Ocjena seizmičke ranjivosti zgrada u nizu parametarskim vizualnim programiranjem

Mužić, Leopold

SVEUČILIŠTE U ZAGREBU

GRAĐEVINSKI FAKULTET

DIPLOMSKI RAD

OCJENA SEIZMIČKE RANJIVOSTI ZGRADA U NIZU PARAMETARSKIM VIZUALNIM PROGRAMIRANJEM

Leopold Mužić

Mentor: doc. dr. sc. Mislav Stepinac

UNIVERSITY OF ZAGREB

FACULTY OF CIVIL ENGINEERING

MASTER'S THESIS

Assessment of seismic vulnerability of aggregate buildings by parametric visual programming

Leopold Mužić

Mentor: doc. dr. sc. Mislav Stepinac

Sadržaj

Sažetak

Tema ovog diplomskog rada je ocjena seizmičke ranjivosti zgrada u nizu tako da se već poznata metoda ocjene seizmičke ranjivosti automatizira, te pojednostavni unos podataka što je vise moguće, kako bi se cijeli postupak ubrzao. Za izradu skripte koristio se Grasshopper, parametarski programski paket sadržan u Rhinoceros 3D-u. U radu se prvo detaljno prolazi kroz elemente skripte, te objašnjava logiku procesa kojom ide. Nakon toga se skripta i primjenjuje u analizi stvarnoga bloka smještenog u Zagrebu. S obzirom da je ovo prva inačica skripte, postoji još prostora za daljnje poboljšanje. Cilj je napravit skriptu koja gotovo automatski može analizirati blok i ocijeniti ranjivost pojedine građevine u bloku, te predložiti najbolje rješenje za sanaciju.

Ključne riječi: seizmička ranjivost, parametarsko programiranje, zgrade u nizu, Grasshopper, Rhinoceros 3D

Abstract

The topic of this thesis is assessment of seismic vulnerability of aggregate buildings by automatization of already established method of assessment of seismic vulnerability, to speed up the whole process. Script is made in Grasshopper it`s parametric programming package included in Rhinoceros 3D. In thesis, firstly are explained elements of the script and logical process behind it. Later on script is used on real life block situated in Zagreb. Considering the fact that this is first iteration of script there is much more room to improve. Goal is to make a script that is capable of almost automatically analysing the block and assessing the vulnerability of each individual building and to propose most effective solution to improve it.

Keywords: seismic vulnerability, parametrical programming, aggregate buildings, Grasshopper, Rhinoceros 3D

1.UVOD

Posljednjih nekoliko godina mozemo primijetiti trend povećanja učestalosti prirodnih katastrofa, posljedice tih katastrofa su ljudske žrtve i golemi ekonomski gubici. Između svih prirodnih katastrofa potres je najrazorniji kako po pitanju ljudskih žrtava tako i ekonomski. Mogli smo svjedočiti potresu na Haitiju 2010. koji je odnio, po procjenama, između 46.000 i 316.000 života. Potres u Japanu, Tohoku godinu dana kasnije, uzrokovao je gubitak 20.475 života, te je ostavio preko milijun ljudi bez doma, ekonomska šteta procjenjuje se na 140 milijardi dolara (Kassem i ostali, 2020). S obzirom na razornu moć potresa javila se potreba za prevencijom te umanjivanjem njegovog štetnog djelovanja na ljude. Prvi pokušaj opisivanja ranjivosti građevina potječe iz 1980. godine u SAD-u, Središnjoj i Istočnoj Europi, točnije u seizmički aktivnim državama kao što su Italija, Bugarska, Rumunjska i Grčka. Procjena rizika ranjivosti građevina u urbanim sredinama je složen proces koji se temelji na kombinaciji tri glavna faktora: Hazard (H), izloženost (E) i ranjivost (V). Adekvatna procjena ranjivosti postojećih građevina te implementacija odgovarajućih rješenja iznimno je bitna za smanjenje negativnog učinka potresa, povećanja sigurnosti ljudi i smanjenja socio-ekonomskih gubitaka u slučaju budućih potresa. U urbanim sredinama, upravljanje rizicima često se provodi bez odgovarajućeg prostornog planiranja. Velika gustoća naseljenosti i neadekvatno obnovljene odnosno sanirane građevine negativno utječu na ukupnu ranjivost cijelog urbanog centra i imaju katastrofalne posljedice u slučaju potresa. Stoga, procjena rizika ranjivosti ima značajnu ulogu u planiranju smanjenja ranjivosti gradova i osiguranju zaštite stanovništva i smanjenju njihove izloženosti potresnom djelovanju. U radu se ocjenjuje seizmička ranjivost zgrada u nizu/bloku pomoću parametarskog vizualnog programiranja. Metoda po kojoj se ocjenjuje se sastoji od petnaest parametara, svaki s određenom težinom na konačni rezultat. Metoda je prvotno razvijena za ocjenu rizika samostalnih građevina. Sastojala se od deset parametara, a kasnije je prilagođena za zgrade u nizu dodavanjem pet novih parametara koji uzimaju u obzir okolne građevine i njihovu međusobnu interakciju čime se pruža kompleksnija i točnija ocjena seizmičkog rizika ranjivosti (Formisano i ostali, 2015)(Petrini & Benedetti, 1984).Obje

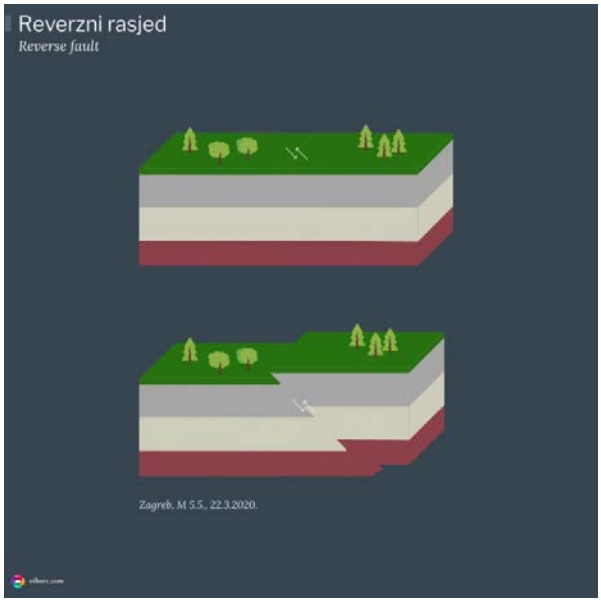
metode razvijene su u Italiji za ocjenu starih gradskih jezgri. S obzirom na potres koji je pogodio područje Hrvatske, odnosno Zagreba i Petrinje 2020. godine, javila se potreba za brzom ocjenom seizmičke ranjivosti građevina. Grad Zagreb je kroz veći dio svoje povijesti bio pod utjecajem Beča, stoga možemo reći da ima sve elemente srednjeeuropskog grada, to se također očituje i u njegovoj arhitekturi pa posljedično i u korištenim tehnikama gradnje i materijalima. Centri gradova, posebice većih, zbog ubrzanog razvoja građeni su bez plana, s ciljem smještaja velikog broja stanovništva na što manjem prostoru. Postojeće građevine nerijetko su se rušile da bi se na njihovom mjestu gradile nove, dizani su katovi, a ukoliko je postojao prostor između postojećih on se koristio za građenje novih i tako su nastajali blokovi građevina u nizu. Takva neplanska gradnja tijekom godina u različitim vremenskim periodima dovela je do građevina u nizu koje variraju po visini, visini etaža, broju katova, korištenim materijalima i tipovima nosive konstrukcije. Blokovi su paralelni s ulicama. U Zagrebu nosivi zidovi su postavljeni isključivo paralelno s ulicom, a okomiti zidovi su samo pregrade ili zabati, što je izrazito nepovoljno prilikom potresa. Generalno gledano to su konstrukcije koje su građene da dobro podnose gravitacijska opterećenja, dok su izrazito osjetljiva na djelovanje potresa. Zbog svoje kulturološke važnosti bitno je razumjeti njihovo seizmičko ponašanje kako bi se mogli provesti zahvati s ciljem očuvanja tih građevina, a i same sigurnosti ljudi koji u njima obitavaju. Jedan takav blok građevina u nizu će se i analizirati u ovom radu.

2. POTRES

Zagreb se nalazi na sjecištu tri velika regionalna tektonske područja: Alpe na sjeverozapadu, Panonski bazen na istoku i Dinaridi na jugu (Slika 2.1). Potresi u tom području uzrokovani su tektonskim pomicanjima u gornjoj kori zbog interakcije između Europske ploče i Jadranske mikro ploče (Markušić i ostali, 2020).

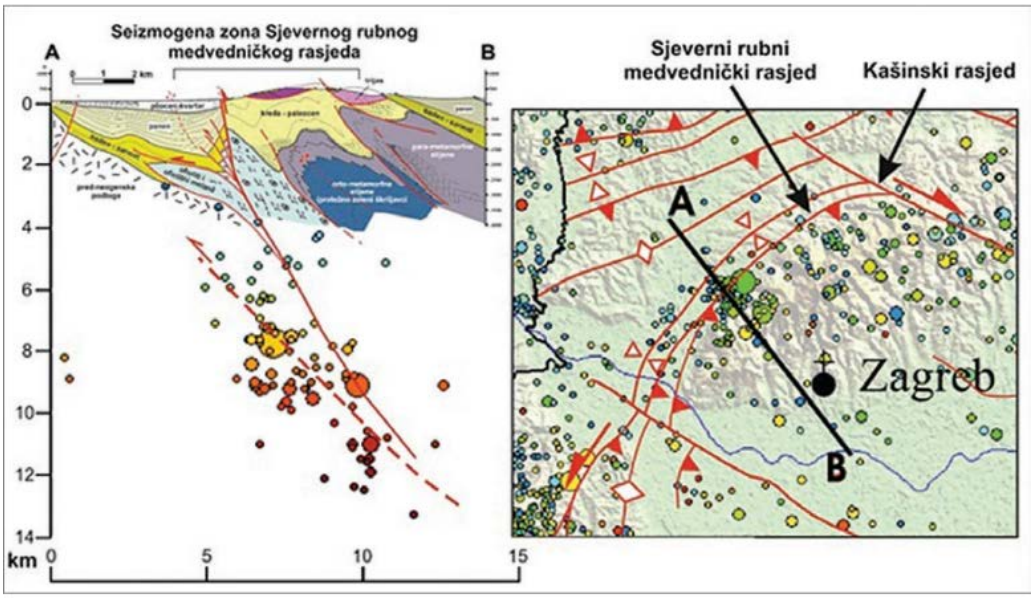
Slika 2.1 Regionalne tektonske ploče (Ćatić, 2022)

Akumulira se energija koja se očituje kao tektonska napetost, a razlog je konvergencija između ploča. Kada naprezanje između ploča postane veće od posmične čvrstoće tla, dolazi do naglog otpuštanja energije, odnosno potresa. Uzrok potresa na Zagrebačkom području su uglavnom seizmičke aktivnosti povezane s reverznim rasjedima (Slika 2.2) s orijentacijom sjeveroistok-jugozapad, koje su rezultat miocenske tektonske inverzije u tom dijelu Panonskoga bazena.



Slika 2.2 Reverzni rasjed (Vibor Cipan, 2021).

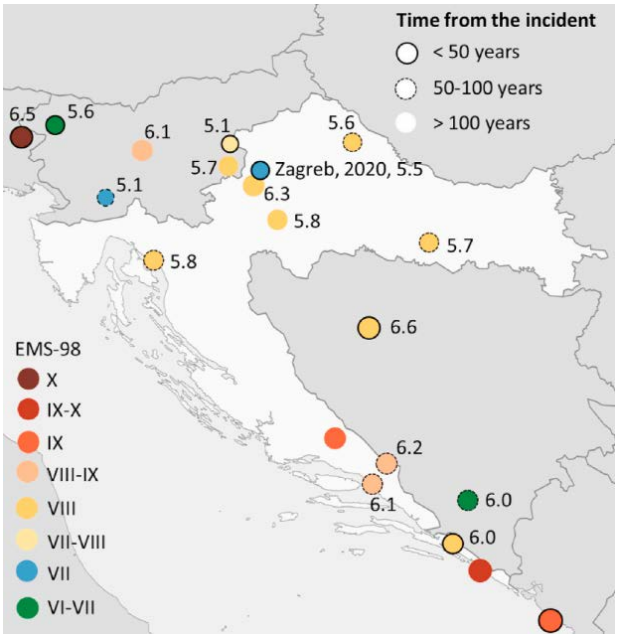
Dva izrazito seizimogena rasjeda na tom području su medvednički rasjed (Slika 2.3) (sa smjerom pada rasjedne plohe od sjeverozapada prema jugoistoku) i Kašinski rasjed koji je okomit na prethodni, pružanju (Bonevska T. i ostali, 2020).



Slika 2.3 Rasjedi u okolici Zagreba (Bonevska T. i ostali, 2020).

Sustavno prikupljanje podatak o potresima u Hrvatskoj pa tako i u Zagrebu počinje u 19. stoljeću, iako je bilo mnogo potresa na zagrebačkom području i prije. Poznati hrvatski geofizičar Mohorovičić došao je do broja od 661 potresa u razdoblju od 1502. do 1883. godine. Iako nije bilo puno podataka o potresima prije 1879. Kišpatić izdaje

publikaciju koja je sadržavala sve informacije o potresima sa epicentrom u okolici Zagreba (Slika 2.4).



Slika 2.4 (Stepinac i ostali, 2021).

1880. godine Zagreb pogađa potres magnitude 6.3 s hipocentrom blizu Kašine. Povijesni izvori nam govore da je u to doba Zagreb imao nešto manje od 30 000 stanovnika i oko 2500 građevina (Markušić i ostali, 2020). Posljedica potresa bile su dvije žrtve i 29 ozlijeđenih. Oštećeno je 1.758 građevina (Slika 2.5), a procijenjena materijalna šteta iznosila je 50% BDP-a tadašnje države. Bilo je potrebno 25 godina da se Zagreb oporavi od tog potresa (Stepinac i ostali, 2021). Medu stanovništvom je vladala velika panika pa su se tako širile priče kako se ispod Zagreba nalazi vulkan te da još veća katastrofa slijedi koja će u potpunosti uništiti grad. Usprkos strahotama koje je taj potres uzrokovao on predstavlja prekretnicu u načinu percipiranja potresa. Počinje se sa sustavnim proučavanjem potresa i potres ne biva više interes samo pojedinaca, znanstvenika, već cijele zajednice (Markušić i ostali, 2020). Također potres 1880. doveo je do promjena praksi u gradnji i urbanom planiranju razvoja grada. Nosioci promjena bili su bankari, trgovci i zemljoposjednici koji su zajedno s arhitektima i inženjerima, uglavnom školovanim u Beču, oblikovali grad u europskom duhu ponajviše po uzoru na Beč. Iako je bilo još potresa nakon tog 1880. niti jedan nije imao tako veliki utjecaj na zajednicu, pa 140 godina nakon tog potresa većina lekcija biva zaboravljeno, a percepcija potresa u javnosti potpuno se izmijenila. Ignoriraju se pozivi stručnjaka da se izradi strategija koja bi imala za ulogu smanjiti

posljedice potresa, osigurati sigurnost stanovnika i smanjiti socio-ekonomske učinke na društvo (Stepinac i ostali, 2021).

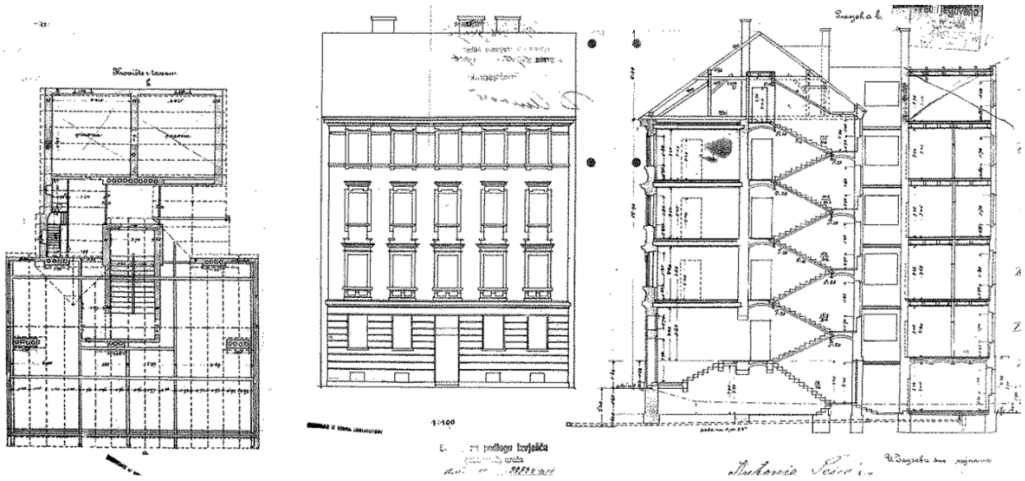
Slika 2.5 Razaranje od potresa 1880. godine (Fabbrocino Giovanni, 2008).

Serija potresa koja je pogodila Zagreb u ožujku 2020. godine bili su prvi značajniji potresi na području Hrvatske. Glavni potres bio je magnitude M_L=5,5 prema Richteru (M_W=5,3), a najjači naknadni iznosio je 4,9 po Richteru (M_W=4,7)(Novak i ostali, 2020). Intenzitet glavnog potresa je bio VII Prema EMS ljestvici, dok je intenzitet pratećeg bio VI. Epicentar je bio 7km od centra grada na dubini od 10km, u Pod sljemenskoj zoni, točnije u Markuševcu (Stepinac i ostali, 2021). Naknadno je bilo još pratećih podrhtavanja tla, u prvih 45 dana od potresa zabilježeno je preko 1400 potresa (Novak i ostali, 2020). U potresu je smrtno stradala jedna osoba, dok ih je 26 bilo ozlijeđeno (18 teže). Nastala materijalna šteta bila je i veća nego sto bi se moglo očekivati za potres takve jačine. Procjenjuje se da je 25.000 građevina bilo oštećeno potresom, te da je gotovo 800.000 ljudi osjetilo potres intenziteta VII (EMS-98), zahvaćena površina oštećenih građevina iznosila je 22,2 milijuna kvadratnih metra(Stepinac i ostali, 2021). Svjetska banka zajedno s Hrvatskom vladom procijenila je štetu na 11,3 milijarde eura. Razlog za toliku štetu s obzirom na jačinu potresa je veliki broj oštećenih građevina koje pripadaju spomenicima kulture. Gotovo cijeli centar grada je zaštićeno kulturno dobro sto samu obnovu čini mnogo zahtjevnijom i skupljom.

3. ZGRADE U BLOKU

3.1 Karakteristike zgrada u nizu u centru Zagreba

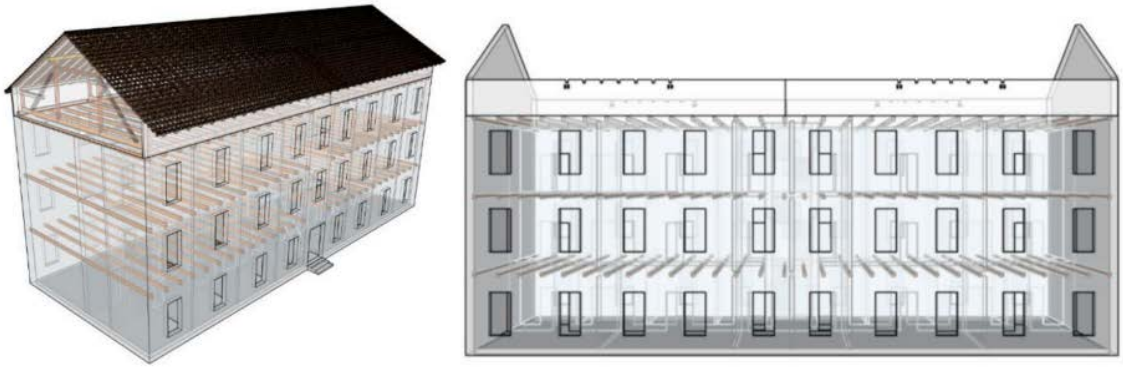
Veliki broju građevina u centru Zagreba premašio je predviđeni projektirani životni vijek (50 godina). Uzimajući u obzir period građenja možemo u grubo procijeniti seizmičku otpornost građevine. Metode građenja te materijal koji se koristio mijenjao se ovisno o razdoblju. Nakon potresa 1880. godine gradilo se po tada pravilima Austro-Ugarskog carstva (Slika 3.1). Građevine možemo s obzirom na vrijeme građenja podijeliti u nekoliko razdoblja. Najveći broj zgrada u donjem gradu su zidane s drvenim međukatnim konstrukcijama i krovovima. Prosječna visina etaže je između 3.5m i 4.7m.



Slika 3.1 Tipična kuća u donjem gradu (Moretić i ostali, 2022).

Zatim, postepeno međukatne konstrukcije počinju biti armirano betonske, no takva gradnja počinje prevladavati tek u razdoblju između 1945 i 1964. Uvođenjem prvih seizmičkih regulativa počinju se graditi zidane konstrukcije s horizontalnim i vertikalnim ukrućujućim elementima odnosno serklažima. Stambene zgrade imale su maksimalno šest etaža, dok su obiteljske kuće imale dvije. Tek nakon 1960. počinju se graditi armirano betonski nosivi sustavi. Od 1981. pa do 2007. koriste se Jugoslavenske norme za potres, nakon 2007. se koriste ENV i tek nakon 2013 se uvodi europski sustav normi, tj. Konstrukcijski Eurokodovi. Možemo zaključiti da je većina građevina građena prije normi za potres (čak 29%) te ne postoje elementi koji mogli prihvatit sile od potresa. Većina međukatnih konstrukcija je drvena, puno su

fleksibilnije od armiranobetonskih, te se može pretpostaviti da će kod njih mehanizam otkazivanja biti otkazivanje van ravnine (Slika 3.3), zbog slabih veza između zidova i međukatnih konstrukcija (Slika 3.2).



Slika 3.2 Konstrukcijska shema tipične stambene građevine (Stepinac i ostali, 2021).

Također većinu starijih građevina stanari su tokom vremena adaptirali, pa tako nije nepoznata praksa primjerice razbijanje nosivih zidova (Slika 3.5), smanjivanje poprečnog presjeka zidova zbog ugradnje ormara (Slika 3.4) itd. Spomenuto za posljedicu ima negativan utjecaj na seizmičku otpornost građevine.

Slika 3.3 Slom van ravnine (Stepinac i ostali, 2021).

Slika 3.4 Smanjenje poprečnog presjeka zida (Stepinac i ostali, 2021).

Slika 3.5 Adaptacija postojećih građevina (Stepinac i ostali, 2021).

Iako su stručnjaci konstantno ukazivali na nedostatak konstrukcijske obnove u centru grada, nije bilo sluha medu političarima kako bi se pokrenuli projekti koji bi to omogućili, radila se samo energetska obnova, posljedice zanemarivanja mogla su se vidjeti nakon potresa. (Stepinac i ostali, 2021)

3.2 Metodologija analize

Blok je analiziran koristeći metodologiju razvijenu od strane talijanskih znanstvenika (Petrini & Benedetti, 1984). Metoda se sastoji od deset parametra po kojima se

ocjenjuje zgrada sa četiri ocjene A,B,C i D svaka ocjena nosi određenu brojčanu vrijednost koja se množi s težinskim faktorom toga parametra (Tablica 3.1) (Tablice 3.3-3.12). Ova metoda je razvijena za ocjenu indeksa ranjivosti pojedinačnih zgrada te je bila potrebna dodatna adaptacija kojom bi se uzele u obzir i zgrade koje se dodiruju (Formisano i ostali, 2015) zato dodaju još pet parametara koji uzimaju u obzir interakciju između susjednih građevina (Tablica 3.2) (Tablice 3.13-3.17).

