

**PROGRAMSKI MODEL PODESIVIH KOMPONENTI ZA FPS IGRE SA
STRATEGIJSKIM MEHANIZMIMA**
**A PROGRAMMING MODEL OF CONFIGURABLE COMPONENTS FOR FPS GAMES
WITH STRATEGY ELEMENTS**

Nedeljko Tešanović, dipl. inž. elektr. i računar, *Fakultet tehničkih nauka, Novi Sad*

**Oblast – ELEKTROTEHNIČKO I RAČUNARSKO
INŽENJERSTVO**

Kratak sadržaj – *Trenutno ne postoje adekvatni modeli za brz razvoj prototipa modernih video igara određenih žanrova. Ovaj rad predlaže programski model podesivih komponenti čija implementacija predstavlja takav alat. Razvijeni alat je testiran kroz proces razvoja dva prototipa igara. Rezultat je potvrda validnosti rješenja za datu problematiku. Rezultat daje osnovu za dalja istraživanja.*

Ključne reči: Programski model, razvoj video igara, FPS, RTS, prototip

Abstract – *There is a lack of adequate models for fast video game prototyping. This paper proposes a programming model of configurable components whose implementation represents such a tool. The tool is tested in the form of two game prototypes. The result confirms the validity of the proposed solution and also serves as a basis for future research.*

Keywords: Programming model, game development, FPS, RTS, prototype

1. UVOD

Video igre su trenutno jedan od najpopularnijih oblika zabave sa ogromnim tržištem i industrijom iza njih [0]. Potreba tržišta za novim igrama je neprestana i povećava se sa vremenom zbog rasta broja igrača, uz složenost tih igara. Zbog toga je za industriju video igara od ogromnog značaja da se proces razvoja novih igara ubrza i pojednostavi koliko god je to moguće.

2. OPIS PROBLEMA I POTREBE ZA RJEŠENJEM

Potreban je programski model koji jasno opisuje skup komponenti čija implementacija olakšava razvoj video igara proizvoljnog žanra. Implementacija ovog modela treba da omogući brz razvoj igara odabranog žanra sa što većoj fleksibilnosti u količini i složenosti igara.

Implementacija ovog modela ne mora da sadrži apsolutno sve njegove komponente, nego samo one komponente koje projektant želi da ima u svojim igrama. Pored toga, ovakav model ne treba da ograničava dizajn igre samo na mehanizme prisutne u ovom modelu, nego da omogućiti dodatna proširenja po potrebama vizije projektanta. Isto ako, ne treba da bude vezan za određen pogon, nego da je

opšte primjenljiv – kako za više pogona, tako i za „domaći“ pogon.

Trenutno, jedan od najpopularnijih žanrova video igara jesu FPS igre (eng. *First Person Shooter*) [0]. Ovaj žanr je danas najsloženiji nego što je ikada bio i njegova složenost se povećava iz dana u dan. Nažalost, šabloni za pravljenje tih igara su ostali relativno jednostavni u odnosu na potrebnu složenost modernih igara. Za razvoj FPS igara u pogonima [1][2][3] je često dostupan samo paket za kontrolu karaktera iz prvog lica, odnosno za kretanje po svijetu konstantnom brzinom, skakanje i gledanje naokolo, ali ne i za ostale prethodno navedene mehanizme.

Zbog toga je za potrebe ovog rada odabran upravo FPS žanr. Tačnije, odabran je spoj FPS i RTS (eng. *Real-Time Strategy*) žanrova. Razlog tome jeste prisustvo određenih strategijskih mehanizama koji se mogu često pronaći u RTS igrama, a čija integracija u FPS igru bi omogućila dodatnu složenost sastava te igre i otvorila nove mogućnosti za dizajn.

