SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Računalna animacija

Proceduralno generiranje gramatikom

Ante Pušić

3. laboratorijska vježba

Zagreb, siječanj 2022.

Sadržaj

1. Teorija

Postupci opisani u ovom dijelu reimplementacija su ideja iz Talton et al. (2012).

1.1. Izgradnja gramatike

Formalizam koji korišteni algoritam rabi za predstavljanje strukture objekta jest stohastička (probabilistička) kontekstno-neovisna gramatika (`https://en.wikipedia.org/wiki/Probabilistic_context-free_grammar`).

Algoritam započinje s obradom primjera i izgrađuje najmanje općenitu konformirajuću gramatiku. Za svaki element u primjeru stvara se nezavršni znak M, te se inicijalizira brojač i. Slijedi iterativni obilazak primjerâ; pri obilasku se za svaki element e s djecom $c_1, ..., c_k$ generira produkcija $A_i \rightarrow eA_{i+1}...A_{i+k}$. Kod elementa e uzima se u obzir i njegov položaj u odnosu na roditelja.

Dobivena gramatika generira svaki od n primjera s vjerojatnosti $1/n$. S druge strane, specifična je za te primjere, pa se proizvoljno kombiniraju dvije produkcije kako bi se uvela varijacija.

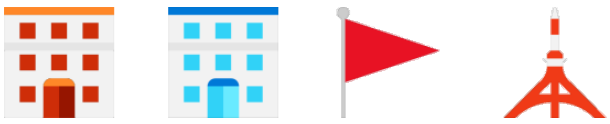
1.2. Postupak generiranja

Nakon što je gramatika izgrađena, objekt se generira sukcesivnom primjenom produkcija. U slučaju da za nezavršni znak postoji više produkcija, vrši se slučajni odabir po njihovoj vjerojatnosti.

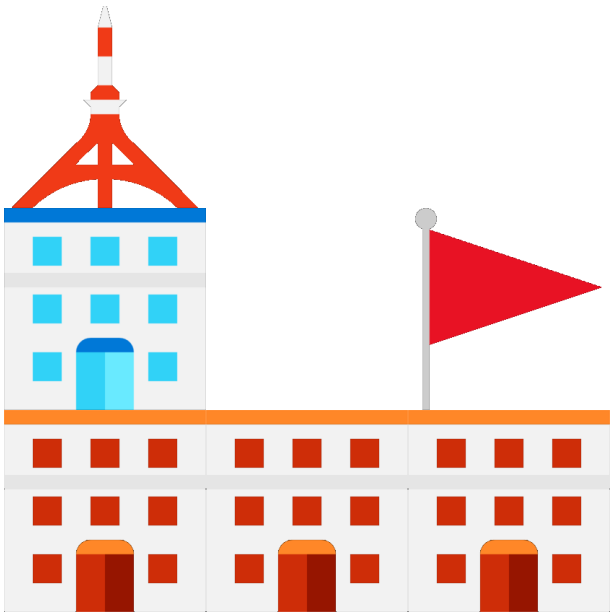
2. Funkcionalnost

Logika projekta implementirana je u `SCFG.py`. Kôd prvo čita primjere iz `examples.py` i na temelju njih izgrađuje gramatiku. Zatim kombinira dvije produkcije i, služeći se tako modificiranom gramatikom, generira nove objekte po uzoru na primjere.

Za svrhu vizualizacije, projekt sadrži unaprijed pripremljene dizajne u direktoriju `resources`. Generirani objekti se vizualiziraju pomoću Pythonove biblioteke `tkinter`, a relativna pozicija elemenata objekta se računa iz pravila pohranjenih u gramatici.



Slika 2.1: Elementi.



Slika 2.2: Jedan generirani objekt.

3. Upute za pokretanje

Izvorni kod ovog projekta nalazi se u GitHub repozitoriju na `https://github.com/antepusic/computer-animation`, u direktoriju `project`. Repozitorij se može klonirati ili preuzeti kao zip arhivu te je zatim raspakirati.

Za pokretanje implementacije potrebno je imati instaliran Python 3.10 ili noviji. Sva funkcionalnost ostvarena je alatima u Pythonovoj standardnoj biblioteci te nije potrebno instalirati druge biblioteke.

Implementacija se može pokrenuti na sljedeća tri načina:

1. U naredbenom retku pozicionirajte se u direktorij `project` i izvršite naredbu `python main.py`.
2. Otvorite direktorij `project` u nekom IDE-ju za Python (npr. PyCharmu) i pokrenite `main.py` kao i bilo koju drugu `.py` datoteku u tom softveru.
3. [Windows] Desnim klikom na `main.py` pojavit će se kontekstni izbornik s opcijom `Edit with IDLE`. Odabirom te opcije datoteka će se otvoriti u okruženju IDLE, a pokreće se pritiskom tipke `F5`.

4. Literatura

Jerry Talton, Lingfeng Yang, Ranjitha Kumar, Maxine Lim, Noah Goodman, & Radomir Mech. Learning design patterns with bayesian grammar induction. *UIST'12 - Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, 10 2012. doi: 10.1145/2380116.2380127.

5. Sažetak

5.1. Hrvatski

Uzorke u postojećim primjerima može se iskoristiti za brzo i raznoliko proceduralno generiranje objekata. Ovaj projekt implementira generator koji izgrađuje formalnu gramatiku strukture objekata iz primjera i stvara nove objekte po uzoru na njih. Projekt reimplementira dijelove rada „Learning Design Patterns with Bayesian Grammar Induction" (Talton et al., 2012) koji se odnose na stvaranje početne gramatike.

5.2. English

Fast procedural generation of diverse objects can be made possible by leveraging the patterns found in sample objects. This project implements a generator that builds a formal grammar of the example objects' structure and uses it to make new objects modeled upon them. The project reimplements the portions of Learning Design Patterns with Bayesian Grammar Induction (Talton et al., 2012) about initial grammar generation.