

PROCEDURALNO GENERISANJE SCENE RAČUNARSKE IGRE ZASNOVANO NA GRAMATICI**PROCEDURAL GENERATION OF GRAMMAR-BASED COMPUTER GAMES**Lazar Anđelić, *Fakultet tehničkih nauka, Novi Sad***Oblast – RAČUNARSTVO I AUTOMATIKA**

Kratak sadržaj – Opis jednog načina primene L-sistema zajedno sa standardnom gramatikom za generisanje 3D scene računarske igre.

Ključne reči: L-sistem, Gramatika, Proceduralno generisanje, Video igra

Abstract – Description of a method of using L-systems combined with a formal grammar to generate a 3D scene usable in video game.

Keywords: L-system, Grammar, Procedural generation, Video game

1. UVOD

Uz napredak tehnologije i povećanje mogućnosti računara, raste i želja korisnika za što detaljnijem i realnijem prikazu. Posebno industrija video igara ima težak zadatak da zadovolji očekivanja za najveće projekte, i troškovi izrade takvih projekata naglo rastu.

Potencijalno rešenje za problem stvaranja sadržaja koji zadovoljava korisnička očekivanja je primena metoda proceduralnog generisanja. Ove metode se već dugo primenjuju u oblasti računarske grafike, i tu imaju širok spektar primene. Samo neki od primera primene su generisanje zemljišta, vodenih površina i biljaka. Kompleksne scene, koje sadrže veliki broj različitih modela, mogu da zahtevaju više meseci da se ručno naprave, ali primenom metoda proceduralnog generisanja one mogu biti generisane i za manje od sat vremena.

Metode proceduralnog generisanja je moguće primeniti i za generisanje entiteta koji su nastali uticajem ljudi, kao što su gradovi. Gradovi su vizuelno kompleksni i rezultat su razvoja pod uticajem mnogobrojnih faktora, koji može da traje i više od stotinu godina. Veliki je izazov napraviti realan model za proceduralno generisanje toliko kompleksnog sistema.

2. FORMALNE GRAMATIKE

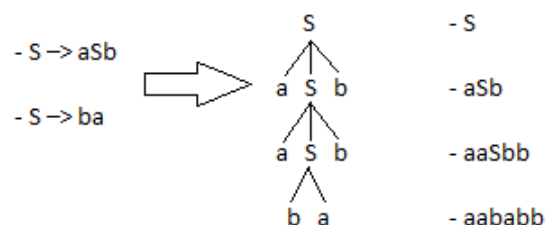
Formalna gramatika je deo formalne teorije jezika. Formalnu teoriju jezika je opisao N. Čomski 1956. godine. U pitanju su velika pojednostavljenja i apstrakcije nad empirijskim domenom prirodnih jezika.

Jedna od ključnih osobina formalne teorije jezika jeste to da je značenje „reči“ koje čine neki jezik u potpunosti ignorisano. Apstrakcije u formalnoj teoriji jezika su veoma uspešno odabrane, što pokazuje široki obuhvat oblasti u kojima je do sada nađena primena iste.

Formalni jezik je neograničen skup nizova karaktera, gde su nizovi karaktera neodređene ali konačne dužine. Nizovi tih karaktera predstavljaju izraze u formalnoj teoriji jezika. Formalna teorija jezika je skup matematičkih alata koji opisuju načine definisanja i obrade formalnih jezika. U formalnoj teoriji jezika se istražuju matematičke osobine formalnih jezika, i načini da se ti jezici opišu na konačan način, čak i ako je neki jezik beskonačan. Uobičajeni način da se takav opis postigne je pomoću formalne gramatike. Takođe, unutar teorije jezika, jezici su hijerarhijski poređani po kompleksnosti. To je *hijerarhija čomskog*. Četiri grupe koje čine hijerarhiju su: računski prebrojivi jezici (tip 0), kontekstno osetljivi jezici (tip 1), kontekstno slobodni jezici (tip 2), i regularni jezici (tip 3).

Formalna gramatika se izražava pomoću matematičkih pravila i logike. Formalna gramatika se sastoji od četiri elementa. To su konačan skup simbola (*terminala*), koji su elementi jezika, konačan skup simbola (*neterminala*), *početni simbol* iz skupa *neterminala*, i konačan skup pravila (produkciona pravila).

