**Pojednostavljena interaktivna simulacija protuoklopnog vođenog raketnog sustava primjenom alata Unity**

**Seminarski rad iz kolegija “Interaktivni simulacijski sustavi”**

**Branimir Tomeljak, Petra Mažar, Antonia Meštrović, Rea Vukasović**   
 **15. 01. 2022.**

Djelovođa: Siniša Popović

***Sažetak***

**Na ovom projektu bavili smo se izradom vrlo pojednostavljenje verzije protuoklopnog vođenog raketnog sustava u 3D pogledu. Ovaj rad prikazuje koncepte ostvarene simulacije te programsku potporu i alate koji su korišteni pri izradi. Cilj simulacije bio je prikazati osnovna obilježja proturaketnog sustava uz poštovanje osnovnih principa sustava. Sustav omogućuje igraču da uređuje postavke prije pokretanja igre kako bi prilagodio scenu. Simulacija se sastoji od objekata „POVRS,“ koji ima mogućnost ispucavanja raketa, i „Tank“, koji uz okolinu čini središnju metu našeg sustava. Cijela simulacija odvija se na 3D terenu koji je ručno izrađen.**

# 1 Uvod

Simulacije su sustavi koji ponašanje nekog objekta ili odvijanje određene situacije promatraju na računalnom modelu. Primjena simulacija pokazala se iznimno praktičnom u slučaju procesa koji uključuje veliki trošak, teško je izvediv u stvarnosti ili rizičan za korisnika. Sukladno tome, simulacijski sustavi najčešće se koriste u stresnim situacijama, prilikom raznih obuka, testiranja sustava koji su u razvoju ([3]). Simulacijski sustav u ovom projektu namijenjen je treningu ljudi i testiranju reakcija korisnika u stresnim situacijama.

Ovaj seminarski rad bavi se izradom igre koja omogućuje igraču da s fiksne pozicije gađa metu („Tank“). Meta može biti statična ili dinamička. O ovoj varijabli odlučuje korisnik pri pokretanju igre odabirom jedne od opcija iz izbornika. Također mu je omogućeno prilagođavanje vidljivosti podešavanjem razine magle u sceni. Igra je ostvarena upotrebom alata Unity ([8]). Osim vizualnih efekata, igra je popraćena i zvučnim efektima poput onih prilikom ispucavanje rakete i pogotka mete.

# 2 Uloge pojedinih članova tima

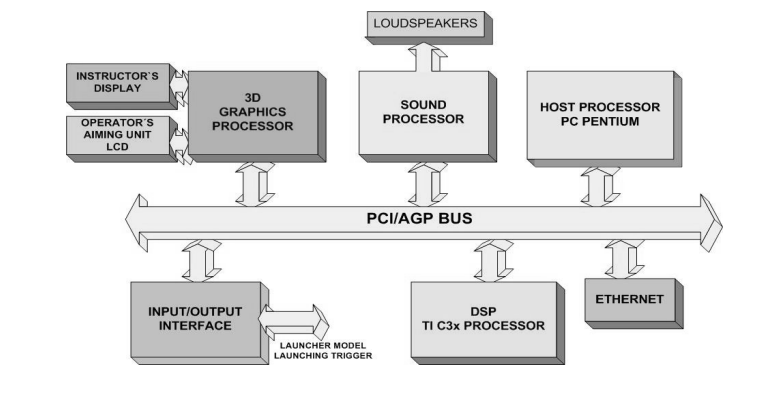
U sveukupnim poslovima na izradi seminarskog rada te pisanju ovog izvješća, članovi tima sudjelovali su na sljedeći način:

* **Branimir Tomeljak** – Obavio točke zadatka **4** (*Simulacija leta rakete*), **7** (*Dizajn mirnog i stresnog scenarija*), **11** (*Efekti rakete tijekom leta*), **13** (*Podešavanje kamere tijekom simulacije*). Pridonio pisanju dokumentacije u sekcijama **2** (*Uloge pojedinih članova*), **3** (*POVRS i arhitektura POVRS simulatora)* i **5** (*Alati i programska potpora*)
* **Petra Mažar** – Obavila točke zadatka **1** (*Izrada terena*), **2** (*3D model tenka*), **3** (*3D model rakete*). Pridonijela pisanju dokumentacije u sekcijama **1** (*Uvod*), **3** (*POVRS i arhitektura POVRS simulatora*) i **4** (*O simulaciji*)
* **Antonia Meštrović** – Obavila točke zadatka **8** (*Reakcija mete na pogodak*), **9** (*Eksplozija i dim pri sudaru*), **10** (*Zvukovi u igri*). Pridonijela pisanju dokumentacije u sekcijama **5** (*Alati i programska potpora*), **6** (*Organizacija projekta*), **7** (*Glavni izbornik*) i **9** (*Literatura*)
* **Rea Vukasović** – Obavila točke zadatka **5** (*Kretanje tenka u dinamičkoj/stresnoj situaciji*), **6** (*Nasumično pozicioniranje tenka i POVRS-a*), **12** (P*odešavanje vremenskih prigoda*). Pridonijela pisanju dokumentacije u sekcijama **6** (*Organizacija projekta*), **8** (*Zaključak*) i **9** (*Literatura*)