U nastavku ovog poglavlja će biti navedeni skupovi mehanizama koji su često sadržani u FPS i RTS igrama i čija unija će predstavljati skup potrebnih mehanizama koje programski model treba da sadrži da bi se smatrao validnim rješenjem prethodno pomenute problematike. Skupovi mehanizama koji će biti navedeni nisu u potpunosti sadržani u svakoj igri, nego su sadržani neki od njihovih podskupova.

2.1. Mehanizmi česti u FPS igrama

Ovi mehanizmi se mogu pronaći u većini FPS igara, od relativno jednostavnih igara, do složenih vojnih simulacija [4][5][6]. Naravno, postoje i drugi mehanizmi koji se mogu naći u nekim FPS igrama, ali pokrivanje svih različitih mehanizama svih tih igara i formiranje modela koji ih sve uključuje je praktično nemoguće. Neki mehanizmi često sadržani u FPS igrama su lokomocija, kamera, interakcija, avatar igrača, oružje (blisko i daljinsko), neigrivi karakteri, logika igre, i korisnički interfejs.

2.2. Mehanizmi česti u RTS igrama

Igre RTS žanra su jedne od najstarijih video igara i obično se bave tematikom ogromnih bitaka i manevrisanja jedinica zarad nadmudrivanja protivnika i uništenja njihove baze komande [7][8]. Sa obzirom na to da FPS igre u prosjeku imaju mnogo manji opseg, tj. akcenat se daje na relativno manjim okršajima, od interesa je razmotriti strategijske mehanizme koji se mogu pronaći u ovim igrama radi potencijalnog poboljšanja i proširenja FPS igara. Neki od tih mehanizama koji već nisu nabrojani u

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Dragan Ivetić, red. prof.

FPS igrama su upravljane jedinicama, baza, resursi, i regrutovanje jedinica. Kamera, logika igre, korisnički interfejsi neigrivi karakteri su takođe prisutni i u većini RTS igara, ali se razlikuju od njihovih predstava u FPS žanru, te je potrebno imati i to na umu prilikom osmišljavanja rješenja.

3. MODEL APSTRAKTHNIH MEHANIZAMA

Imajući u vidu mehanizme nabrojane i pojašnjene u poglavlju 2.1, kao i specifikaciju problema u poglavlju 2, predlaže se sledeće rješenje objašnjeno u ovom poglavlju: **programski model podesivih komponenti za FPS igre sa strategijskim mehanizmima.**