Tablica 3.1 Tablica parametara (Petrini & Benedetti, 1984).

Parametri	Klasa ranjivosti				Težinski faktor
	A	B	C	D	
1. Organizacija vertikalnih konstrukcija	0	5	20	45	1.00
2. Karakteristike vertikalnih konstrukcija	0	5	25	45	0.25
3. Lokacija građevine i tip temeljena	0	5	25	45	0.75
4. Tlocrtna distribucija elemenata otpornosti	0	5	25	45	1.50
5. Jednostavnost oblika	0	5	25	45	0.50
6. Vertikalna pravilnost	0	5	25	45	0.80
7. Tip međukatne konstrukcije	0	5	25	45	0.80
8. Tip krovišta	0	15	25	45	1.00
9. Detalji	0	5	25	45	0.25
10. Stanje konstrukcije	0	5	25	45	1.00

Tablica 3.2 Tablica parametara (Formisano i ostali, 2015).

Parametri	Klasa ranjivosti				Težinski faktor
	A	B	C	D	
1. Visine okolnih građevina u odnosu na promatranu	0	5	20	45	1.00
2. Položaj građevine u nizu	0	5	25	45	1.50
3. Broj stepeničasto raspoređenih međukatnih konstrukcija	0	5	25	45	0.50
4. Konstrukcijska ili tipska heterogenost susjednih zgrada	0	5	25	45	1.20
5. Postotak razlike između površina otvora susjednih građevina	0	5	25	45	1.00

Tablica 3.3 Organizacija vertikalnih konstrukcija

Klasa	Ocjena	Opis
A	0	Zgrada građena u skladu sa seizmičkim propisima
B	5	Opeka s AB stropnom konstrukcijom i AB gredama
C	20	Opeka s AB stropnom konstrukcijom
D	45	Opeka bez AB stropne konstrukcije

Tablica 3.4 Karakteristike vertikalnih konstrukcija

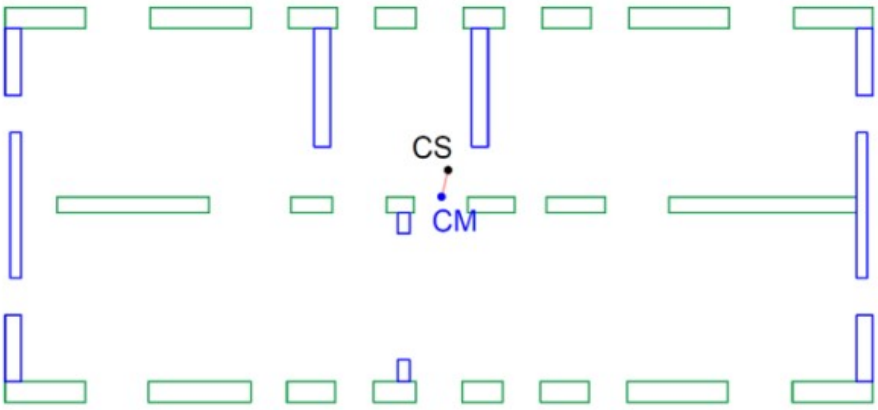
Klasa	Ocjena	Opis
A	0	Pravilna blok opeke (homogeni vez) ili zidovi s horizontalnim vezama
B	5	Pravilna blok opeke (nehomogen) ili nehomogeni zidovis horizontalnim vezama
C	25	Poluzavršena blok opeke (loša kvaliteta) s heterogenim materijalima bez horizontalnih veza
D	45	Nepravilni vez blok opeke (loša kvaliteta) saheterogenim materijalom bez horizontalnih veza

Tablica 3.5 Lokacija građevine i tip temelja

Klasa	Ocjena	Opis
A	0	Građevine s AB trakastim temeljem - stabilno tlo s nagibom manjim od 5°
B	5	Građevina s AB trakastim temeljem - stabilno tlo s nagibom manjim od 15°; Građevina bez trakastih temelja -loša stijena s nagibom manjim od 10°
C	25	Građevina s AB trakastim temeljem - opće tlo s nagibom većim od 15°; Građevina bez trakastih temelja - glinena tla s nagibom manjim od 15°
D	45	Građevine bez AB trakastog temelja - glinena tla s nagibom većim od 15° ili loša stijena sa nagibom većim od 25°

Tablica 3.6 Tlocrtna distribucija elemenata otpornosti

Klasa	Ocjena	Opis
A	0	Građevine sa $\alpha \geq 1$
B	5	Građevine sa $0.6 \leq \alpha \leq 1$
C	25	Građevine sa $0.4 \leq \alpha \leq 0.6$
D	45	Građevine sa $\alpha \leq 0.4$



Slika 3.6 Nosivi zidovi u x i y smjeru

$$a_0 = \frac{A}{A_t} \quad y = \frac{B}{A} \Rightarrow$$

- a_0 = 0.085
- y = 1.8
- A: $min\ \{A_X; A_Y\}$
- B: $max\ \{A_X; A_Y\}$

$$q = \frac{(A+B)\times h}{A_t} \times P_m + P_S = \frac{(6.75+12.15)\times 3}{79.3} \times 22 + 2 = 17.16\ KNm^{-2}$$

$$C = \frac{a_0 \times \tau_K}{q \times N} \times \sqrt{1 + \frac{q \times N}{1.5 \times a_0 \times \tau_K \times (1+y)}} = 0.18$$

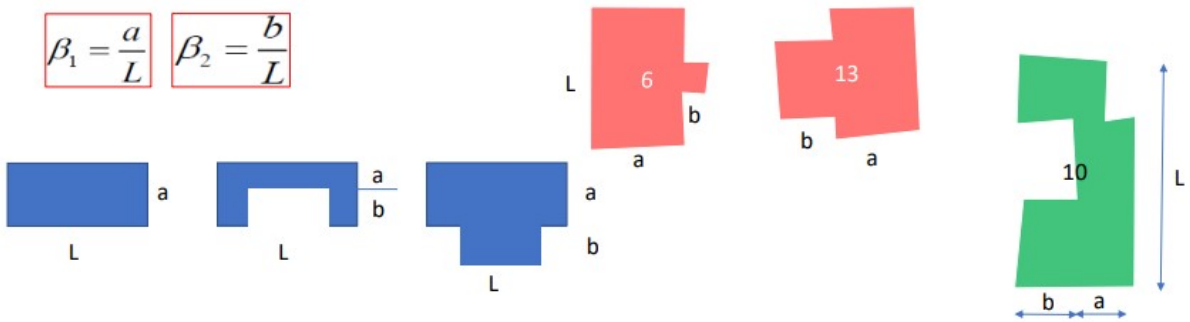
$$\alpha = \frac{C}{0.4} = 0.45$$

- Class C: 0.4 ≤ α ≤ 0.6

Slika 3.7 Proračun α

Tablica 3.7 Jednostavnost oblika

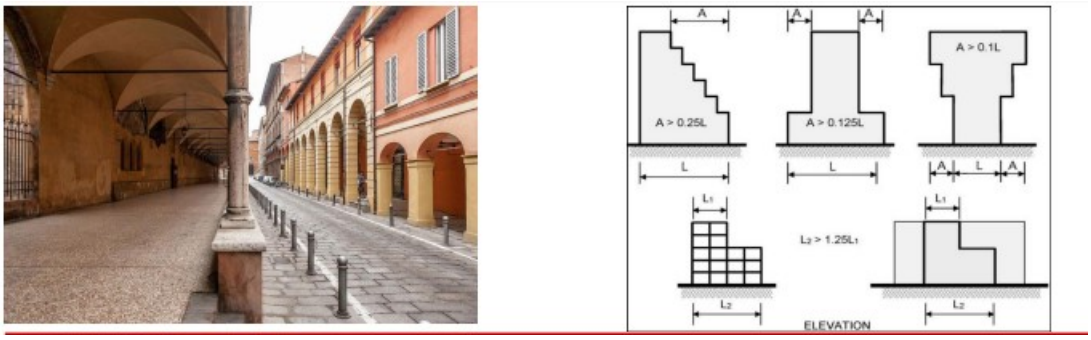
Klasa	Ocjena	Opis
A	0	β1≥0.8; β2≤0.1
B	5	.8>β1≥0.6; 0,1<β2≤0.2;
C	25	0.6>β1≥0.4; 0.2<β2≤0.3;
D	45	0.4>β1; 0.3<β2;



Slika 3.8 Tlocrtni oblik građevina

Tablica 3.8 Vertikalna pravilnost

Klasa	Ocjena	Opis
A	0	Građevine s uniformnom distribucijom mase
B	5	Građevine s arkadama/svodovima/velikim otvorima prema ulici
C	25	Građevine s arkadama koje prekrivaju 10% do 20% ukupne tlocrtne površine
D	45	Građevine s arkadama koje prekrivaju preko 20% ukupne tlocrtne površine



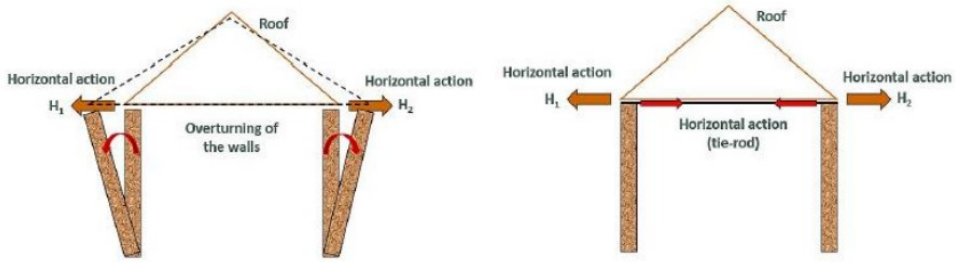
Slika 3.9 Vertikalna pravilnost

Tablica 3.9 Tip međukatne konstrukcije

Klasa	Ocjena	Opis
A	0	Građevine s međukatnom konstrukcijom bilo kakve prirode koja zadovoljava tri uvjeta: (1) Kruta međukatna konstrukcija; (2) Dobra povezanost sa zidovima (3) Odsustvo nepravilno raspoređenih međukatnih konstrukcija
B	5	Građevine s međukatnom konstrukcijom bilo kakve prirode koja zadovoljava tri uvjeta: (1) Kruta međukatna konstrukcija (2) Dobra povezanost sa zidovima (3) Odsustvo nepravilno raspoređenih međukatnih konstrukcija
C	25	Građevine s deformabilnim međukatnim konstrukcijama, ali dobro povezane sa zidovima
D	45	Građevine s različitim međukatnim konstrukcijama i loše povezane

Tablica 3.10 Tip krovišta

Klasa	Ocjena	Opis
A	0	Krovišta sa vlačnim zategama ili veznim gredama
B	5	Krovišta bez potiska, bez vlačnih zatega i bez veznih greda
C	25	Krovišta s malim potiskom bez vlačnih zatega ili veznih greda; Krovišta s potiskom sa vlačnim zategama ili veznim gredama
D	45	Krovišta sa potiskom bez vlačnih zatega ili veznih greda



Slika 3.10 Tip krovišta

Tablica 3.11 Detalji

Klasa	Ocjena	Opis
A	0	Građevine s dobro povezanim okvirima, malim dimnjacima i dobro povezanim balkonima
B	5	Građevine s dobro povezanim okvirima prozora, malim do srednje velikim dimnjacima i dobro povezanim balkonima
C	25	Građevine s loše povezanim znakovima sa zidovima, vanjske instalacije loše povezane sa zidovima
D	45	Građevine s lošim sponama, loše povezani okviri prozora, loše povezani oglasne ploče i visoki dimnjaci

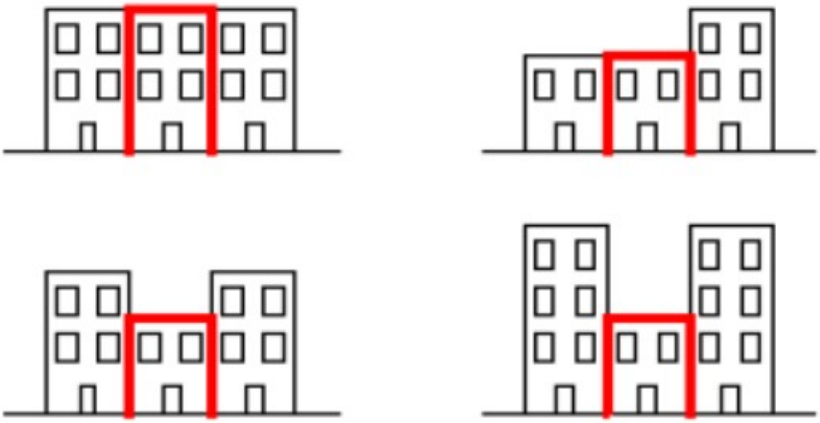
Tablica 3.12 Stanje konstrukcije

Klasa	Ocjena	Opis
A	0	Bez pukotina
B	5	Neproširene kapilarne pukotine
C	25	Pukotine srednje veličine (2-3mm) ili pukotine od seizmičke aktivnosti
D	45	Velike pukotine, parcijalni kolaps zidova, nesigurne ili loše očuvane građevine

Slika 3.11 Stanje građevine

Tablica 3.13 Visine okolnih građevina u odnosu na promatranu

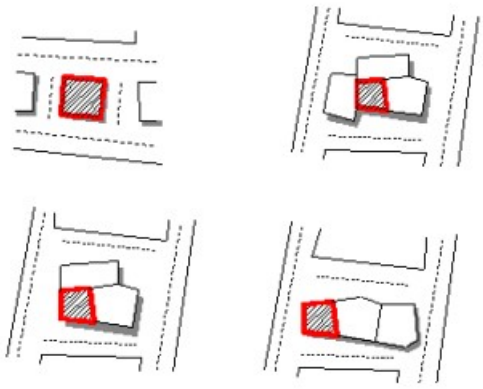
Klasa	Ocjena	Opis
A	-20	Građevine sa susjednim građevinama iste visine
B	0	Građevine susjedne s višim građevinama; Građevine susjedne s višim građevinama i jednom iste visine
C	15	Građevine susjedne s nižim građevinama i jednom iste visine; Građevine susjedne s višim i nižim građevinama
D	45	Građevine susjedne s nižim građevinama



Slika 3.12 Visine susjednih građevina

Tablica 3.14 Položaj građevine u nizu

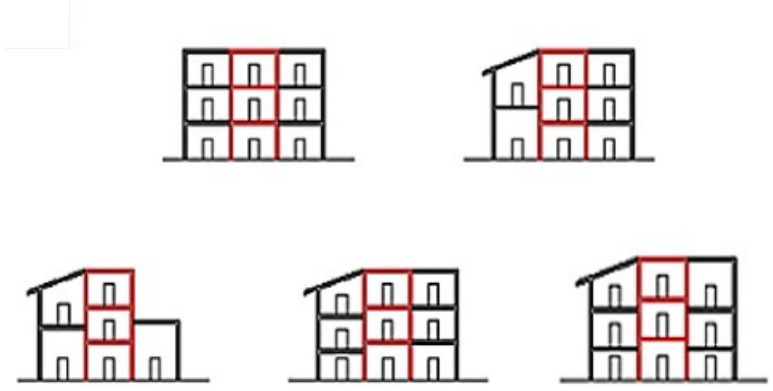
Klasa	Ocjena	Opis
A	-45	Građevina zauzima poziciju u sredini i okružena je s tri strane
B	-25	Građevina zauzima poziciju u sredini i okružena je s dvije strane
C	-15	Građevina se nalazi u kutu bloka
D	0	Građevina zauzima čelnu poziciju u bloku



Slika 3.13 Položaj u bloku

Tablica 3.15 Broj stepeničasto raspoređenih međukatnih konstrukcija

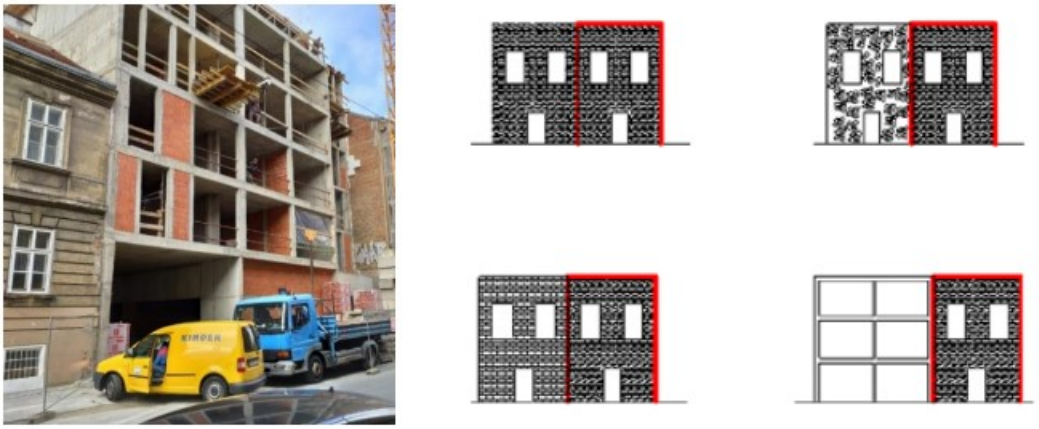
Klasa	Ocjena	Opis
A	0	Nema stepeničastog rasporeda međukatnih konstrukcija
B	15	Par stepeničasto raspoređenih međukatnih konstrukcija
C	25	Dva para stepeničasto raspoređenih međukatnih konstrukcija
D	45	Više parova stepeničasto raspoređenih međukatnih konstrukcija



Slika 3.14 Stepeničasta raspodjela međukatne konstrukcije

Tablica 3.16 Konstrukcijska ili tipska heterogenost susjednih zgrada

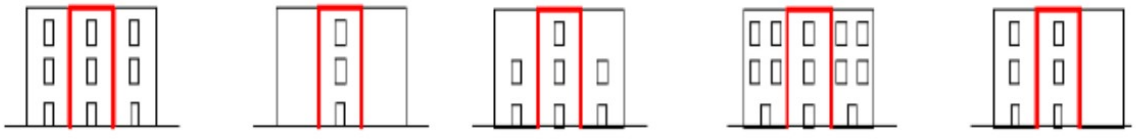
Klasa	Ocjena	Opis
A	-15	Tipološki i konstrukcijski kontinuitet susjednih građevina
B	-10	Građevina građene od istog materijala kao i susjedna, ali drugačijom tehnikom
C	0	Građevina građena od različitog materijala kao susjedna, ali se isto ponašaju
D	45	Građevina pokazuje konstrukcijsku heterogenost u odnosu na susjednu



Slika 3.15 Heterogenost u bloku

Tablica 3.17 Postotak razlike između površina otvora susjednih građevina

Klasa	Ocjena	Opis
A	-20	Postotak otvora u odnosu na susjednu građevinu manji od 5%
B	0	Postotak otvora u odnosu na susjednu građevinu između 6% i 10%
C	25	Postotak otvora u odnosu na susjednu građevinu između 11% i 20%
D	45	Postotak otvora u odnosu na susjednu građevinu veći od 21%

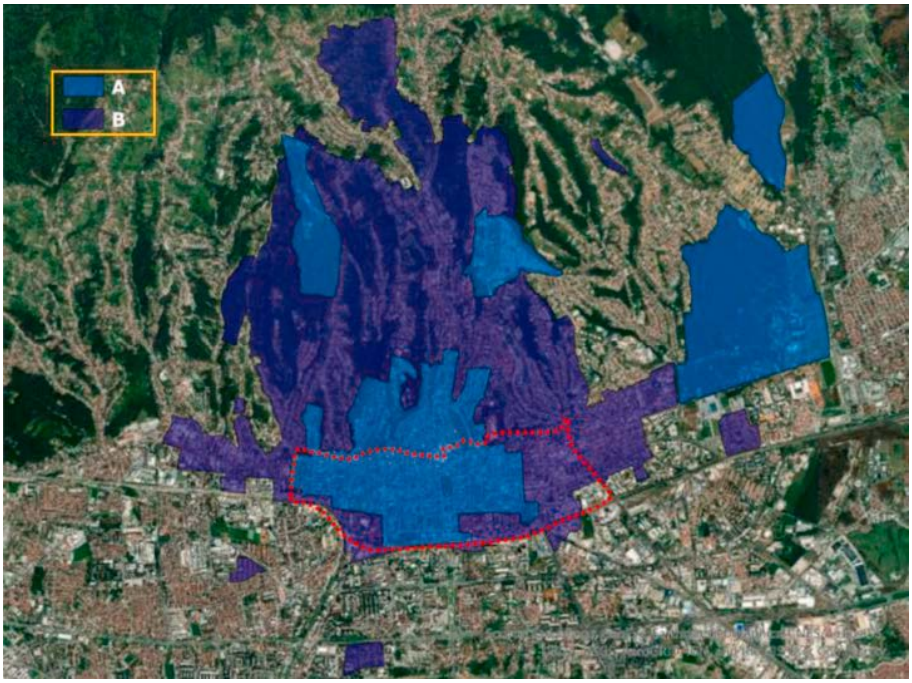


Slika 3.16 Površina otvora

3.3 Analizirani blok

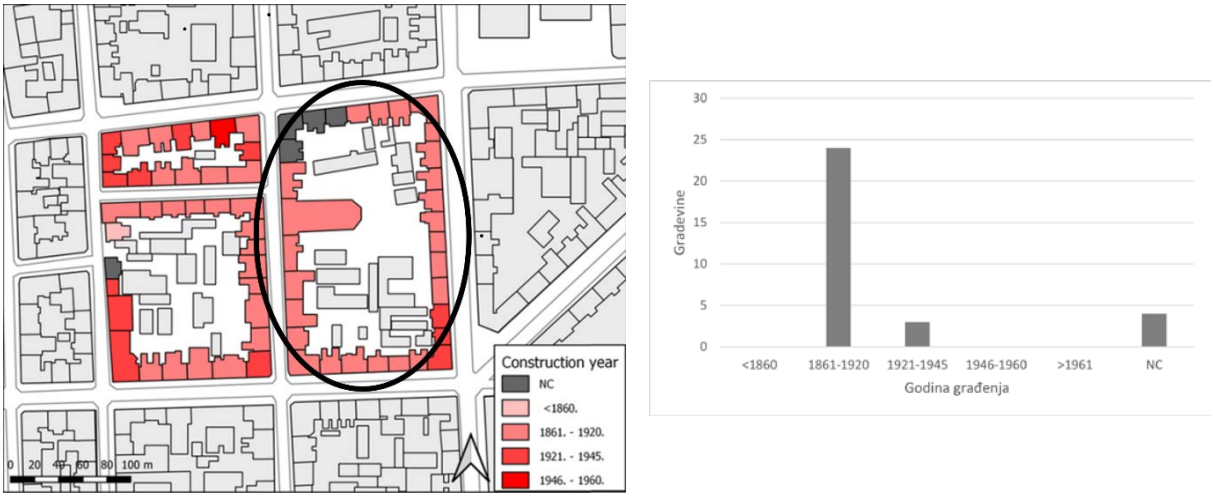
Promatrani blok zgrada u nizu (slika 3.17) koji će se analizirati u ovom radu nalazi se u donjem gradu koji je ujedno zaštićeno područje, zona B (Slika 3.18).