# 3 POVRS i arhitektura POVRS simulatora

Protuoklopni vođeni raketni sustav je namijenjen za uništavanje tenkova, oklopnih transportera u drugih vozila ([3]). Zbog jednostavnosti i prenosivosti, njihovom uporabom je i pješaštvo razvilo mogućnost borbe protiv oklopništva. Te rakete ispaljuju se s ramena ili prenosivog vozila, helikoptera ([2]). Prva generacija vođenih raketnih sustava navodi se ručno, operator kroz durbin promatra cilj i let rakete sve do njenog pogotka navodi upravljačkom palicom. Druga generacija je poluautomatski vođena, dok je treća generacija radi na principu ispali i zaboravi gdje se rakete navode infracrvenim zrakama ili nekim drugim sustavom navođenja ([1]).

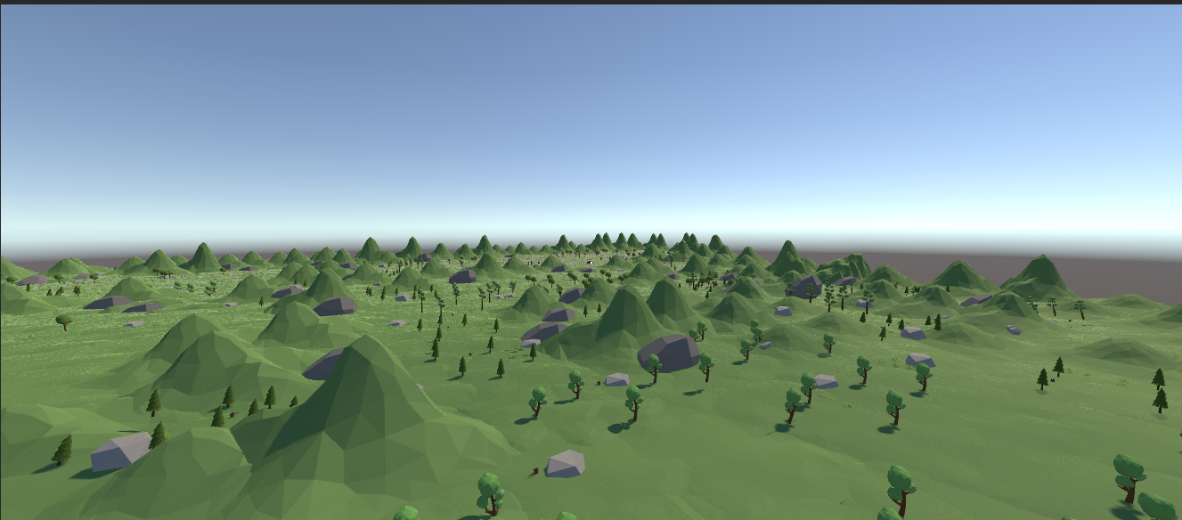
Simulator POVRS sastoji se od instrukcijske stanice, modela streljačkog mjesta topnika te nišana s LCD ekranom. Višeprocesorska arhitektura POVRS simulatora sastoji se od ulaznog i izlaznog sučelja i procesora, sabirnicom povezanih s 3D grafičkim procesorom, procesorom zvuka i glavnim procesorom. Dodatno, zvučnici su povezani s procesorom zvuka, a ekran instruktora i LCD ekran korisnika simulatora s 3D grafičkim sučeljem ([3]).



