameHolder.transform.rotation = Quaternion.identity;
198         }
199
200     }
```

Slika 3. Isječak koda ControllableAirplane skripte. Ostvarenje funkcionalnosti rotacije zrakoplova.

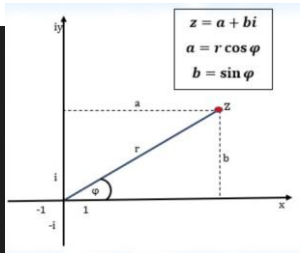
U cijeloj funkciji rotiramo isključivo tijelo zrakoplova dok canvas i nameHolder pomoću Quarternion.identity ne rotiramo tako da ime zrakoplova i upravljačka ploča uvijek budu pozicionirane horizontalno, neovisno o zakretanju zrakoplova.

Također je važno opisati način na koji smo računali novi vektor smjera zrakoplova nakon što je korisnik zadao određenu rotaciju zrakoplova. Koristili smo formule za pretvorbu kuta zadanog u stupnjevima u vektorski zapis s

koordinatama, uz naznaku da vektor smatramo jediničnim, pa je u formuli r = 1. Za izračun novog vektora smjera iz starog vektora smjera i zadanog kuta koristili smo formulu skalarnog umnoška dva vektora. Naime, sada kad smo kut zakretanja također pretvorili u vektor, iz tog vektora i starog vektora smjera pomoću formule skalarnog umnoška dva vektora izračunali smo kut između ta dva vektora, njega pomoću prve formule pretvorili u vektorski zapis i tako smo dobili novi vektor smjera, mDirection.

```
xNovi=Mathf.Abs(Mathf.Cos(dropdown.getNum()*Mathf.PI/180));
yNovi=Mathf.Abs(Mathf.Sin(dropdown.getNum()*Mathf.PI/180));
xStari = myBody.transform.position.x;
yStari = myBody.transform.position.y;
float duljinaNovi = Mathf.Sqrt(xNovi*xNovi + yNovi*yNovi);
float duljinaStari = Mathf.Sqrt(xStari*xStari + yStari*yStari);
fi =  Mathf.Acos((xNovi * xStari + yNovi * yStari)/(duljinaNovi*duljinaStari));
x = Mathf.Abs(Mathf.Cos(fi));
y = Mathf.Abs(Mathf.Sin(fi));
mDirection = new Vector3(x, y, 0);
```

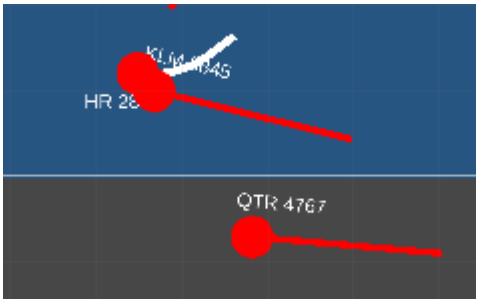
Slika 4. Isječak koda za računanje novog vektora smjera.



$$\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\| \|\mathbf{b}\| \cos\theta.$$

Slike 5. i 6. Korištene matematičke formule (izvori [3] i [4]).

Implementirane su i mogućnosti detekcije kolizije zrakoplova. Svakom kolizijom zrakoplova, oni prolaze jedan kroz drugi, ali se brojač kolizija uvećava, a korisniku naglasimo koliziju tako što se tijelo zrakoplova zacrveni.



Slika 7. Kolizija kontroliranog (bijeli) i nekontroliranog (crveni) zrakoplova. Tijelo kontroliranog zrakoplova se zacrvenilo dok kolizija traje.

Ako kontrolirani zrakoplov u pravilnom redoslijedu prođe kroz navigacijsku točku koja mu je zadana u željenoj putanji u konfiguracijskoj datoteci simulacije, tijelo zrakoplova postane zeleno.

Slika 8. Kontrolirani zrakoplov prolazi kroz navigacijsku točku NAV0 koja mu je bila zadana kao iduća točka oputanje. Tijelo je postalo zeleno.

