

**TEHNIČKO VELEUČILIŠTE U ZAGREBU**  
**STRUČNI STUDIJ RAČUNARSTVA**

Antonio Peršin

JMBAG: 0246082708

**IZRADA DRUŠTVENE IGRE IGRANJA ULOGA**  
**TE IZRADA PROGRAMA ZA VIRTUALIZACIJU**  
**ISTE U PYTHONU**

ZAVRŠNI RAD br. 1305

Zagreb, rujan 2021.



## Sažetak

Cilj ovog rada je bio izrada novog sustava društvene igre igranja uloga i virtualne ploče koja bi se koristila kao alat u prikazu iste. Razvijen je sustav pod nazivom *404th Assault Squad*. Svijet i mehanike sustava su inspirirane raznim videoigrama i filmovima od kojih su velike inspiracije bile serijali videoigara *Fallout* i *Titanfall*. Glavna vodilja u izradi sustava i u tome kako prilagoditi mehanike koje dolaze iz videoigara je bilo peto izdanje knjige pravila za igrače društvene igra igranja uloga *Dungeons & Dragons*. Ploča je izrađena za izvanmrežnu uporabu, ali se uz neke druge vanjske alate za konferencijske pozive i dijeljenje ekrana, sustav također može igrati na daljinu. Aplikacija je u potpunosti programirana u Pythonu korištenjem objektno orijentirane paradigme. Sustav je prije implementacije u Pythonu razrađen pomoću Microsoft Office Excela te su dodani elementi koji olakšavaju unapređenje aplikacije i snalaženje voditelja igre.

## Sadržaj

1. UVOD.....	1
1.1. Društvene igre igranja uloga.....	1
1.2. Sustavi <i>TTRPG</i> igara.....	2
1.3. Dostupni alati.....	4
1.4. Izrada novog sustava .....	6
2. Nova društvena <i>RPG</i> igra i virtualna ploča.....	6
2.1. Preuzimanje i modificiranje mehanika iz videoigara .....	7
2.2. Dizajniranje i razrada sustava .....	8
2.2.1. Naziv sustava.....	8
2.2.2. Heroji .....	9
2.2.3. Statistike .....	10
2.2.4. Oprema.....	13
2.2.5. Ostali likovi u sustavu .....	14
2.2.6. Okoliš.....	14
2.2.7. Ploča.....	15
2.3. Implementacija sustava .....	16
2.3.1. Korišteni alati .....	16
2.3.2. Prijenos podataka iz Excela u aplikaciju .....	19
2.3.3. Model aplikacije .....	20
2.3.4. Konačni izgled aplikacije.....	25
2.4. Vrednovanje sustava i aplikacije.....	27
2.4.1. Kako koristiti aplikaciju i igrati <i>404th Assault Squad</i> .....	28
2.4.2. Rezultati prvih ispitivanja .....	33
2.4.3. Istaknuti problemi i njihova rješenja .....	34

3. Zaključak .....	37
4. Popis literature .....	39
Prilozi.....	41

## Popis kratica:

<i>RPG</i>	<i>role playing game</i> , (hrv. igra igranja uloga)
<i>TTRPG</i>	table-top role playing game, (hrv. društvena igra igranja uloga)
<i>404th</i>	<i>404th Assault Squad</i> , autorov novi <i>TTRPG</i> sustav
<i>D&amp;D</i>	<i>Dungeons &amp; Dragons</i> , (hrv. Tamnice & Zmajevi) prvi <i>TTRPG</i> sustav/igra
<i>dX</i>	igrača kocka sa X strana odnosno X mogućih ishoda
<i>DM</i>	<i>dungeon master</i> (hrv. voditelj igre, u <i>D&amp;D</i> sustavu)
<i>GM</i>	<i>game master</i> (hrv. voditelj igre, opći naziv)
<i>NPC</i>	<i>non-playable character</i> (hrv. lik kojeg igrač ne može kontrolirati)
<i>PC</i>	<i>playable character</i> (hrv. lik kojeg igrač kontrolira, heroj)
<i>TB</i>	<i>turn-based</i> (hrv. na poteze, način igre)
<i>RT</i>	<i>real-time</i> (hrv. u stvarnom vremenu, način igre)
<i>XP</i>	<i>experience points</i> (hrv. iskustveni bodovi)
<i>HP</i>	<i>hit points / health points</i> (hrv. životni bodovi)
<i>AC</i>	<i>armor class</i> (hrv. (klasa) oklop(a), kolike su šanse da lik izbjegne štetu)
<i>PDF</i>	<i>Portable Document Format</i> (hrv. format prenosivih dokumenata)
<i>GUI</i>	<i>Graphical User Interface</i> (hrv. grafičko korisničko sučelje)
<i>JSON</i>	<i>JavaScript Object Notation</i> (hrv. zapis JavaScriptovih objekata)
<i>UML</i>	<i>Unified Modeling Language</i> (hrv. unificirani jezik modeliranja)

## Ključne riječi:

Python aplikacija, društvena igra igranja uloga, virtualizacija, objektno-orijentirana paradigma, sustav TTRPG, virtualna ploča, mehanike video igara

## Popis slika

Slika 1: Set kockica za TTRPG igre, Amazon trgovina [5] .....	2
Slika 2: Ploča s minijaturama i figuricama za dočaravanje mjesta radnje, picclick.com [6] .....	3
Slika 3: Prikaz sučelja aplikacije Roll20 iz perspektive igrača, Roll20 [7] .....	5
Slika 4: Dijagram entiteta u igri sa prikazanim ovisnostima, autorski rad .....	13
Slika 5: Primjer postavljene ploče unutar aplikacije, autorski rad .....	16
Slika 6: Primjer tekstualne datoteke s podacima iz Excel tablice, autorski rad .....	19
Slika 7: Isti podatci sa slike 6 nakon formatiranja pomoću napisane skripte, autorski rad .....	20
Slika 8: UML dijagram klasa entiteta korištenih u aplikaciji, autorski rad .....	21
Slika 9: UML dijagram klase glavnog prozora, autorski rad .....	22
Slika 10: Izgled glavnog prozora aplikacije, autorski rad .....	25
Slika 11: Ikona koja predstavlja uništenog meka, autorski rad .....	26
Slika 12: Primjer prozora za kreiranje likova, autorski rad .....	30
Slika 13: Prikaz detalja NPC pilota, autorski rad.....	31
Slika 14: Uređivanje opreme lika u istom prozoru prikazanom na slici 13, autorski rad .....	31
Slika 15: Izbornik za kreiranje ploče željene veličine, autorski rad .....	32
Slika 16: Prozor za prikaz statistika bitnih za bitke, autorski rad.....	33

# 1. UVOD

## 1.1. Društvene igre igranja uloga

„Once upon a time, long, long ago, in a realm called Midwestern United States - specifically the states of Minnesota and Wisconsin - a group of friends gathered together to forever alter the history of gaming.“ [1]

Knjiga pravila prve društvene igre igranja uloga započinje paragrafom koji na specifičan način sumira svoju povijest, a i povijest drugih sličnih društvenih igara, pa iz toga i videoigara. Bez računala, bez ekrana, kako ući u svijet fantazije? Nikada nam ne treba ništa osim volje da uđemo u bilo koji svijet mašte i upoznamo nekog vilenjaka. Tako se grupa prijatelja iz američkih saveznih država Minnesote i Wisconsinina sakupila kako bi se zajedno upustili u pustolovinu unutar svijeta mašte te nenamjerno zaintrigirali brojne buduće generacije. Započeli su prvu kampanju, prve društvene igre igranja uloga, *Dungeons & Dragons – D&D*, u prijevodu „Tamnice & zmajevi“. Svijet *D&D* igre, kako je najstariji, tako je i najpopularniji svijet, točnije sustav igranja društvene igre igranja uloga, engl. *table-top role playing game - TTRPG*. Ne samo da je *D&D* prva društvena igra igranja uloga, nego je i prva igra igranja uloga uopće [1]. Godine 1974., kada je izmišljena [2], na računalu smo u najboljem slučaju mogli igrati Pong. [3]

*D&D* igra i sve ostale *TTRPG* igre se temelje na tri glavna koraka koji se cijelo vrijeme ponavljaju i čine temelj igre. To su: opisivanje, odlučivanje i bacanje [4].

Opisivanje je ključan element jer se cijela igra odvija u našim maštama. Bitno je opisati što se događa i kako nešto izgleda kako bi svi igrači imali sličnu predodžbu o tome gdje se nalaze i što se događa.

Odlučivanje je element koji pokreće igru. Svi igrači upravljaju nekim likovima i s njima rade što misle da je najbolje za njih. Odluke u *TTRPG* igrama, za razliku od onih u videoigramama nisu ograničene dizajnom igre [4]. Videoigre imaju dizajn koji igračima omogućuje ili onemogućuje određene akcije, ali kako mašti nema kraja, tako nema niti ograničenja u dizajnu. Likovi u *TTRPG* igrama mogu pokušati napraviti bilo što čega se igrač sjeti.



Likovi mogu baš to - pokušati. Budući da nemaju svi likovi iste sposobnosti i kako niti u stvarnom svijetu ne ide baš sve uvijek kako smo zamislili, uvodi se treći temelj *TTRPG* igre, bacanje igračih kocaka. Kocke dolaze u mnogim oblicima, jer se ne može uvijek sve svesti na šest različitih ishoda. Kocke se označavaju oznakom  $NdX$ . Gdje je  $N$  broj kocaka koji se baca,  $d$  oznaka da se radi o kocki, a  $X$  je broj strana, tj. vrijednosti koje se mogu dobiti bacanjem iste. Oznaka za broj kocaka koji se baca može biti i izostavljena ako se pretpostavlja da se radi o jednoj kocki. Glavna kocka, koja se najčešće koristi, je ona s 20 strana, dakle  $d20$ . Primjer jednog seta kockica za *TTRPG* igru prikazan je na slici 1. Bacanjem kocki i uspoređivanjem rezultata vidimo jesu li odluke koje smo donijeli ranije uspješno provedene. Tako se zatvara ciklus, bilo da je naša odluka uspješno provedena ili nije te se ponovno kreće u opisivanje novonastale situacije [4].



Slika 1: Set kockica za *TTRPG* igre, Amazon trgovina [5]

## 1.2. Sustavi *TTRPG* igara

Budući da se igra razvija, kao što danas programi imaju zakrpe i ažuriranja, tako i *TTRPG* igre imaju svoja ažuriranja. Ona dolaze više kao smjernice, nego kao pravila, u pismenom ili digitalnom obliku. Na primjer, specifično za *D&D* igru postoje priručnici

*Player's Handbook* i *Dungeon Master's Guide*. Danas je za *D&D* igru aktivno peto izdanje knjige *Player's Handbook*, iz 2014. godine, te stoga kažemo da je aktivno peto izdanje igre *D&D*. Sama ta knjiga ne sadrži neku specifičnu priču ili kampanju koju bismo mogli odmah zaigrati, nego sve potrebno za stvaranje jedne takve priče i svijeta, ili način kako igrati već kreirani svijet iz nekih javno objavljenih kampanja.

Skup pravila, entiteta i svijeta zove se sustav. Pravila se odnose na svijet, neprijatelje i heroje. Heroji su likovi u igri koje igrači kontroliraju i kako su oni najbitnija stavka igre, uglavnom se cijeli sustavi grade na mogućnostima heroja, pa onda i na svijetu u kojem se ti heroji nalaze. Jedan igrač nema heroja i on predstavlja sve ostale likove u igri. On se zove voditelj igre (engl. *game master* – *GM* ili engl. *dungeon master* – *DM* u sustavu *D&D* igre). Voditelj igre predstavlja sve ostale likove u igri koje igrači koji igraju heroje ne kontroliraju (engl. *non-playable character* – *NPC*). On također kontrolira tijek priče, dočarava drugim igračima svijet kroz priču i možda neke druge alate. Danas je dosta popularno korištenje igraće ploče s figuricama i minijaturama (slika 2), kartice entiteta u igri, pa tako i miksete sa zvučnim efektima i brojni drugi. Voditelji igre su najviše zaduženi za opisivanje situacija u kojima se nalaze heroji. Ostali igrači su više zaduženi za odlučivanje, a svi igrači bacaju kockice kako bi saznali ishod odluke koju su donijeli.



Slika 2: Ploča s minijaturama i figuricama za dočaravanje mjesta radnje, [picclick.com](https://picclick.com) [6]

Zapamtimo, *D&D* je igra u kojoj igrači sami grade svoju priču, stoga nas sami autori i u knjizi pravila potiču na kreativnost i potiču nas da napravimo svoje svjetove, svoje heroje i neprijatelje. Godine 1981. je na temelju već popularnog *RPG* sustava *D&D* izmišljen *RPG* u svijetu u kojem istražitelji istražuju misterij Katulu kulta, *Call of Cthulhu*. To je danas drugi najpopularniji *TTRPG* sustav, po *Dicebreakerovom* članku sa samog početka 2020. godine. [2] Uz navedena dva sustava igre, postoje i mnogi drugi. Cilj ovog rada je uz virtualizaciju jednog takvog sustava, također i izmisliti novi sustav.

### 1.3. Dostupni alati

Društvene *RPG* igre se razlikuju od današnjih videoigara po tome što im je glavna svrha okupiti prijatelje na jednom stvarnom mjestu, kao klasične društvene igre. Ovisno o sustavu igre koriste se različiti alati, ali kada bi govorili o klasičnim sustavima, u koje spadaju svi ranije spomenuti, za sve se koriste različiti tipovi kocaka te papir i olovka (ili neka aplikacija za praćenje statistika). Igraće kocke odlučuju sve, od štete, zdravlja, pa do toga je li naš heroj uspio uvjeriti nekog drugog lika o svojim namjerama.