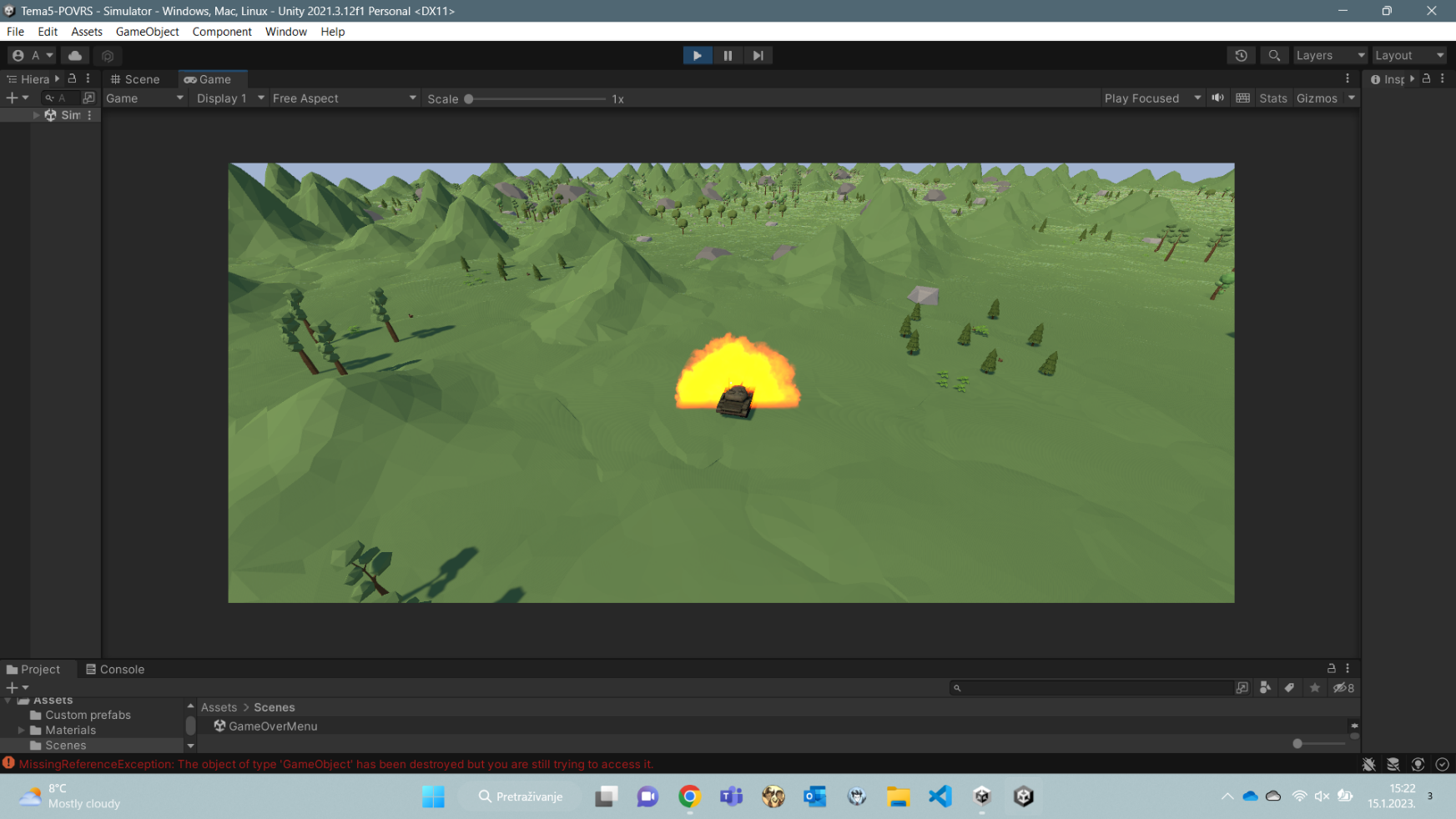
Slika 1. arhitektura POVRS Simulatora([3])