3.2) Konfiguracijske datoteke simulacije

Korisnik prije samog pokretanja simulacije treba postaviti inicijalne parametre simulacije, njihovim upisivanjem u konfiguracijsku tekstnu datoteku “config”.

```
navPoints=5
controllableAirplanes=3
redAirplanes=2
1,2,3,4,5
1,2,3,4,5
1,2,4,3,5
```

Slika 9. Config datoteka sa zadanim inicijalnim parametrima simulacije.

U datoteku možemo zapisati broj navigacijskih točaka, broj kontroliranih i nekontroliranih zrakoplova, a zatim za svaki kontrolirani zrakoplov navodimo redoslijed obilaska navigacijskih točaka koje zrakoplov mora poštivati. Svi navedeni elementi se pojave na nasumično odabranim lokacijama na sceni.

Po završetku simulacije u datoteci "data" korisnik može pogledati zapise performanci protekle simulacije. Simulacija se izvršava na frekvenciji od 10 Hz, no ispis performanci se zapisuje svakih 0.5 Hz zbog čitljivosti. Svi ispisi u datoteci imaju timestamp formata yyyy.dd.mm-hh.mi.sc.msc na početku svog zapisa.

U datoteci se prvo zapisuju pozicije navigacijskih točaka u zajedničkom koordinatnom sustavu svih elemenata. Ako je korisnik u konfiguracijsku datoteku "config" prije početka simulacije zadao n navigacijskih točaka, u ovoj datoteci će se ispisati n pozicija. Svakoj je dodijeljena nasumična pozicija i ime.

Nakon popisa navigacijskih točaka zapisane su širina i visina ekrana u zajedničkom koordinatnom sustavu svih elemenata simulacije.

```
[2023.18.01-09:12:17:084] Navigation point 1 is at position (3.17624,0.6682229)
[2023.18.01-09:12:17:084] Navigation point 2 is at position (0.4327383,-1.715052)
[2023.18.01-09:12:17:084] Navigation point 3 is at position (-1.276746,-2.941489)
[2023.18.01-09:12:17:084] Navigation point 4 is at position (-0.6254845,2.15634)
[2023.18.01-09:12:17:084] Navigation point 5 is at position (1.237252,0.317912)
[2023.18.01-09:12:17:084] Screen height is 434 pixels
[2023.18.01-09:12:17:084] Screen width is 737 pixels
```

Slika 10. Početni zapisi "data" datoteke. Zapisi o navigacijskim točkama, visini i širini ekrana.

Nakon ovih zapisa, u datoteku se, uz timestamp, spremaju i zapisi:

- položaja i smjera svih zrakoplova u zajedničkom koordinatnom sustavu simulacije, pa čak i oni koji se trenutno ne vide na ekranu jer su u daljini i upravo dolaze ili odlaze iz scene,

- poruke odabira određenog zrakoplova (click miša) kada se korisniku u simulaciji otvara upravljački izbornik,

```
[2023.18.01-17:32:46:727] Airplane HR 3867 is at position (1.414336,-1.222618) in direction (4.19, -3.57, 0.00).
[2023.18.01-17:32:47:520] You clicked on HR 3867
[2023.18.01-17:32:51:728] Airplane HR 5438 is at position (-3.357557,-0.581004) in direction (0.34, 0.94, 0.00).
[2023.18.01-17:32:51:728] Airplane HR 2247 is at position (-0.8588822,0.7366772) in direction (-5.67, 4.99, 0.00).
[2023.18.01-17:32:51:728] Airplane HR 3867 is at position (2.385312,-2.048954) in direction (4.19, -3.57, 0.00).
[2023.18.01-17:32:56:730] Airplane HR 5438 is at position (-4.21875,0.3592007) in direction (0.09, 1.00, 0.00).
[2023.18.01-17:32:56:730] Airplane HR 2247 is at position (-1.815728,1.579358) in direction (-5.67, 4.99, 0.00).