Uz ove glavne „alate“ većina voditelja igre koristi knjige ili vlastite bilješke kako bi mogli dočaravati svijet i odraditi svoj zadatak vođenja igre što lakše. Već je spomenuto da ovisno o sustavu igre i tome koliko je tko spreman potrošiti na neke detalje, *online* trgovine nude brojne dodatke kako bi se iskustvo poboljšalo. Na primjer, igraće ploče služe kako bi mogli svi igrači zamisliti sličnu situaciju i točne pozicije neprijatelja i saveznika u bitci.

Nekada su nečije glavne prednosti ujedno i glavne mane. Tako i društvene igre imaju manu to što su društvene. Kako bismo svi mogli zaigrati igru morali bismo se sastati na jednom mjestu i izvaditi papir i olovku. Danas to naravno više nije tolika prepreka. Imamo mnogo mogućnosti, na primjer konferencijske pozive pomoću raznih programa, pa tako i cijele *web* aplikacije koje su osmišljene za igranje društvenih igara, time i *RPG* igara, na daljinu. U nastavku ću ukratko predstaviti dvije mrežne aplikacije za igranje društvenih *RPG* igara, *Roll20* i *Astral tabletop*.

Mrežni alati zahtijevaju internetsku vezu kako bi svi igrači mogli sudjelovati. Prednost je u tome što nisu potrebne nikakve instalacije dodatnih programa na našem osobnom računalu. Nedostatak je to što ovisimo o našoj internetskoj vezi jer o njezinoj kvaliteti može ovisiti korisničko iskustvo. Jedna od prednosti koju također imaju obje aplikacije je velika mogućnost personalizacije svih elemenata igre, koju možda ne bi imali na gotovoj ploči, danim figuricama i preuzetim, gotovim pričama, tj. kampanjama. To naravno zahtijeva dobro poznavanje odabrane aplikacije, ali je krajnji rezultat u većini slučajeva vrijedan truda. Današnje mrežne aplikacije ne bi bile uspješne da nisu vrlo prilagodljive. Zato i *Roll20* i *Astral tabletop* aplikacije pružaju isto iskustvo neovisno o uređaju na kojem im se pristupa i s kojeg sudjelujemo u igri. Obje aplikacije su prilagodljive i sa strane igrača, pa zato nude mnogo sustava koje igrači mogu igrati te biblioteke s gotovim kampanjama i drugim predefiniranim elementima igre. Obje aplikacije nude većinu svojih funkcionalnosti potpuno besplatno, no isto tako obje imaju plaćeno članstvo koje nudi nekolicinu dodatnih opcija i mogućnosti. [7], [8] Primjer jedne postavljene virtualne ploče u aplikaciji Roll20 je prikazan na slici 3. Većinu ekrana zauzima sami prikaz ploče, dok se s desne strane nalazi prostor za razmjenu tekstualnih poruka u kojem se također prikazuju ishodi bacanja virtualnih kockica.



Slika 3: Prikaz sučelja aplikacije Roll20 iz perspektive igrača, Roll20 [7]



#### 1.4. Izrada novog sustava

Već je rečeno kako svaki sustav ima neko svoje vrijeme i mjesto radnje koje odlučuje o tome što se sve nalazi u svijetu i ograničava mogućnosti *NPC* i *PC* likova. Isto tako svaki sustav ima nešto drugačija pravila i nešto drugačije statistike igrača te svih ostalih likova u svijetu. S obzirom na dosadašnje iskustvo igranja sustava koji su bili neslužbeni i na to da stvaranje ploče za neki postojeći sustav zahtijeva podosta iskustva s tim određenim sustavom, u ovom radu je, prije izrade ploče, izrađen vlastiti sustav.

Na temelju vlastitog iskustva i *Player's Handbook* priručnika dobivene su glavne smjernice za izradu jer je bitno prije razrade sustava znati na što treba obratiti pozornost. Elementi igre koji su bitni za shvatiti su razlike između videoigara i *TTRPG* igara: kretanje likova, interakcija likova s predmetima iz svijeta u kojem se nalaze i napredak likova kroz priču. Uz *Player's Handbook* priručnik kao vodilju u tome ŠTO treba napraviti, korištena su znanja dugog iskustva s videoigrama za KAKO napraviti određene elemente. Glavne inspiracije iz videoigara su došle iz *Fallout* serijala videoigara te drugih igara poput *Titanfall*, *Warframe*, *Overwatch*, ...

## 2. Nova društvena *RPG* igra i virtualna ploča

U ovom poglavlju će biti detaljnije opisan razvijeni *TTRPG* sustav te implementacija istog u istoimenu aplikaciju. Aplikacija služi kao virtualna ploča koja će vizualno reprezentirati dijelove igre te tako zamijeniti potrebu za stvarnim pločama i figuricama.

## 2.1. Preuzimanje i modificiranje mehanika iz videoigara

Za ovaj sustav igre preuzete su najbolje i najprikladnije mehanike iz nekolicine videoigara koje su spojene u jednu igrivu cjelinu.

Prva navedena mehanika koja je nešto drugačija od uobičajenih videoigara je pomicanje likova u svijetu. Pomicanje likova u društvenim *RPG* igrama ovisi o stanju u kojem se nalaze. Glavno pitanje kod stanja lika je da li je u opasnosti ili ne. Uglavnom se to pitanje odnosi na to da li je lik u bitci ili izvan nje, ali može se odnositi na stajanje na zamci ili drugim situacijama. Ako su likovi u bitci onda se igra treba odvijati sporije kako bi svaki igrač mogao odlučiti i opisati voditelju igre i drugim suigračima što radi njegov heroj i po potrebi baciti igraće kocke za odluku je li uspio u svojoj namjeri. Takav sustav bitke zovu se potezi, engl. *turn-based* - *TB*. Najlakše je takav sustav zamisliti kao da vrijeme stane, zatim se svi igrači dogovore što će napraviti i zatim se vrijeme nastavi. Nakon nekoliko sekundi u svijetu igre, vrijeme ponovno stane te se igrači ponovno dogovaraju što će učiniti. Taj ciklus se ponavlja, sve dok se ne razriješi sukob ili druga slična situacija. *TB* sustav kretanja ne bi imao smisla ako naš heroj nije u opasnosti, dok recimo radi svakodnevne stvari. Zato se u takvim situacijama vrijeme i radnje izvršavaju u stvarnom vremenu, engl. *real-time* – *RT*. Postoji nekoliko videoigara koje baš na taj način omogućavaju kretanje igrača, sve one zapravo do neke mjere vuku korijene iz originalne *D&D* igre.

Sljedeća mehanika su interakcije likova sa svijetom oko njih koje opisuju kako likovi djeluju na svijet oko sebe i kako svijet koji ih okružuje djeluje na njih same. U ljudskom iskustvu to se događa kroz osjetila mirisa, okusa, vida, sluha i dodira. *Fallout* serijal je prva videoigra koja je navedene procese, još krajem 90-ih, sumirala u sedam glavnih statistika lika. To su bili snaga, percepcija, izdržljivost, karizma, inteligencija, agilnost i sreća. Skraćeno se za takav sustav koristi naziv *SPECIAL*, po akronimu koji proizlazi iz engleskih naziva tih statistika: ***Strength***, ***Perception***, ***Endurance***, ***Charisma***, ***Intelligence***, ***Agility***, ***Luck***. [9] Tako je i u ovom radu preuzet i dorađen sustav kako bi odgovarao aktualnom vremenu radnje i, za razliku od videoigre, sustavu *TTRPG* igre.

Posljednji element je sam napredak likova kroz igru koji je već ustaljen u mnogim kako videoigrama tako i društvenim *RPG* igrama. To je sustav iskustvenih bodova, engl.

*experience points* – *XP*. Heroji napreduju tako što završavanjem zadataka dobivaju *XP* bodove koje troše na unapređenje likova, a u mom sustavu i opreme.

## 2.2. Dizajniranje i razrada sustava

U nastavku će detaljno biti objašnjeni elementi sustava. Već je navedeno da je cijeli sustav izmišljen uzimajući inspiraciju iz petog izdanja *D&D* i prikladnih videoigara te drugih medija. *D&D* sustav potiče svoje igrače na izmišljanje novih svjetova i priče, pa čak i prilagodbu svojih pravila kako bi pristajale tim svjetovima. Takvi elementi sustava koji se razlikuju od objavljenih pravila i kampanja, u zajednici igrača zovu se *homebrew* (hrv. domaći, domaće izrađeni) elementi. Budući da ovdje razvijeni sustav samo uzima glavne smjernice iz originalne *D&D* igre, možemo reći da je cijeli sustav *homebrew*, jer je veći dio sustava izmišljen ili prilagođen, a ne izravno preuzet odnekuda.

### 2.2.1. Naziv sustava

Svaki sustav ima ime koje odražava priču koju prati. *Dungeons & Dragons* je sada već toliko velik svijet da ime ne reflektira sve likove i priče u njemu. Dok su one najranije kampanje iz 70-ih i 80-ih uglavnom uključivale pustolovine u tamnicama (engl. *dungeons*) ili ubijanje zmajeva (engl. *dragons*), u *Call of Cthulhu* se radi o detektivima koji istražuju misterije vezane uz Katulu kult. Katulu je fiktivno mistično biće koje je u davnoj, fiktivnoj prošlosti bilo smatrano božanstvom i koje počiva ispod Pacifičkog oceana [10], prvi je puta spomenuto u Howard Philips Lovecraftovoj horor priči iz 1920-ih godina [11]. Sustav razvijen u ovom radu se naziva *404th Assault Squad*, a naziv je inspiriran kodom greške koja se javi kada se zatraže podatci koji nisu dostupni, kôd 404. Sustav 404th reflektira priču naših heroja koji su nakon problema s prethodnom vojnom organizacijom, čijim su bili dijelom, morali napustiti istu. Unovači ih mala skupina plaćenika koja im nudi novu priliku i svu potrebnu opremu, u zamjenu za

njihove vještine koje su stekli u prošlosti. Kako su plaćenici imali opremu, ali nisu imali ljude, tj. vojnike, naši heroji zajedno s novopronađenim kolegama uspiju stvoriti autonomni odred koji zajedno putuje sunčevim sustavom i rješavaju misije. Misije *GM* može sam dizajnirati tako da su međusobno neovisne jedna o drugoj ili kao ulančani niz misija koje zajedno tvore priču koju *GM* želi ispričati igračima. To mogu biti misije spašavanje neke zajednice, zaštite konvoja ili spašavanja likova.

Isto kao i kod *D&D* sustava, naziv sustava je proizašao iz određene ideje o svijetu ili likovima. U ovom slučaju to je ideja o tajnom redu plaćenika koji spašavaju svijet, ili svjetove, od raznih zlikovaca. No, ime se ne mora reflektirati u drugim kampanjama koje bi se igrale u sustavu *404th Assault Squad*.

### 2.2.2. Heroji

Heroji su zamišljeni kao plaćenici koji imaju prethodno vojno iskustvo. Radili su za vojnu organizaciju u kojoj su stekli vještine rukovanja oružjem i upoznali se s raznim tehnologijama kojima se svakodnevno služe. Glavna tehnologija koja im pomaže u borbi su veliki vojni roboti kojima mogu upravljati iz kabine unutar samog robota. Ovakva tehnologija je popularizirana videoigrama iz serijala *Titanfall* i filmovima poput *Avatara* i serijalom *G. I. Joe*. Naši heroji se nazivaju pilotima jer svaki od njih ima vlastitog robota kojim može upravljati. Isti naziv se koristi i u igri *Titanfall*. Kada su heroji izvan svojih robota, igrači upravljaju pilotima. Kada heroji pozovu i uđu u svog robota, igrači upravljaju robotom, tj. mekom (engl. *mech*). Igrači tako zapravo imaju dva heroja kojima upravljaju, pilotima i mekovima, ali nikada oba u isto vrijeme. Piloti i mekovi se razvijaju posebno te svaki igrač može birati što i kako će razviti svoje heroje. To dovodi do povezanosti igrača sa svojim herojima jer su heroji u potpunosti izrađeni na temelju igračevih odluka i razmišljanja. Također, igrači sami biraju izgled svojih likova i tu imaju najveću moguću slobodu, sve dok ostanu unutar smjernica koje im zada voditelj igre, ovisno o tome kakav svijet izrađuje.



### 2.2.3. Statistike

Sposobnosti likova za interakciju sa svijetom oko njih se odražavaju u njihovim primarnim statistikama. Iako bacanje kockica odlučuje je li neka akcija uspješna ili ne, primarne statistike odlučuju kolika je šansa za uspjehom, a kolika za neuspjehom. Već je navedeno da se primarne statistike temelje na *S.P.E.C.I.A.L.* statistikama iz *Fallout* serijala te da su za potrebe sustava modificirane kako bi imale smisla u svijetu igre. Između svih primarnih statistika jedino je karizma (engl. *C - charisma*) lika izbačena iz sustava radi poticanja interakcije između voditelja igre i igrača. Karizmatičnost lika se očituje kroz interakcije igrača s drugim likovima u igri dok govore ili djeluju u liku, odnosno dok rade ono što njihov lik radi trenutno u svom svijetu. Karizma bi se najrijeđe koristila kao primarna statistika koja se provjerava te bi bilo nepotrebno poboljšavati istu. Igrači bi ju uglavnom zanemarivali ili se loše osjećali kada bi shvatili koliko malo ima utjecaja na igru. Ostale primarne statistike su ostale iste i uglavnom pokrivaju ista područja kao i u videoigri *Fallout*. Minimalna vrijednost svih primarnih statistika je jedan, a najveća je 10. Takav raspored odgovara desetostranoj igračkoj kocki koja se koristi pri odlučivanju je li uspjela neka akcija koja zahtijeva određenu statistiku.

Snaga (engl. *S - strength*) mjeri fizičku snagu pilota. Piloti mogu pomicati teške stvari u svijetu i snaga odlučuje kolike su šanse da pomaknu takav predmet. Sva oružja koja koriste piloti također zahtijevaju određenu snagu kako bi se mogla koristiti.