# 4 O simulaciji

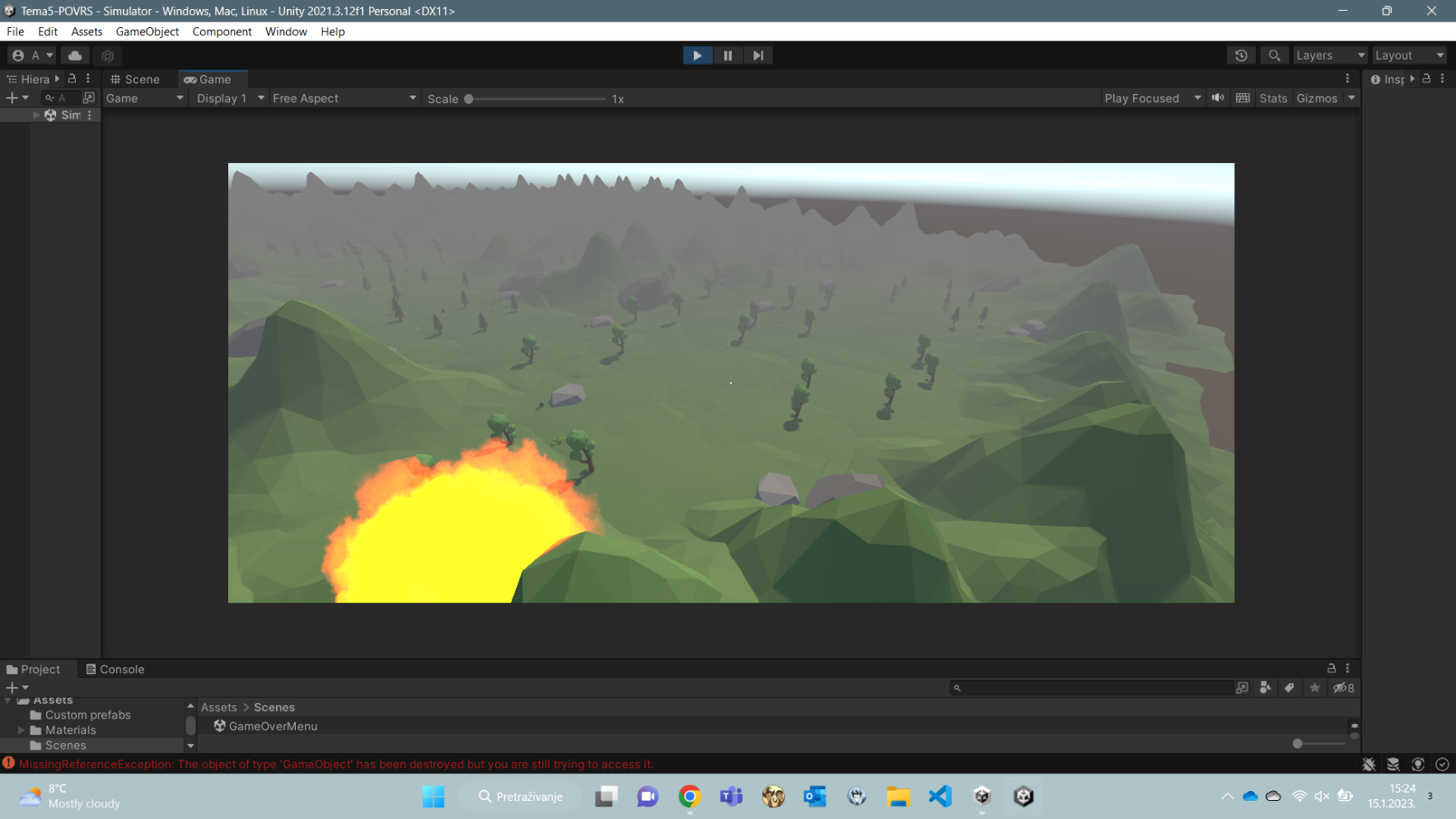
Simulacija kreće s pogledom na okolinu u prvom licu gdje se nalazimo u ulozi operatora rakete. Pri svakom pokretanju simulacije POVRS i meta bit će smješteni na različitim pozicijama. Operator ima mogućnost naciljati metu te je ispaljena raketa automatski prati. U našem slučaju, meta se prati ciljnikom u centralnoj točki ekrana označenim bijelom točkom. Ukoliko je raketa ispaljena u malom radijusu oko tenka, prikaz, to jest kamera će se prebaciti u bliži pogled na sam tenk te će se prikazati eksplozija sa zvučnim efektom i meta će nestati. U slučaju pogotka tenka simulacija završava, dok u slučaju da je meta promašena, kamera se vraća na početni pogled kod POVRS. Sama raketa može reagirati sa svim elementima u okolišu poput drveća, tla i kamenja uzrokujući eksploziju sa zvučnim efektom. U slučaju pogotka drveća, meta će nestati. Kretnja tenka ovisi o opciji odabranoj u glavnom izborniku, ukoliko je odabrana opcija da je neprijatelj statičan, tenk se neće micati, a ukoliko je odabrana opcija da je tenk dinamičan, tenk će nakon svakih par sekundi kretnje slučajnim odabirom promijeniti smjer. U dinamičnom radu pojavljuju se i popratne eksplozije koje pridonose stresnom okruženju. Isto tako, veliku vizualnu promjenu u simulaciji čini i količina magle koja se također podešava u glavnom izborniku.