Slika 3.17 Analizirani blok

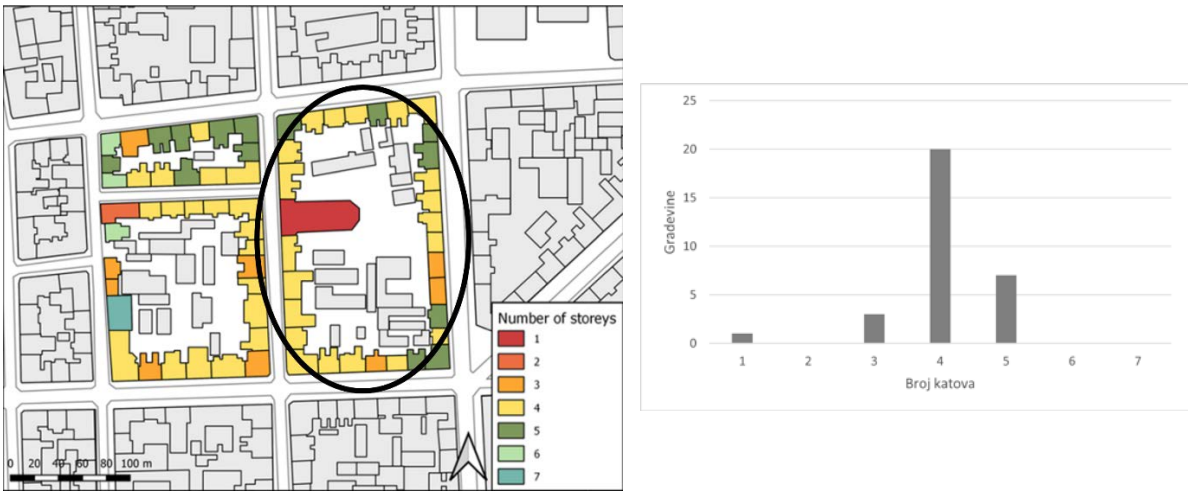


Slika 3.18 Zaštićene zone

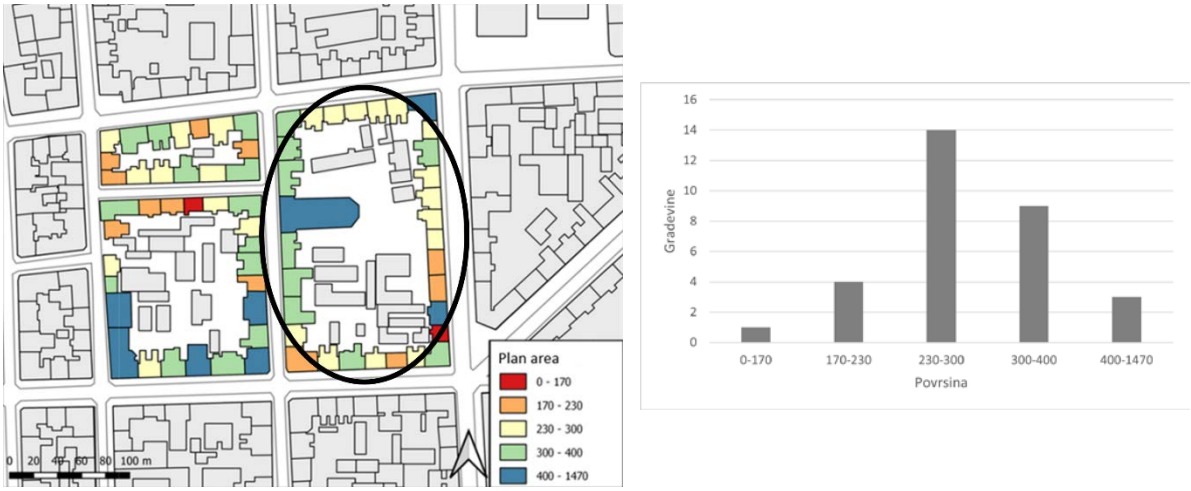
Građevine su s obzirom na konstrukcijska obilježja klasificirana na M1 (građevine s drvenom međukatnom konstrukcijom), M2 gdje su međukatne konstrukcije armiranobetonske i na RC odnosno armiranobetonske konstrukcije koje su očekivano, puno bolje odgovorile na potresna opterećenja, oštećenja su bila uglavnom kozmetičke prirode poput površinskih pukotina na žbuci. M1 građevine su starije od M2, te pripadaju razdoblju u kojem se nije gradilo po seizmičkim normama. Većinu građevina u bloku možemo klasificirati kao M1. Blok je također klasificiran po još tri obilježja: godini građenja (Slika 3.19), broju katova (Slika 3.20) i tlocrtnoj površini (Sliku 3.21). Godina građenja nam govori o tehnikama građenja koje su korištene u tom razdoblju te materijalima koji su korišteni (Slika 3.22). S obzirom na činjenicu da je visina susjednih zgrada, odnosno broj katova bitan zbog negativnog učinka. Sve tri karakteristike imaju utjecaj na krutost i seizmički odgovor građevine. Cjelokupna metodologija opisana je u radu (Moretić i ostali, 2022).
Iz rada su korišteni podaci o građevinama u bloku i kako bi se mogle usporediti rezultati.



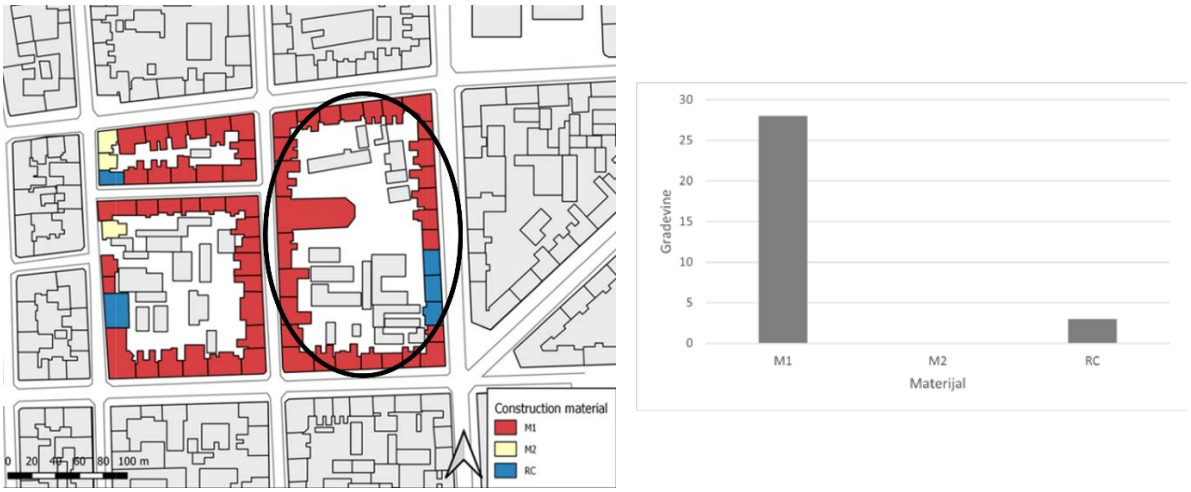
Slika 3.19 Klasifikacija po godini izgradnje



Slika 3.20 Klasifikacija po broju katova



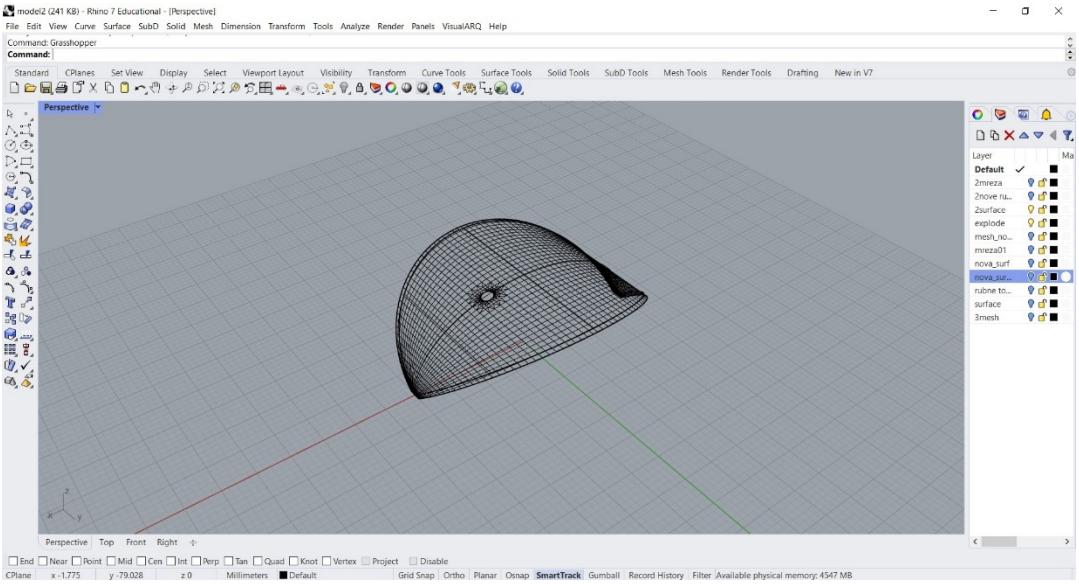
Slika 3.21 Klasifikacija po površini



Slika 3.22 Klasifikacija po materijalu

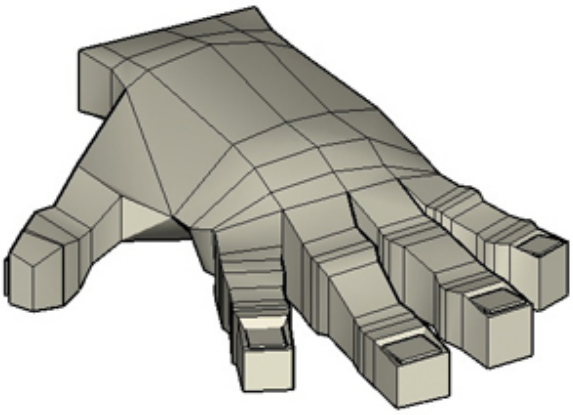
4. PARAMETARSKO PROGRAMIRANJE

U analizi bloka korišten je programski paket „Rhinoceros“ (Slika 4.1). Razvio ga je Robert McNeel sa suradnicima. To je profesionalni 3D CAD program kojem se geometrija zasniva na preciznim matematičkim krivuljama (NURBS- Non Uniform Rational B-Splines) zahvaljujući tome moguće je točno opisati od jednostavnih 2d linija pa sve do složenih 3D površina, za razliku od nekih drugih 3D programa koji koriste poligonalnim mrežama za opis geometrije.

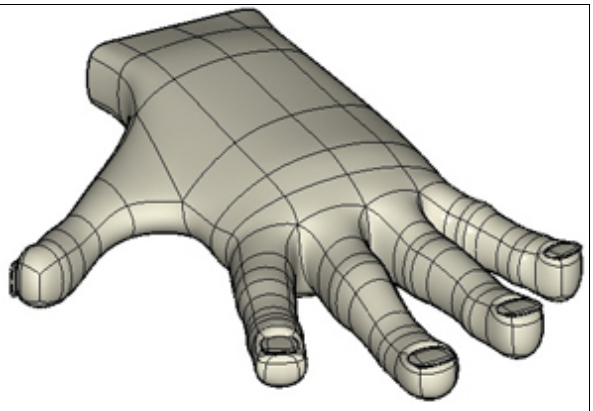


Slika 4.1 Rhino geometrija mreža

U zadnjoj verziji programa, Rhino 7 predstavljena je subD (Slike 4.2 i 4.3) (Subdivision Surface Modelling).(*Rhinoceros*, 2022) Bazira se na tome da napravi pojednostavljenu mrežu, kao svojevrsni kavez, te onda kreira pod dijelove, odnosno „finije“ mreže te tako omogućuje veću preciznost modela i bolju kontrolu elemenata. Bazirana je na Catmull-Clark algoritmu.(*SubD Surface*, 2022)

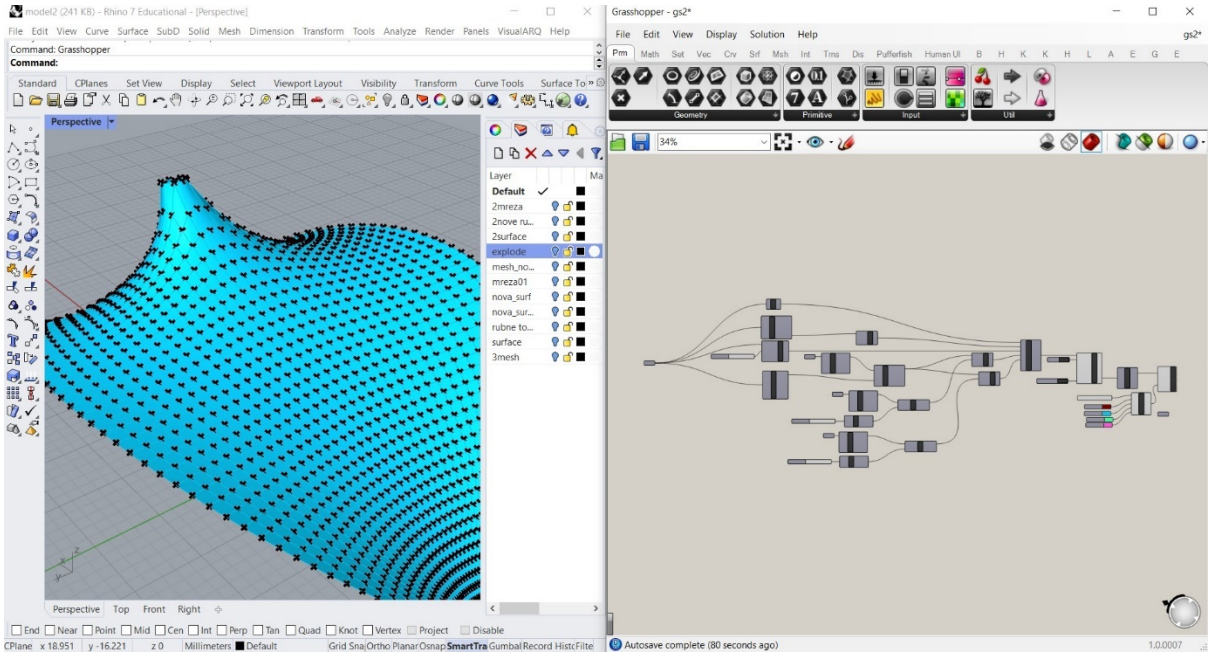


Slika 4.2 Model bez subD geometrije

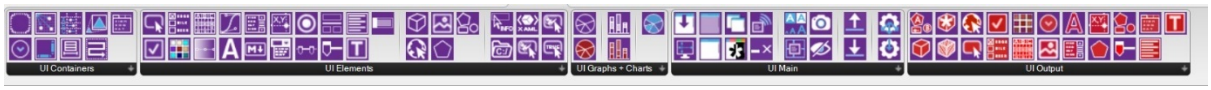


Slika 4.3 Model sa subD geometrijom

U građevini i arhitekturi se često koristi dodatak za „Rhino“, „Grasshopper“ (Slika 4.4). Omogućuje nam parametarsko modeliranje odnosno vizualno programiranje. Tako je i u ovom slučaju korišten primarno Grasshopper s dodacima kao sta su „Human UI“ (Slika 4.5), „Putterfish“ i „Anemone“ i drugi. „Rhino“ je služio za unos podatak te modul na kojem su se prikazivao dobiven rezultat.



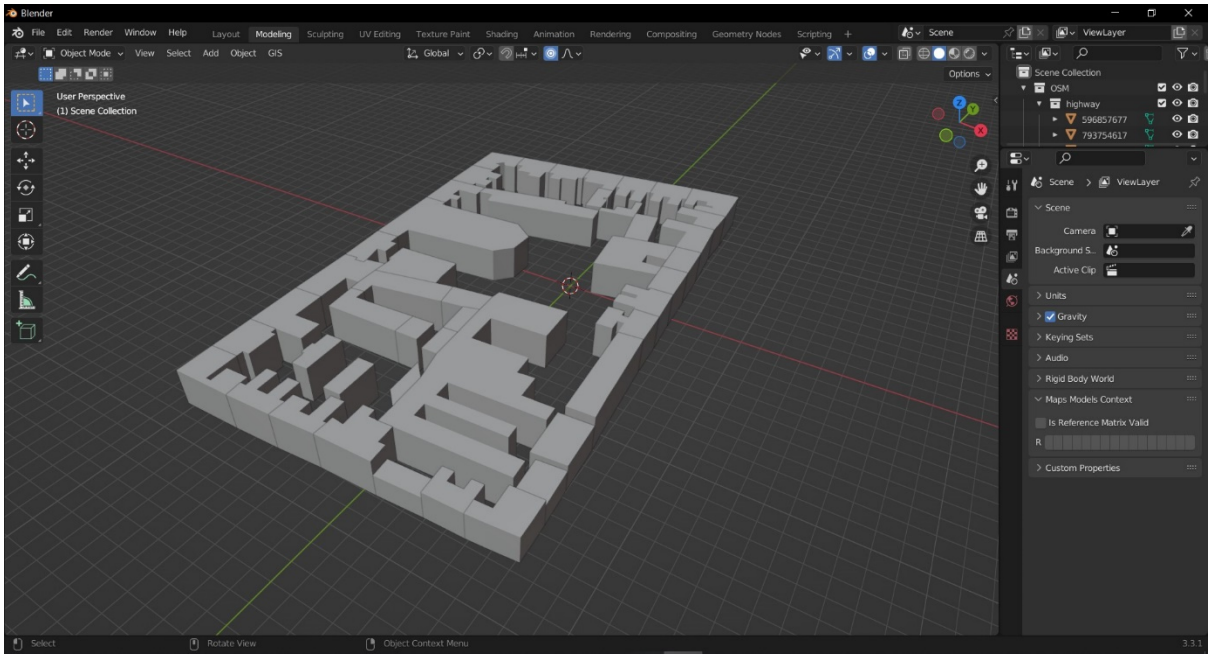
Slika 4.4 Rhino i Grasshopper



Slika 4.5 Human UI dodatak za Grasshopper

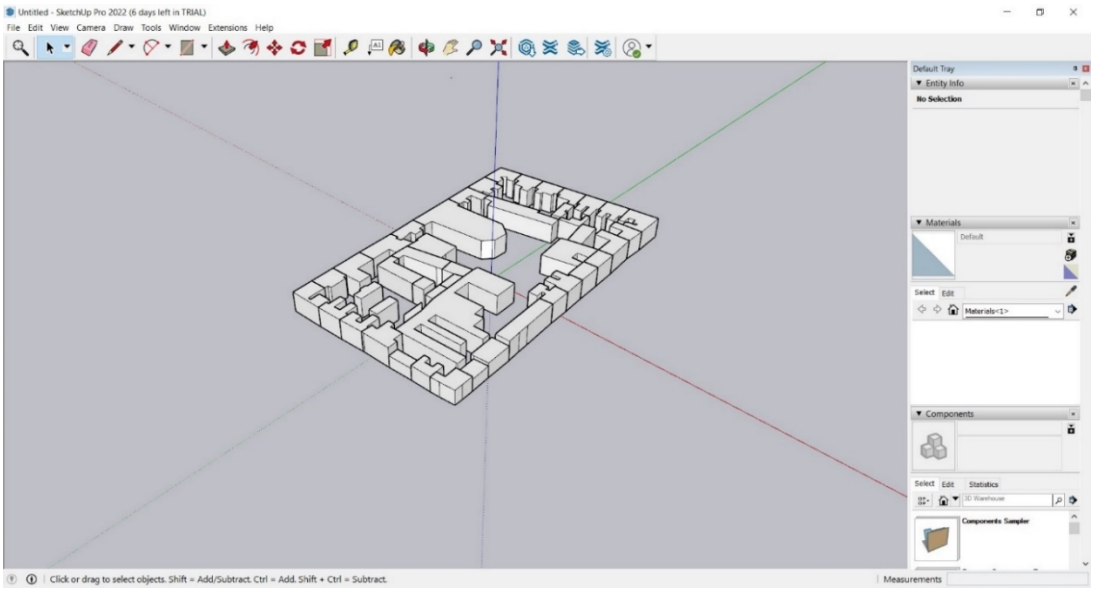
4.1 Pojašnjenje procesa analize

Proces započinje lociranjem traženog bloka u programu „Blender“ (Slika 4.6), tamo pomoću GIS dodatka možemo učitat karte s nekoliko poslužitelja usluga (Google, Bing, Osm). GIS dodatak nam omogućuje da u blenderu učitamo traženi blok s dostupnim visinama.



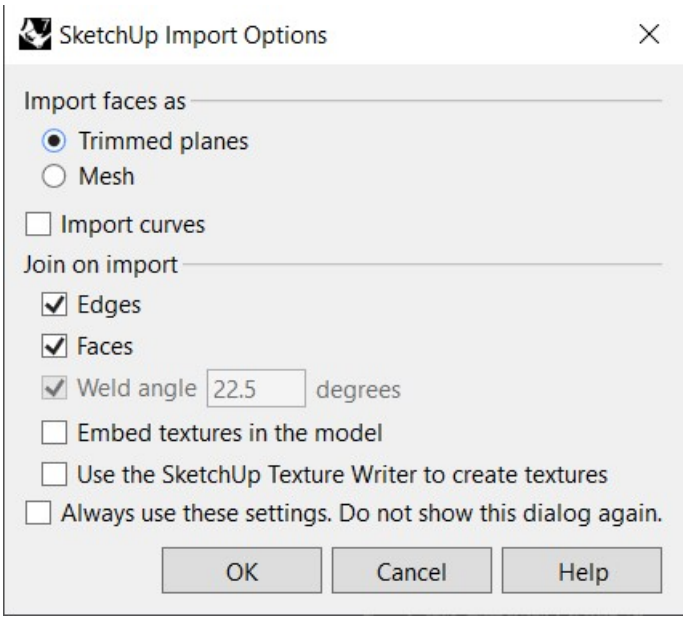
Slika 4.6 Blok u Blenderu

Mana ovoga postupka je što visine često nisu točne pa ne možemo koristiti te visine kasnije u „Rhinu“, odnosno „Grasshopperu“ za analizu bloka. Potom, traženi blok unesemo u „SketchUp“ (Slika 4.7).

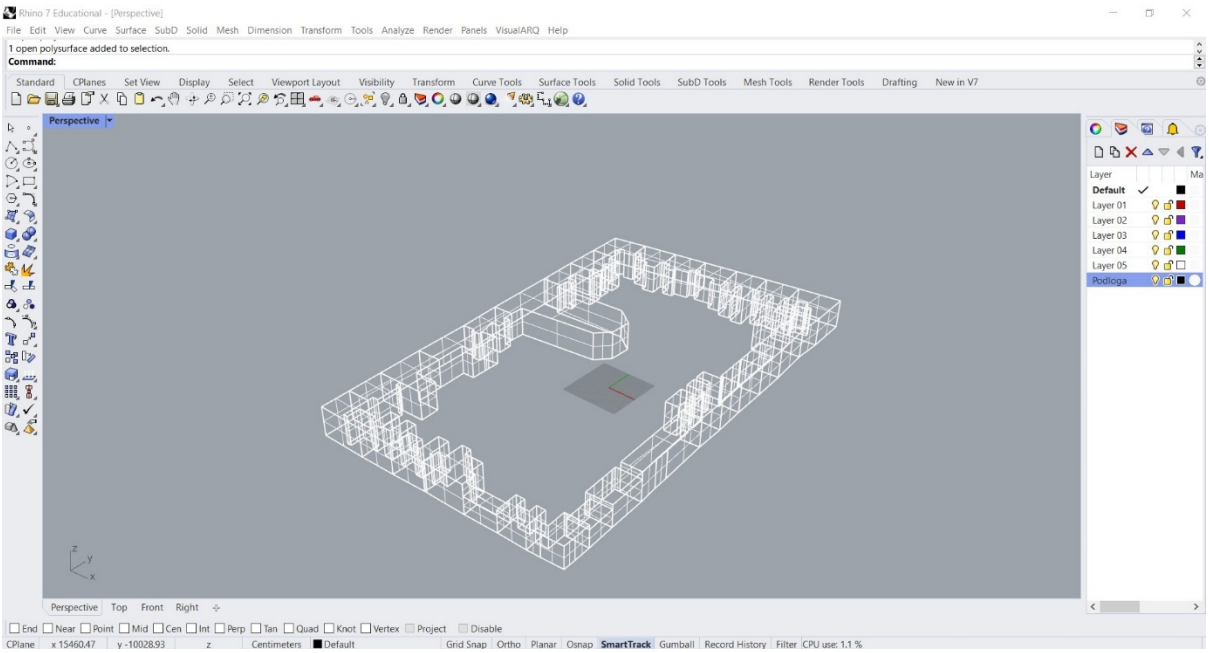


Slika 4.7 Blok u „SketchUp-u“

Razlog zašto nam SketchUp služi kao posrednik je zbog velike kompatibilnosti s ostalim programima pa nam tako omogućuje da imamo veću razinu detalja nego da se direktno prebacio set podataka iz Blendera u Rhino (Slika 4.8). U Rhinu se onda uklanjaju nepotrebne građevine iz bloka tako da ostanu samo one koje se planiraju analizirati i stavlja se u layer „Podloga“ (Slika 4.9).