Percepcija (engl. *P - perception*) označava koliko dobro piloti primjećuju stvari oko sebe i koliko dobro mogu razaznati predmete u daljini. Oružja koja ispucaavaju projekte, kao i određene nadogradnje odijela imaju zadane minimalne vrijednosti percepcije kako bi se mogli koristiti.

Izdržljivost (engl. *E - endurance*) prikazuje koliko je pilot otporan na fizičku štetu koja mu se nanosi. To uglavnom uključuje projekte i razna druga oružja. Iz izdržljivosti proizlaze nove dvije statistike, životni bodovi (engl. *hit points / health points – HP*) i oklop (engl. *armor class – AC*). Uz podnošenje ozljeda, izdržljivost ukazuje i na sposobnosti tijela pilota na ugradnju modifikacija. Piloti mogu ugraditi kibernetičke proteze u svoje tijelo kako bi poboljšali svoje sposobnosti.

Inteligencija (engl. *I - intelligence*) okrija mentalne sposobnosti pilota. Pametniji piloti mogu koristiti kompliciranije nadogradnje na svojim odijelima i mekovima. Piloti s

većom inteligencijom brže stječu nova znanja tako da im je potrebno manje iskustvenih bodova za unapređenje lika. Likovi mogu unaprijediti svoje statistike i opremu koristeći resurs koji dobivaju prilikom svakog unapređenja razine ili kada ga pronađu u svijetu. Ti resursi se nazivaju *tech points*. Oni se koriste za unapređivanje opreme, robota i statistika igrača, ali igrači to ne mogu napraviti bilo kada. Svako od različitih unapređivanja likova se mora napraviti na posebnim mjestima predviđenim za to. Npr. mekovi se mogu unaprijediti jedino u radionici koja je za to namijenjena.

Agilnost (engl. *A - agility*) mjeri koliko je lik brz i sposoban pri savladavanju prepreka. Brzina lika je bitna u dijelovima igre kada likovi djeluju po potezima. Agilniji likovi mogu napraviti više radnji od onih manje agilnih. Neke modifikacije odijela također imaju zahtjeve za određenom vrijednošću agilnosti. Ako se u svijetu heroj treba nekamo popeti, agilnost odlučuje hoće li to biti uspješno, povećavajući šansu uspjeha.

Na kraju, bilo da nam nešto ide jako dobro ili loše, sreća (engl. *L - luck*) uvijek može malo okrenuti stvari. Sreća se manifestira u gotovo svim prethodno navedenim aspektima. Recimo, neuspjeli pokušaj penjanja ne mora uvijek završiti sa slomljenom nogom, no može završiti i puno gore. U klasičnom *D&D* sustavu ne postoji sreća kao statistika, ali ima ulogu u igri. Kada bi igrač dobio 20 na *d20* to se zove kritični uspjeh (engl. *critical success*), a kada bi dobio 1 to se naziva kritični neuspjeh (engl. *critical fail* ili *fumble*). Ovdje su zadržani kritični uspjesi i neuspjesi, ali također je i sreća kao statistika dodana u igru pa kritični uspjesi i neuspjesi ne znače automatski i apsolutne uspjehe ili neuspjehe. Ovisno o statistici sreće likovi imaju veće ili manje šanse za izbjegavanjem kritičnih neuspjeha ili izbjegavanjem posljedica kritičnog uspjeha postignutog od strane protivnika nad likom u pitanju. Odluka se donosi na temelju bacanje *d20* kockice, a veća statistika sreće znači veću šansu za izbjegavanje neželjenih posljedica kritičnih uspjeha ili neuspjeha.

Već su spomenuti životni bodovi. Kada lik dođe na nula životnih bodova ili bi otišao u negativne vrijednosti, lik bespovratno umire. Heroji se mogu spasiti ugrađivanjem čipova koji im nakon što pilot umre omogućavaju da prenesu statistike na novog pilota. Nažalost, pilot koji umre je i dalje mrtav. Mekovi kada dođu do nula životnih bodova u sljedećem potezu eksplodiraju. Pilotima je dan jedan potez kako bi se spasili. Iste su mehanike implementirane u videoigri *Titanfall*. [12] Klasični *D&D* kao i mnogi drugi sustavi imaju različite načine kako tretiraju smrti, kako heroja tako i neprijatelja. U

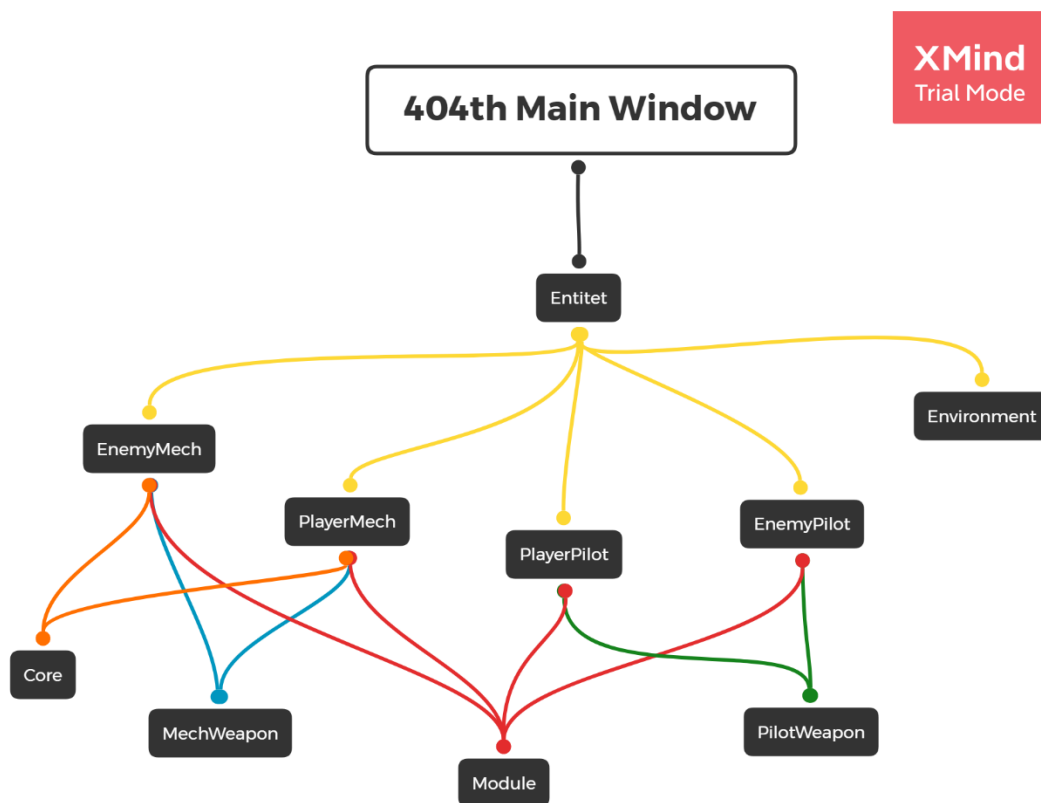
mnogima smrt ne mora značiti kraj igre za tog heroja, a pogotovo ne za igrača. U krajnjem slučaju se uvijek može napraviti novi lik.

Teško bi bilo balansirati igru kada bi svi likovi samo primali štetu i zacjeljivali od iste. Zato se uvodi način izbjegavanja štete. U videoigrama to se omogućuje korisničkim kretnjama i statistikama heroja. Kako u *TTRPG* igrama ne možemo točno odrediti poziciju svakog dijela tijela u svakom trenutku te tako vidjeti je li došlo do pogotka ili ne, moramo uvesti statistike koje nam to govore. Različiti sustavi igre imaju različite načine kako to implementiraju. U ovom slučaju sustava *404th* se radi kombinaciju oklopa iz *D&D* sustava [1] i klase oklopa (*AC*) iz starijih videoigara serijala *Fallout* [13]. U oba slučaja *AC* utječe na šansu da lik biva pogođen od neprijatelja. Neprijatelj baca *d20* i ako dobije veći iznos od iznosa *AC* nanosi štetu meti. Ako dobije 20, dobiva *critical success* čime može nanijeti više štete ili čak oštetiti odijelo ili mek toliko da ograničava sposobnosti žrtve. Ako pak dobije 1 može oštetiti svoje oružje.

U svakom potezu likovi imaju određeni broj akcija koje mogu napraviti u trenutnom potezu. Spomenuto je kako na to utječe agilnost, no također utječu i drugi faktori koji ovise o situaciji. Dok se u *D&D* sustavu radnje u potezu dijele na pomicanje i akcije, u sustavu *404th* akcije se dijele u tri kategorije: pomicanje, napadi i akcije. Pomicanje se odnosi samo na pomicanje likova po polju. Kako bi svi likovi znali koliko se daleko smiju pomaknuti u jednom pomicanju potrebno je odrediti mjerne jedinice za kretanje. Sustavi društvenih *RPG* igara uglavnom koriste stvarne mjerne jedinice poput metara i stopa, ali u ovom sustavu *404th* se gleda više u smjeru videoigara te je osmišljena mreža koja će u svakom trenutku dijeliti okruženje heroja na jedinične cjeline. Tako je riješeno kretanje, ali i domet oružja te veličina likova na ploči. Likovi su veličine 1/9 jediničnog kvadrata ploče. Kretanje tijela na mjestu nisu uključene i uglavnom ne troše akcije. Napadi se odnose samo na napadanje meta. U to je uključeno i hakiranje, hvatanje i drugi oblici svladavanja mete. Akcije se odnose na sve ostale radnje, ali mogu biti korištene i za bilo koju od navedenih radnji, napad ili pomak. Nakon što svi likovi potroše svoje akcije, ili ih odluče preskočiti, gotov je potez te se krug ponavlja dok se konflikt ne razriješi.

## 2.2.4. Oprema

Mekovi i piloti mogu koristiti dva tipa opreme: oružja i module. Uz opremu, svakog od mekova pokreće jedna od dva tipa jezgara: standardna ili nuklearna. Sva oprema je izmišljena i inspirirana različitim videoigrama, filmovima i stvarnim svijetom. Oružja su podijeljena u kategorije. Mekovi i piloti ne mogu koristiti ista oružja, no neka imaju svoje inačice za pilote i za mekove. Oružja koja koriste projektele koriste različite tipove streljiva koje se može pronaći u svijetu. Moduli su različiti alati koji se ugrađuju u mekove i pilotovo odijelo. Nemaju neki univerzalni oblik u kojem dolaze jer mogu doći kao čip koji se ugrađuje u kacigu ili stabilizator koji se montira na meka. Postoji preko 30 oružja za pilote, isto tako i za mekove, te preko 60 modula. Ovakav fleksibilan sustav omogućava igračima da eksperimentiraju s različitom opremom koju lako mogu izmijeniti između misija. Neke kombinacije mogu biti vrlo učinkovite, a neke malo manje. Na slici 4 su jednostavnim dijagramom predstavljene sve vrste opreme koje su povezane s likovima koji je koriste. Na isti način će oprema i entiteti biti povezani u aplikaciji, što će biti objašnjeno kasnije.



Slika 4: Dijagram entiteta u igri sa prikazanim ovisnostima, autorski rad

### 2.2.5. Ostali likovi u sustavu

Osim igrača svaka igra ima i neprijatelje protiv kojih se igrači bore. U sustavu *404th* *NPC* likovi se dijele na neprijatelje i saveznike, te pilote (ili druge humanoide koji ne moraju nužno moći upravljati mekovima) i mekove (ili druge velike robote koji mogu biti i autonomni). Kod izrade likova se odabire pojavljuje li se lik u igri kao saveznik ili neprijatelj, a ovisno o tome je li pilot ili mek likovi se izrađuju na drugačije načine. Nije potrebno da svi likovi imaju sve navedene statistike, a neki čak ne trebaju niti ime. Ovisno o tome koliko bi oni bili bitni igračima toliko im se statistika može odrediti. Ako će oni biti samo prolaznici, bitan je samo općeniti izgled. Ako su neki bitni ljudi poput zapovjednika, onda je bitno i ime i osobnost. Ako će sudjelovati u bitci, onda moraju imati sve statistike koje su bitne za odvijanje bitke.

Za likove koji sudjeluju u bitci od glavnih statistika je bitna jedino sreća, jer ona jedino izravno sudjeluje u ishodima koji su bitni u igri. Sve ostale statistike su ili izvedene ili nebitne. Za snajperista možemo reći da ima dovoljno veliku snagu i preciznost da koristi snajper, a za hakera da ima dovoljno veliku inteligenciju za korištenje modula za hakiranje. Ostale sekundarne statistike kao životni bodovi (*HP*) i klasa oklop (*AC*) se mogu fiksno odrediti ako govorimo o *NPC* likovima. Kod *PC* likova te statistike se izračunavaju iz glavnih po određenoj formuli.

### 2.2.6. Okoliš

Ostale prepreke koje su nežive nazivaju se okoliš. Na slici 1 je on označen kao *Environment* (engl. okoliš) zato što će se taj naziv koristiti u implementaciji aplikacije. Okoliš ograničava kretanje likova po bojnom polju, a time i ploči. Okoliš nema potrebe imati statistike kao i živi likovi, ali ima dva glavna svojstva. Neki okoliš je moguće uništiti i time otvoriti novi put ili otkriti neprijatelja. Određena oružja imaju mogućnost uništavanja okoliša ili zaklona. Može li neko oružje uništiti određeni okoliš ovisi o razornoj moći samog oružja, koja nije posebna statistika, već sami igrači mogu to

logikom zaključiti. Recimo, top može napraviti rupu u zidu, dok obična sačmarica ne može. Primjeri okoliša su zid, drvo, stol i slično.