Slika 2. Prikaz situacije u kojoj je meta naciljana



Slika 3. Prikaz mete iz drugog kuta te uništenje mete



Slika 4. Prikaz stresne situacije s okolnim eksplozijama

i povećanom razinom magle

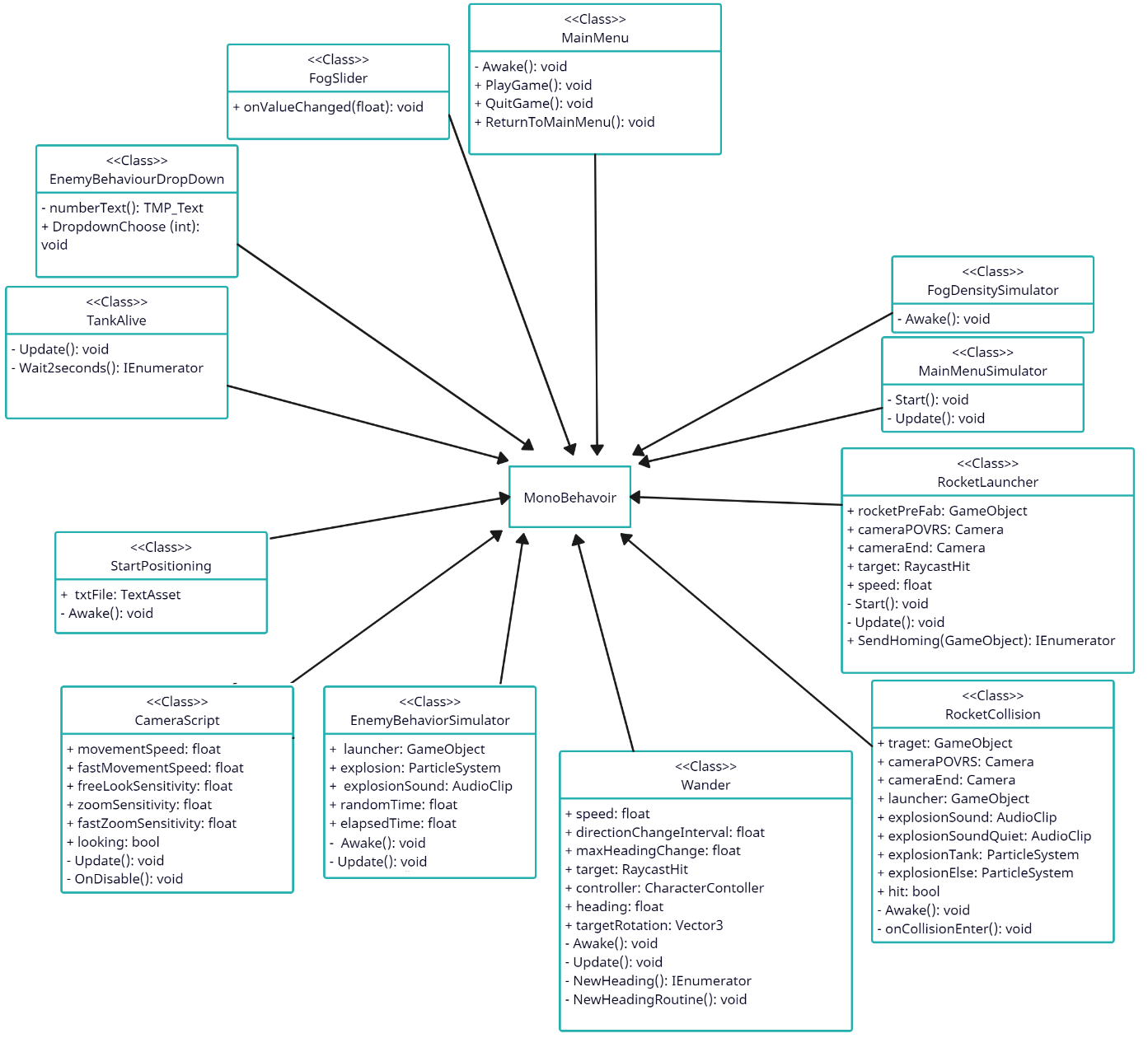
# 5 Alati i programska potpora

Za izradu računalne igre koristili smo programski alat Unity ([8]) koji se često koristi za izradu sličnih aplikacija. Sve skripte unutar projekta pisane su u programskom jeziku C# što nam je odgovaralo s obzirom na to da je to objektno orijentiran jezik. Prilikom uređivanja postojećih skripti i pisanje svojih koristili smo Visual Studio Code ([9]). Modele za „Tank“ i „POVRS“ smo preuzeli iz Unity Asset Store-a ([10]) i dodatno ih doradili za svoje potrebe. Također, korišteni su i asseti Unity Particle Pack ([11]) za eksplozije i Free Missile za raketu

# 6 Organizacija projekta

Naš Unity projekt organiziran je u dvije glavne mape Assets i Packages. U mapi Assets bitnije za spomenuti su mape Materials u kojoj se nalaze objekti i zvučni efekti koje smo koristili, mapa Scenes u kojoj se nalaze tri scene, GameOverScene za završni izbornik, MainMenuScene za glavni izbornik te SimulatorScene za kreiranu simulaciju. Najvažnija mapa je ipak Scripts koja sadrži sve kodove napisane u programskom jeziku C# potrebne za pokretanje cjelokupne simulacije. Svi razredi iskorišteni u projektu nasljeđuju klasu MonoBehavior što se vidi na prikazu dijagrama klasa u nastavku.