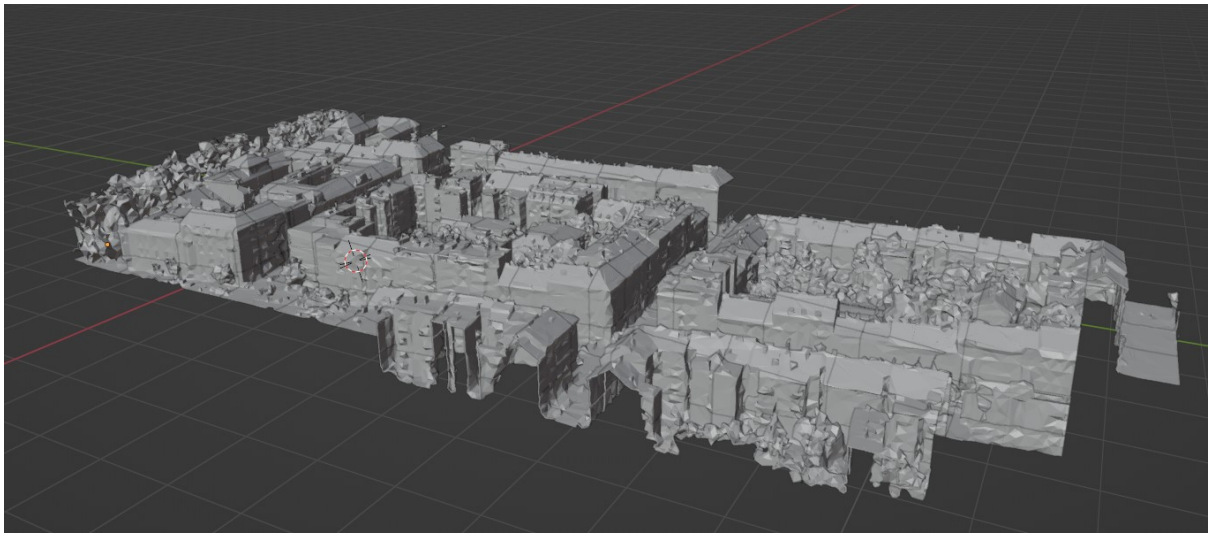


Slika 4.8 Postavke za učitavanje SketchUp modela u Rhino



Slika 4.9 Očišćeni blok u Rhinu

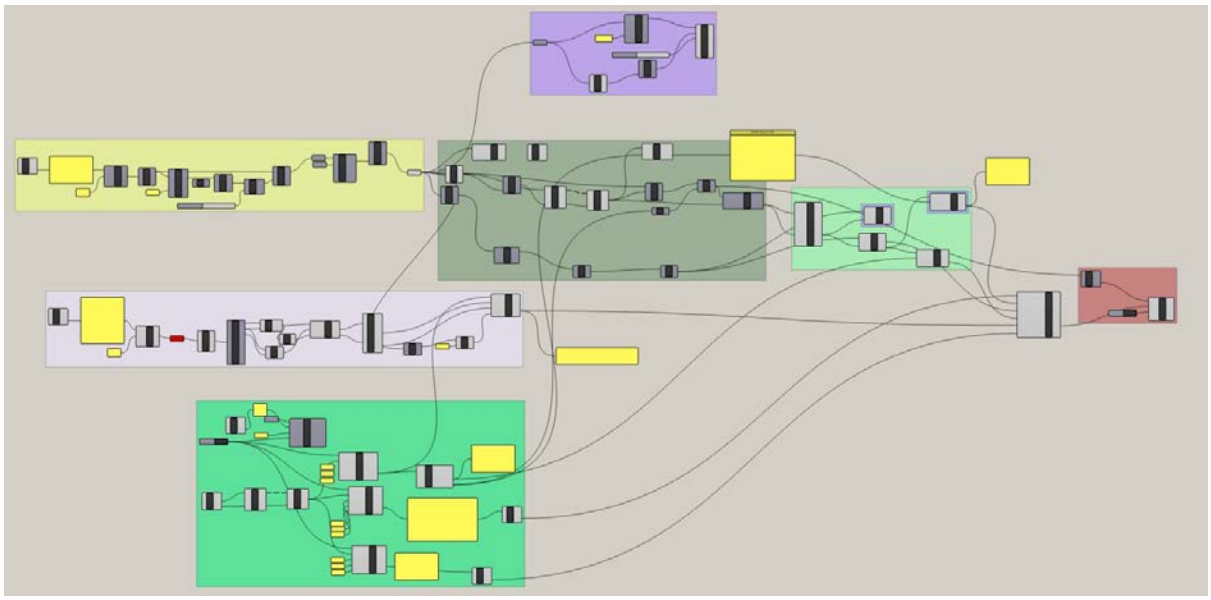
Također postoji još jedna metoda učitavanja bloka u blender, međutim ta metoda ima previše kompliciranu odnosno detaljnu geometriju objekata, što sami model čini nepraktičnim za daljnju manipulaciju (Slika 4.10). Stoga se ova metoda nije koristila u ovoj analizi.



Slika 4.10 Primjer druge metode

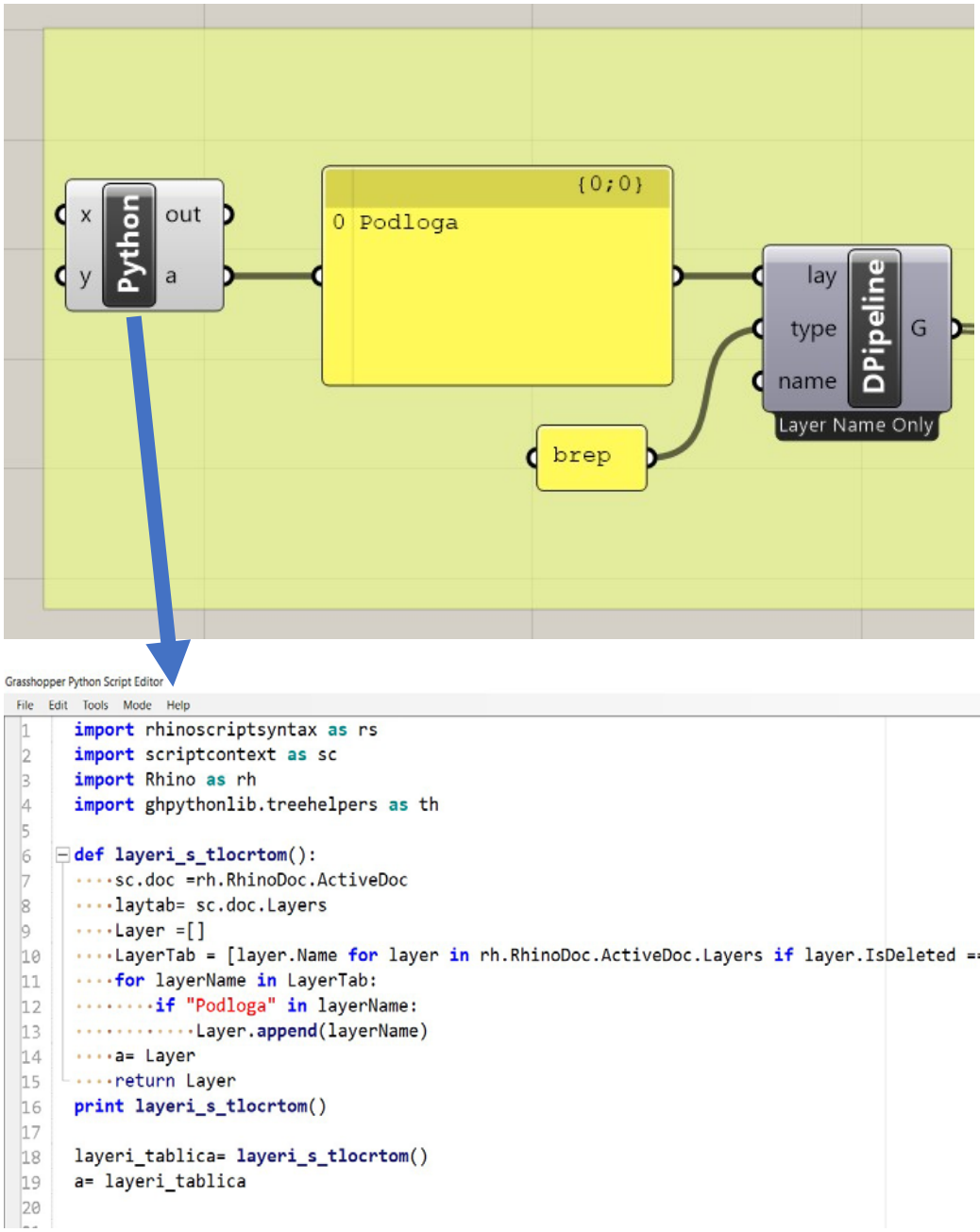
Nakon brisanja građevina koje ne pripadaju bloku ili koje ne možemo analizirati ovom metodom otvaramo skriptu u Grasshopperu (Slika 4.11). Skripta se može podijeliti u nekoliko većih cjelina:

- Generiranje površine,
- numeriranje površina,
- Excel – tablični podaci,
- obrada podataka za proračun parametara,
- proračun otpornosti zidnih elemenata,
- proračun samih parametara i ispis rezultata.



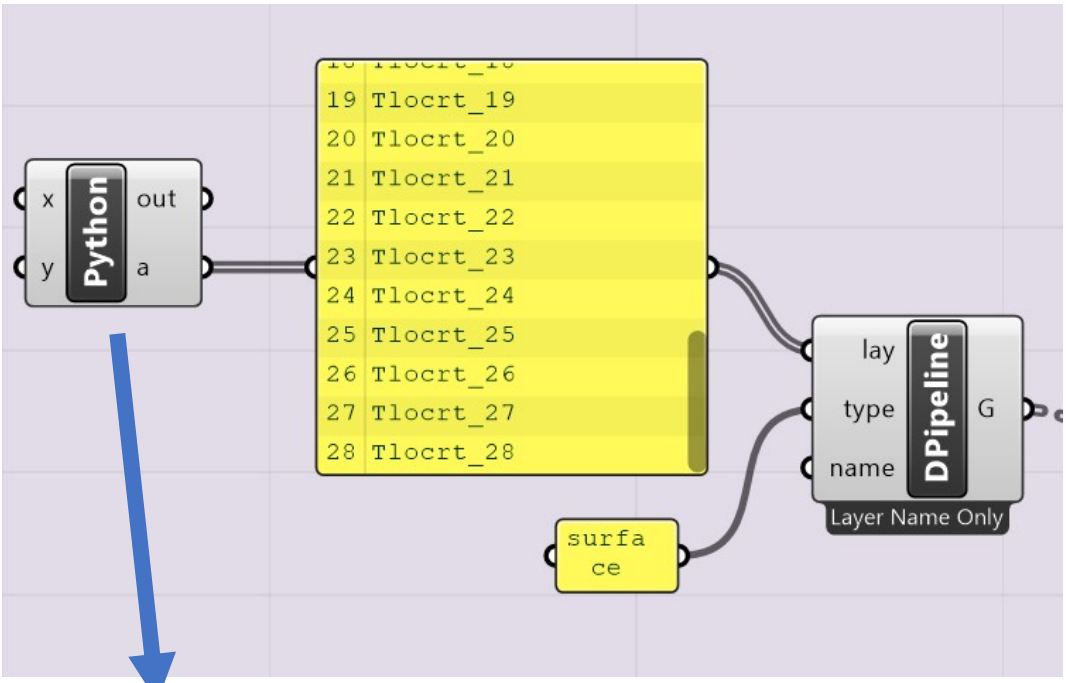
Slika 4.11 Grasshopper mreža

Podaci se unose preko tri inputa: dva su geometrijskog karaktera u obliku površina i „brepsa“ (boundary representation) i tekstualni ili numerički podaci iz Excela. Također i Excel očitava podatak o broju građevina u bloku.



Slika 4.12 Provjera Layera podloga

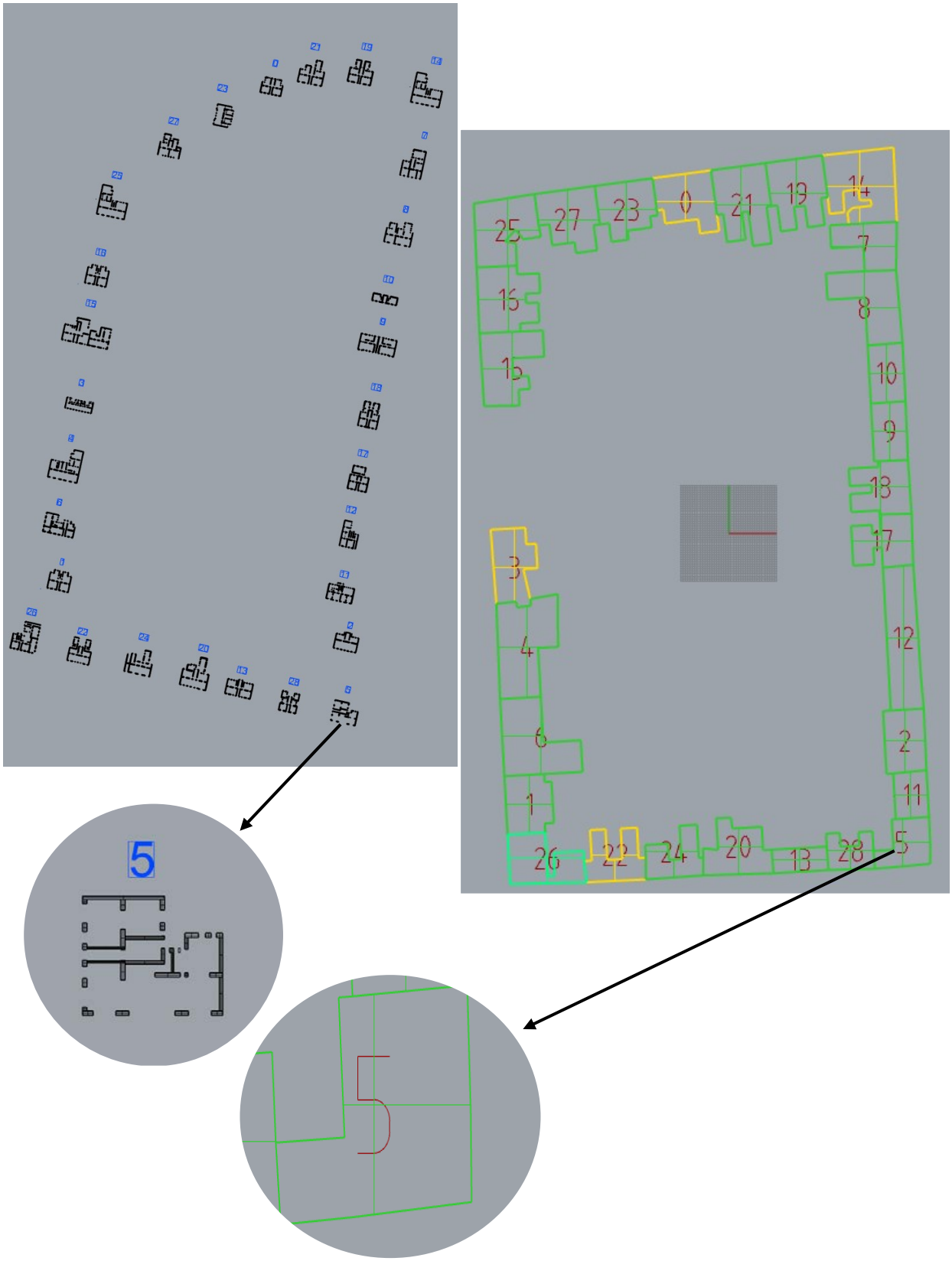
GHpython skripta pronalazi layer s imenom „Podloga“ kojeg smo prije definirali, ispisuje u panel. DPipeline je Grasshopper komponenta koja na temelju imena Layera i vrste geometrije označuje sve elemente koji zadovoljavaju navedene uvjete (Slika 4.12).



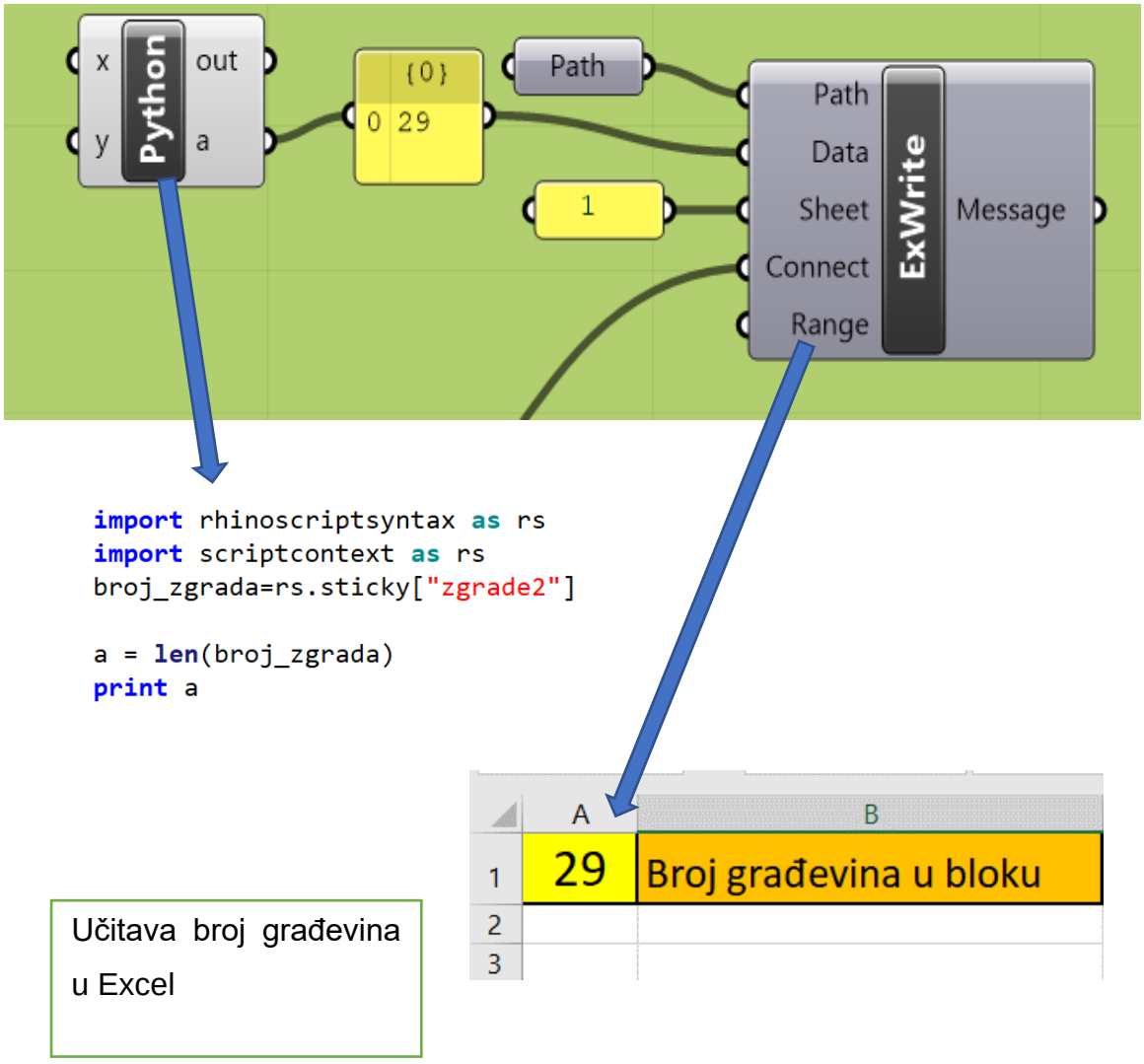

```python
import rhinoscriptsyntax as rs
import scriptcontext as sc
import Rhino as rh
import ghpythonlib.treehelpers as th

def layeri_s_tlocrtom():
    sc.doc =rh.RhinoDoc.ActiveDoc
    laytab= sc.doc.Layers
    Layer =[]
    LayerTab = [layer.Name for layer in rh.RhinoDoc.ActiveDoc.Layers if layer.IsDeleted == False]
    for layerName in LayerTab:
        if "Tlocrt" in layerName:
            Layer.append(layerName)
    a= Layer
    return Layer
print layeri_s_tlocrtom()

layeri_tablica= layeri_s_tlocrtom()
a= layeri_tablica
```

Slika 4.13 Unos tlocrta u skriptu

Sličan je princip i za učitavanje površina, učitavaju se svi layeri s „Tlocrti“ (Slika 4.13). U imenu razlika je samo u traženoj geometriji, ovdje se traži „surface“. Svaki Layer sadrži geometriju nosivih zidova za pojedinu zgradu. Zato je važno pravilno ih poredat kako bi skripta znala za koju građevinu se tlocrt odnosi (Slika 4.14).