#### 2.2.7. Ploča

Sustav je predviđen da se igra na dva načina, ovisno o tome postoji li opasnost ili ne. Ako heroji nisu u bitci i samo prolaze svijetom bez opasnosti voditelj igre opisuje svijet i objašnjava što se događa oko njih. Mogu se koristiti bilješke i druga pomagala. Kada se likovi nalaze u opasnosti i postoje neprijatelji oko heroja, koristi se igraća ploča za predočavanje situacije (engl. *encounter*). Ploča je dvodimenzionalni pregled polja odozgo na kojem se događa bitka. Na ploču će voditelj igre moći postaviti neprijatelje, heroje i prepreke kako bi što bolje dočarao pozicije na bojnom polju. Heroji dolaze u dva oblika koja će kasnije biti spomenuta. Neprijatelji će biti prethodno izrađeni te će se lako moći postaviti i pomicati po ploči, isto kao i heroji. Prepreke uključuju sve objekte na fiktivnom bojnom polju koji ograničavaju pomicanje, poput zidova, ili iza kojih se likovi mogu stati u zaklon, poput niskih zidova, uglova, drva i sl.

Primjer ploče u nekoj situaciji igranja igre prikazan je na slici 5. Na slici su označeni predstavnici većine entiteta koji se mogu postaviti na ploču. Ovisno o pripadnosti likovi su označeni različitim bojama, crvena predstavlja neprijatelje, plava saveznike, a zelena heroje. Piloti su prikazani portretom, s kacigama na glavama, dok su mekovi prikazani gotovo u cijelosti. Iako su iste veličine na ploči, mekovi su do nekoliko puta veći od pilota, ali radi jednostavnosti, na ploči svi zauzimaju istu količinu prostora. Na slici je također označeno nekoliko entiteta koji predstavljaju okoliš. Gornje označeni okoliš sastavljen je od više istih dijelova koji zajedno tvore kameni zid. Donje označeni okoliš je zapravo drvo. Ono je predstavljeno samo deblom, kako krošnja ne bi zakrivala dio ploče, a time i dio prostora u igri po kojem bi se likovi trebali moći kretati (prostor ispod krošnje drveta).



programskog jezika, prethodno iskustvo i jedna autorska igra, u potpunosti izrađena u Pythonu, činili su bitne preduvjete za izradu jedne takve aplikacije. Kombinacija objektno orijentirane paradigme i proceduralnog programiranja omogućavaju potpunu implementaciju, čak i bez baze podataka. Baze podataka nisu neophodne jer sustav u trenutnom stanju ima tek oko dvjestotinjak entiteta kao što su oružja i moduli za heroje. No i dalje su entiteti implementirani sa svojim identifikacijskim brojevima i indeksima u listama. Cijela aplikacija se relativno lako može unaprijediti implementacijom baze podataka. Python i sama implementacija entiteta to jednostavno dopuštaju.

Python je interpreterski jezik visoke razine koji se fokusira na čitljivost koda. Implementira više paradigmi, od kojih je za ovu aplikaciju najvažnija objektno orijentirana paradigma. Njegova lakoća pisanja, ali i velike mogućnosti koje dolaze u obliku modula omogućile su mu integraciju s gotovo svim tehnologijama. Danas uporabu Pythona možemo pronaći na web stranicama, u kinematografiji, kod administracije sustava i brojnim drugim tehnologijama. [14] Prethodna upoznatost s tim programskim jezikom i njegovim modulom za izradu grafičkog korisničkog sučelja, Tkinter, olakšala je izradu virtualne ploče. Tkinter je sučelje između Tk *GUI* alata i Pythona. Tk *GUI* je alat za izradu grafičkog sučelja pisan u C-u. [15]

Već je spomenuto kako umreženost, koliko je i prednost, može biti i mana. Glavni nedostaci su ovisnost o poslužitelju i internetskoj vezi na obje strane. Ako postoje poteškoće bilo kod korisnika ili poslužitelja, aplikacija, ako će uopće raditi, to će biti otežano. Nadalje, uz virtualne igraće ploče gubi se dio čari društvenih igara, druženje s drugim ljudima i veća mogućnost interakcije. Jedna od prednosti izgradnje izvanmrežne virtualne igraće ploče je lakoća korištenja kod lokalnih sjednica igranja igre. Lokalnim sjednicama su situacije kada igrači stvarno svi sjede za jednim stolom i zajedno igraju igru. Kod lokalnih sjednica aplikacija može biti otvorena na jednom računalu i funkcionirati će kao fizička igraća ploča s dvodimenzionalnim pogledom na igru odozgo. Ploča može biti kontrolirana od strane voditelja igre, dok ostali igrači koriste papir i olovku za praćenje vlastitih statistika. Tako se tehnologija minimalno miješa u klasični koncept *TTRPG* igara.

Aplikacija se također vrlo lako može koristiti i u virtualnim sjednicama na daljinu. Trenutno opće stanje u svijetu zbog pandemije korona virusa nas je zapravo odvelo ka virtualnom svijetu i većinu poslova koja se može obavljati na daljinu poticalo se tako i obavljati. Uvid u prednosti i nedostatke takvog načina igranja na temelju iskustva



*online* kampanje *D&D* igre bio je motivacija i za izradu ovog alata. Aplikacija se koristi na isti način kao i kod lokalnih sjednica, a igranje na daljinu se ostvaruje preko programa za konferencijske pozive i dijeljenja ekrana. Osvrćući se na stanje pandemije, gotovo svako računalo danas ima instaliran jedan takav program neophodan za rad na daljinu. Budući da su tako lako dostupni, za aplikaciju ovakve složenosti, dobro ih je iskoristiti. Voditelj igre i dalje kontrolira ploču, te dijeli ekran svim ostalim sudionicima, koji su pak slobodni pratiti statistike svojih heroja ili na računalu ili opet papirom i olovkom.

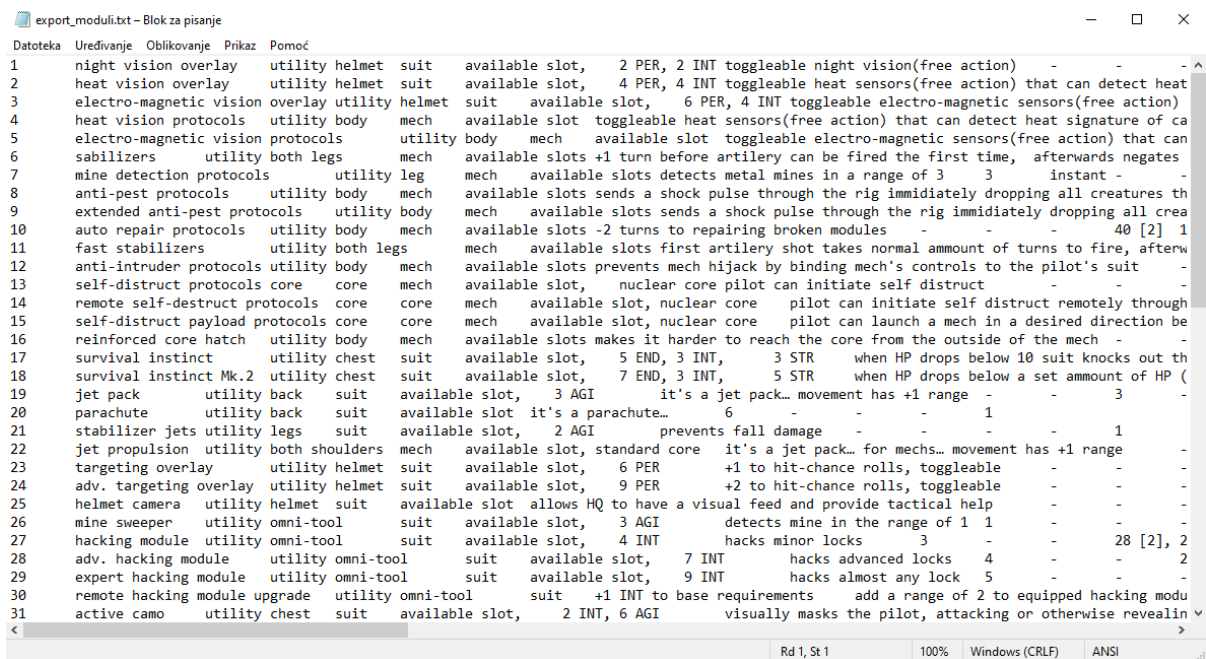
U duhu *TTRPG* igara većina pravila su zapisana u svojevrzni priručnik, dok aplikacija u Pythonu samo omogućava vizualizaciju igre. Voditelj igre vodi brigu o tome da se pravila poštuju, pa će ta sloboda i odgovornost na njima i ostati. U Microsoft Excel dokumentu se nalaze i ostale bilješke o pravilima igre koje se ne nalaze u aplikaciji, a mogu se pronaći u Excel datoteci koja je priložena uz ovaj rad. Osim toga, pohranjuju se i podatci o oružjima, modulima, neprijateljima i drugim elementima igre. Svaki entitet ima svoje statistike koje je bitno zapisati: šteta, učinak i sl. Dok se eventualno ne napravi implementacija s bazom podataka Excel se koristi za nacрте i prijedloge entiteta, prije unosa u bazu. Svakom entitetu je pridijeljen i identifikacijski broj kako bi se voditelju igre lako snašao u pronalasku traženog modula ili oružja. Identifikacijski brojevi također olakšavaju implementaciju baze podataka u daljnjem razvoju sustava i same aplikacije te omogućavaju lako dodavanje novih entiteta koji se dodaju u sustav, npr. novih modula.

Programski alat Microsoft Word se može koristiti za pisanje priče i drugih tekstualnih podataka. *D&D* svjetovi mogu postati jako veliki ako se gradi na njima i ako se kampanje nadovezuju jedne na drugu. Stoga je neophodno imati sve informacije na jednom mjestu. Jedan Wordov dokument je lako prenosiv i, uz dobro organizaciju, čitljiv. Veliki broj sustava, priča i kampanja *TTRPG* igara se također mogu preuzeti s interneta. Postoje besplatni resursi i oni koji se plaćaju. Većina ih dolazi u *PDF* formatu.

Izrada same priče i kampanje nisu bili cilj ovog projekta, već izrada potrebnih alata i niza pravila. Kao i za knjige originalne *D&D* igre, to može biti poticaj za buduće igrače i voditelje igre da izrađuju svoje priče i mjesta te da koriste pravila *TTRPG* igre *404th* kao smjernice pri izradi svijeta i likova. Iz tog razloga u materijalima projekta nisu uključeni tekstovi koji se odnose na ekonomiju, izgled svijeta (osim izgleda glavnih elemenata *404th*, pilota i mekova) te priču kampanje.

### 2.3.2. Prijenos podataka iz Excela u aplikaciju

Uporabom pomoćne skripte koja je razvijena nezavisno od glavne aplikacije, ali je korištena pri njezinoj izradi, mogu se dodavati novi entiteti u sustav iz Excelovih tablica spomenutih u prethodnom poglavlju. Skripta radi tako da se najprije podatci iz Excelovih tablica pretvaraju u tekstualne dokumente prikazane na slici 6. Skripta zatim pročita tekstualni dokument te liniju po liniju pretvara u kôd, dodajući navodnike, zagrade i sl. te mičući nepotrebne razmake i tabulatore. Ishod skripte se zatim liniju po liniju zapisuje u novi tekstni dokument prikazan na slici 7. Tako iz izvezene tablice iz Excela, dobivamo kôd koji za svaki red u Excelovoj tablici kreira objekt spajajući Pythonov kôd i tekst iz tablice. Taj se kôd sprema u novu tekstualnu datoteku iz koje se lako kopira u Pythonovu datoteku, a mogao bi se i izravno čitati iz tekstne datoteke uz nekoliko promjena. Podatci bi se također mogli spremati na sličan način kako se spremaju i podatci o kreiranim entitetima unutar aplikacije te serijalizirati u posebne datoteke koristeći *Pickle*, koji će biti pojašnjen kasnije.



```
export_moduli.txt - Blok za pisanje
Datoteka  Uređivanje  Oblikovanje  Prikaz  Pomoć
1  night vision overlay  utility helmet  suit  available slot,  2 PER, 2 INT toggleable night vision(free action)  -  -  -
2  heat vision overlay  utility helmet  suit  available slot,  4 PER, 4 INT toggleable heat sensors(free action) that can detect heat
3  electro-magnetic vision  overlay utility helmet  suit  available slot,  6 PER, 4 INT toggleable electro-magnetic sensors(free action)
4  heat vision protocols  utility body  mech  available slot  toggleable heat sensors(free action) that can detect heat signature of ca
5  electro-magnetic vision  protocols  utility  mech  available slot  toggleable electro-magnetic sensors(free action) that can
6  sabalizers  utility both legs  mech  available slots +1 turn before artillery can be fired the first time, afterwards negates
7  mine detection protocols  utility leg  mech  available slots detects metal mines in a range of 3  3  instant -
8  anti-pest protocols  utility body  mech  available slots sends a shock pulse through the rig immediately dropping all creatures th
9  extended anti-pest protocols  utility body  mech  available slots sends a shock pulse through the rig immediately dropping all crea
10 auto repair protocols  utility body  mech  available slots -2 turns to repairing broken modules  -  -  40 [2] 1
11 fast stabilizers  utility both legs  mech  available slots first artillery shot takes normal ammount of turns to fire, afterw
12 anti-intruder protocols  utility body  mech  available slots prevents mech hijack by binding mech's controls to the pilot's suit  -
13 self-destruct protocols  core  core  mech  available slot,  nuclear core pilot can initiate self destruct  -  -
14 remote self-destruct protocols  core  core  mech  available slot, nuclear core  pilot can initiate self destruct remotely through
15 self-destruct payload protocols  core  core  mech  available slot, nuclear core  pilot can launch a mech in a desired direction be
16 reinforced core hatch  utility body  mech  available slots makes it harder to reach the core from the outside of the mech  -  -
17 survival instinct  utility chest  suit  available slot,  5 END, 3 INT,  3 STR  when HP drops below 10 suit knocks out th
18 survival instinct Mk.2  utility chest  suit  available slot,  7 END, 3 INT,  5 STR  when HP drops below a set ammount of HP (
19 jet pack  utility back  suit  available slot,  3 AGI  it's a jet pack... movement has +1 range  -  -  3  -
20 parachute  utility back  suit  available slot  it's a parachute...  6  -  -  -  1  -
21 stabilizer jets  utility legs  suit  available slot,  2 AGI  prevents fall damage  -  -  -  1  -
22 jet propulsion  utility both shoulders  mech  available slot, standard core  it's a jet pack... for mechs... movement has +1 range  -
23 targeting overlay  utility helmet  suit  available slot,  6 PER  +1 to hit-chance rolls, toggleable  -  -  -
24 adv. targeting overlay  utility helmet  suit  available slot,  9 PER  +2 to hit-chance rolls, toggleable  -  -  -
25 helmet camera  utility helmet  suit  available slot  allows HQ to have a visual feed and provide tactical help  -  -  -
26 mine sweeper  utility omni-tool  suit  available slot,  3 AGI  detects mine in the range of 1  1  -  -  -
27 hacking module  utility omni-tool  suit  available slot,  4 INT  hacks minor locks  3  -  -  28 [2], 2
28 adv. hacking module  utility omni-tool  suit  available slot,  7 INT  hacks advanced locks  4  -  -  2
29 expert hacking module  utility omni-tool  suit  available slot,  9 INT  hacks almost any lock  5  -  -  -
30 remote hacking module upgrade  utility omni-tool  suit  +1 INT to base requirements  add a range of 2 to equipped hacking modu
31 active camo  utility chest  suit  available slot,  2 INT, 6 AGI  visually masks the pilot, attacking or otherwise revealin
```