Hijerarhijska struktura projekta podijeljena je na više komponenti, a kod scene SimulatorScene imamo: Directional Light koji dovodi svjetlo kao dnevno svjetlo koje ispunjava velike prostore iz jednog kuta, Ground koji kreira tlo, Enviroment Objects u kojem se nalaze sva drveća, kamenje i ostali manji elementi okoliša, Sounds koji sadrži sve zvučne elemente kao pozadinske zvukove i zvukove eksplozije, MainMenu povezan sa skriptom MainMenuSimulator.cs, Fog povezan sa skriptom FogDensitySimulator.cs i CheckTankAlive povezan sa skriptom TankAlive.cs. Komponente od velike važnosti su Tank i POVRS povezani s najviše skripti. Hijerarhijska struktura kod MainMenuScene i GameOverScene je slična, oboje sadrže komponente Main Camera, Directional Light, Background Sounds identične kao i kod SimulatorScene, te dodatno istoimene komponente s pripadajućim elementima.



Slika 4. Dijagram razreda mape Scripts

# 6.1 POVRS

Komponenta POVRS sadrži tri elementa za pokretanje, dodatno sadrži i element sa assetom Modular First Person Controller kojim je namještena kamera da bude u željenoj poziciji.

Komponenta RocketSpawnPoint ima na dodanu skriptu RocketLauncher.cs koja omogućuje ispucavanje rakete, preko GameObject objekta rocketPrefab povezana je s missile objektom koji se nalazi u direktoriju Materials. Rotiranjem kamere možemo izviđati scenu u potrazi za metom. Dodatna funkcionalnost je da se prilikom pogotka tenka mijenja perspektiva iz one igrača na drugu kameru kako bismo izbliza vidjeli eksploziju.

# 6.2 Tank

„Tank“ se sastoji od komponenti body, track\_1, track\_2, turret, wheels. Ova je komponenta preuzeta iz Unity Asset Store ([10)]. Na nju je dodana skripta Wander.cs koja je zadužena za upravljanje kretanjem tenka. Tenk se kreće prema naprijed, a nakon zadanog vremenskog intervala generira se nasumični stupanj rotacije koji mijenja pravac kretanja. Dodana je i skripta EnemyBehaviourSimulator.cs koja prima informacije o odabranom načinu kretanja tenka iz glavnog izbornika.

# 6.2.1 3D model tenka

3D model je preuzet s Unity Asset Store ([5]). Koristili smo ovaj model zbog toga što je jednostavan, a sadrži sve što nam je potrebno.

# 6.3 Missile

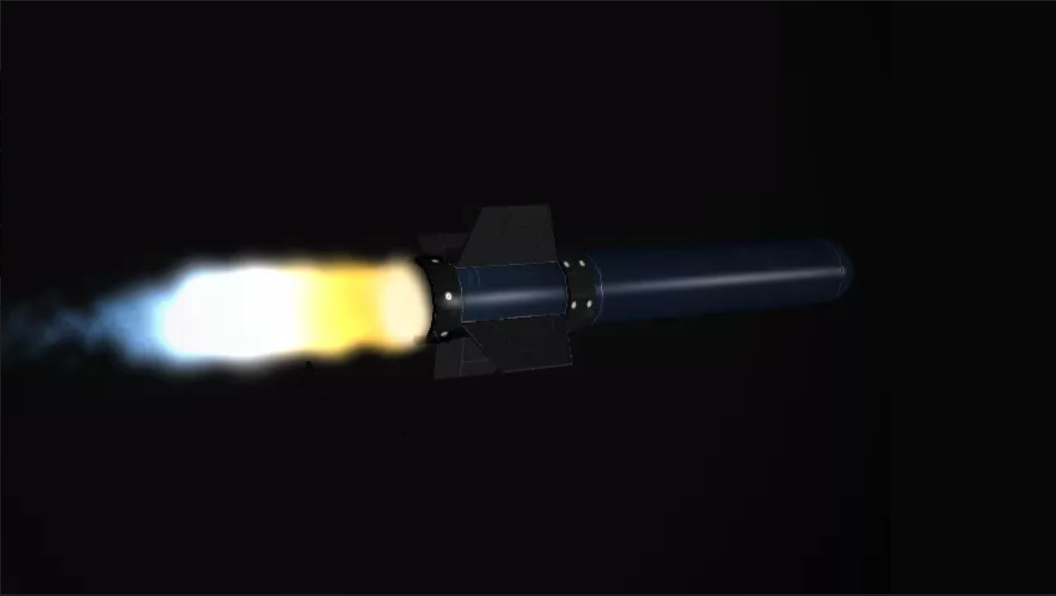
Komponenta „Missile“ ima dodanu skriptu RocketCollision.cs koja upravlja ponašanjem rakete pri ispucavanju i pogotku mete. U slučaju sudara rakete s tenkom ili scenom doći će do eksplozije i u slučaju tenka do uništenja mete. Ova komponenta također ima dodan zvučni i čestični efekt eksplozije. Povezani su i elementi smoke i fire koji se pojavljuju u simulaciji kada dođe do sudara s drugim objektima te pri ispaljivanju rakete.