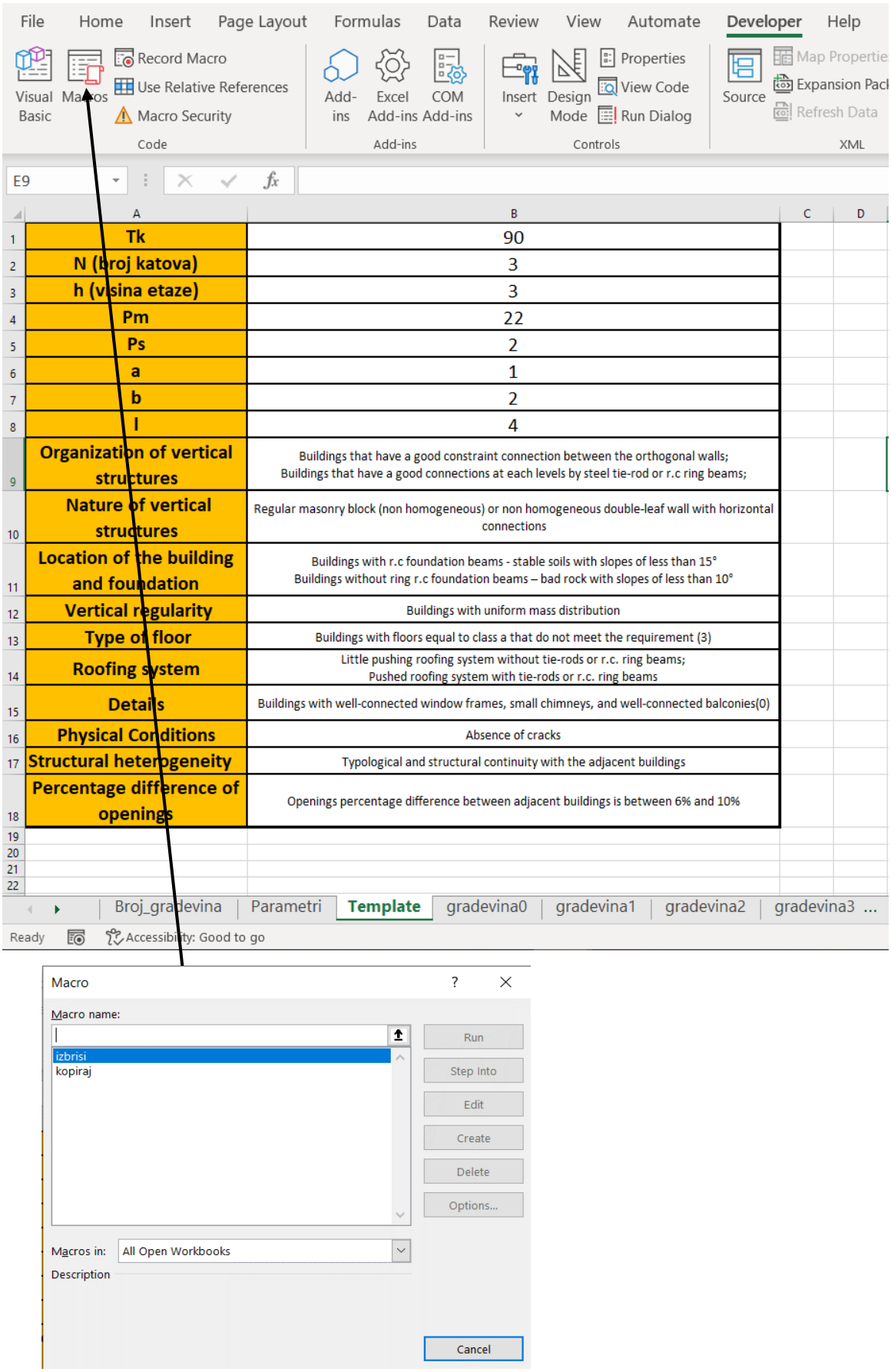


Slika 4.14 Tlocrti pojedinih građevina i bloka



Slika 4.15 Očitavanje broja građevina

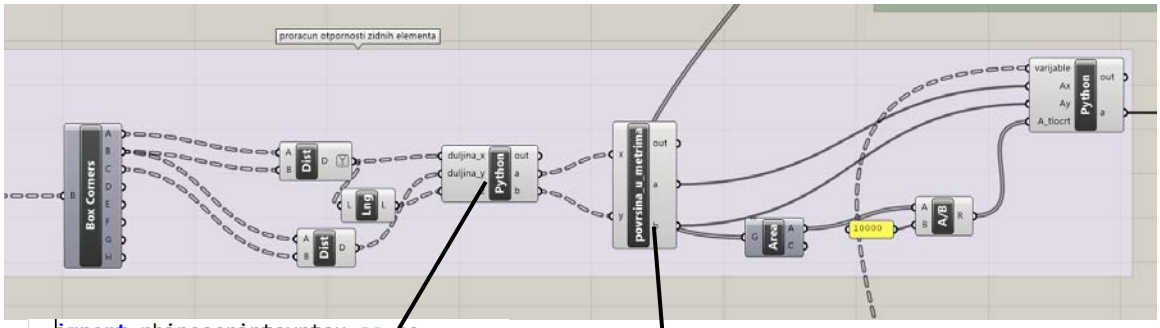
Excel se sastoji od tri fiksna lista i ostalih koji variraju ovisno o broju građevina, a generira ih Excel pomoću VB skripte (Slike 4.16 i 4.20). Prva lista sadrži broj građevina, druga parametre koji služe za kreiranje padajućih izbornika u listu „Template“ (Slika 4.15).



	A	B	C	D
1	Tk	90		
2	N (broj katova)	3		
3	h (visina etaze)	3		
4	Pm	22		
5	Ps	2		
6	a	1		
7	b	2		
8	l	4		
9	Organization of vertical structures	Buildings that have a good constraint connection between the orthogonal walls; Buildings that have a good connections at each levels by steel tie-rod or r.c ring beams;		
10	Nature of vertical structures	Regular masonry block (non homogeneous) or non homogeneous double-leaf wall with horizontal connections		
11	Location of the building and foundation	Buildings with r.c foundation beams - stable soils with slopes of less than 15° Buildings without ring r.c foundation beams – bad rock with slopes of less than 10°		
12	Vertical regularity	Buildings with uniform mass distribution		
13	Type of floor	Buildings with floors equal to class a that do not meet the requirement (3)		
14	Roofing system	Little pushing roofing system without tie-rods or r.c. ring beams; Pushed roofing system with tie-rods or r.c. ring beams		
15	Details	Buildings with well-connected window frames, small chimneys, and well-connected balconies(0)		
16	Physical Conditions	Absence of cracks		
17	Structural heterogeneity	Typological and structural continuity with the adjacent buildings		
18	Percentage difference of openings	Openings percentage difference between adjacent buildings is between 6% and 10%		
19				
20				
21				
22				

Slika 4.16 Macro izbornik i tablica u Excelu

Broj katova N i visina građevine h su pretpostavljeni, na temelju slika iz „Google Earth-a“. Na žalost ne postoji javno dostupna GIS karta iz kojih bi se mogle automatski učitati građevine sa visinama. Vrijednosti „a“ i „b“ predstavljaju duljine istaka, prikazano na slici 3.8. Za proračun parametra četiri (Slika 4.17) potrebne su nam gustoća materijala (Pm), trajno opterećenje (Ps) i tangencijalna čvrstoća (Tk) (Slika 4.18).




```python
import rhinoscriptsyntax as rs
import ghpythonlib.treehelpers as th
#x=th.tree_to_list(duljina_x)
#y=th.tree_to_list(duljina_y)
x = duljina_x
y = duljina_y
Duljina = z
povX= 0
povY=0
output = []
sum_pov_y=0
sum_pov_x=0
for i in range(Duljina):
    if x < y:
        povY=x*y
        sum_pov_y=+povY
    elif x > y:
        povX=x*y
        sum_pov_x=+povX
print sum_pov_x
print sum_pov_y
a = sum_pov_x
b = sum_pov_y
```

Prvo se podijelila površina/zid na rubne točke, a potom se provjerava udaljenost između njih kako bi znali odrediti da li je zid u x ili y smjeru, te pripadajuća površina

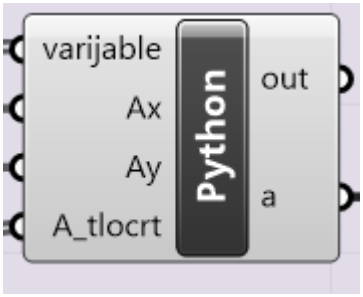
```python
import rhinoscriptsyntax as rs
import ghpythonlib.treehelpers as th
xd=th.tree_to_list(x)
yd=th.tree_to_list(y)
x_sumP=0
y_sumP=0
x_sum={}
y_sum={}
povX={}
povY={}
#print yd[0][0]
for i in range(len(xd[0])):
    x_sumP=0
    for j in range(len(xd[0][i])):
        x_sumP=x_sumP+xd[0][i][j]
        x_sum[i]=x_sumP
#print x_sum

for i in range(len(yd[0])):
    y_sumP=0
    for j in range(len(yd[0][i])):
        y_sumP=y_sumP+yd[0][i][j]
        y_sum[i]=y_sumP
#print y_sum[0]
for i in range(len(x_sum)):
    povX[i]=(x_sum[i]/10000)
#print povX[0]
for i in range(len(y_sum)):
    povY[i]=(y_sum[i]/10000)
#print povY[0]
print povX
```

Zbraja površine zidova u x i y smjeru

```python
class Data:
    pass

a = Data()
a.dictionary=povX
b=Data()
b.dictionary=povY
```

Slika 4.17 Proračun zidova u x i y smjeru

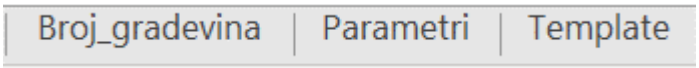
```
import rhinoscriptsyntax as rs
import math
import ghpythonlib.treehelpers as th
varijable=th.tree_to_list(varijable)
#print varijable
Ax=dict(Ax.dictionary)
#print Ax
Ay=dict(Ay.dictionary)
#print Ay
A=B=Tk=N=h=Pm=Ps=a0=y=q=C=a=alpha=rezultat={}

for i in range(len(Ax)):
    A[i]=min(Ax[i],Ay[i])
    B[i]=max(Ax[i],Ay[i])

for i in range(len(varijable[0])):
    Tk[i]=varijable[0][i][0]
    N[i]=varijable[0][i][1]
    h[i]=varijable[0][i][2]
    Pm[i]=varijable[0][i][3]
    Ps[i]=varijable[0][i][4]
    a0[i]=A[i]/A_tlocrt[i]
    y[i]=B[i]/A[i]
    q[i]=((((A[i]+B[i])*h[i])/A_tlocrt[i]))*Pm[i]+Ps[i]
    C[i]=((a0[i]*Tk[i])/(q[i]*N[i]))*math.sqrt(1+(q[i]*N[i])/(1.5*a0[i]*Tk[i]*(1+y[i])))
    a[i]=C[i]
    alpha[i]=C[i]/0.4
    if alpha[i]>=1:
        rezultat[i]=0
    elif 0.6<=alpha[i]<=1:
        rezultat[i]=5
    elif 0.4<=alpha[i]<=0.6:
        rezultat[i] =25
    elif alpha[i]<=0.4:
        rezultat[i]=45
print rezultat

class Data:
    pass
a=Data()
a.dictionary=rezultat
b=alpha
```



Proračun α po postupku sa slike 3.7

Slika 4.18 Proračun α

Excel se sastoji od tri osnova radna lista „Broj_gradevina“, „Parametri“ i „Template“ (Slika 4.19), u Broj_gradevina se učitava broj građevina u bloku, u „Parametri“ se nalaze informacije potrebne za padajuće izbornike iz „Template“. Tablica u „Template“ se nalazi tablica koja se onda pomoću makro tipke kopira x puta.



Slika 4.19 Osnovni listovi

```
Sub izbrisi()
Dim gradevina As String, i As Integer
 gradevina = ThisWorkbook.Sheets("broj_gradevina").Range("A1") - 1
Application.DisplayAlerts = False
For i = 0 To gradevina
    Sheets("gradevina" & i).Delete
Next
Application.DisplayAlerts = True

End Sub
```

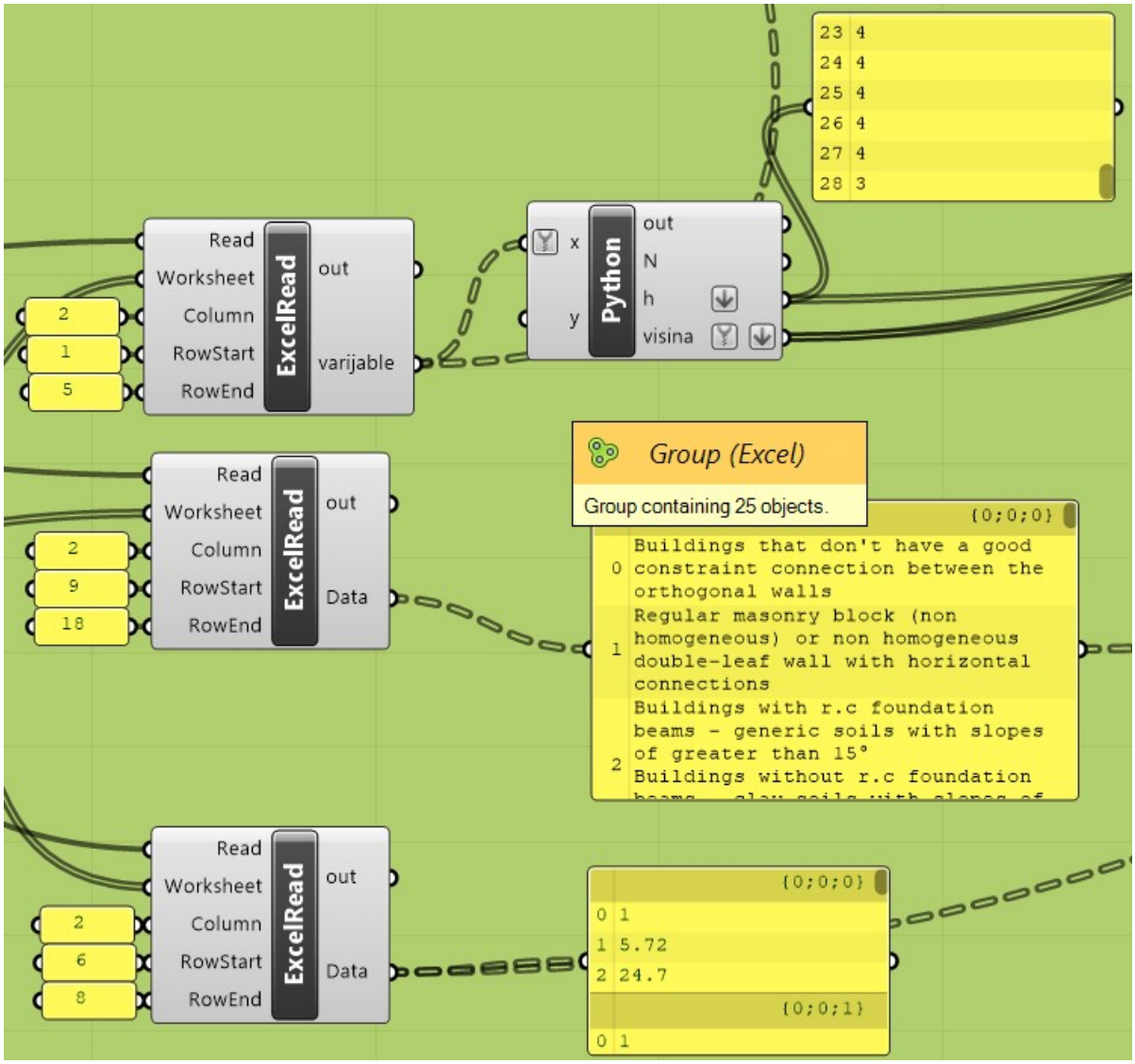
Funkcija „izbriši“ briše sve listove osim tri osnovna

```
Sub kopiraj()

Dim i As Long, Template As String, active_ws As Worksheet

    On Error Resume Next
gradevina = ThisWorkbook.Sheets("broj_gradevina").Range("A1") - 1
    Application.ScreenUpdating = False

    Set active_ws = ActiveSheet

    For i = 0 To gradevina

        Template = ActiveSheet.Name

        active_ws.Copy After:=ActiveWorkbook.Sheets(Template)

        ActiveSheet.Name = "gradevina" & i

    Next

    active_ws.Activate

    Application.ScreenUpdating = True

End Sub
```

Funkcija „kopiraj“ kopira list N broj puta i daje ime listu „gradevinaN“

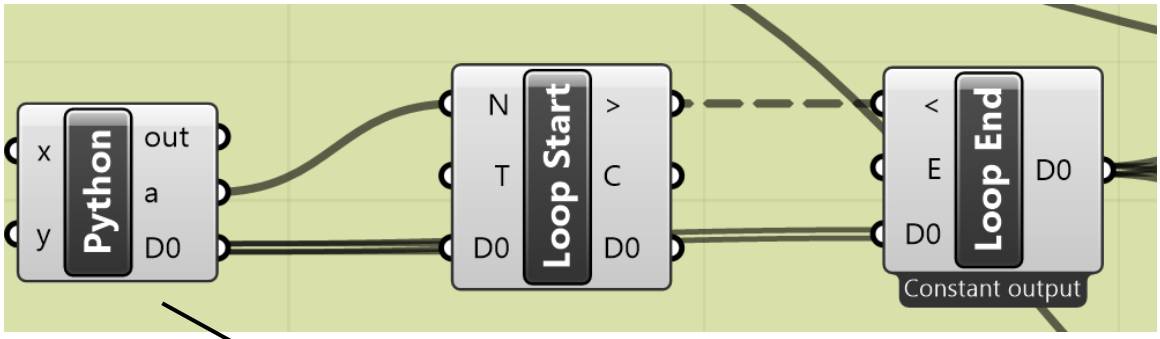
Slika 4.20 VB skripta „macro“ tipki

Excelu se pristupa s tri različita modula (Slika 4.21), jedan modul je za parametre iz padajućeg izbornika u Excelu, drugi učitava parametre potrebne za ocjenjivanje pravilnosti tlocrta i treći kojeg koristimo za proračun parametara, ali i za ekstradiranje površine i kreiranje „breps-a“ koje nam služe za grafički prikaz bloka, ali i generiranje informacija za daljnju analizu.



Slika 4.21 Moduli za očitavanje podataka iz Excela

Cijeli postupak omogućuje petlja koja prolazi kroz svaki list i omogućuje modulima iz dodatka „GhExcel“ da očitaju i unesu podatke u Grasshopper. Petlja je iz dodatka „Extra“. Ovaj dio je i najzahtjevniji računalu za obradu te traje dvije to tri minute za blok od 30 građevina (Slika 4.22).

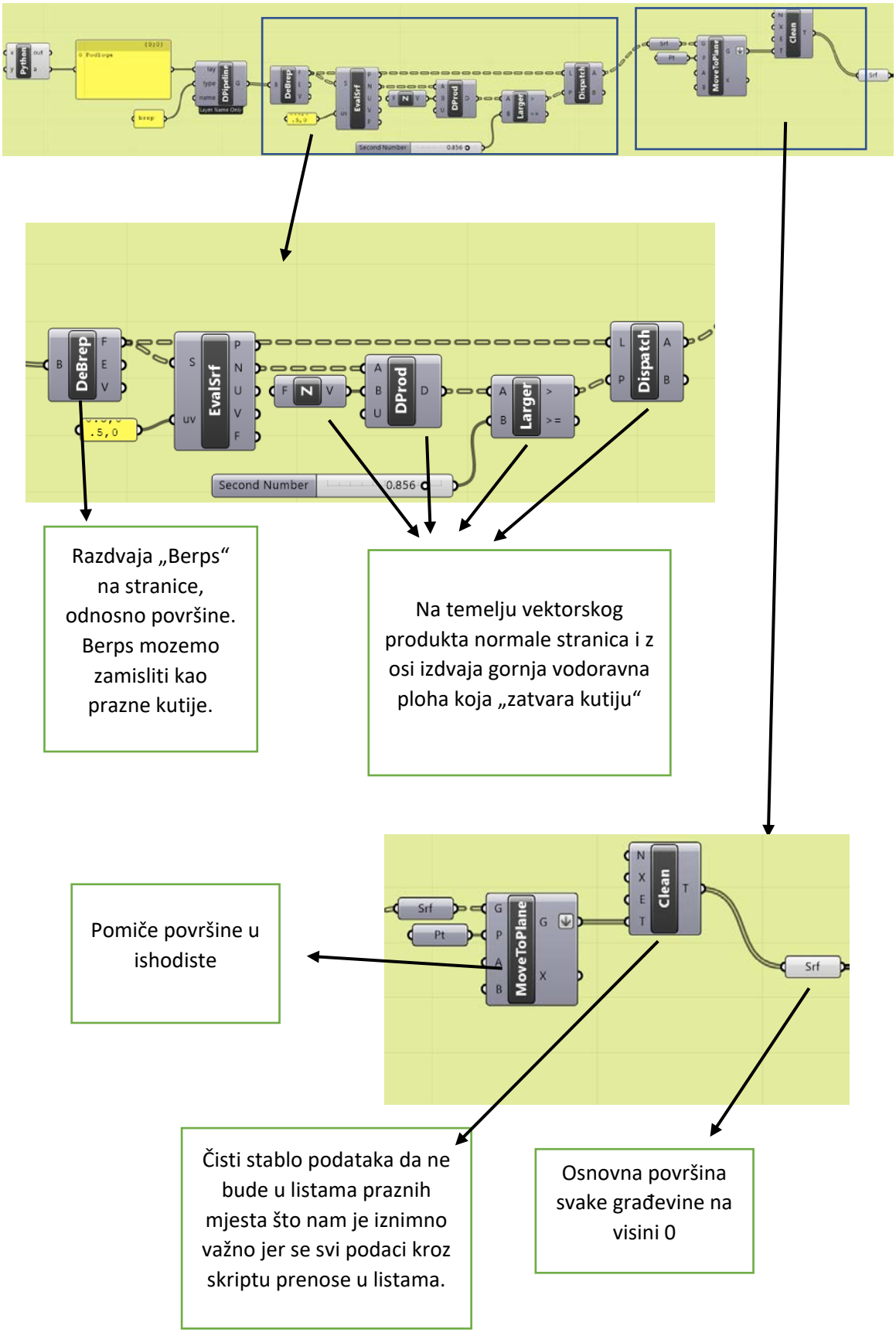



```
import scriptcontext as rs
brojac=rs.sticky["zgrade2"]
a=len(brojac)
b=len(brojac)+2
i=2
lista = []
for i in range(4,len(brojac)+4):
    lista.append(i)
print lista
D0=lista
```

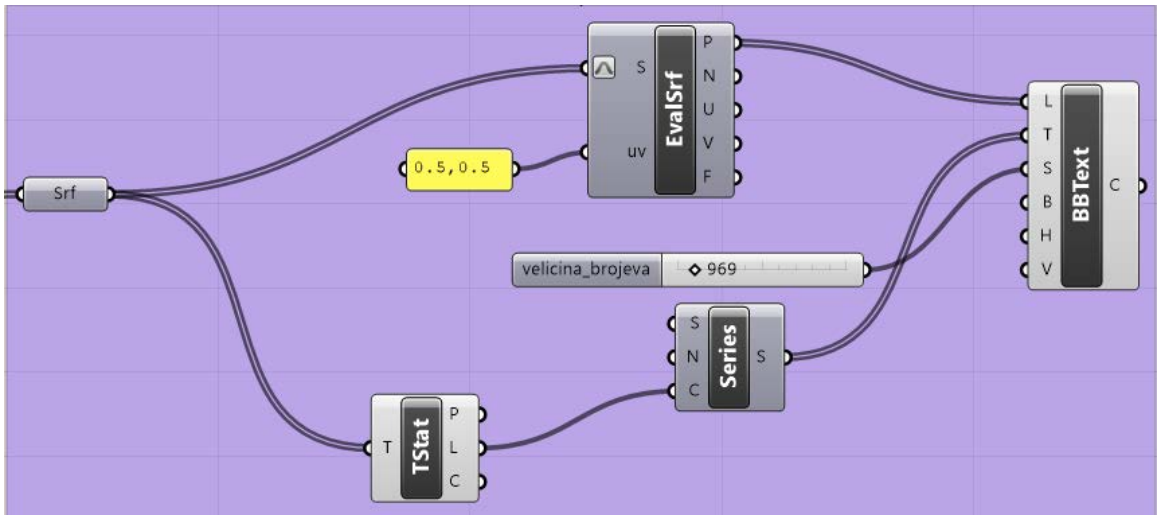
Python modul zadaje broj iteracija kroz koju petlja prolazi, a jednak je broju građevina. Petlja kreće od radnog lista 4, odnosno od lista s podacima o prvoj građevini.

Slika 4.22 Petlja za ispis rezultata iz Excela

Nakon što smo objasnili ulazne informacije, možemo početi s modulima koji generiraju informacije na osnovu ulaznih parametara (Slike 4.23-4.29).