Slika 6: Primjer tekstualne datoteke s podacima iz Excel tablice, autorski rad

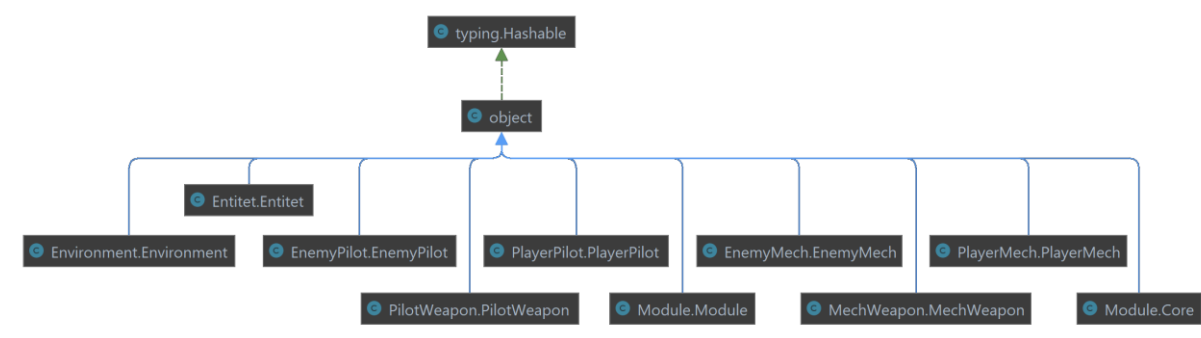


Slika 7: Isti podatci sa slike 6 nakon formatiranja pomoću napisane skripte, autorski rad

### 2.3.3. Model aplikacije

Aplikacija je u potpunosti pisana u Pythonu. Glavni modul koji je bio korišten je *Tkinter*. *Tkinter* se koristi za kreiranje svih elemenata sučelja i njihove funkcionalnosti. Također se koristio i *Pickle*. *Pickle* je modul koji sprema kolekcije, u ovom slučaju liste, objekata u datoteke na određenoj putanji. [16] Zapravo ima ulogu *JSON*-a u svijetu Pythona. Svi entiteti, oružja, moduli, neprijatelji, su implementirani kao objekti neke klase. Svi entiteti iste kategorije, npr. svi moduli, su objekti iste klase i dio jedne Pythonove liste. Liste su kolekcije podataka koje mogu sadržavati podatke različitih vrsta. U ovom slučaju ti podatci su objekti odgovarajućih klasa koje su predstavljene na UML (engl. *Unified Modeling Language*) dijagramu klasa na slici 8. Svaka klasa koja predstavlja neki entitet u sustavu sadrži varijable koje su svojstva tog entiteta. Za oružja su to: identifikacijski broj, ime, tip oružja, šteta koju radi, doomet, tip streljiva, kapacitet okvira, koliko se okvira može najviše nositi, koliko metaka ispali po jednom rafalu, koliko se rafala može napraviti u jednom napadu, koliko akcija treba da se promijeni okvir, koje se oružje dobije kada se trenutno poboljša, unaprijedi (po identifikacijskom broju), te kojeg je reda to oružje. Budući da je ovo puno informacija, neće sve biti vidljive u

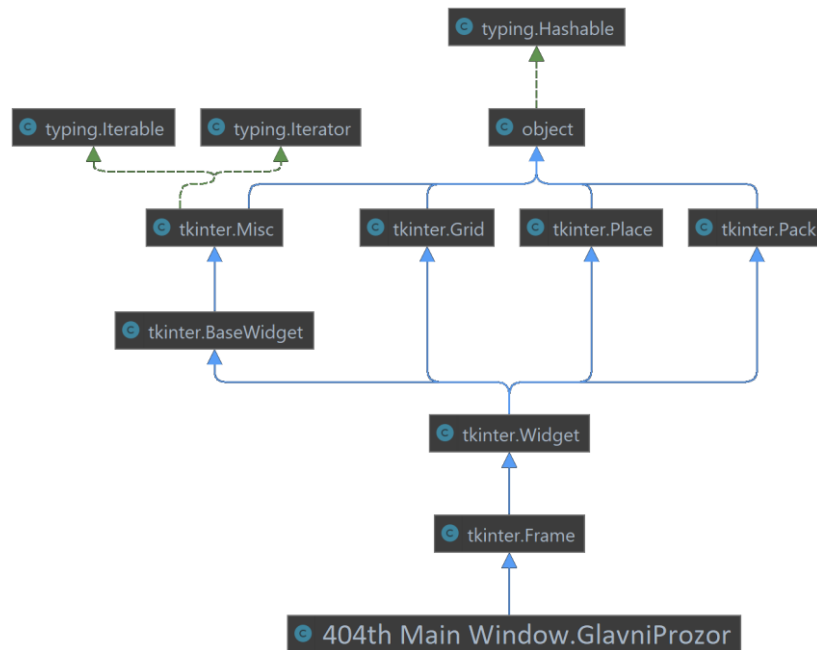
svakom trenutku nego samo one bitne. Korisnik nema previše koristi od toga da zna identifikacijski broj, a program nema koristi od toga da zna kako se oružje zove. Na isti način su implementirani i moduli te neprijatelji i heroji. Heroji, naravno, imaju više evidentiranih podataka od neprijatelja.



Slika 8: UML dijagram klasa entiteta korištenih u aplikaciji, autorski rad

Aplikacija se sastoji od nekoliko datoteka koje su povezane za jednu glavnu datoteku koja se izvršava prilikom pokretanja aplikacije. Glavne datoteke su podijeljene u nekoliko cjelina. Gotovo sve sadrže klase koje predstavljaju elemente aplikacije. Čak je i sam glavni prozor naslijeđena klasa.

Datoteka „404th Main Window.py“ je, kako je već rečeno, glavna datoteka aplikacije koja se treba pokrenuti pri pokretanju aplikacije. Ona sadrži klasu GlavniProzor, što je klasa glavnog prozora aplikacije. Ta klasa sadrži metode za kreiranje raznih elemenata sučelja i njihove funkcionalnosti, kao na primjer mogućnost pomicanja likova. Klasa također sadrži kôd za spremanje i učitavanje elemenata pomoću *Pickle* modula. Na slici 9 je prikazan UML dijagrama klasa sa klasom glavnog prozora i klasama koje predstavljaju *Tkinter*ove elemente prozora.



Slika 9: UML dijagram klase glavnog prozora, autorski rad

Datoteka „*Entitet.py*“ sadrži klasu *Entitet* koja je nadklasa ostalim klasama koje predstavljaju entitete sustava, točnije objekte koji se mogu pronaći na ploči, uključujući žive i nežive entitete. Ona ima ulogu sučelja između glavne klase *Prozor* i specifičnih klasa entiteta sustava, ali također sadrži i određene informacije o svakom entitetu, njegovu ikonu u aplikaciji i određene zastavice.

Datoteka „*podatci.py*“ sadrži čiste podatke o entitetima. Bez ikakve klase, u datoteci se samo nalaze liste objekata koji predstavljaju entitete. Dio tih listi se učitava kroz *Pickle*, a dio se ondje nalazi zapisan u kodu. Entiteti koji su zamišljeni da se mijenjaju se spremaju i učitavaju pomoću *Picklea*, a entiteti koji bi trebali biti stalni su zapisani u kodu. Kôd koji se koristi kod inicijalizacije lista koje se ne učitavaju pomoću *Pickle* modula je zapravo niz konstruktora odgovarajućih klasa koji inicijaliziraju objekte. Taj kôd je bio generiran pomoću pomoćne skripte koja je objašnjena ranije.

Sve ostale datoteke predstavljaju određeni tip entiteta u sustavu, od oružja do okoliša. U svakoj se nalazi jedna do dvije klase koje predstavljaju entitete, a svaka od klasa ima svoje metode koje su potrebne za funkcionalnosti ili opisivanje iste. Neke metode su slične jer predstavljaju slične entitete. Na primjer klasa *EnemyMech* i klasa

*PlayerMech*, od kojih obje predstavljaju istu stvar, mekove, ali zbog različitih pripadnosti sadrže neke različite podatke.

Ukratko o svakoj od ostalih datoteka s navodom bitnijih funkcionalnosti:

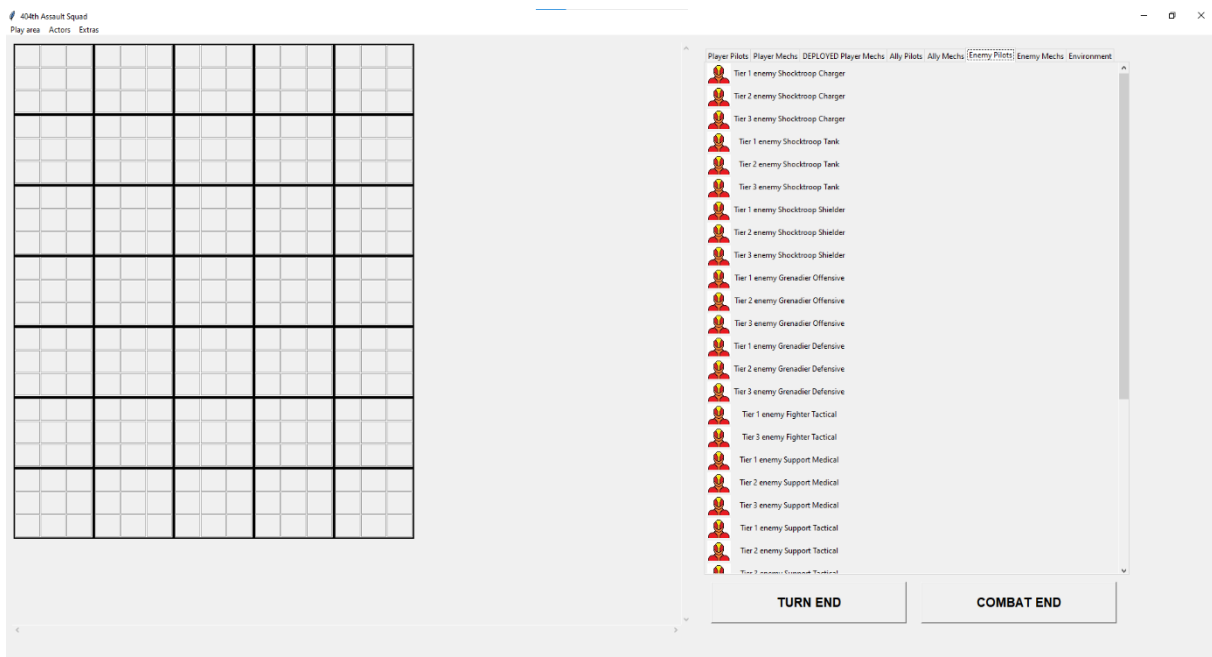
- *EnemyMech.py* – sadrži istoimenu klasu koja opisuje *NPC* mekove s metodama:
  - *showLoadoutDetails* i *showDetails* – pri pozivu iz glavnog prozora prikazuju statistike,
  - *endTurn* i *endCombat* – pri pozivu iz glavnog prozora klikom na gumb daju do znanja opremi kako je prošao jedan potez ili završila bitka,
  - s prefiksom *eval\*\*\** - prilikom promjene opreme ili drugih uvjeta mijenjaju i provjeravaju statistike na koje promjena treba utjecati (HP, AC, broj akcija).
- *EnemyPilot.py* – sadrži istoimenu klasu koja opisuje *NPC* pilote i druge humanoide s metodama:
  - *showSuitDetails* i *showDetails* – pri pozivu iz glavnog prozora prikazuju statistike,
  - *endTurn* i *endCombat* – pri pozivu iz glavnog prozora klikom na gumb daju do znanja opremi kako je prošao jedan potez ili završila bitka,
  - s prefiksom *eval\*\*\** - prilikom promjene opreme ili drugih uvjeta mijenjaju i provjeravaju statistike na koje promjena treba utjecati (HP, AC, broj akcija).
- *PlayerMech.py* – sadrži istoimenu klasu koja opisuje *PC* mekove s metodama:
  - *showLoadoutDetails* i *showDetails* – pri pozivu iz glavnog prozora prikazuju statistike,
  - *endTurn* i *endCombat* – pri pozivu iz glavnog prozora klikom na gumb daju do znanja opremi kako je prošao jedan potez ili završila bitka,
  - *eval\*\*\** - prilikom promjene opreme ili drugih uvjeta mijenjaju i provjeravaju statistike na koje promjena treba utjecati (HP, AC, broj akcija).
- *PlayerPilot.py* – sadrži istoimenu klasu koja opisuje *PC* pilote, odnosno heroje, s metodama:
  - *showSuitDetails* i *showDetails* – pri pozivu iz glavnog prozora prikazuju statistike,