Produkciona pravila su oblika $\alpha \rightarrow \beta$, što znači „ α se može zameniti sa β “, i gde su α i β nizovi sastavljeni od terminala i neterminala. U slučaju da je na niz karaktera „ $x\alpha y$ “ primenjeno pravilo $\alpha \rightarrow \beta$, dobija se niz karaktera „ $x\beta y$ “. Primer upotrebe formalne gramatike je na Slici 1.



Slika 1. Primena pravila formalne gramatike

2.1. Lindenmajerovi sistemi

Lindenmajerovi sistemi, često skraćeno na L-sistemi, su nastali kao matematički model za predstavljanje razvoja biljaka. Fokusiraju se na topologiju biljaka, odnosno odnose susednih ćelija ili većih gradivnih jedinica u sklopu biljke. L-sistemi su gramatika koja funkcioniše na principu zamene, odnosno iterativno zamenjuju elemente

NAPOMENA:

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

trenutnog stanja objekta pomoću predefinisanih *pravila zamene*. Pravila zamene su u stvari *produkciona pravila*. L-sistemi se definišu kao uređeni par $G=(V, \omega, P)$, gde je V skup simbola koji sadrži *konstante* (terminale) i *promenljive* (neterminale), ω aksiom, odnosno početno stanje sistema, a P skup produkcionih pravila, odnosno pravila zamene.

Pravila zamene se primenjuju iterativno, počevši od početnog stanja. Primenjuje se svako pravilo koje je moguće primeniti u datoj iteraciji. To je glavna razlika u odnosu na ostale formalne gramatike, koje primenjuju samo jedno pravilo pri svakoj iteraciji.

Pravila zamene takođe određuju da li je L-sistem deterministički. Deterministički L-sistem je takav da za svaki simbol koji se nalazi sa leve strane produkcionog pravila postoji tačno jedno produkciono pravilo. Ako L-sistem nije deterministički, onda je stohastički.

L-sisteme je moguće primeniti za generisanje dvodimenzione i trodimenzione grafike. Ova funkcionalnost se nadograđuje na generisanje nizova karaktera, tako što se proces generisanja tih nizova modifikuje tako da se u nizovima nađu određeni kontrolni simboli. Kontrolni simboli služe za upravljanje interpretacijom generisanih nizova karaktera.

Interpreter se definiše kao uređena trojka (x, y, α) . Simboli x i y predstavljaju trenutne koordinate na kojima se nalazi kursor interpretera. Simbol α predstavlja ugao rotacije kursora, i tako je određen pravac u kom će se on kretati. Pored ta tri parametra, potrebno je definisati dužinu koraka d i povećanje ugla δ . Kada su svi pomenuti parametri definisani, u osnovnom, dvodimenzionom, obliku važe sledeće oznake: F kao kretanje napred za korak d i crtanje linije između prethodne i nove pozicije, f kao pomeraj napred za korak d , $+$ i $-$ kao povećanje ili smanjenje ugla α za δ , i i $|$ kao obrtanje smeru kretanja. Osim osnovnih funkcionalnosti interpretera, moguće je uvesti još dve. Njihovi simboli su $[$ i $]$, a značenje im je sledeće: $[$ kao čuvanje trenutnog stanja kursora na *LIFO* („pushdown“) steku, i kao uzimanje stanja sa steka i korišćenje istog kao trenutno stanje kursora. Iako se pri uzimanju stanja sa steka kursor pomera, linija se ne crta.

3. PROCEDURALNO GENERISANJE

Proceduralno generisanje sadržaja je automatsko stvaranje digitalnih objekata za video igre, simulacije ili filmove, zasnovano na predefinisanim algoritmima i šablonima koji zahtevaju minimalnu korisničku interakciju [1]. Može smanjiti pritisak na dizajnere tako što automatizuje deo dizajnerskog procesa. Isto tako, može biti i previše vremenski zahtevno za implementirati i konfigurirati neki kompleksan algoritam koji bi zadovoljio potrebe video igre.

Postoji mnogo kategorija proceduralnog generisanja, i skoro sve imaju isti problem: nedostatak kreativnosti i izvesnosti. Teško je sa sigurnošću potvrditi da je sadržaj stvoren nekim algoritmom novitet, ili da je koristan i visokog kvaliteta. Ipak, ta neizvesnost je takođe i dobra strana proceduralnog generisanja. Pomoću proceduralnog generisanja moguće je dobiti neočekivano kvalitetan sadržaj, kao i neupotrebljivo loš sadržaj. Zato treba biti veoma pažljiv pri razvoju metoda proceduralnog

generisanja i napraviti algoritam koji je na granici između fiksnog, potpuno predvidivog sistema, i haotičnog, nepredvidivog sistema.