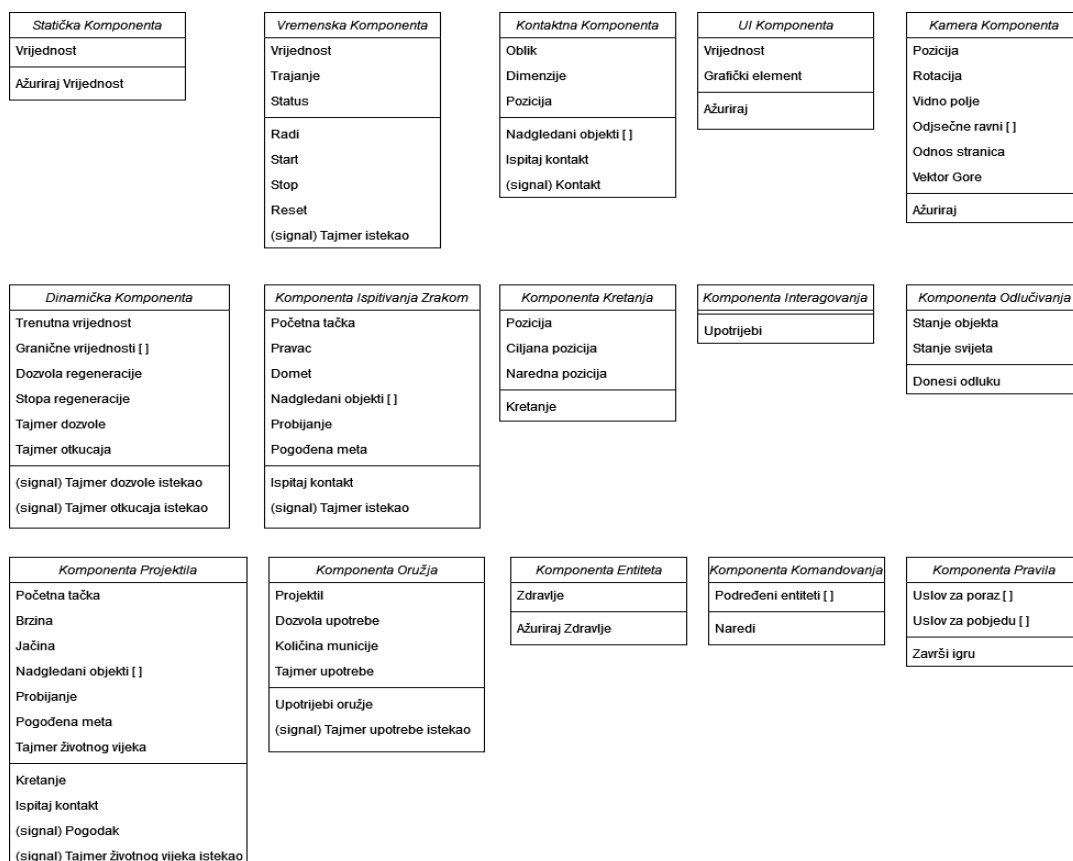
Komponente ovog rješenja (prikazane na slici 1) su samo smjernice za funkcionalnosti koje bi trebale biti implementirane, a sam način implementacije je ostavljen projektantu na slobodan izbor. Akcenat je stavljen na upotrebi šema naslijeđivanja i kompozicije za ostvarenje komponenti čija implementacija može da obezbjedi ogromnu fleksibilnost u smislu varijanti stvari u igrama koje bi se pravile tim alatom. Komponente rješenja su one koje svojim kombinacijama ili uključivanjem omogućavaju implementiranje mehanizama koji se često nalaze u FPS i RTS igrama, odnosno onih opisanih u poglavlju 2. Bitno je naglasiti da se ovo rješenje odnosi na apstraktni nivo programskog dijela razvoja igre ili alata – implementiranje ovakvih komponenti podrazumijeva već postojeći pogon igre sa osnovnim funkcijama (programska petlja, renderer, animacije, 3D modeli, zvukovi, itd), bilo komercijalni ili domaći, za koji se ove komponente razvijaju.

Rješenje je modularno po prirodi, tako da projektant može da odabere koje komponente će da implementira u svom

proizvodu i koje će da budu sadržane u nekoj formi u igrama razvijenim upotrebom tog proizvoda. Puna ili djelimična implementacija bi trebala da obezbjedi mogućnost lakog i ubrzanog razvoja video igara sa elementima FPS i RTS žanrova, ali i igara striktno FPS, odnosno RTS žanra. Drugim riječima, projektantu je na raspolaganju prošireni skup komponenti, ali ukoliko projektant želi da pravi samo FPS igre u retro stilu, dovoljno je da implementira podskup komponenti ovih rješenja koje on smatra potrebnim za ostvarenje tog cilja. Naravno, moguće je i dopuniti ovaj model sopstvenim komponentama - što je broj implementiranih komponenti veći, to je moguće i lakše pravljenje većeg broja i složenijih igara sa mnogo manje potrebnog truda i rada u fazama dizajna i implementacije prilikom razvoja igre.