- *endTurn* i *endCombat* – pri pozivu iz glavnog prozora klikom na gumb daju do znanja opremi kako je prošao jedan potez ili završila bitka,
- s prefiksom *eval/\*\** - prilikom promjene opreme ili drugih uvjeta mijenjaju i provjeravaju statistike na koje promjena treba utjecati (HP, AC, broj akcija).
- *Environment.py* – sadrži istoimenu klasu koja opisuje nežive objekte koji se mogu dodati na ploču:
  - Metoda *showDetails* – pri pozivu iz glavnog prozora prikazuje detalje o specifičnom tipu okoliša.
- *MechWeapon.py* – sadrži istoimenu klasu koja opisuje oružja koja se mogu montirati na mekove, bilo oni *NPC* ili *PC* likovi, s metodama:
  - *fire* i *reload* – prilikom ispaljivanja i punjenja municije izmjenjuje potrebne vrijednosti,
  - *breakF* – onemogućuje korištenje oružja.
- *PilotWeapon.py* – sadrži istoimenu klasu koja opisuje oružja koja mogu koristiti piloti i drugi humanoidi, bilo oni *NPC* ili *PC* likovi, s metodama:
  - *fire* i *reloadFunc* – prilikom ispaljivanja i punjenja municije izmjenjuje potrebne vrijednosti,
  - *breakF* i *fixF* – onemogućuje odnosno omogućuje korištenje oružja.
- *Module.py* – sadrži istoimenu klasu koja opisuje sve module koji daju mekovima i pilotima određene moći koje su poduprte tehnologijom, npr. termalna ili elektromagnetska vizija. Također sadrži klasu *Core* koja predstavlja jezgre mekova. Neke od metoda unutar njih su:
  - Metode *activate* i *startCooldown* – označavaju početak korištenja odnosno početak hlađenja modula te izmjenjuju potrebne vrijednosti,
  - Metode *endTurn* i *endCombat* – izmjenjuju vrijednosti ovisno o tome je li modul u korištenju ili na hlađenju.

Međusobne ovisnosti entiteta koje predstavljaju likove i njihovu opremu mogu se vidjeti na slici 4 u poglavlju 2.2.4.

### 2.3.4. Konačni izgled aplikacije

Konačna implementacija virtualne ploče sastoji se od glavnog prozora i nekolicine drugih koji će detaljnije biti objašnjeni u sljedećem poglavlju. Glavni prozor (slika 10) je podijeljen na dva dijela: na lijevoj strani se nalazi ploča, a na desnoj svi elementi sortirani po karticama koji se postavljaju na ploču. Jedno polje ploče je predstavljeno jednim okvirom Tkinterove klase *Frame*. Svako takvo polje u sebi sadrži devet manjih polja u rasporedu 3x3 koja su predstavljena klasom *Label*. Etikete će sadržavati ikone koje predstavljaju entitete, a one prazne predstavljaju prazan prostor po kojem se likovi mogu pomicati. Glavni prozor je napravljen tako da je optimalne veličine kada se otvori na ekranima razlučivosti 1920x1080 piksela, što je najraširenija razlučivost ekrana na današnjim osobnim računalima [18]. Gumbi koji kontroliraju pravilno korištenje aktivnih sposobnosti likova se nalaze ispod kartica. Na gornjem rubu prozora se nalaze izbornici koji služe za generiranje ploče, stvaranje likova te ostale, manje bitne funkcije u izborniku „Extras“.



Slika 10: Izgled glavnog prozora aplikacije, autorski rad





*Slika 11: Ikona koja predstavlja uništenog meka, autorski rad*

Kako je rečeno, svi entiteti su predstavljeni svojom ikonom koje će se smještati unutar etiketa na ploči. Primjer jedne ikone nalazi se na slici 11. Sve su ikone napravljene u aplikaciji Bojanje u veličinama 32x37 piksela i spremljene u .png format jer s tim formatom *Tkinter* najlakše radi.

Nadalje, napravljena su četiri različita prozora za kreiranje likova, koji odgovaraju svim mogućim kombinacijama ovisno o tome kreira li se mek ili pilot te je li on *PC* ili *NPC*. Pri kreiranju likova odabiru se osnovni podatci kao ime, određene statistike i početna oprema. Ostale statistike se izračunavaju po određenim formulama o kojima korisnici neće morati voditi računa. Npr. domet koliko daleko pilot može baciti granate jest zaokružena aritmetička sredina snage i agilnosti pilota. Sve se formule mogu pronaći u priloženoj Excelovoj datoteci s bilješkama, a primjer prozora će biti prikazani kasnije.

Uz prozore za kreiranje likova, postoje i prozori koji prikazuju detalje o entitetima u igri, a time i likova. Sadržaji tih prozora specifični su za likove ili entitete koje pregledavamo te ovise u kojem kontekstu otvaramo detalje o entitetu.

Ako se prozor s detaljima otvara na desnoj strani glavnog prozora (u jednoj od kartica) onda će on prikazivati sve detalje o određenom entitetu. Sadržavati će sve informacije koje korisnik, bilo voditelj igre ili igrač, odredi pri izradi entiteta te one koje se izvedu iz tih informacija kroz ranije navedene formule. U istom prozoru igrači mogu unaprijediti heroje i uređivati svoju opremu.

Nasuprot tome, ako se prozor s detaljima otvara na lijevoj strani glavnog prozora, dok je entitet na ploči, sadržaj prozora će biti skraćen na samo one informacije koje su bitne tijekom bitke. Time igrači i voditelj igre nisu preopterećeni informacijama u slučaju kada je entitetu potrebno promijeniti vrijednosti životnih bodova ili promijeniti stanje okvira oružja.

Naposljetku postoji još nekoliko prozora sa specifičnim funkcijama kao što su prikaz verzije aplikacije te prozor za kreiranje ploče sa prilagođenim parametrima. Neki od njih će biti spomenuti kasnije, kada će biti govora o korištenju aplikacije.

Gotovo cijelom aplikacijom se upravlja samo mišem i tri klika, lijevi, desni i srednji. O tome kako se koriste, biti će govora malo kasnije. Iz kartica možemo pristupiti detaljima

svakog entiteta čiji su prozori prethodno spomenuti, a detaljnije će biti objašnjeni kasnije. Također, možemo uzeti, tj. odabrati entitet i postaviti ga na ploču te zatim pristupiti skraćenim detaljima.

## 2.4. Vrednovanje sustava i aplikacije

*404th Assault Squad* sustav je imao svoje pretke. Iskustvo razvoja prethodna dva sustava pokazalo je da se neke prednosti i nedostaci manifestiraju tek pri ispitivanju i igranju igre. U odnosu na *404th* mogli bi reći da su prethodni sustavi bili prototipovi uzimajući u obzir njihovu složenost. Nedostatci i nebalansirane mehanike uglavnom dolaze do izražaja pri prolasku igrača kroz kampanju odnosno priču. Kao i u svakoj igri, iako se ispituje i ispravlja, igrači će pronaći nove nedostatke koji se trebaju popraviti.

Svijet je jako fleksibilan i omogućuje voditelju igre da ga proširi svojim likovima i pričama za koje misli da će zaintrigirati njegove igrače. Budući da u *404th* ne postoje ograničenja koliko heroji mogu napredovati i uz jednostavne mogućnosti dodavanja novih entiteta u igru poput oružja i modula, igrači dobivaju puno mogućnosti igranja igre. Baza operacija je zamišljena kao svemirski brod. Time se omogućava lako proširenje svijeta novim planetima i misijama te lako pridruživanje novih igrača već postojećem timu. Inače je to dosta teško jer ako se novi igrač pridruži timu koji je više razine, novi heroji će imati poteškoća u borbi jer nisu opremljeni za tako jake neprijatelje. U sustavu *404th* su neprijatelji podijeljeni u samo tri razine težine, a veći spektar težina se postiže promjenom broja neprijatelja i promjenom tipova neprijatelja. Sve modifikacije sustava se lako mogu implementirati u izrađenu aplikaciju. Budući da je korišten objektno orijentirani pristup, novi entiteti se lako dodaju u kolekcije već postojećih. Bojna polja se na ploči generiraju prema parametrima voditelja igre, tako da već u početku podržavaju velike bitke s puno entiteta, a novi entiteti će odmah postati dostupni s obzirom da aplikacija ne čeka ažuriranja s poslužitelja.

Aplikacija koja sadrži ploču je izvanmrežna. To znači da nije potrebna internetska veza za korištenje tog alata. No, ipak ima nekih ovisnosti. Budući da je izrađena u Pythonu

u potpunosti, aplikaciju nije moguće kompajlirati u izvršnu datoteku koja bi omogućila pokretanje neovisno o instaliranom programskom jeziku. Stoga je prije pokretanja na računalu potrebno imati instaliran programski jezik Python. To je zasigurno jedna od glavnih mana programa. Izvanmrežnost, kako je već objašnjeno, može biti i prednost i mana. Ako se igra na daljinu potrebna je aplikacija koja omogućava dijeljenje ekrana, a s time i internetska veza. No, ako se igra igra lokalno, kao što se uglavnom igraju *TTRPG* igre, internetska veza nije potrebna i aplikacija i dalje zadržava sve funkcionalnosti te ne škodi duhu društvenih igara. Svi i dalje mogu bacati prave igraće kocke i mogu koristiti olovku i papir za praćenje statistika igrača, iako aplikacija nudi i tu funkciju. Aplikacija primarno postoji samo za vizualizaciju bojnog polja te tako zamjenjuje figurice i igraću ploču. Sve figurice i ploča su zamijenjene jednom aplikacijom na laptopu ili osobnom računalu. Jedan od nedostataka izvanmrežnosti je problem ako imamo više kopija aplikacije te ako se nešto želi promijeniti. Promijenjena kopija se ponovno mora redistribuirati ili se identične promjene moraju napraviti na svakoj od već postojećih kopija.

Druga činjenica koju je bitno napomenuti je da je aplikacija izmišljena za specifičan sustav igre. Bez nekih promjena nije moguće igrati neki drugi sustav *TTRPG* igre koristeći ploču ove aplikacije. Iako se lako nadograđuje što se tiče sustava za kojeg je napravljena, ima vrlo ograničene mogućnosti ako se želi koristiti za neki drugi sustav.

#### 2.4.1. Kako koristiti aplikaciju i igrati *404th Assault Squad*

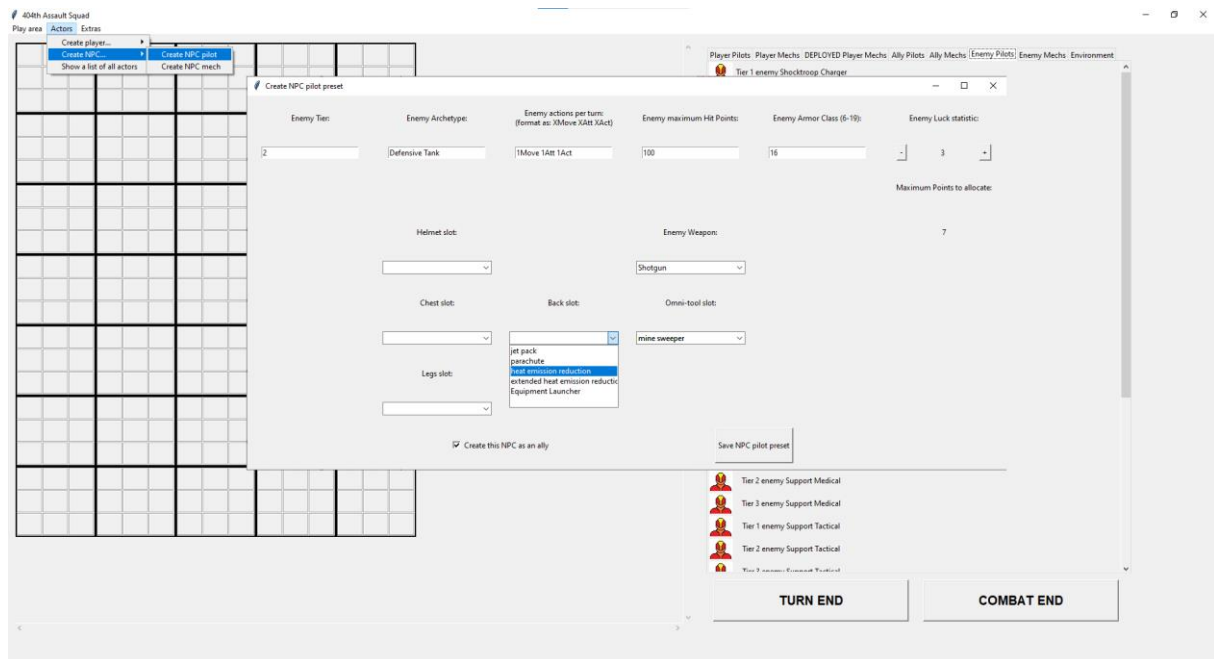
U ovom poglavlju neće biti opisana sva pravila ili postupak igranja same igre, već će biti opisan način korištenja aplikacije i što predstavljaju pojedini dijelovi aplikacije u odnosu na sustav. Detaljnije o pravilima se može pročitati u natuknicama koje se nalaze u priloženoj Excelovoj datoteci, ili izmisliti vlastita pravila te prilagoditi sustav svojim željama i potrebama.

Za izvanmrežnu igru u lokalnoj sjednici potrebno je na jednom računalu imati instaliran Python, poželjno je koristiti posljednju inačicu tog programskog jezika. Mrežne sjednice se izvode na isti način, jedina razlika je što voditelj igre uz pomoć nekog od aplikacija

za konferencijske pozive dijeli svoj ekran s otvorenom aplikacijom drugim sudionicima. Nadalje, igrači i voditelj igre dodatno sa strane moraju voditi bilješke o određenim elementima igre, kao praćenje inventara ili bilješke o priči i sl.