# 6.3.1 Matematički model gibanja rakete

Matematički model gibanja ispaljene rakete vrlo je jednostavan. Model prati klikom odabranu točku odabranog 3D model (*target.point*) i prilikom svakog osvježavanja slike ažurira poziciju rakete prateći trenutnu poziciju (*rocket.transform.position*), zadanu brzinu (*speed*) i razliku u vremenima ažuriranja pozicije (*Time.deltaTime*). Pritom vodimo brigu o tome da je prednji dio 3D modela rakete uvijek uperen prema meti (rocket.transform.LookAt(target.point)).

# 6.3.2 3D model rakete

3D model je preuzet s Unity asset store ([6]). Koristili smo ovaj model zbog toga što osim osnovnog izgleda koji nam odgovara, sadrži i čestične efekte dima, kao što i ostavlja trag letenja.

 Slika 5. Prikaz preuzetih 3D modela

# 6.4 3D modeliranje terena

Zbog efikasnosti izrade dijelovi terena preuzeti su s Unity asset store ([7]). Teren je hijerarhijski organiziran u mapama Ground i EnvironmentObjects. U mapi Ground mogu se primijetiti 3 osnovne vrste terena (Ground\_01, Ground\_02, Ground\_03) koje se naknadno dorađivalo, dupliciralo i transformiralo. Transformacije su najčešće uključivale translaciju i rotaciju. Također su vidljive i elevacije terena. Područje terena je ručno popunjeno i s elementima koje možemo naći u podmapama Tree, Rock, Bush, Stump, Branch. Elemente tih podmapa bilo je potrebno dodatno translatirati i rotirati kako bi bile u skladu sa x,y,z ravninom već stvorenog terena.

# 6.5 Detekcija sudara

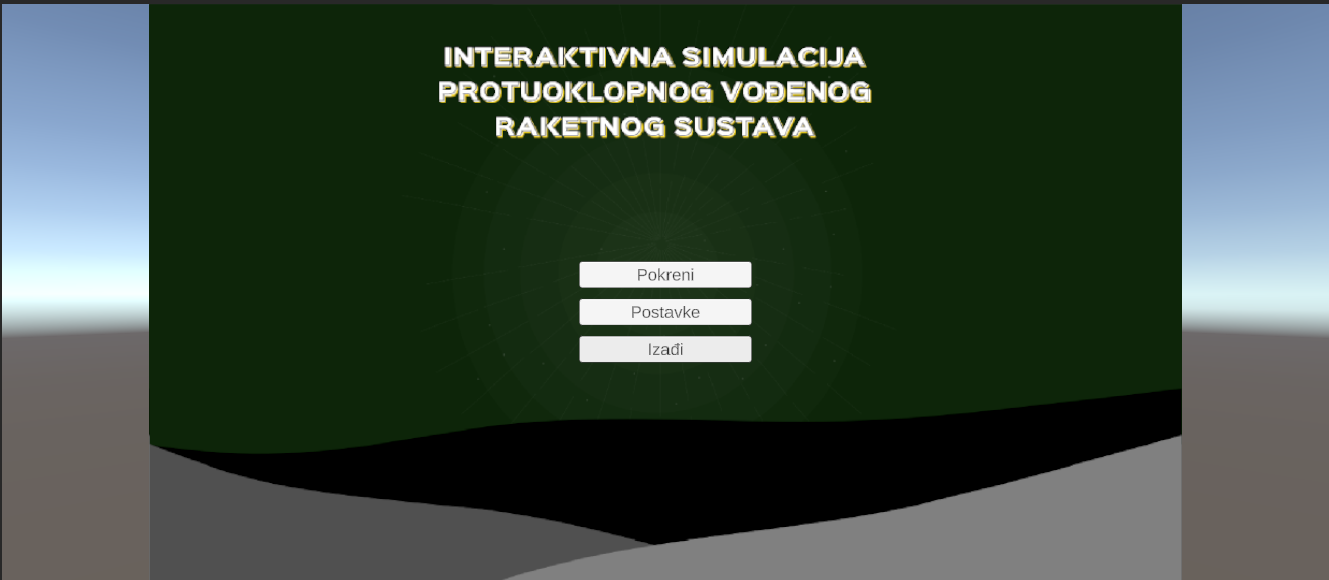
Detekcija sudara je tako da se implementira Unityjeva funkcija OnCollisionEnter() koja prati collider (komponenta objekta koja služi za detekciju sudara) 3D modela rakete i poziva se u slučaju kada dođe do sudara collidera rakete s colliderom drugog 3D objekta. U slučaju sudara poziva se zvuk eksplozije (AudioSource.PlayClipAtPoint(explosionSound, launcher.transform.position)), kao i efekt eksplozije na mjestu sudara (Instantiate(explosionTank, target.transform.position, Quaternion.identity)), a raketa se uništava nakon 0.7 sekundi (Destroy(target, 0.7f)). Ako je pogođen tenk onda se kamera približava i vidimo sudar iz blizine. Ako se pogodi drugi objekt okoliša, a da to nije tlo, taj objekt se uništava.