3.1. Neizvesnost u video igrama

Neizvestan rezultat procesa je potpuno nepoznat, dok je izvestan rezultat predodređen. Na granici između ta dva, nalazi se rizik, koji ima rezultate sa poznatom verovatnoćom dobijanja istih. Na osnovu toga, Salen i Cimerman su definisali neizvestan sistem kao sistem u kojem akcije uključuju neki nivo rizika [2]. Osim toga, identifikovali su nekoliko osobina takvih sistema koje su relevantne za proceduralno generisanje u video igrama. Prva je da, čak i bez upotrebe metoda proceduralnog generisanja, video igra može imati „osećaj“ nasumičnosti. Druga je da čak i igre koje su u potpunosti zasnovane na sreći mogu ponuditi smisao za nastavak igre, dok god igrači imaju šansu da iskoriste neke smislene prilike tokom igre. Treća osobina neizvesnih sistema je identifikovana kao prikrivanje rezultata koje bi igrači inače mogli da predvide. Sa takvim osobinama, igrač mora, barem donekle, rizikovati u svojim akcijama da bi došao do uspeha.

Nedostatak izvesnosti je neophodan deo u dizajnu video igre. Iz tog razloga je dobro što je proceduralno generisanje neizvesno. Neizvesnost se često pojavljuje kao jedan od ključnih elemenata igre. To igračima daje osećaj svrhe za nastavak igranja. Ako igrači u potpunosti poznaju logiku iza igre, i znaju šta se tačno dešava u istoj, gube osećaj neizvesnosti i interesovanje i prestaju da troše vreme na istu. Igračima treba dozvoliti da donesu smislene odluke, koje imaju neizvesne posledice. Metode proceduralnog generisanja uvode neizvesnost u video igre na jedinstven način. Na primer, nivoi koji se generišu pseudo nasumično, ili dijaloz i ponašanja likova koje kontroliše kompjuter, svi mogu da zavise od odluka koje igrač donese, ali da ne postoji jedan siguran rezultat za svaku odluku igrača.

Neizvesnost u video igrama ne mora da se ispoljava u obliku neizvesnosti događaja tokom trajanja igre. Mnoge video igre arkadnog tipa, kao „*Space Invaders*“ (1978), uvek imaju isti završetak, a to je da igrač izgubi. Ipak, igra poseduje neizvesnost u obliku rezultata koji igrač ostvari pre nego što se igra završi. Igrač ne može biti siguran koliki će mu biti maksimalan rezultat u partiji koju igra, niti zna da li će oboriti svoj prethodni maksimalan rezultat.

4. GENERISANJE SCENE POMOĆU GRAMATIKA

Potrebno je generisati trodimenzionu scenu upotrebljivu u video igri. Scena treba da sadrži urbanu i/ili prirodnu sredinu, koja treba biti raznolika. Scena se sklupa od predefinisanih trodimenzionih oblika koji su određeni kao elementarni oblici. Ti oblici mogu, uz odgovarajuće tekstone, predstavljati deo bilo čega što se nađe na sceni. Generisana scena mora biti smisljena predstava njoj odgovarajuće sredine u realnom svetu. Obavezno je upotrebiti gramatiku u toku implementacije rešenja. Za primere ranijih rešenja sličnih problema, pogledati reference [3, 4].

Scena se generiše u više koraka, i u pitanju je urbana sredina. Prvi korak je generisanje saobraćajnica. Zatim se

na osnovu saobraćajnica formira matrica koja predstavlja scenu. Kada je matrica formirana, kroz nju se iterira da bi se popunio ostatak scene. Tokom interpretacije, kursor se uvek kreće u istoj ravni.

Saobraćajnice se generišu pomoću L-sistema. L-sistem je implementiran sa osnovnim funkcionalnostima koje su opisane u Poglavlju 2.1. Aksiom, pravila zamene i broj iteracija su ulazne vrednosti koje korisnik može da specifikuje, i smeju sadržati osnovne karaktere opisane u Poglavlju 2.1. L-sistem generiše tekstualnu predstavu saobraćajnice, koja se zatim interpretira i crta na sceni. Tokom crtanja se obavlja i teksturisiranje saobraćajnice.

Sledeći korak je napraviti dvodimenzioni niz, odnosno matricu, objekata koji sadrže informacije o koordinatama polja scene na kojima se generišu njeni delovi, kao i informacije o tipu tih polja na osnovu kojih se zna šta je generisano na tom polju. Tip polja može biti prazno polje, saobraćajnica, građevina, zelena podloga i slično.