4. PRIMJER IMPLEMENTACIJE MODELA

Kao što je već pomenuto, za ovaj rad je odabran spoj FPS i RTS žanrova i rješenje je orjentisano ka funkcionalnostima ta dva žanra, a radi jednostavnosti primjera, ograničeno je na jednog igrača. Fokus ove implementacije je na određenoj miješavini ta dva žanra gdje je najveći fokus na FPS dijelu, a strategijski mehanizmi igraju sekundarnu ulogu i služe da pomognu mehanizmima iz FPS žanra. Kretanje, perspektiva, akcija i interaktivne tehnike su preuzete iz FPS žanra, a ponašanje NPC-a (eng. *Non-Playable Character*) je uslovljeno mješavinom ponašanja čestim u FPS igrama, ali i raznim strategijskim mehanizmima - pogotovo koordinacija većeg broja NPC-a. Taj hibrid FPS i RTS žanrova će se u nastavku ovog rada skraćeno zvati FPSS (eng. *First Person Strategy Shooter*).



Slika 1: Apstraktne komponente modela

Implementacija modela je sastavljena od podskupa komponenti modela opisanog u poglavlju 3, odnosno njihovih implementacija specifičnih za odabrani pogon igre. Pogon koji je odabran za implementaciju modela je Godot [3], specifično trenutna verzija – Godot 4.

4.1. Godot 4

Godot je pogon za razvijanje igara opšte namjene. Glavni razlog za izbor Godota, kao i njegova glavna prednost u odnosu na druge pogone jeste njegova modularna arhitektura koja cijelu igru organizuje na bazi čvorova, stabla, scena i signalnih funkcija sa filozofijom maksimizacije upotrebe šema nasljeđivanja i kompozicije, odnosno minimizacije broja komponenti. Sve to se poklapa sa načinom realizacijom ovog programskog modela, i trebalo bi da obezbjedi olakšanu implementaciju programskog modela. Ovi čvorovi, scene, stablo i signali su četiri glavna koncepta Godota.

Čvorovi su najsitniji, ugrađeni elementi Godota koji se redaju u hijerarhijska **stabla**. Svaki čvor ima određen skup funkcionalnosti i parametara za podešavanje tih funkcionalnosti. Pored toga, u slučaju da je potrebna dodatna funkcionalnost nekog čvora, moguće je zakačiti skriptu za njega, i pozivati njene funkcije, odnosno podešavati njene promjenljive. Skripte su pisane u GDScript jeziku, domaćem programskom jeziku Godot pogona. **Scene** se sastoje od čvorova i drugih scena i mogu se instancirati. **Signali** su Godotova implementacija *Observer* šeme [9].

4.2. Implementacija modela u obliku paketa za razvoj FPS i RTS igara - „FPSS paket“

Ovaj paket je namijenjen prvobitno za razvoj FPS igara sa strategijskim mehanizmima. Kao takav, ima potrebne komponente modela koje su krojene specifično za tu namjeru, od kojih su mnoge već potpuno ili djelimično sadržane u predefinisanim čvorovima Godot pogona. U nastavku ovog potpoglavlja će ukratko biti predstavljene implementirane komponente, bez detaljne razrade njihove implementacije, sa obzirom na to da postoji više načina za realizaciju svake komponente.

Statička komponenta je ograničena vrijednost sa funkcijom za ažuriranje i prvobitno se koristi kao brojač municije. **Vremenska komponenta** je već sadržana u predefinisanim *Timer* čvoru. **Dinamička komponenta** je kompozicija statičke i vremenske komponente i automatski mijenja svoju vrijednost tokom vremena. **Kontakt komponenta** je kombinacija Godotovih *Area3D* i *CollisionShape3D* čvorova i koristi se za ispitivanje dodira objekata. **UI komponenta** je predefinisani *Control* čvor sa prpratnom skriptom. **Komponenta ispitivanja zrakom** je Godotov čvor *RayCast3D*. **Kamera komponenta** je takođe već prisutna kao *Camera3D* čvor. **Komponenta interagovanja** je implementirana kroz kod skripti zakačenih za druge čvorove i sastoji se jedne funkcije koja definiše posljedice upotrebe datog objekta. **Komponenta kretanja** sadrži *NavigationAgent3D* čvor i kod za kretanje po stazama stvorenim upotrebom *NavigationRegion3D* čvora. **Komponenta odlučivanja** je implementirana u formi funkcija koje diktiraju ponašanje entitetskih čvorova. **Komponenta projektila** se sastoji od vremenske komponente, *RigidBody3D* čvora i kontakt komponente i