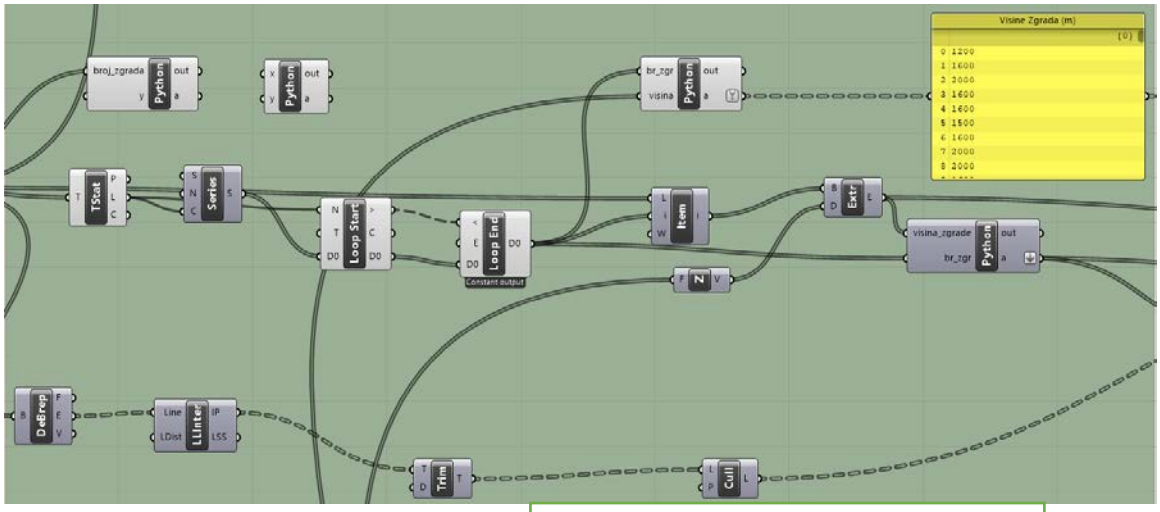


Slika 4.23 Izdvajanje površine iz učitanog modela



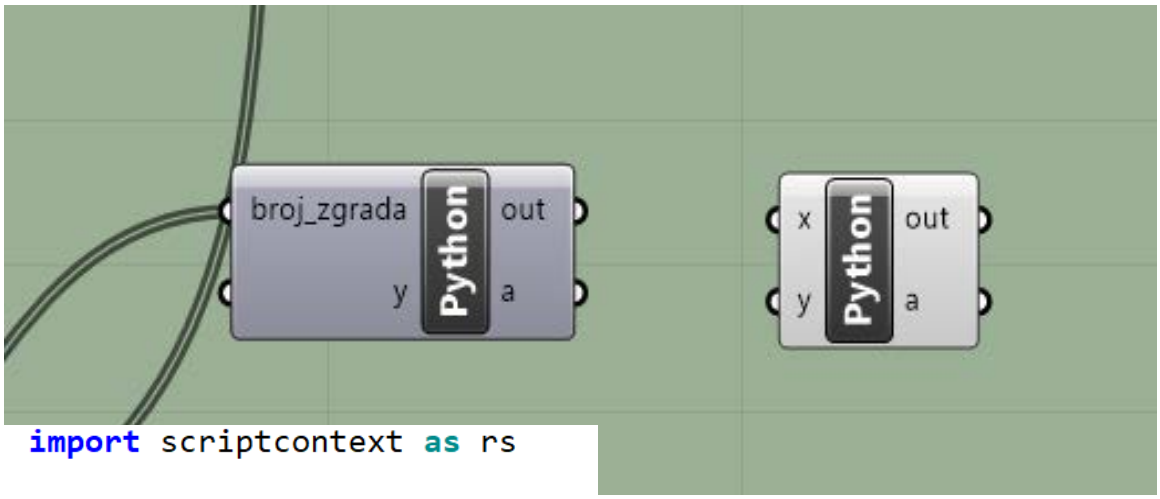
Ovaj dio skrpite zapisuje brojeve građevina na početnu površinu. Numeriranje je nasumično, međutim svaka površina ide u listu pod brojem koji piše i tada to postaje ključ za tu površinu odnosno građevinu.

Slika 4.24 Generiranje broja zgrade



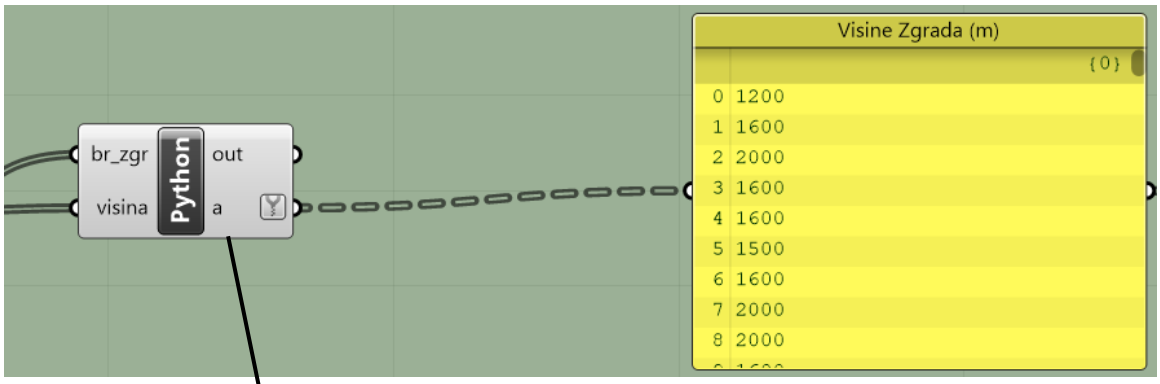
Dio skripte koji generira informacije koje se kasnije koriste za proračun parametra

Slika 4.25 Dio skripte koji generira informacije




```
import scriptcontext as rs

duljina=rs.sticky["zgrade2"]
lista=[]
for i in range(len(duljina)):
    lista.append(" ")

rs.sticky["zgrade"]=lista
```

Ova dva modula se koriste za izradu globalnih varijabli koje se uvijek mogu pozivat



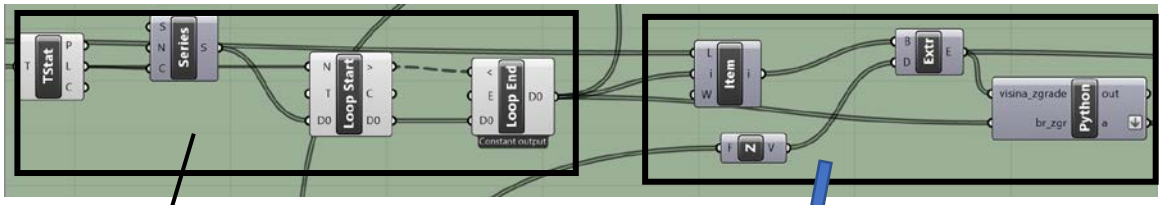

```
import math
import scriptcontext as rs

zgrade_lista=rs.sticky["zgrade"]
print zgrade_lista
zgrade_lista[br_zgr]=visina
a=zgrade_lista[0]
```

Generira listu visina zgrada. Pozicija u listi je ujedno i „ID“ ili ključ građevine

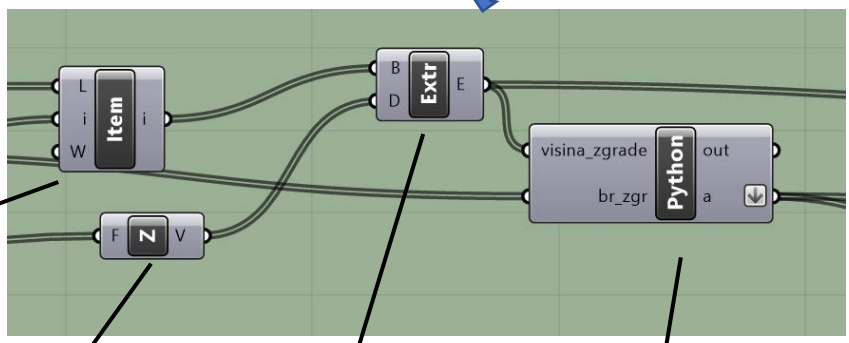
```
import rhinoscriptsyntax as rs
N=x[1]
h=x[2]
visina=N*h*100
print N
print h
```

Visina iz podataka iz Excela

Slika 4.26 Lista visina zgrada



Generiranje indeksa liste i broja ponavljana za petlju i petlja



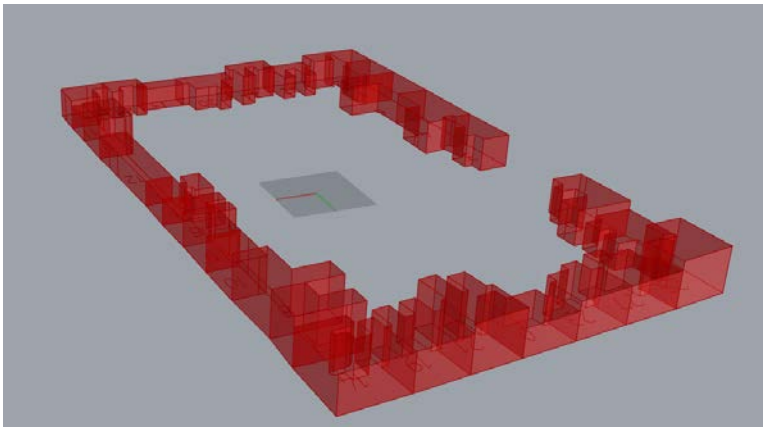
Pristupa listi pojedinačno po indeksu

Pretvara visinu vektor paralelan s osi Z

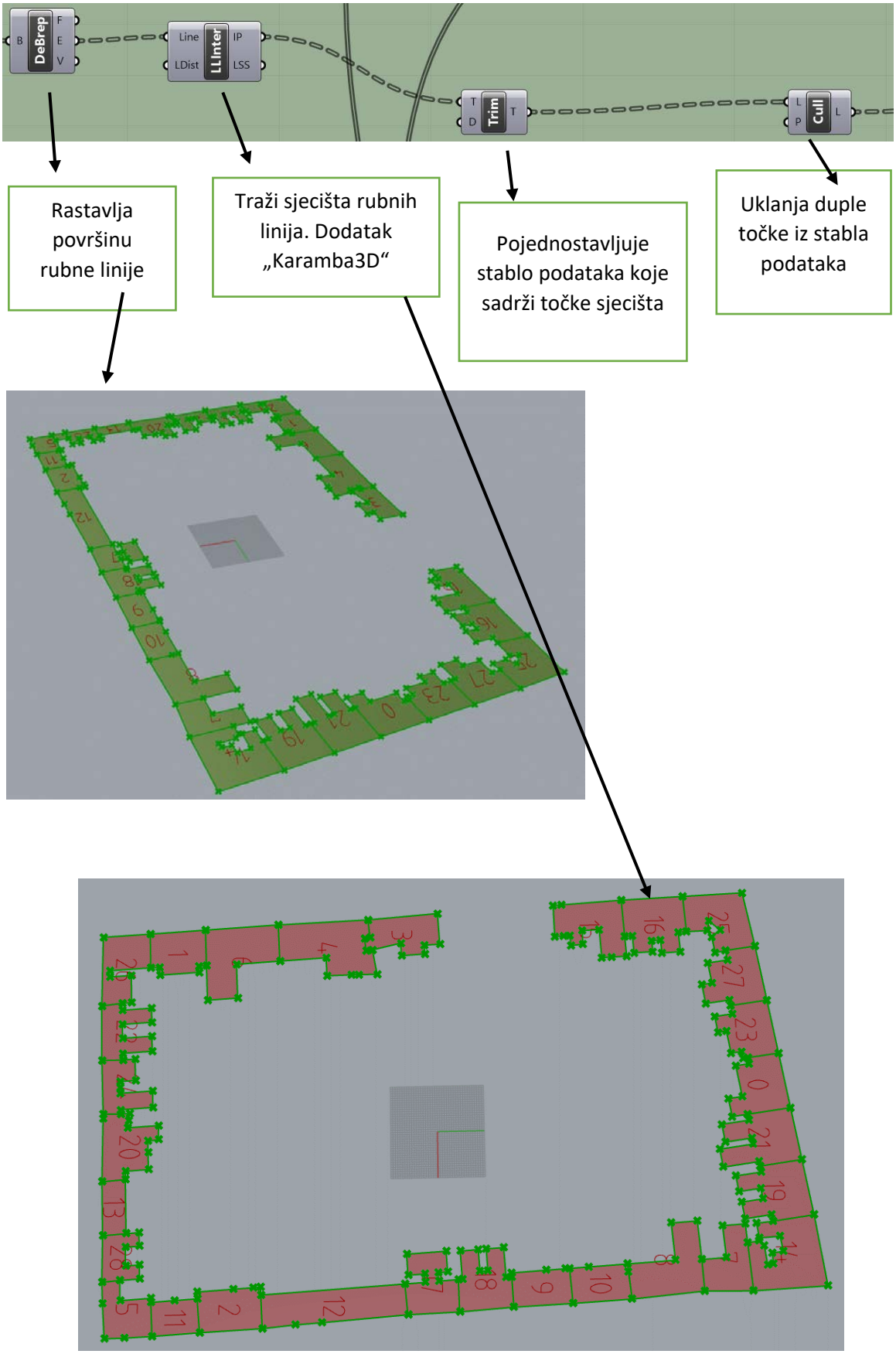
Ekstrudira površinu u smjeru za vrijednost vektora

```
import math
import scriptcontext as rs
import Rhino.Geometry as rg
visina_pojedine_zgr=rs.sticky["zgrade2"]
visina_pojedine_zgr[br_zgr]=visina_zgrade

a = visina_pojedine_zgr
```

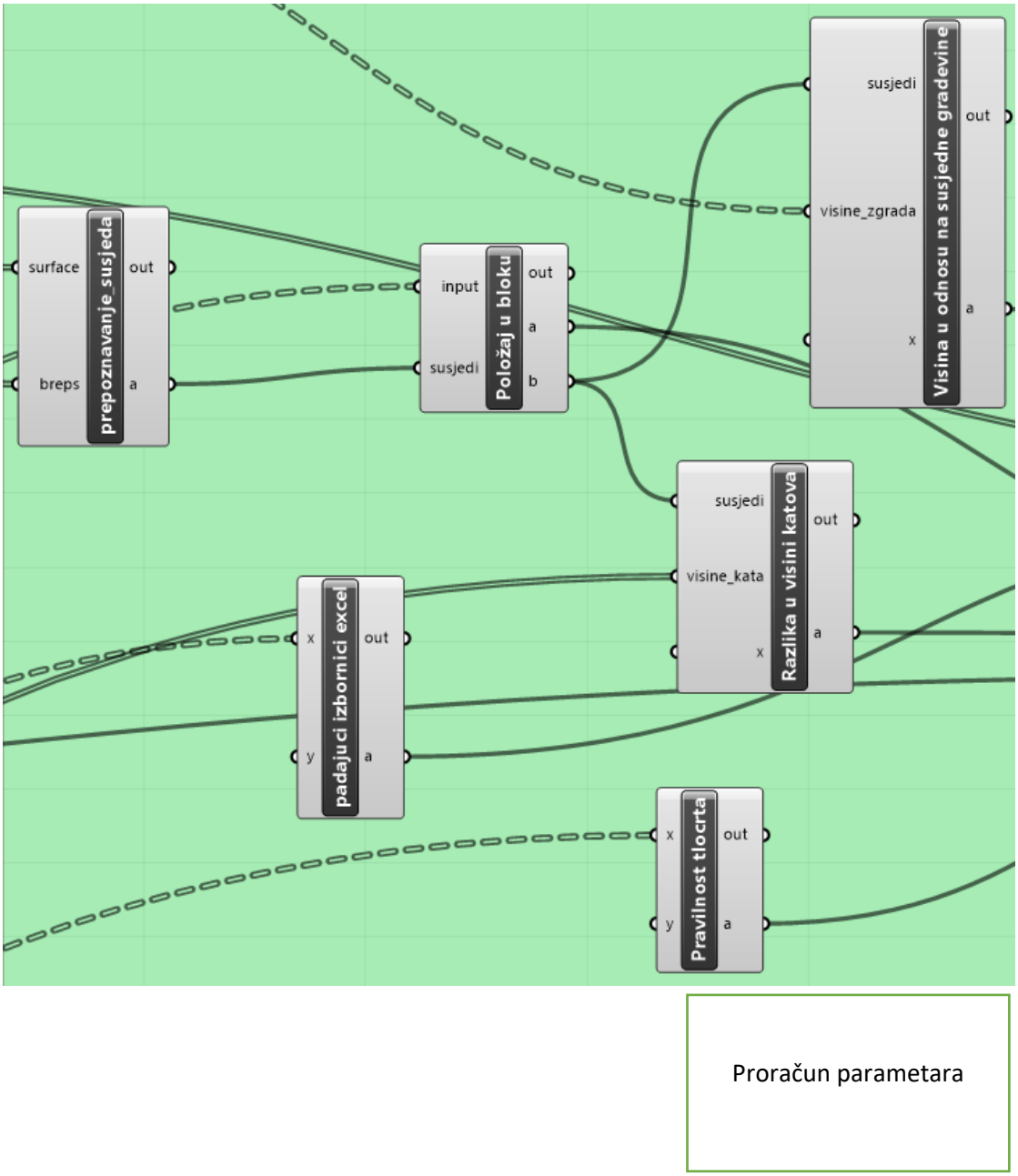
Radi listu koja pridružuje zgradu sa ekstrudiranom površinom koja se pretvara u „Brep“



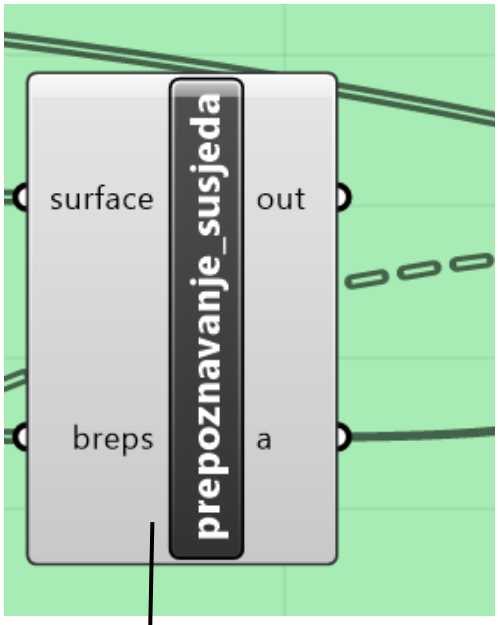
Slika 4.27 Ekstradiranje površine



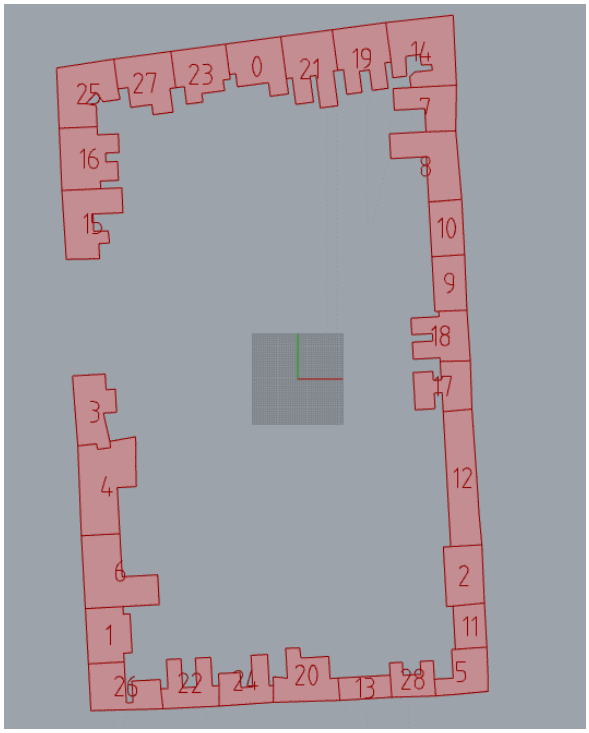
Slika 4.28 Generiranje točaka



Slika 4.29 Moduli za proračun parametara



Python modul za prepoznavanje susjeda




```
import Rhino
import scriptcontext as rs
duljina=rs.sticky["zgrade2"]
lista_pov=[]
lista_brep=[]
def generate_collission(breps,surface): #radi listu koje se sve povrsine doticu
    for i in range(len(duljina)):
        for j in range(len(duljina)):
            success, crvs, pts = Rhino.Geometry.Intersect.Intersection.BrepBrep(breps[i], surface[j], Rhino.RhinoDoc.ActiveDoc.ModelAbsoluteTolerance)
            if success and (crvs or pts):
                lista_pov.append(i)
                lista_brep.append(j)
                lista_sudaranja=zip(lista_pov,lista_brep)
    return lista_sudaranja
#a=generate_collission(breps,surface)

gc=generate_collission(breps,surface)
```

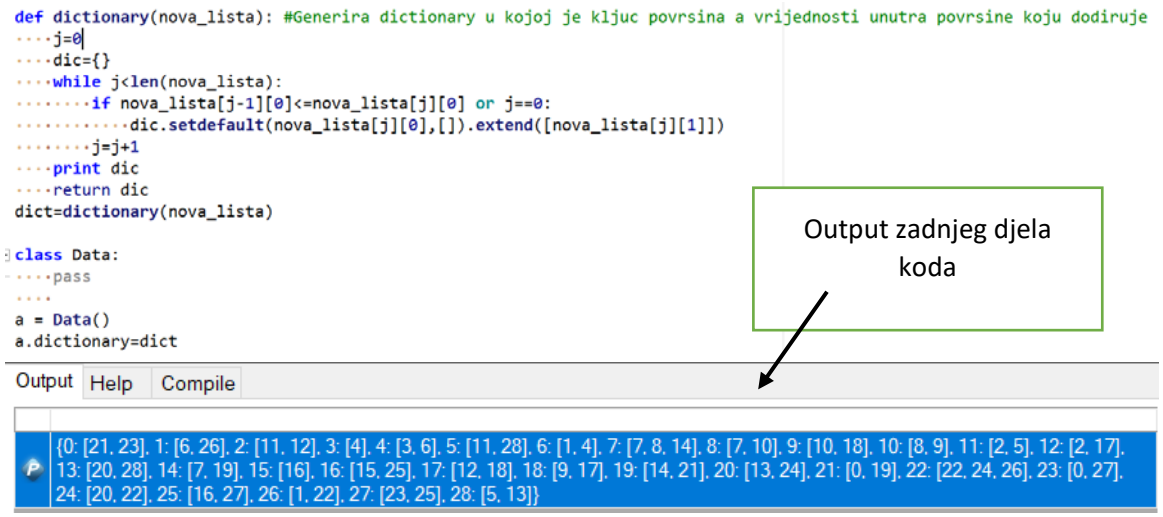
Output prvog djela koda

```
Output  Help  Compile
[(0, 0), (0, 21), (0, 23), (1, 1), (1, 6), (1, 26), (2, 2), (2, 11), (2, 12), (3, 3), (3, 4), (4, 3), (4, 4), (4, 6), (5, 5), (5, 11), (5, 28), (6, 1), (6, 4), (6, 6), (7, 7), (7, 8), (7, 14), (8, 7), (8, 8), (8, 10), (9, 9), (9, 10), (9, 18), (10, 8), (10, 9), (10, 10), (11, 2), (11, 5), (11, 11), (12, 2), (12, 12), (12, 17), (13, 13), (13, 20), (13, 28), (14, 7), (14, 14), (14, 19), (15, 15), (15, 16), (16, 15), (16, 16), (16, 25), (17, 12), (17, 17), (17, 18), (18, 9), (18, 17), (18, 18), (19, 14), (19, 19), (19, 21), (20, 13), (20, 20), (20, 24), (21, 0), (21, 19), (21, 21), (22, 22), (22, 24), (22, 26), (23, 0), (23, 23), (23, 27), (24, 20), (24, 22), (24, 24), (25, 16), (25, 25), (25, 27), (26, 1), (26, 22), (26, 26), (27, 23), (27, 25), (27, 27), (28, 5), (28, 13), (28, 28)]
```

```
def collission(gc): #update-a listu tako da izbaci iste povrsine
    nova_lista=gc[:] #lista bez preklapanja
    i=0
    while i<=len(nova_lista):
        k=0
        prva_povrsina=nova_lista[i][k] #prvi broj u listi
        druga_povrsina=nova_lista[i][k+1] #drugi broj u listi
        if prva_povrsina == druga_povrsina and k!=k+1:
            del nova_lista[i]
        i=i+1
    return nova_lista
#a=collission(gc)

nova_lista=collission(gc)
```