[2023.18.01-17:32:56:730] Airplane HR 3867 is at position (3.326571,-2.892284) in direction (0.66, 0.75, 0.00).
```

Slika 11. Zapisi "data" datoteke. Zapis o korisničkom upravljanju zrakoplovom HR 3867.

- poruka promjene vektora smjera nakon što je korisnik u upravljačkom izborniku odabrao kut i smjer zakretanja za određeni zrakoplov i zapis broja stupnjeva i smjera (L ili R)

```
[2023.18.01-17:51:57:335] Airplane HR 2452 is at position (3.57953,0.46995) in direction (0.61, 0.79, 0.00).
[2023.18.01-17:51:57:335] Airplane HR 2452 has new rotation angle 120, Left
```

Slika 12. Zapis "data" datoteke. Zapis o zadanom novom kutu skretanja i smjeru skretanja.

- zadnji zapis u datoteci je zapis broja kolizija koji se dogodio između zrakoplova.

```
[2023.18.01-17:35:11:768] Airplane HR 3867 is at position (-17.96663,-29.32871) in direction (0.85, 0.52, 0.00).
[2023.18.35-17:35:16:002] Simulation finished, total number of collides is 4
```

Slika 13. Zadnji zapis "data" datoteke. Zapis o broju "sudara" zrakoplova.

* Napomene: Svi zrakoplovi i navigacijske točke imaju svoje ime kojim ih razlikujemo. Na početku simulacije svi zrakoplovi dobiju nasumično odabranu lokaciju u zajedničkom koordinatnom sustavu, te nasumično odabran vektor smjera kretanja. Lokacije navigacijskih točki su također nasumično dodijeljene.

3.3) Upravljanje završetkom simulacije

Simulacija se prekida ako svi kontrolirani zrakoplovi uspiju proći kroz sve zadane navigacijske točke svojih putanja u pravilnom redoslijedu. Napomena: zrakoplov unutar svoje putanje može proći i kroz navigacijske točke koje nisu iduće u njegovoj zadanoj putanji, njih zanemarujemo, a zrakoplov pri nailasku na iduću putanjom zadanu točku postaje zelen. Važno je da tijekom simulacije prođe u pravilnom redoslijedu kroz sve navigacijske točke zadane putanje.

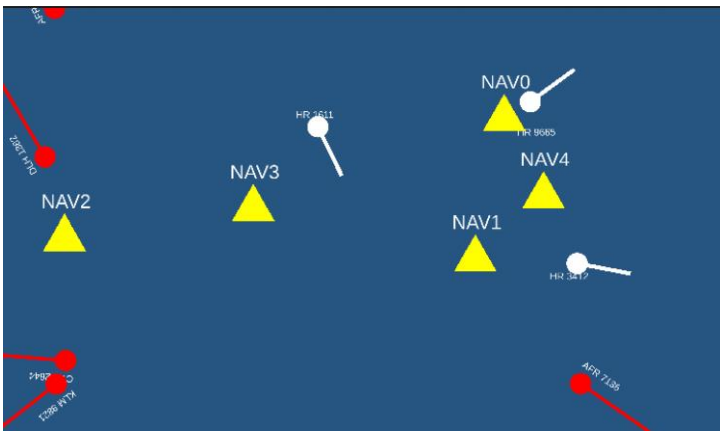
Ako se prethodno opisani scenarij ne dogodi unutar 3 minute od početka pokretanja simulacije, simulacija se prekida.

3.4) Scenariji simulacije

Na samom početku implementacije koristili smo jednostavnije scenarije za testiranje. Krenuli smo od nekontroliranog zrakoplova. Nakon toga smo se fokusirali na kontrolirani zrakoplov. Kada su oba imala implementirane planirane funkcionalnosti scenarij smo malo otežali unosom jednog kontroliranog i jednog nekontroliranog zrakoplova. Zatim smo dodali i funkcionalnost zadavanja ruta, tj. slijeda prolaska navigacijskim točkama kontroliranom zrakoplovu. Scenarij simulacije se tada dodatno otežao jer se korisnik sada treba pobrinuti da zrakoplov prođe kroz sve navedene navigacijske točke pravilnim redoslijednom u zadanom vremenskom intervalu trajanja simulacije.