# 6.6 Obrada ulaznih informacija

Ulaze u sustav korisnik unosi aperiodički. Desnim klikom miša korisniku se omogućuje bliži pogled prema cilju bez pomicanja tijela, a zatim kada je naciljao lijevim klikom miša se daje znak računalnom sklopovlju da može ispaliti raketu. Kao izlaznu informaciju računalo nam pokazuje let rakete i ako je meta uspješno pogodila pokazuje se eksplozija. Metoda Input.GetMouseButtonDown(0) u skripti RocketLauncher prati se je li korisnik kliknuo miš ili još nije zadao nikakve ulaze.

# 7 Glavni izbornik

Kada korisnik pokrene igru prvo što će se prikazati bit će izbornik. To je početna scena koja sadrži Unityjevo korisničko sučelje. U glavnom izborniku moguće je odabrati postavke po kojima će se kreirati simulacija. Korisnik može izabrati hoće li meta biti statička ili dinamička. Također se može podesiti i razina maglovitosti. Pritiskom na gumb *Natrag* vraćamo se u glavni izbornik nakon čega možemo ponovno podesiti scenu ili pokrenuti igru pritiskom na pokreni. Pri pogotku mete, simulacija završava te se prikazuje izbornik s opcijama ponovnog pokretanja, povratka na glavni izbornik te izlaza. Također je i prikazana prigodna poruka “Tenk je uništen!”.



Slika 5. Prikaz glavnog izbornika



Slika 6. Prikaz izbornika nakon pogotka mete

# 8 Zaključak

U ovome radu opisana je izrada pojednostavljene interaktivne simulacije protuoklopnog vođenog raketnog sustava primjenom alata Unity. Objašnjen je rad simulatora POVRS-a i što je to POVRS uopće. Cilj ovog projekta bio je upoznati se s procesom izrade računalnih igara u željenom programskom jeziku koji je u našem slučaju bio Unity. Za dobar konačan proizvod potrebno je točno utvrditi zahtjeve te kako će se oni na najefikasniji način ispuniti, stoga smo na početku se fokusirali upravo na to. Naišli smo na mnoge izazove prilikom izrade simulacije, no uz timski rad i prethodno znanje uspješno smo ih riješili.

# 9 Literatura

[1] Exclusive: Evolution of the Russian Federation's guided anti-tank and anti-personnel weapons [Online]. [https://pbrasil.wordpress.com](https://pbrasil.wordpress.com/2010/07/28/evolucao-dos-armamentos-anticarro-e-antipessoal-guiados-da-federacao-russa/) Datum pristupa: 30. prosinca 2022.

[2] raketno oružje [Online]. [https://enciklopedija.hr](https://enciklopedija.hr/natuknica.aspx?ID=51683) Datum pristupa: 30. prosinca 2022.

[3] ISS\_Lectures\_2022Fall\_1-7\_ISS\_Web.pdf [Online]. [https://www.fer.unizg.hr](https://www.fer.unizg.hr/_download/repository/ISS_Lectures_2022Fall_1-7_ISS_Web.pdf) Datum pristupa: 30. prosinca 2022.

[4] El-gabri, A. N. O., Ahmed, A. S., Elhalwagy, Y. Z., & Khamis, A. (2018). Command Guidance Missile Tracking Algorithms Evaluation Based On Visual Simulation. International Journal of Engineering Research and Technology, 7, 110-116. [Online]. <https://www.ijert.org/research/command-guidance-missile-tracking-algorithms-evaluation-based-on-visual-simulation-IJERTV7IS080043.pdf> Datum pristupa: 30.prosinca 2022.

[5] Unity asset store, <https://assetstore.unity.com/packages/3d/vehicles/land/tank-3d-model-225955>

[6] Unity asset store, <https://assetstore.unity.com/packages/3d/props/weapons/free-missile-72692>

[7] Unity asset store, https://assetstore.unity.com/packages/3d/environments/landscapes/low-poly-simple-nature-pack-162153

[8] Unity, <https://unity.com/>

[9] Visual Studio Code, <https://code.visualstudio.com/>

[10] Unity Asset Store, <https://assetstore.unity.com/>