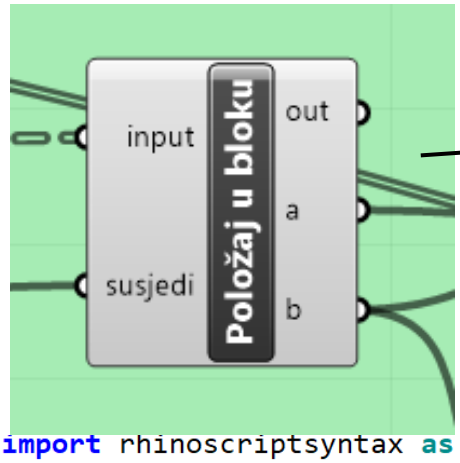
Output drugog djela koda

```
Output  Help  Compile
[(0, 21), (0, 23), (1, 6), (1, 26), (2, 11), (2, 12), (3, 4), (4, 3), (4, 6), (5, 11), (5, 28), (6, 1), (6, 4), (7, 7), (7, 8), (7, 14), (8, 7), (8, 10), (9, 10), (9, 18), (10, 8), (10, 9), (11, 2), (11, 5), (12, 2), (12, 17), (13, 20), (13, 28), (14, 7), (14, 19), (15, 16), (16, 15), (16, 25), (17, 12), (17, 18), (18, 9), (18, 17), (19, 14), (19, 21), (20, 13), (20, 24), (21, 0), (21, 19), (22, 22), (22, 24), (22, 26), (23, 0), (23, 27), (24, 20), (24, 22), (25, 16), (25, 27), (26, 1), (26, 22), (27, 23), (27, 25), (28, 5), (28, 13)]
```

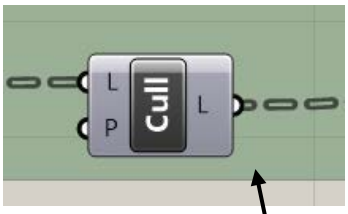


Slika 4.30 Prepoznavanje susjeda

Prvi dio koda traži sjecišta između „Brepova“, tako da za svaki „Brep“ provjerava ako ga i jedan drugi sječe, ukoliko ga sječe zapisuje njihov ID (ID je broj gradevine), kako provjerava sjecište i sa samim sobom u rezultatu ispisuje i par istih ID-eva. U drugom djeu izbacujemo par ID-eva koji se sijeku sa samim sobom i konačno zadnji dio koda kreira „dictionary“ u kojoj je ključ ID građevine, a IDevi površina koje se dodiruju su smješteni u listu unutar „dictionary-a“. Rezultat se pretvara u objekt i onda kao takav prenosi dalje unutar Grasshoppera jer Grasshopper nema sposobnost čitanja „dictionary-a“ (Slika 4.30).



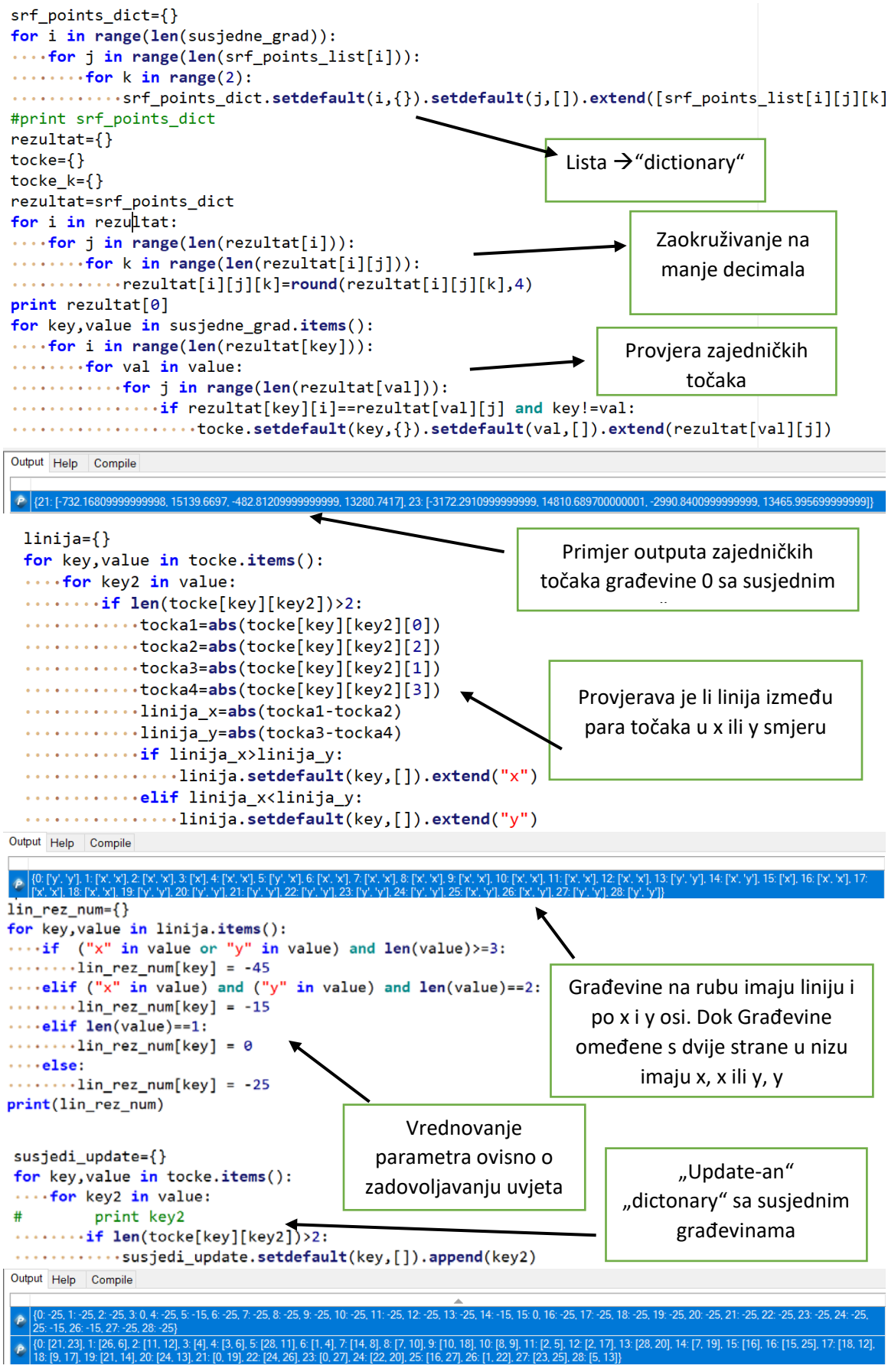
Python modul za određivanje položaja zgrade u bloku




```
import rhinoscriptsyntax as rs
import ghpythonlib.treehelpers as th
import Rhino.Geometry as rg
import Rhino

susjedne_grad=dict(susjedi.dictionary)
print susjedi.dictionary

srf_points_list=th.tree_to_list(input)
```

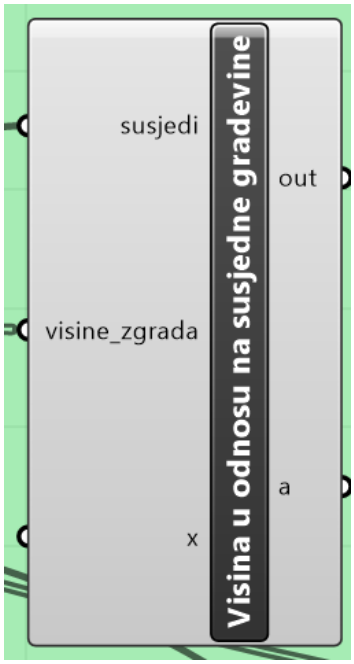
Lista svih točaka na rubovima površine. Generirane korak prije.

Slika 4.31 Modul za određivanje položaja



Slika 4.32 Kod modula položaja

Uzima se lista točka generirana na površini komponentom prije. Lista se zatim pretvara u „dictionary“. Zbog mogućnosti greške prilikom „importiranja“ iz sketchupa, zajednička točka može varirati na nekoj od zadnjih decimala. Provjeravaju se točke promatrane građevine s točkama susjedne građevine i ukoliko postoji, smješta se u „nested dictionary“ gdje je prvi ključ ID građevine, a drugi ključ je ID susjedne građevine unutar koje se nalazi par točaka koji je zajednički. Par točaka se provjerava pružaju li se u više u x ili y smjeru. Zatim se položaj boduje po uvjetima iz metode. Na kraju se još „update-a“ lista susjeda jer po prvotnoj se može dogodit da kao susjede prikazuje i građevine s kojom ima samo jednu zajedničku točku (Slike 4.31 i 4.32).



Python modul za vrednovanje razlika u visini građevina

```
import rhinoscriptsyntax as rs
import copy
import ghpythonlib.treehelpers as th

susjedne_grad=dict(susjedi.dictionary)
#print susjedi.dictionary
new_dict = {}
visine_zgrada=th.tree_to_list(visine_zgrada)

def convert(visine_zgrada): #pretvara listu visina u dictionary
    d_visina_zgrada={i:visine_zgrada[i] for i in range(0,len(visine_zgrada))}
    return d_visina_zgrada
#a= convert(visine_zgrada)

print convert(visine_zgrada)
visina=convert(visine_zgrada)
```

Lista visina u „dictionary“

```
Output  Help  Compile
{0: 1200, 1: 1600, 2: 2000, 3: 1600, 4: 1600, 5: 1500, 6: 1600, 7: 2000, 8: 2000, 9: 1600, 10: 1600, 11: 1200, 12: 1200, 13: 1200, 14: 1600, 15: 1600, 16: 1200, 17: 1200, 18: 1200, 19: 1600, 20: 1200, 21: 2000, 22: 1200, 23: 1600, 24: 1600, 25: 2000, 26: 1600, 27: 1600, 28: 1500}
```

Slika 4.33 Prvi dio modula za visinu građevina

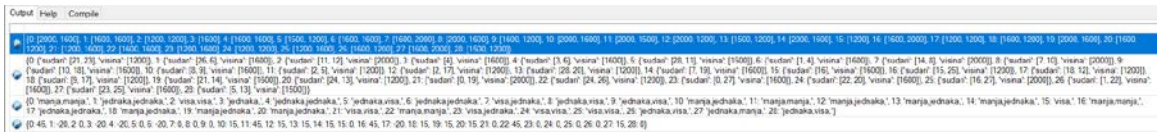
```
new_dict={}
for sudari in susjedne_grad:
    for i in  range(len(susjedne_grad[sudari])):
        for vis in visina:
            if susjedne_grad[sudari][i]== vis:
                new_dict.setdefault(sudari,[]).extend([visina.get(vis)])
print new_dict

dic_visina=visina
#print dic_visina
zgrade_sudari_i_visine={}
for i in range(len(visine_zgrada)):
    zgrade_sudari_i_visine.update({i:{"visina":[dic_visina[i]],"sudari":susjedne_grad[i]}})
print zgrade_sudari_i_visine
```

Kreira „dictionary“ koji sadrži visinu susjednih građevina

```
for i in range(len(new_dict)):
    for j in range(len(new_dict[i])):
        if visina[i]>new_dict[i][j]:
            a=a+1
            dict_rez.setdefault(i,[]).extend("visa,")
        elif visina[i]<new_dict[i][j]:
            a=a-1
            dict_rez.setdefault(i,[]).extend("manja,")
        else:
            a=a+0
            dict_rez.setdefault(i,[]).extend("jednaka,")
#print dict_rez

for i in range(len(dict_rez)):
    delimiter = ''
    dict_rez[i]=delimiter.join(dict_rez[i])
print dict_rez
dict_rez_num={}
A=-20
B=0
C=15
D=45

for key,value in dict_rez.items():
    if ("visa" in value and "jednaka" in value) or "visa" in value:
        dict_rez_num[key] = B
    elif ("manja" in value and "jednaka" in value) or ("visa" in value and "manja" in value):
        dict_rez_num[key] = C
    elif "manja" in value:
        dict_rez_num[key] = D
    else:
        dict_rez_num[key] = A
print(dict_rez_num)
```

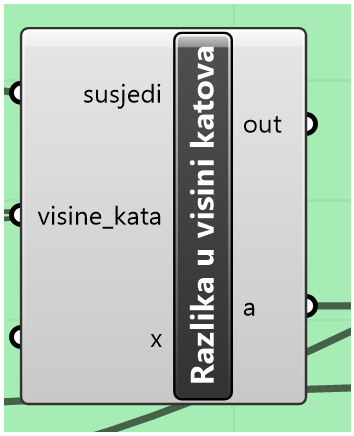
Uvjet koji provjerava ako je građevina veća, manja ili jednaka od susjedne

Bodovanje ovisno o uvjetu iznad



Slika 4.34 Ostatak koda modula za visine građevina

Provjera visine u odnosu na susjedne se vrši tako da se učita „dictionary“ susjeda. Potom se radi novi „dictionary“ koji sadrži listu s vrijednostima visina susjeda, zatim se uspoređuju i kreira se novi „dictionary“ s rezultatima usporedbe ne temelju koje se onda i boduje (Slike 4.33 i 4.34).

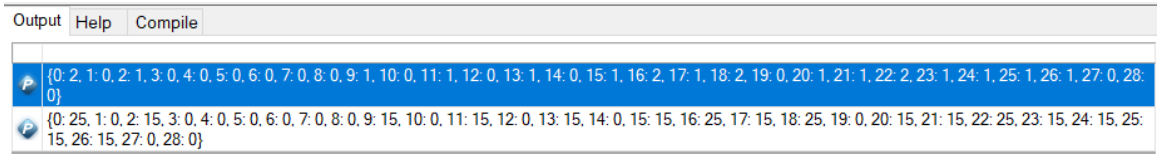


Python modul za provjeru visine etaža susjednih građevina

```
for i in range(len(new_dict)):
    brojac=0
    for j in range(len(new_dict[i])):
        if visina[i]>new_dict[i][j] or visina[i]<new_dict[i][j]:
            brojac=brojac+1
            dict_rez[i]= brojac
        else:
            brojac=brojac+0
            dict_rez[i]=brojac
print dict_rez

dict_rez_num={}
for i in dict_rez:
    if dict_rez[i]==0:
        dict_rez_num[i]=0
    elif dict_rez[i]==1:
        dict_rez_num[i]=15
    elif dict_rez[i]==2:
        dict_rez_num[i]=25
    else:
        dict_rez_num[i]=45
print dict_rez_num
```

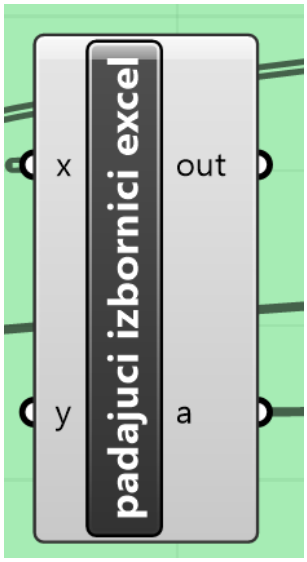
Broji koliko susjeda ima različitu visinu etaža

Brojčana ocjena u odnosu na uvjet iznad



Slika 4.35 Modul za razlike u etažama

Postupak je isti kao kod provjere visina susjednih građevina, razlika je jedino u uvjetu i bodovanju (Slika 4.35).



Python modul za bodovanje vrijednosti iz padajućeg izbornika u Excelu

```
import rhinoscriptsyntax as rs
import ghpythonlib.treehelpers as th

vrijednosti=th.tree_to_list(x)
#print vrijednosti[0][0]
k=-1
vr_dict_char={}
vr_dict={}
vr=[]
#PARAMETAR_1
par1_1="Buildings built in accordance with the seismic design code"
par1_2="Buildings that have a good constraint connection between the orthogonal walls\nBuildings that have a good connections at each levels by steel tie-rod or r.c ring beams"
par1_3="Buildings that have a good constraint connection between the orthogonal walls but they don't have\nsteel tie-rod or r.c ring beams at each level"
par1_4="Buildings that don't have a good constraint connection between the orthogonal walls"
#PARAMETAR_2
par2_1="Regular masonry block (homogeneous) or double-leaf wall with horizontal connections"
par2_2="Regular masonry block (non homogeneous) or non homogeneous double-leaf wall with horizontal\nconnections"
par2_3="Semi finished masonry block (bad quality) with heterogeneity materials without horizontal\nconnections"
par2_4="Irregular masonry block (bad quality) with heterogeneity materials without horizontal connections"
#PARAMETAR_3
par3_1="Buildings with r.c foundation beams - stable soils with slopes of less than 5°"
par3_2="Buildings with r.c foundation beams - stable soils with slopes of less than 15°\nBuildings without ring r.c foundation beams - bad rock with slopes of less than 10°"
par3_3="Buildings with r.c foundation beams - generic soils with slopes of greater than 15°\nBuildings without r.c foundation beams - clay soils with slopes of less than 15°"
par3_4="Buildings without r.c foundation beams - clay soils with slopes of greater than 15° or bad rock with slope\ngreater than 25°"
#PARAMETAR_6
par6_1="Buildings with uniform mass distribution"
par6_2="Buildings with arcades of modest size"
par6_3="Buildings with modest-sized arcades that cover 10% to 20% of the total floor area"
par6_4="Buildings with modest-sized arcades that cover more than 20%of the total floor area"
#PARAMETAR_7
par7_1="Buildings with floors of any nature that meet the three conditions:\n(1) Rigid floor;\n(2) Effective connections with walls;\n(3) Absence of staggered floors in the same building"
par7_2="Buildings with floors equal to class a that do not meet the requirement (3)"
par7_3="Buildings with deformable floors but well connected to the walls"
par7_4="Buildings with various floors and poorly connected"
#PARAMETAR_8
par8_1="Roofing with tie-rods or r.c. ring beams"
par8_2="Non pushed roofing system without tie-rods or r.c. ring beams"
par8_3="Little pushing roofing system without tie-rods or r.c. ring beams;\nPushed roofing system with tie-rods or r.c. ring beams"
par8_4="Pushed roofing system without tie-rods or r.c. ring beams"
#PARAMETAR_9
par9_1="Buildings with well-connected window frames, small chimneys, and well-connected balconies(0)"
par9_2="Buildings with well-connected window frames, small chimneys, and well-connected balconies(5)"
par9_3="Buildings with badly connected signs connected to the walls, external fixtures poorly connected to the walls"
par9_4="Buildings with poorly tied roofs, poorly connected window frames, poorly connected signs and tall chimneys"
#PARAMETAR_10
par10_1="Absence of cracks"
par10_2="Not widespread capillary cracks"
par10_3="Cracks of medium size (2-3 mm) or cracks due to seismic action"
par10_4="Large cracks, partially collapsed walls, unsafe or poorly preserved buildings"
#PARAMETAR_14
par14_1="Typological and structural continuity with the adjacent buildings"
par14_2="Building made of the same material as the adjacent building but\n with different techniques (example: two types of masonry)"
par14_3="Building made of different material from that of the adjacent building, but both have the same behavior"
par14_4="Building presents structural heterogeneity with the adjacent\n building"
#PARAMETAR_15
par15_1="Openings percentage difference between adjacent buildings is less than 5%"
par15_2="Openings percentage difference between adjacent buildings is between 6% and 10%"
par15_3="Openings percentage difference between adjacent buildings is between 11% and 20%"
par15_4="Openings percentage difference between adjacent buildings is greater than 20"
```

Slika 4.36 Popis sadržaja tablice Excela

```
for i in vrijednosti[0]:
    k=k+1
    for j in range(len(i)):
        vr_dict.setdefault(k,{}).setdefault(j,[]).extend(i[j])

for i in range(len(vr_dict)):
#    print i
    for j in range(len(vr_dict[i])):
#        print j
        delimiter = ''
        vr_dict[i][j]=delimiter.join(vr_dict[i][j])
print vr_dict[10][0]
print par1_3
for i in range(len(vr_dict)):
    if vr_dict[i][0]==par1_1:
        vr_dict[i][0]=0
    elif vr_dict[i][0]==par1_2:
        vr_dict[i][0]=5
    elif vr_dict[i][0]==par1_3:
        vr_dict[i][0]=20
    elif vr_dict[i][0]==par1_4:
        vr_dict[i][0]=45
    if vr_dict[i][1]==par2_1:
        vr_dict[i][1]=0
    elif vr_dict[i][1]==par2_2:
        vr_dict[i][1]=5
    elif vr_dict[i][1]==par2_3:
```

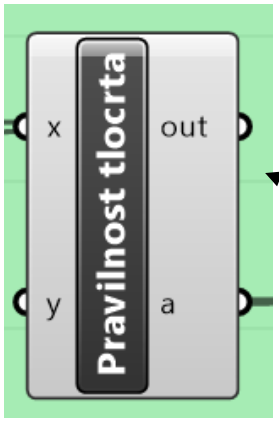
Usporedba odabranih vrijednosti sa tablicom te bodovanje sukladno odabiru

Output Help Compile

{0: {0: 45, 1: 5, 2: 20, 3: 0, 4: 20, 5: 20, 6: 45, 7: 0, 8: -15, 9: -20}, 1: {0: 45, 1: 5, 2: 20, 3: 0, 4: 20, 5: 20, 6: 45, 7: 0, 8: -15, 9: -20}, 2: {0: 5, 1: 0, 2: 20, 3: 0, 4: 5, 5: 20, 6: 45, 7: 0, 8: 0, 9: -20}, 3: {0: 45, 1: 5, 2: 20, 3: 0, 4: 20, 5: 20, 6: 45, 7: 5, 8: -15, 9: 0}, 4: {0: 45, 1: 5, 2: 20, 3: 0, 4: 45, 5: 20, 6: 45, 7: 5, 8: -15, 9: -20}, 5: {0: 5, 1: 0, 2: 20, 3: 0, 4: 5, 5: 20, 6: 45, 7: 0, 8: 0, 9: 0}, 6: {0: 45, 1: 5, 2: 20, 3: 0, 4: 20, 5: 20, 6: 45, 7: 0, 8: -15, 9: -20}, 7: {0: 45, 1: 5, 2: 20, 3: 0, 4: 20, 5: 20, 6: 45, 7: 5, 8: -15, 9: -20}, 8: {0: 45, 1: 5, 2: 20, 3: 0, 4: 20, 5: 20, 6: 45, 7: 0, 8: -15, 9: 0}, 9: {0: 45, 1: 5, 2: 20, 3: 0, 4: 20, 5: 20, 6: 0, 7: 0, 8: -15, 9: -20}, 10: {0: 20, 1: 5, 2: 20, 3: 0, 4: 20, 5: 20, 6: 20, 7: 0, 8: -15, 9: -20}, 11: {0: 5, 1: 0, 2: 20, 3: 0, 4: 5, 5: 20, 6: 45, 7: 0, 8: -15, 9: 0}, 12: {0: 20, 1: 5, 2: 20, 3: 0, 4: 20, 5: 20, 6: 0, 7: 0, 8: -15, 9: 45}, 13: {0: 20, 1: 5, 2: 20, 3: 0, 4: 20, 5: 20, 6: 45, 7: 5, 8: -15, 9: 0}, 14: {0: 45, 1: 5, 2: 20, 3: 0, 4: 20, 5: 20, 6: 45, 7: 5, 8: -15, 9: 25}, 15: {0: 20, 1: 5, 2: 20, 3: 0, 4: 20, 5: 20, 6: 45, 7: 0, 8: -15, 9: -20}, 16: {0: 20, 1: 5, 2: 20, 3: 0, 4: 20, 5: 20, 6: 45, 7: 5, 8: -15, 9: -20}, 17: {0: 20, 1: 5, 2: 20, 3: 0, 4: 20, 5: 20, 6: 45, 7: 0, 8: -15, 9: 25}, 18: {0: 20, 1: 5, 2: 20, 3: 0, 4: 20, 5: 20, 6: 45, 7: 0, 8: -15, 9: -20}, 19: {0: 20, 1: 5, 2: 20, 3: 0, 4: 20, 5: 20, 6: 45, 7: 0, 8: -15, 9: -20}, 20: {0: 20, 1: 5, 2: 20, 3: 0, 4: 20, 5: 20, 6: 45, 7: 0, 8: -15, 9: -20}, 21: {0: 20, 1: 5, 2: 20, 3: 0, 4: 20, 5: 20, 6: 45, 7: 0, 8: -15, 9: -20}, 22: {0: 20, 1: 5, 2: 20, 3: 0, 4: 20, 5: 20, 6: 45, 7: 5, 8: -15, 9: 0}, 23: {0: 20, 1: 5, 2: 20, 3: 0, 4: 20, 5: 20, 6: 45, 7: 0, 8: -15, 9: -20}, 24: {0: 20, 1: 5, 2: 20, 3: 0, 4: 20, 5: 20, 6: 20, 7: 0, 8: -15, 9: -20}, 25: {0: 20, 1: 5, 2: 20, 3: 0, 4: 20, 5: 20, 6: 45, 7: 0, 8: -15, 9: 0}, 26: {0: 20, 1: 5, 2: 20, 3: 0, 4: 20, 5: 20, 6: 20, 7: 0, 8: -15, 9: -20}, 27: {0: 20, 1: 5, 2: 20, 3: 0, 4: 20, 5: 20, 6: 45, 7: 0, 8: -15, 9: -20}, 28: {0: 20, 1: 5, 2: 20, 3: 0, 4: 20, 5: 20, 6: 20, 7: 5, 8: -15, 9: -20}}

Slika 4.37 Kod za davanje brojčane vrijednosti podacima iz tablice

Uspoređuju se odabrane vrijednosti s tekstom u skripti, gdje svaki tekst ima predodređenu vrijednost (Slike 4.36 i 4.37).