Kada je opisana matrica napravljena, iteriranjem kroz nju se dolazi do polja koja još ne sadrže deo scene. Zatim se na njih, sa određenom šansom postavljaju objekti ili odgovarajuće podloge (travnjak i sl.). Pošto objekti mogu da se nalaze samo uz saobraćajnicu, šansa da na takvo polje bude postavljen objekat je veća nego šansa da se postavi podloga. Objekti i podloge se, kao i saobraćajnice, teksturišu prilikom generisanja. Za objekte se odabira jedna od petnaest ponuđenih tekstura, a za podloge se bira tekstura u odnosu na tip podloge. U slučaju da je generisana podloga definisana tako da može da podrži rast vegetacije, uz određenu šansu se postavljaju drva na istu. Primeri generisane scene iz perspektive igrača („*first person*“) su na Slici 2.



Slika 2. Generisana scena iz perspektive igrača

Ako su spoljni faktori kao što su rezolucija i broj tekstura izuzeti, brzina rada algoritma zavisi od tri faktora, i to su broj iteracija u radu L-sistema, sadržaj aksioma i broj promenljivih i njima odgovarajućih pravila zamene.

Ni jedan od ovih faktora nije značajniji od druga dva, već su svi međusobno zavisni. Dovoljno visok broj iteracija će uvek loše uticati na performanse algoritma, ali u slučajevima kada se zahteva kompleksan aksiom i/ili kompleksna pravila zamene, čak i mali broj iteracija (manje od 10) može znatno da utiče na performanse. Iz tog razloga nije moguće tvrditi da će za svaku situaciju manji broj iteracija obezbediti bolje performanse algoritma.

U Tabeli 1 predstavljene su performanse za jednu promenljivu „F“, aksiom „FFF-FFF-FFF-FFF“, i jedno pravilo zamene „F=FFF-FFF+FFF-FFF-FFF“.

Tabela 1. Performanse algoritma kroz različit broj iteracija

Broj iteracija	Prosečno trajanje [s]
1	0,05
2	0,75
3	65,3

Iz tabele je moguće videti da za dati primer aksioma i pravila zamene performanse naglo opadaju sa povećanjem broja iteracija. Ipak, za ovaj primer je dovoljno samo dve iteracije da bi se generisala scena dovoljna za neke video igre. Primer rezultata generisanja sa dve iteracije je na Slici 2. Trenutna implementacija ostavlja dosta opcija za unapređenja, među kojima su: upotreba stanja kursora interpretera, upotreba stohastičkog L-sistema, upotreba L-sistema za generisanje objekata i vegetacije, generisanje tekstura i vodenih površina, kretanje kursora u prostoru i uvođenje segmentacije scene u obliku octree stabla radi jednostavne ručne izmene delova scene ili dodele semantike istima.

5. ZAKLJUČAK

U ovom radu je prikazan jedan pristup proceduralnom generisanju urbane okoline za video igru. Prvi deo algoritma se sastoji od L-sistema koji generiše mrežu saobraćajnica, a drugi deo od jednostavne gramatike koja definiše pravila generisanja ostatka scene. U zavisnosti od parametara L-sistema, algoritam može biti primenjiv u realnom vremenu.

Da bi sa trenutnom implementacijom rezultati generisanja bili raznovrsniji, potrebno je menjati ulazne parametre između svaka dva generisanja. Ulazni parametri su broj iteracija, aksiom i pravila zamene. Dovoljno je promeniti jedan od njih da bi se dobila različita mreža saobraćajnica, ali potrebno je paziti da se ne zadaju parametri koji će onemogućiti rad u realnom vremenu, u slučaju da je to jedan od uslova. Potencijalna unapređenja algoritma su navedena u Poglavlju 4.

6. LITERATURA

- [1] J. Freiknecht, W. Effelsberg, „*A Survey on the Procedural Generation of Virtual Worlds*“, 2017.
- [2] K. Salen, E. Zimmerman, „*Rules of Play: Game Design Fundamentals*, 2004“.
- [3] S. Ince, „*Automatic Dynamic Content Generation for Computer Games*“, 1999.
- [4] Y. I. H. Parish, P. Mueller, „*Procedural Modeling of Cities*“, 2001.

Kratka biografija:



Lazar Anđelić rođen je u Novom Sadu 1993. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Računarstvo i automatika odbranio je 2018.god.

kontakt: cyber.laki@gmail.com