Scenarije smo postupno dodatno otežavali unosom većeg broja navigacijskih točaka i kontroliranih zrakoplova, produžavanjem zahtijevanih ruta zrakoplova, pa i povećavanjem broja nekontroliranih zrakoplova jer se tada trebalo obratiti još više pozornosti na sprječavanje kolizije zrakoplova.

Završni scenarij kojeg ćemo i demonstrirati jest scenarij u kojem koristimo 3 kontrolirana zrakoplova i 5 navigacijskih točaka.



Slika 14. Završni scenarij simulacije; 5 navigacijskih točaka i 3 kontrolirana zrakoplova

3.5) Korištene tehnologije i alati

- Glavni alat koji smo koristili za izradu našeg projektnog zadatka je Unity (verzija 2021.3..11f1).
- Za pisanje koda smo koristili razvojnu okolinu Visual Studio Code.
- Dokumentacija – Word.
- Komunikacija – WhatsApp, Discord, MS Teams.
- GitHub – udaljeni direktorij projekta.

4 Zaključak

Ovim projektnim zadatkom pobliže smo se upoznali s postojećim simulacijama zrakoplovnog prometa. Razvili smo našu pojednostavljenu simulaciju zračnog prometa. Projekt je odlično poslužio kao uvod u široko područje razvoja simulatora. Koristili smo nekima, u potpunosti nov alat, Unity. Također smo razvijali sposobnosti timskog rada i planiranja. Projekt nam je omogućio da se i sami izložimo stresnim situacijama kontroliranja zrakoplovnog prometa u vlastitoj simulaciji. U budućnosti bi ovaj rad mogli nadograditi mogućnostima nekih još složenijih scenarija. Kontrolirani zrakoplovi bi također mogli imati različite brzine kretanja. Također bi mogli implementirati opciju gdje svi zrakoplovi ne lete nužno na istoj visini, što bi značilo da neće uvijek doći do kolizije.

5 Literatura

[1] Šarlija, M., Popović, S., Jagodić, M., Jovanovic, T., Ivkovic, V., Zhang, Q., ... & Ćosić, K. (2020). Prediction of task performance from physiological features of stress resilience. IEEE Journal of Biomedical and Health Informatics, 25(6), 2150-2161.

[2] Imbert, J. P., Hodgetts, H. M., Parise, R., Vachon, F., & Tremblay, S. (2014, September). The LABY microworld: A platform for research, training and system engineering. In Proceedings of the Human Factors and Ergonomics Society Annual Meeting (Vol. 58, No. 1, pp. 1038-1042). Sage CA: Los Angeles, CA: SAGE Publications.

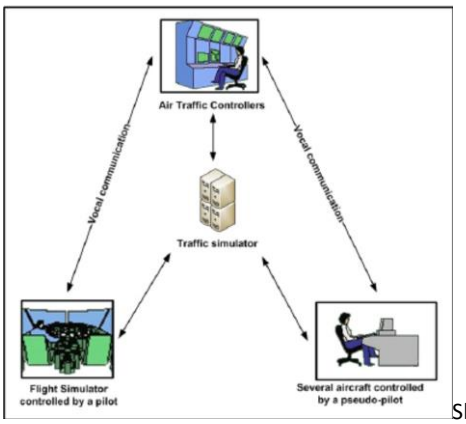
[3] https://www.researchgate.net/publication/278965052_Increasing_Air_Traffic_Control_simulations_realism_through_voice_transformation#pf2

[4] https://ematematika.hr/kompleksni/bpid/85

[5] https://slideplayer.com/slide/14780152/

[6] https://www.researchgate.net/figure/ATC-simulation-architecture_fig1_278965052

6 Dodatak



Slika 15. Primjer arhitekture ATC simulatora [5]