omogućava definisanje ponašanja projektila od prave putanje do krivih putanja balističkog proračuna. **Komponenta oružja** sadrži komponente interagovanja i projektila i definiše ponašanje oružja u igrama. **Komponenta entiteta** ima dinamičku komponentu zdravlja, kontakt komponentu i komponentu kretanja. **Komponenta komandovanja** podređenim entitetima zadaje instrukcije kroz svoje brojne funkcije u zavisnosti od "ličnosti" koja predstavlja stil komandovanja. **Komponenta pravila** je implementirana kroz kod i upravlja logikom za dodjeljivanje poena i završetak igre.

5. PRIMJER IGARA RAZVIJENIH IMPLEMENTACIJOM MODELA

U ovom poglavlju je pokazana primjena razvijenog FPSS starter paketa iz poglavlja 4 za razvoj dvije različite igre. Te dvije igre se razlikuju po tematici, kao i funkcionalnostima, što prikazuje fleksibilnost i valjanost paketa, i samim tim valjanost programskog modela na kome je on zasnovan. Ovaj rad neće ulaziti u sve detalje dizajna ovih igara, nego samo one komponente za koje je bio iskorišten FPSS starter paket.

5.1. FPS igra sa strategijskim mehanizmima

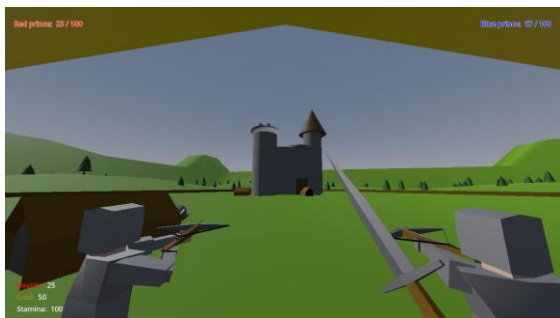
Kraljevina Jugoslavija je okupirana od strane sila osovine, a igrač se stavlja u ulogu partizanskog komandira i nadmudruje neprijatelje manevrisanjem svojih jedinica direktno sa bojnog polja. Sam igrač je hendikepiran u upotrebi oružja, te nije dovoljno jak da ide u direktan okršaj protiv neprijateljskih vojnika kao u klasičnoj FPS igri, nego mora da se uzda u svoje komandirske vještine, strategijsko razmišljanje i prijateljske jedinice.



Slika 2: Slika iz FPS igre sa strategijskim mehanizmima

5.2. RTS igra u prvom licu

Kraljevstvo je ostavljeno u krizi sa praznim prijestolom i bez imenovanog nasljednika. Postoje mnogi pretendenti, ali samo jedan od njih može postati kralj. Igrač se stavlja u ulogu jednog od princeva koji vriebeju prijesto. Radi bolje legitimacije i morala, princevi moraju da vode svoje armije u bukvalnom smislu, izdavajući im naredjenja direktno sa bojnog polja. Igrači imaju borbenu opremu u vidu oklopa, a mogu da koriste i svoje mačeve, samostrijele, katapulte i druga oružja, kao i da naređuju svojim jedinicama i radnicima.



Slika 3: Slika iz RTS igre u prvom licu

6. ZAKLJUČAK

Ovaj rad je predstavio programski model podesivih komponenti za FPS igre sa strategijskim mehanizmima, čija je arhitektura predložena u poglavlju 3. Ovaj programski model se pokazao kao sasvim validan alat za dopunjavanje funkcionalnosti već postojećih pogona za razvoj video igara, odnosno dopunjavanja izvornog koda domaćih pogona igara.