Dijelovi igre koji se igraju u stvarnom vremenu su predviđeni da se odvijaju bez upotrebe aplikacije. Jedino se u aplikaciju moraju unijeti promjene koje su bitne za bitke, kao što su promjene životnih bodova, te dobivanje iskustvenih bodova i promjene na odijelima pilota ili mekovima, npr. montiranje novog oružja na meka.

Izgled glavnog prozora već je predstavljen u poglavlju 2.3.4. te prikazan na slici 10. Prije igranja kampanje potrebno je napraviti likove. Voditelj igre kreira *NPC* likove, a svatko od igrača kreira vlastitog *PC* lika, odnosno heroja. To se radi u odgovarajućim prozorima koji se otvaraju odlaskom na „*Actors > Create player...*“ ili „*Actors > Create NPC...*“. Odaberemo opciju za kreiranje meka ili pilota ovisno o tome što kreiramo te se otvara prozor za kreiranje lika. Na slici 12 je prikazan prozor za kreiranje *NPC* pilota. Ostali prozori za kreiranje likova imaju sličan ili isti izgled ovisno o tome koje su informacije potrebne za kreiranje istih. Igrači mogu imati određene restrikcije o tome kako mogu kreirati svoje mekove i koje module i oružja mogu koristiti sa svojim pilotima. O tome odlučuje voditelj igre, stoga aplikacija nudi sve moguće kombinacije bez restrikcija, a na igračima je da se dogovore o tome kako mogu konfigurirati likove na početku igre. Neke restrikcije i dalje stoje što se tiče kreiranja heroja, ali kod uređivanja istih one se mogu ignorirati. Nakon što se likovi kreiraju i zatvori se prozor za kreiranje lika, isti će se pojaviti u odgovarajućoj kartici na desnoj strani glavnog prozora s odgovarajućom ikonom.



Slika 12: Primjer prozora za kreiranje likova, autorski rad

Svi se živi entiteti igre mogu uređivati i detaljnije pregledavati klikom na srednju tipku miša. Tada se otvara prozor u kojem se prikazuju detalji pojedinih likova i opcije za uređivanje i brisanje istih. Pritiskom na odgovarajući gumb prikazati će se dio prozora za pregledavanje detalja o opremi (slika 13) ili dio prozora s nizom izbornika za uređivanje opreme lika (slika 14). Izbrisani likovi će biti zauvijek izgubljeni ako se izađe iz aplikacije nakon što su izbrisani. Stoga je moguće vratiti likove ako se napravi sigurnosna kopija datoteka u koje se likovi spremaju prije nego što se igra zatvori. To se odnosi na likove koji nisu kreirani u trenutnoj sjednici aplikacije.

Tier 2 enemy Shocktroop Charger

Tier 2 enemy Shocktroop Charger, HP: 67, AC (base): 11

Affiliation: Enemy

DELETE THIS ENTITY

Warning! This will permanently delete this entity!

Tier: 2
Pilot archetype: Shocktroop Charger
Hit Points: 67/67

Actions per Turn

Movement (2 squares = 1 movement action): 2  
Attack (1 attack = attack action): 2  
Action (any other action or prolonged gesture): 2

Luck (LCK): 3  
Current cover bonus: 0  
Character is not under suppressing fire and can act normally.

Base Armor Class (Total Armor Class): 11 (11)

PILOT SUIT setup

Helmet module:

There is nothing on this slot.

Chest module:

Tier 1 utility suit module: [31] active camo
Requirement to equip: available slot, 2 INT, 6 AGI
Duration: 2 (turns)  
Cooldown: 3 (turns)  
Infinite uses.

Effect: visually masks the pilot, attacking or otherwise revealing yourself ends the effect

Back module:

Tier 1 utility suit module: [19] jet pack
Requirement to equip: available slot, 3 AGI
Infinite (toggle) duration or instant activation.  
No cooldown.  
Uses: 3 (ammount per Mech deployment)

Effect: it's a jet pack... movement has +1 range

Suit Legs module:

Tier 1 utility suit module: [21] stabilizer jets
Requirement to equip: available slot, 2 AGI
Infinite (toggle) duration or instant activation.  
No cooldown.  
Infinite uses.

Effect: prevents fall damage

Omni-Tool module:

There is nothing on this slot.

[13] Power fist
Weapon tier: 2
Weapon class: melee
Damage: 20
Maximum range: 1

Ammo type: -  
Magazine : None/None  
Base magazine reserves size: None  
Reload status: Ready

Upgrades

Power Fist Mk.2 at the cost of 2 TECH points

Hide suit modules' details

Edit suit modules

Slika 13: Prikaz detalja NPC pilota, autorski rad

Tier 2 enemy Shocktroop Charger

Tier 2 enemy Shocktroop Charger, HP: 67, AC (base): 11

Affiliation: Enemy

DELETE THIS ENTITY

Warning! This will permanently delete this entity!

Tier: 2
Pilot archetype: Shocktroop Charger
Hit Points: 67/67

Actions per Turn

Movement (2 squares = 1 movement action): 2  
Attack (1 attack = attack action): 2  
Action (any other action or prolonged gesture): 2

Luck (LCK): 3  
Current cover bonus: 0  
Character is not under suppressing fire and can act normally.

Base Armor Class (Total Armor Class): 11 (11)

PILOT SUIT setup

Helmet slot:

Chest slot:

active camo

Back slot:

jet pack

Omni-tool slot:

Legs slot:

stabilizer jets

Power fist

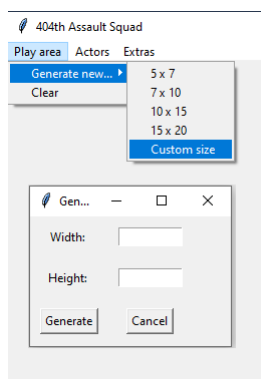
Save preset

cancel editing

Warning! This will overwrite your current setup!

Slika 14: Uređivanje opreme lika u istom prozoru prikazanom na slici 13, autorski rad

31

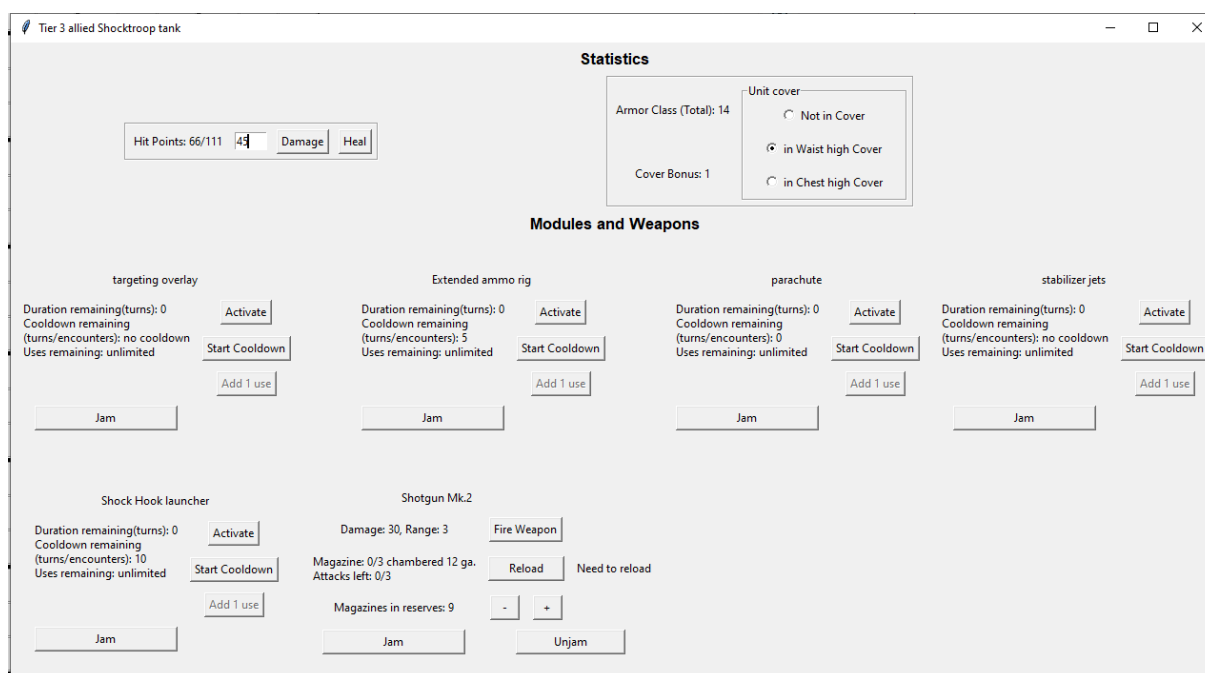


Slika 15: Izbornik za kreiranje ploče željene veličine, autorski rad

Kada dođe do bitke ili je zbog drugog razloga potrebno korištenje načina igre na poteze (*TB*), prvo se kreira i postavlja ploča. To se radi u izborniku „*Play area > Generate new...*“, prikazanom na slici 15. Izbornik nudi nekoliko zadanih veličina ploče te opciju izrade ploče veličine koju korisnik, odnosno voditelj igre, odredi unoseći vrijednosti u prozor koji se otvori na ekranu. Ploča se na kraju može izbrisati opcijom „*Play area > Clear*“.

Nakon kreiranja ploče i likova, na ploču se entiteti postavljaju lijevim klikom miša tako da se prvo klikne na ikonu entiteta kojeg želimo odabrati, a zatim ga se postavi na željeno polje na ploči. Tim načinom se postave svi željeni entiteti na ploču te bitka, engl. *encounter*, može početi. Primjer postavljene ploče može se vidjeti na slici 5 u poglavlju 2.2.7.

Dok su entiteti na ploči, možemo pristupiti istim detaljima koji su dostupni srednjim klikom na desnoj strani prozora. Ovdje se to radi kombinacijom tipke *Control (Ctrl)* i srednjeg klika miša, dok je samo klik srednje tipke miša zamijenjen posebnim prozorom koji prikazuje vrijednosti koje se mogu mijenjati tijekom bitke. U tom prozoru se mogu mijenjati vrijednosti životnih bodova (*HP*), klase oklopa (*AC*) kroz zaklon, iskustvenih bodova (*XP*) te se mogu aktivirati moduli i pucati iz oružja. Primjer takvog prozora je prikazan na slici 16. Desnim klikom miša entitet mičemo s ploče, tj. uzimamo ga u ruku. Nakon toga ga lijevim klikom miša možemo postaviti na drugo mjesto na ploči, a ako ne učinimo ništa entitet je praktički izbrisan s ploče.



Slika 16: Prozor za prikaz statistika bitnih za bitke, autorski rad

Nakon svakog poteza potrebno je kliknuti na gumb „*Turn End*“, a nakon bitke na tipku „*Combat End*“ kako bi moduli dobili potrebnu informaciju i izvršili zadane funkcije koje se trebaju izvršiti u tim okolnostima.

#### 2.4.2. Rezultati prvih ispitivanja

Sustav i aplikacija su ispitani tako da se igralo kroz prolog kampanje. Kampanja nije bila dio ovog projekta te je trenutno samo prolog iste dizajniran u svrhu ispitivanja. Prolog se sastoji od dvije bitke s po nekoliko različitih tipova neprijatelja te tri heroja s predodređenim statistikama. Između bitaka igrači imaju i dijelove koji se igraju u stvarnom vremenu. Svaki od tri igrača je izabrao tip lika kojeg želi igrati i koji mu je i dodijeljen. U normalnim okolnostima igranja igrači će sami kreirati svoje likove, dok su im za potrebe prologa i ispitivanja dodijeljeni već kreirani likovi. Ispitivanje se odvijalo na daljinu. Sve statistike koje se ne mogu pratiti u aplikaciji, primarno inventar, igrači su morali pratiti sami. Ostale statistike su se nalazile u aplikaciji. Aplikacija je bila pokrenuta na računalu voditelja igre koji je dijelio snimku svog ekrana drugim igračima



koristeći aplikaciju *Discord*. *Discord* aplikacija omogućava i grupni poziv te slanje grupnih i privatnih poruka kako bi svi igrači stalno bili u kontaktu. [19] Igrači su po prvi puta vidjeli sustav, stoga je bilo potrebno neko vrijeme dok nisu usvojili sva pravila i značenje svih statistika. Iz tog razloga je većina dijelova prologa bila duža nego predviđeno, ali se je sve uspjelo završiti u jednoj sjednici. Nije bilo nikakvih problema za vrijeme izvršavanja aplikacije. Međutim, sami sustav je imao nekoliko manjih poteškoća. No, to poteškoće su se ticale balansiranja i jasnoće nekih mehanika, a ne samih mehanika i kako one djeluju na igru. Takve poteškoće se lako otklone jasnijim objašnjavanjem i povećanjem upoznatosti igrača sa sustavom te promjenom određenih vrijednosti i konstanti u formulama za izračunavanje problematičnih statistika.

Drugi oblik ispitivanja je bilo kreiranje igračevih likova. Iste likove bi igrači trebali koristiti kao svoje heroje u nadolazećoj kampanji. Aplikacija je za te potrebe proslijeđena svim igračima nakon što su instalirali najnoviju inačicu programskog jezika Python. Voditelj igre je vodio igrače kroz aplikaciju te objašnjavao sve detalje vezane uz kreiranje njihovih likova. Namjera je više bila ispitivanje aplikacije nego sustava jer igrači su bili zadani samo s kreiranjem heroja koji nisu još imali interakcije sa svijetom. Aplikacija se i ovdje pokazala pouzdanom jer je svaki od tri igrača kreirao je svog heroja, tj. pilota, te svog osobnog meka bez većih poteškoća.

#### 2.4.3. Istaknuti problemi i njihova rješenja

U ovom poglavlju će detaljnije biti opisani neki od problema koji su se javili pri dizajniranju sustava i izradi aplikacije. Također će biti opisano kako su isti riješeni ili izbjegnuti tijekom razvoja.