Modularne podesive komponente ovog alata, odnosno njihova pravilna implementacija je omogućila stvaranje paketa za brzi razvoj prototipa igara, a čija funkcionalnost je prikazana u poglavlju 4.2, koji sadrži osnovne elemente za stvaranje igara ne samo FPS žanra, nego i RTS žanra, kao i podžanra koji kombinuje njihove elemente. Iako je početni paket demonstrativno razvijen za pogon Godot, ne bi trebalo da postoji razlog zašto opšti programski model ne bi bio primjenljiv i na druge pogone, kao što su Unity, Unreal i slično.

Upotrebljivost implementacije programskog modela, odnosno početnog paketa FPSS je demonstrirana u poglavlju 5 na dvije igre drastično različite tematike – Drugog svjetskog rata i srednjeg vijeka. Ova varijacija u razvijenim igrama je pokazala valjanost i fleksibilnost paketa koji je podskup funkcija programskog modela. Mogućnosti razvoja dodatnih igara je praktično neograničena, a implementacijom dodatnih komponenti programskog modela bi se samo povećala.

Implementacija ovakvog modela bi trebala biti od koristi ne samo već uspostavljenim studijima za razvoj video igara, nego i nezavisnim projektantima koji žele da brzo razvijaju prototipe, odnosno razrađuju koncepte igara.

Ovaj programski model, iako skoncentrisan na FPS i RTS žanrove, je veoma opširan u svojim funkcionalnostima. Međutim, sa porastom složenosti današnjih video igara kao oblika zabave, javlja se veća i veća potreba za složenijim mehanizmima, kao što je objašnjeno u uvodnom poglavlju 1. Ovo stvara priliku za proširenje ovog programskog modela tako da obuhvata druge žanrove, npr. RPG, itd.

Čak i sama implementacija početnog paketa u Godot pogonu ima prostora za dalji rad i proširenje funkcijama, odnosno mehanizmima FPS i RTS igara, prvobitno onima koji se tiču režima igre sa više igrača, ali i drugima – mehanizmima za virtuelnu realnost, naprednijim algoritmima za pronalaženje staza i donošenje odluka. Interesantno bi bilo posmatrati i kako bi generativne neuronske mreže osmišljale i koristile ovakve modele, kao

i kako ovakvi modeli utiču na brzinu i kvalitet razvoja igara u takmičarskom okruženju.

7. LITERATURA

- [0] <https://medium.com/spring-2024-information-expositions/video-games-popularity-and-longevity-over-time-72a748146f24> (pristupljeno u septembru 2024)
- [1] <https://unity.com/> (pristupljeno u septembru 2024)
- [2] <https://www.unrealengine.com/en-US> (pristupljeno u septembru 2024)
- [3] <https://godotengine.org/> (pristupljeno u septembru 2024)
- [4] <https://www.gameopedia.com/evolution-of-first-person-shooter-fps-games/> (pristupljeno u septembru 2024)
- [5] https://ultimatepopculture.fandom.com/wiki/First-person_shooter (pristupljeno u septembru 2024)
- [6] <https://reedkolbe.medium.com/an-analysis-of-first-person-shooters-what-makes-a-successful-fps-and-where-the-market-is-headed-e7b333ef5d05> (pristupljeno u septembru 2024)
- [7] <https://brokenkingdom.medium.com/what-are-real-time-strategy-games-6ddffd806ed1> (pristupljeno u septembru 2024)
- [8] <https://invogames.com/blog/what-is-real-time-strategy-games/> (pristupljeno u septembru 2024)
- [9] https://docs.godotengine.org/en/stable/getting_started/introduction/key_concepts_overview.html (pristupljeno u septembru 2024)

Kratka biografija:



Nedeljko Tešanović je rođen 1998. godine u Bijeljini. Diplomski ispit na temu „Generisanje realističnih portreta upotrebom dubokih konvolucijskih generativno-protivničkih mreža“ je odbranio 2021. godine sa ocjenom 10,00. kontakt: nedeljkoTESANOVIC@UNS.AC.RS