Jedan od većih problema balansiranja igre je bila razlika između moći pilota i mekova. Piloti imaju u prosjeku pet puta manje životnih bodova (HP) i rade isto toliko manje štete sa svojim oružjima. Problem u uravnoteženosti nastaje kada bi se u bitci borili piloti protiv mekova. Mekovi bi nadjačali pilote možda čak i u bitkama jedan protiv pet.

Stoga je bilo potrebno pronaći način kako pilote ojačati kada se bore protiv mekova te kako mekove oslabiti kada se bore protiv ljudi.

- Rješenje problema je došlo u obliku nekoliko novih ili preuzetih mehanika sustava. Kao prvo, oružja mekova su dobila penale ako je meta iz lakše kategorije, odnosno humanoid ili slične veličine. Penalizira se šteta koju oružja rade u takvim situacijama te meta dobiva dodatni AC, kako bi ju bilo teže pogoditi uopće. Sljedeća mehanika bitna za napomenuti je stanje „pod paljbom“ u kojem kada se pilot nađe kada ga se više puta gađa mekovim oružjem. U tom stanju pilot ne može djelovati na svom potezu, ali dobiva veliku količinu dodatne vrijednosti klase oklopa. To također otvara nove strategije koje bi igrači mogli ukomponirati u bitke u kojima se nalaze, a imaju problema s pobjeđivanjem. Posljednja nova mehanika koja je uvedena iz tog razloga jest rodeo. Preuzet iz videoigre *Titanfall*, rodeo je stanje kada pilot skoči na leđa meka kojim upravlja neki drugi pilot. Pilot koji je u rodeu može direktno gađati jezgru meka i tako nanijeti višestruko više štete nego na bilo koji drugi način. Kako bi se kompenziralo za to, mekovi u svojoj opremi mogu imati module koji sprečavaju i otežavaju neprijateljske pilote u tome da ih jašu.

Prvi problem kod izrade same aplikacije je bila odluka kako će se entiteti moći pomicati i postavljati na ploču. Isprva je bilo zamišljeno kao *drag-and-drop* metoda, ali se pokazala teškom za implementirati u *Tkinteru*.

- Pomicanje „figura“ je iz tog razloga riješeno tako da se željeni entitet kojeg pomičemo izbriše s ploče te se zapamti koji je entitet to bio, zajedno s njegovim trenutnim statistikama. Kada zapamtimo entitet možemo ponovno postaviti na novo polje i on će zadržati trenutni *HP* i *AC*. Iako takvo rješenje možda zvuči nespretno, ima i svoje prednosti. Za postavljanje više istih entiteta, npr. više istih neprijatelja ili zidova, potrebno je entitet samo jednom odabrati na sučelju te ga možemo postaviti na ploču koliko god puta želimo.

Posljednji problem koji je vrijedan spomena je bio vezan za samu logiku Pythona i kako on radi s referencama i varijablama. Naime, Python svaku novu varijablu kojoj pridodajemo isti objekt smatra referencom ona taj objekt. Problem s više referenci na isti objekt je što mijenjanjem svojstava objekta preko jedne varijable, mijenja se i

svojstvo kod druge varijable, zato što su obje varijable reference na isti objekt. U struci se takvo kopiranje referenci naziva plitko kopiranje, što narušava integritet aplikacije. Ako bez ikakvog razmišljanja samo inicijaliziramo varijable mekova i oružja, sve će one pokazivati na isti objekt meka ili oružja. Tu se javlja problem ako na jednom meku želimo imati dva ili više istih oružja, što je u potpunosti dopušteno. Oba oružja, iako je svako montirano na zasebnom mjestu na meku, a mjesta su predstavljena varijablama, će koristiti isti okvir za municiju. To ne odgovara tome kako bi ona trebala funkcionirati. Svako oružje, iako je isto ima svoj okvir, a time i svoje stanje koliko metaka ima u svom okviru. Isti se problem događa ako postavimo dva ista tipa neprijatelja na ploču i oštetimo jednoga, oštetio bi se i drugi.

- Rješenje je bilo duboko kopiranje jednog objekta u varijablu koja bi trebala predstavljati drugu instancu istog objekta. Pythonov modul *Copy* nam daje metodu za duboko kopiranje [17], ona kopira željeni objekt u zadanu varijablu i miče međusobne reference. Tako izmjenom drugog objekta više ne utječemo na prvi. Time smo omogućili mnogo željenih funkcionalnosti sustava u aplikaciji. Dubokim kopiranjem sada možemo imati više istih oružja na jednom meku, a da su pritom oba oružja potpuno neovisna jedno o drugom. Također, na ploči se sada može nalaziti više instanci istog tipa neprijatelja, a utjecanje na statistike jednog, neće utjecati na statistike drugog. Nasuprot tome, korištenjem plitkog kopiranja herojima će se stanje prenositi iz jedne bitke u drugu, a time i iz jedne sjednice u drugu, nakon zatvaranja aplikacije. Za njih nije namijenjeno da se nakon svake bitke u potpunosti oporave, a što je još gore, znači da ne bi mogli pratiti iskustvene bodove, razine niti unapređenja likova. Svakim novim postavljanjem ploče bi PC likovi bili vraćeni na početno stanje kao da su napravili novog lika. Što bi bilo neželjeno ponašanje.

### 3. Zaključak

Cilj ovoga rada je bila izrada *TTRPG* sustava/igre i izrada aplikacije koja digitalizira dio igre. *TTRPG* sustav koji je dizajniran naziva se *404th Assault Squad* u kojem su naši heroji zamišljeni kao plaćenici koji mogu koristiti velike robote, zvane mekovi, kako bi uspješno izvršili zadatke na koje su poslani iz baze. Njihova baza je svemirski brod na kojem se nalazi cijeli njihov tim i sva oprema koja im je potrebna. Sustav je zamišljen da bude vrlo fleksibilan vezano za samu priču i svijet, ali i za mogućnosti koje su stavljene pred igrače što se tiče njihovih heroja. Sustav se izgradio na ideji prve *TTRPG* igre „*Dungeons & Dragons*“. S jednim takvim sustavom kao inspiracijom, u sustav su ugrađene mehanike iz nekih poznatijih videoigara kao *Fallout* i *Titanfall*. Neke od mehanika su izmijenjene, a neke u potpunosti izmišljene kako bi činile jedan uravnotežen i igriv *TTRPG* sustav.

Drugi dio rada je bila aplikacija koja igračima olakšava vizualizaciju svijeta u kojem se nalaze. Aplikacija je implementirana u Pythonu pomoću Pythonovog modula *Tkinter* za izradu *GUI*-a te sadrži ploču na kojoj se nalaze entiteti koji su trenutno na polju, osnovne opcije za uređivanje i generiranje ploče i likova, postavljanje svih entiteta na istu te praćenje njihovog statusa. Aplikacija je izvanmrežna i u potpunosti programirana u Pythonu. To sa sobom nosi određene prednosti i mane koje su obrazložene u razradi. Ona može biti unaprijeđena na brojne načine, isto kao i sam sustav, ali prije daljnjeg razvoja ključno je ispitati i aplikaciju i cijeli sustav kroz primijenjene testove. Osnovni testovi prilikom izrade te kratki prolog kampanje koji uključuje upotrebu aplikacije su prošli uspješno te se nakon daljnje uporabe, igranja kampanje s nekoliko igrača i prikupljanja njihovih iskustava i dodatne analize očekuju daljnje izmjene i nadogradnje.

Neki od potencijalnih nadogradnji koje bi imale minimalan utjecaj na igračevo iskustvo, ali bi pomogle u daljnjem razvoju i proširenju sustava, su implementacija baze podataka za entitete i prilagodba aplikacije kako bi se ista mogla staviti na poslužitelj ili kako bi aplikacija mogla funkcionirati i u klijent-poslužitelj okruženju. Python nudi nekoliko modula koji bi mogli olakšati implementaciju mrežne komunikacije između instanci aplikacije te također nudi nekoliko modula za komunikaciju s bazom podataka. Ova dva unapređenja bi omogućila laku pristupačnosti novim korisnicima, a ne bi narušila korisničko iskustvo igranja, tj. korištenja aplikacije.

Još jedno potencijalno unapređenje je napuštanje Pythona kao glavnog programskog jezika za izradu aplikacije te prelazak u trodimenzionalni svijet. Besplatni *game engine* poput Unityja bi omogućili da je cijela igračka ploča izgleda mnogo bolje i privlačnije, a svaki bi entitet mogao imati puno više detalja.

#### 4. Popis literature

- [1] Wizards RPG Team, "Player's Handbook", peto izdanje, Wizards of the Coast, 2014.
- [2] Elsam S., Chiodini J., "10 best tabletop role playing games out right now", Dicebreaker, 13.2.2020., s Interneta, <https://www.dicebreaker.com/categories/roleplaying-game/best-games/best-tabletop-roleplaying-games> (pristupano 27. 12. 2020.).
- [3] PongGame.org, "Pong Game.", s Interneta, <https://www.ponggame.org/> (pristupano 4.9.2021.).
- [4] Vox, 14.6.2018., "Dungeons and Dragons, explained" (video), Youtube. [https://www.youtube.com/watch?v=2PEt5RdNHNw&feature=emb\\_logo&ab\\_channel=Vox](https://www.youtube.com/watch?v=2PEt5RdNHNw&feature=emb_logo&ab_channel=Vox) (pristupano 4.8. 2021.).
- [5] FLASHOWL za Amazon, "FLASHOWL Pearl D&D Dice Set RPG Dice Polyhedral Dice D20 Dice DND Dice , D12, D10, D8, D6, D4 DND RPG MTG Double Colors Game Dice DND Dice Sets 7 Set Dice for Table Gaming(7 Pieces Blue Black)", s Interneta, <https://www.amazon.com/FLASHOWL-Polyhedral-Double-Colors-Gaming/dp/B07YKQCQJHS> (pristupano 4.9.2021.).
- [6] PicClick, "BIG DUNGEON CHUNKS Deluxe Set Terrain 28mm Dungeons & Dragons Pathfinder d&d rpg", s Interneta, <https://picclick.com/Big-Dungeon-Chunks-Deluxe-Set-Terrain-28mm-Dungeons-333418258876.html#&gid=1&pid=2> (pristupano 4.9.2021.).
- [7] Tim Roll20, "Roll20: Online virtual tabletop for pen and paper RPGs and board games.", s Interneta, <https://roll20.net/> (pristupano 20.8.2021.).
- [8] Astral Tabletop, "The easiest way to play tabletop games online", s Interneta, <https://www.astraltabletop.com/> (pristupano 20.8.2020).
- [9] "Fallout series | Fallout Wiki", Fandom, s Interneta, [https://fallout.fandom.com/wiki/Fallout\\_series#Fallout](https://fallout.fandom.com/wiki/Fallout_series#Fallout) (pristupano 30.12.2020.).
- [10] "Cthulhu | The H.P. Lovecraft Wiki", Fandom, s Interneta, <https://lovecraft.fandom.com/wiki/Cthulhu> (pristupano 1.1.2021.).

- [11] "The H.P. Lovecraft Wiki", Fandom, s Interneta, [https://lovecraft.fandom.com/wiki/Main\\_Page](https://lovecraft.fandom.com/wiki/Main_Page) (pristupano 1.1.2021.).
- [12] Respawn Entertainment (2016.), Titanfall 2 (računalna igra), dostupno na: <https://www.origin.com/irl/en-us/store/titanfall/titanfall-2> (pristupano/preuzeto 2.1.2021.).
- [13] "Armor Class | Fallout Wiki", Fandom, s Interneta, [https://fallout.fandom.com/wiki/Armor\\_Class](https://fallout.fandom.com/wiki/Armor_Class) (pristupano 2.1.2021.).
- [14] Python Software Foundation, "PSF Python Brochure Vol. I", 2015., vol. 1, str. 32, preuzeto s Interneta, <https://brochure.getpython.info/media/releases/psf-python-brochure-vol-i-final-download.pdf/view> (preuzeto 28.7.2021.).
- [15] "What is Tk." Tcler's Wiki, s Interneta, <https://wiki.tcl-lang.org/page/What+is+Tk> (posljednja izmjena 21.10.2018.), (pristupano 15.8.2021.).
- [16] Python Software Foundation, "pickle — Python object serialization — Python 3.9.7 documentation.", s Interneta, <https://docs.python.org/3/library/pickle.html> (posljednja izmjena 13.9.2021.) (pristupano 13.9.2021.).
- [17] Python Software Foundation, "copy — Shallow and deep copy operations — Python 3.9.7 documentation.", s Interneta, <https://docs.python.org/3/library/copy.html> (posljednja izmjena 13.9.2021.) (pristupano 13.9.2021.).
- [18] "Desktop Screen Resolution Stats Worldwide", StatCounter Global Stats, StatCounter (2021.), s Interneta, <https://gs.statcounter.com/screen-resolution-stats/desktop/worldwide> (pristupano 24.8.2021.).
- [19] Discord tim (2015.), "Discord | Tvoje mjesto za razgovor i druženje.", dostupno na: <https://discord.com/> (pristupano 5.9.2021.).

## Prilozi

Prilozi se nalaze na CD-u koji se nalazi na stražnjoj korici tvrdog uveza. Prilozi će ovdje biti pobrojani po istom redoslijedu kako se nalaze i njihove mape na mediju.

Prilog 1: Digitalni format ovog Wordovog dokumenta,

Prilog 2: Aplikacija *404th Assault Squad*,

Prilog 3: Izvršna datoteka za instalaciju izdanja 3.9.7. programskog jezika Python za Windows 10,

Prilog 4: Excelov dokument s bilješkama o pravilima sustava, formulama i skicama te skripta korištena za prebacivanje podataka iz Excelovih tablica u program zajedno sa tekstnim datotekama izvoza iz Excela i rezultata nakon prolaska istih kroz skriptu.