Python modul za proračun pravilnosti tlocrta, koristeći ulazne podatke iz Excela

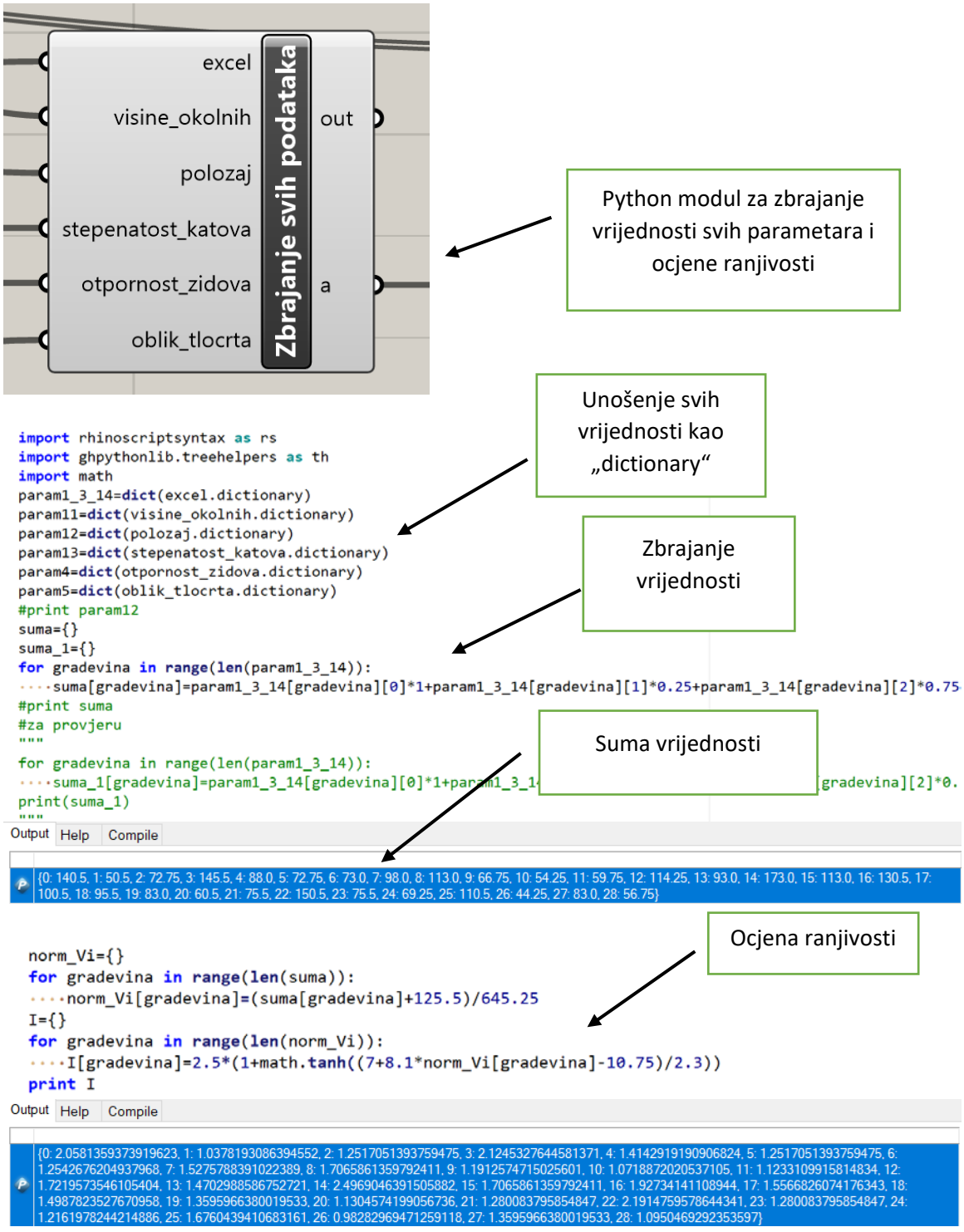
```
import rhinoscriptsyntax as rs
import ghpythonlib.treehelpers as th
bla=th.tree_to_list(x)
print bla[0][0][0]
rez={}

for i in range(len(bla[0])):
    beta1=0
    beta2=0
#    beta1=bla[0][i][0]/bla[0][i][2]
    beta2=bla[0][i][1]/bla[0][i][2]
    rez.setdefault(i,[]).extend([beta1,beta2])
print rez
rez_br={}
print rez[0][1]
for i in range(len(rez)):
    if rez[i][1]<=0.1:
        rez_br[i]=0
    elif 0.1<rez[i][1]<=0.2:
        rez_br[i]=5
    elif 0.2<rez[i][1]<=0.3:
        rez_br[i]=25
    elif 0.3<rez[i][1]:
        rez_br[i]=45
print rez_br
```

Uvjet iz metode

```
Output  Help  Compile
{0: 25, 1: 0, 2: 0, 3: 5, 4: 25, 5: 5, 6: 45, 7: 45, 8: 45, 9: 0, 10: 0, 11: 0, 12: 0, 13: 0, 14: 45, 15: 45, 16: 45, 17: 45, 18: 45, 19: 45, 20: 45, 21: 45, 22: 45, 23: 45, 24: 45, 25: 45, 26: 25, 27: 45, 28: 25}
```

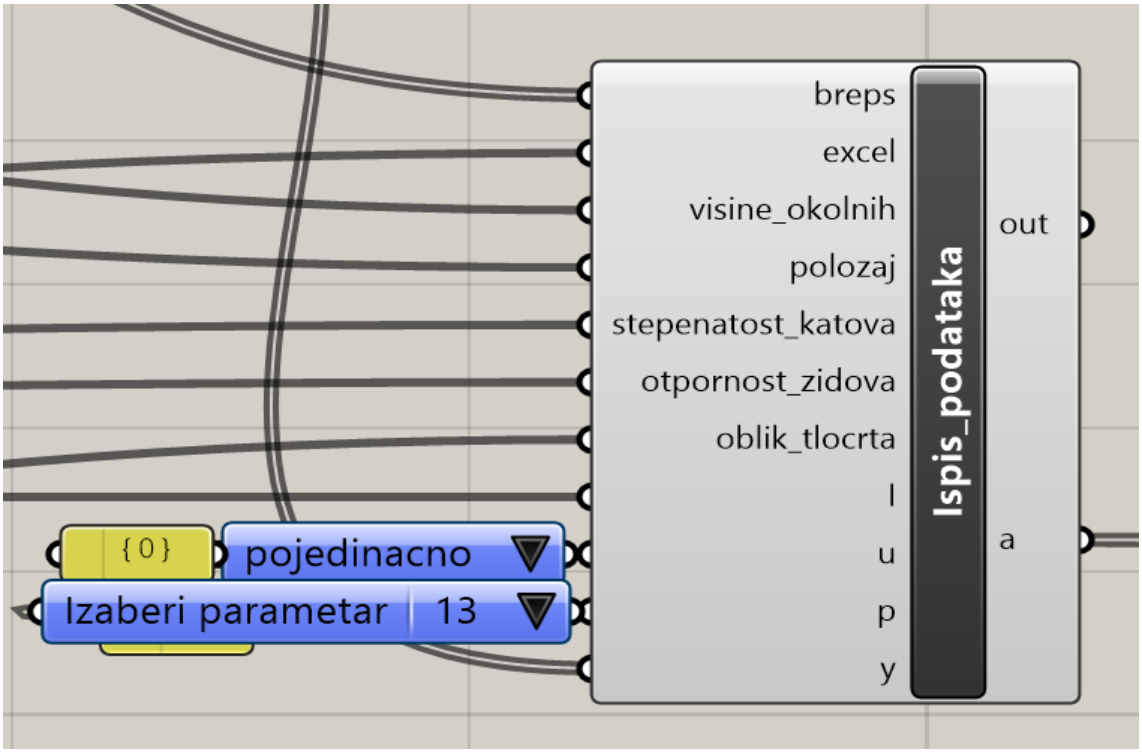
Slika 4.38 Kod za proračun pravilnosti površine

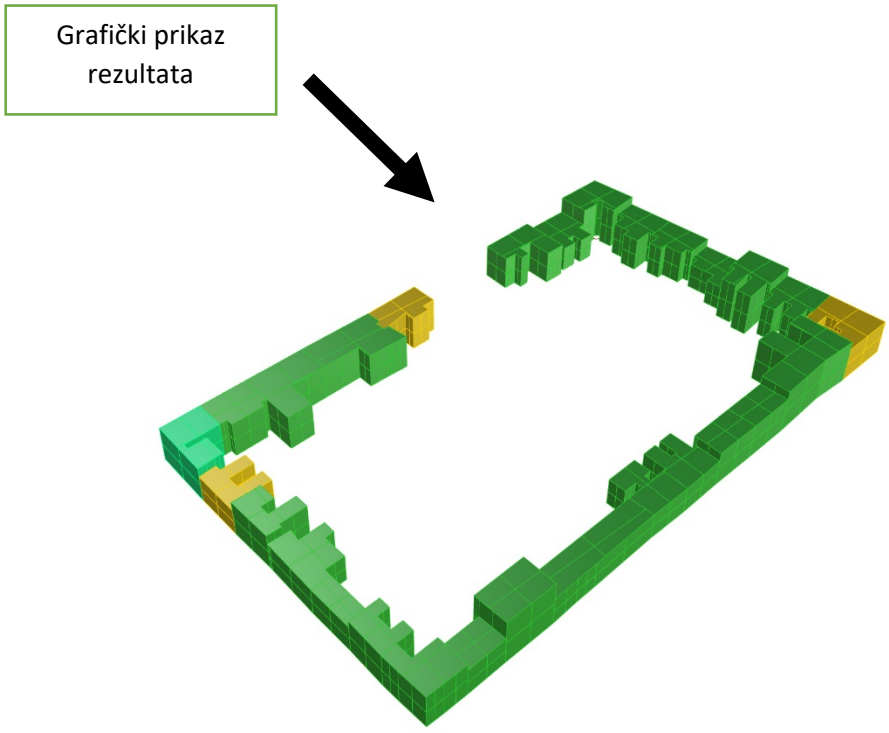
S obzirom da je prva verzija skripte, za proračun pravilnosti površine koriste se ulazni podaci iz Excela, međutim ideja je pokušati to automatizirati tako da skripta sama to može odrediti (Slika 4.38).



Slika 4.39 Kod za zbrajanje vrijednosti svih parametara

Zbrajaju se vrijednosti svih parametra u jedan dictionary, te se onda računaju indeksi ranjivosti (Slike 4.39).





Slika 4.40 Grafički prikaz rezultata

```
import rhinoscriptsyntax as rs
import Rhino
import scriptcontext
import System.Guid
import ghpythonlib.treehelpers as th
Iv=dict(I.dictionary)
brojac1=0
d={}
if x:
    for i in range(len(breps)):
        if 0<=Iv[i]<0.5 and brojac1<1:
            b=scriptcontext.doc.Objects.AddBrep(breps[i])
            color=rs.CreateColor(34,139,34)
            rs.ObjectColor(b,color)
        elif 0.5<=Iv[i]<1:
            b=scriptcontext.doc.Objects.AddBrep(breps[i])
            color=rs.CreateColor(0,255,127)
            rs.ObjectColor(b,color)
        elif 1<=Iv[i]<2:
            b=scriptcontext.doc.Objects.AddBrep(breps[i])
            color=rs.CreateColor(50,205,50)
            rs.ObjectColor(b,color)
        elif 2<=Iv[i]<3:
            b=scriptcontext.doc.Objects.AddBrep(breps[i])
            color=rs.CreateColor(255,215,0)
            rs.ObjectColor(b,color)
        elif 3<=Iv[i]<4:
            b=scriptcontext.doc.Objects.AddBrep(breps[i])
            color=rs.CreateColor(255,140,0)
            rs.ObjectColor(b,color)
        elif 4<=Iv[i]<5:
            b=scriptcontext.doc.Objects.AddBrep(breps[i])
            color=rs.CreateColor(255,0,0)
            rs.ObjectColor(b,color)
```

Slika 4.41 Uvjeti bojanja građevine

Kreira se novi „Berp“ iz starog koji je obojan ovisno o vrijednosti seizmičke ranjivosti građevine. Ideja u prvim sljedećim iteracijama skripte bi bio grafički ispis u pdf te grafički prikaz po parametrima. (Slike 4.40 i 4.41). Rezultati se ne podudaraju u potpunosti zbog nedostataka tlocrta. Naime nije bilo moguće pribaviti sve potrebne tlocrte iz arhive, a jedan od parametara zahtijeva i površine zidova. Dobivene rezultate moguće je vidjeti u prilogu, rezultati su prikazani pojedinačno po parametrima i ukupno.

5. Zaključak

Potresi postoje od vremena nastanka Zemlje, pojava je to velike razorne moći koja može sravniti čitave gradove u trenutku. Upravo u vrijeme pisanja ovoga rada dogodio se snažan potres u Turskoj, broj poginulih mjeri se u tisućama, a ozlijeđenih u desecima tisuća, uz milijune ljudi koji su ostali bez krova nad glavom. Međutim ne treba ići u Tursku da bi se uvjerili u razornu moć potresa. Tri godine nakon niza potresa koji su pogodili Hrvatsku i to bezobzira sto su bili višestruko slabiji od ovog Turskog ostavili su posljedice koje još uvijek osjećamo, pa tako možemo vidjeti ljude koji žive u kontejnerskim naseljima, privremenima domovima i slično, zatim veliki broj oštećenih ili porušenih građevina. To nam još jednom služi kao podsjetnik koliko opasno može biti zanemarivanje i podcjenjivanje potresa. Potres se ne može predvidjeti, ne može se spriječiti, ali može se naučiti živjeti s njime. Pravilno upravljanje rizikom od potresa izradom strateških planova koji bi imali funkciju prikupljanja informacija i pravilne/stvarne procjene stanja građevina, ali prije nego se potres dogodi. Na taj način bi se omogućila izrada alata i metoda kojima bi se moglo planski pristupiti problemu pronalaženja najboljih tehničkih rješenja te implementaciji istih. Takav način rukovođenja doveo bi do velikog smanjenja eventualne ekonomske štete nastale potresom, ali ono još važnije doprinijelo bi sigurnosti ljudi. Jedan od alata koji vjerujem može biti ključan u takvom pristupu je i parametarsko programiranje odnosno projektiranje. Upravo se ono koristilo u ovome radu kakao bi se metoda (Formisano i ostali, 2015; Petrini & Benedetti, 1984) pokušala automatizirati i kako bi se ubrzao postupak s ciljem ocjene što većeg broja građevina na seizmičku ranjivost u što kraćem vremenu. U radu se na primjeru bloka zgrada već analiziranom u radu (Moretić i ostali, 2022) demonstrira „Grasshopper“ skripta sa svojim mogućnostima. Postoje određena ograničenja, pa tako parametar u kojem se uzima u obzir složenost tlocrta, skripta ne može automatski izračunati. Također pristup bazi podataka koji sadrži podatke o visini i broju etaža građevine bi poboljšao kvalitetu rezultata koji se dobiju, te bi znatno ubrzao postupak. Za usporedbu se uzeo se već analizirani blok kako bi se lakše procijenio rezultat generiran skriptom. Ovaj rad demonstrira mogućnosti koje parametarsko programiranje pruža. Skripta može poslužiti kao osnova za neka daljnja istraživanja implementacije parametarskog programiranja u građevinarstvu, koje je zasigurno nešto što čeka građevinske inženjere u budućnosti.

Literatura

Ćatić, S. (2022). *TEMPORAL AND SPATIAL ANALYSIS OF EARTHQUAKE IN STOLAC ON APRIL 22nd, 2022*.

Fabbrocino Giovanni. (2008). *researchgate*. https://www.researchgate.net/profile/Giovanni-Fabbrocino/publication/270686266/figure/fig18/AS:669092595916802@1536535514318/Building-devastated-during-1880-Zagreb-earthquake-Pinta-2008.ppm

Formisano, A., Florio, G., Landolfo, R., & Mazzolani, F. M. (2015). Numerical calibration of an easy method for seismic behaviour assessment on large scale of masonry building aggregates. *Advances in Engineering Software*, *80*(C), 116–138. https://doi.org/10.1016/j.advengsoft.2014.09.013

Kassem, M. M., Mohamed Nazri, F., & Noroozinejad Farsangi, E. (2020). The seismic vulnerability assessment methodologies: A state-of-the-art review. U *Ain Shams Engineering Journal* (Sv. 11, Izdanje 4, str. 849–864). Ain Shams University. https://doi.org/10.1016/j.asej.2020.04.001

Markušić, S., Stanko, D., Korbar, T., Belić, N., Penava, D., & Kordić, B. (2020). The Zagreb (Croatia) M5.5 earthquake on 22 March 2020. *Geosciences (Switzerland)*, *10*(7), 1–21. https://doi.org/10.3390/geosciences10070252

Moretić, A., Chieffo, N., Stepinac, M., & Lourenço, P. B. (2022). Vulnerability assessment of historical building aggregates in Zagreb: implementation of a macroseismic approach. *Bulletin of Earthquake Engineering*. https://doi.org/10.1007/s10518-022-01596-5

Novak, M. Š., Uroš, M., Atalić, J., Herak, M., Demšić, M., Baniček, M., Lazarević, D., Bijelić, N., Crnogorac, M., & Todorić, M. (2020). Zagreb earthquake of 22 March 2020 – Preliminary report on seismologic aspects and damage to buildings. *Gradjevinar*, *72*(10), 843–867. https://doi.org/10.14256/JCE.2966.2020

Petrini, V., & Benedetti, D. (1984). *) A method for evaluating the seismic vulnerability of masonry buildings*.

Rhinoceros. (2022). https://rhino3d.software/history-of-rhino

Stepinac, M., Lourenço, P. B., Atalić, J., Kišiček, T., Uroš, M., Baniček, M., & Šavor Novak, M. (2021). Damage classification of residential buildings in historical downtown after the ML5.5 earthquake in Zagreb, Croatia in 2020. *International Journal of Disaster Risk Reduction*, *56*. https://doi.org/10.1016/j.ijdrr.2021.102140

SubD Surface. (2022). http://www.formz.com/manuals/formz9/Layout/WebHelp/10850_Subdivision_surfaces.html#:~:text=Subdivision%20Surface%20modeling%20is%20a,create%20a%20smoother%20derivative%20surface.

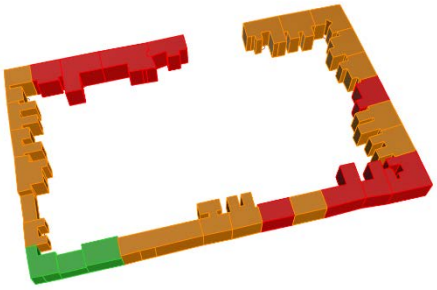
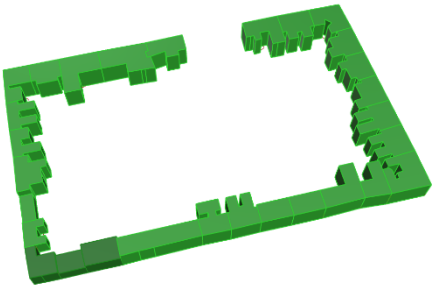
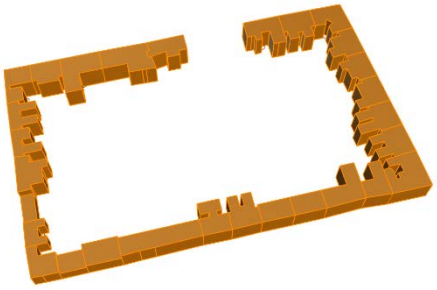
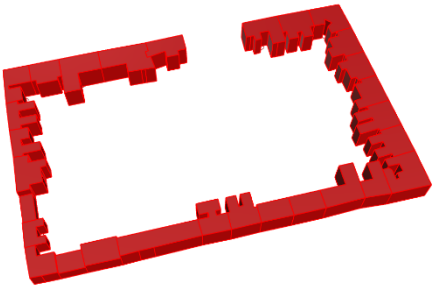
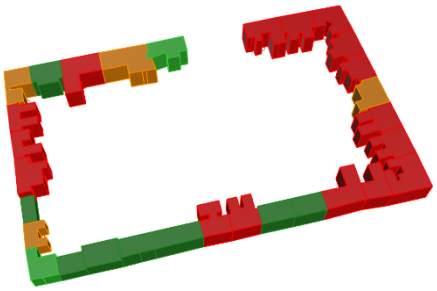
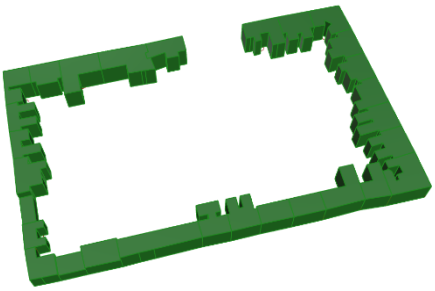
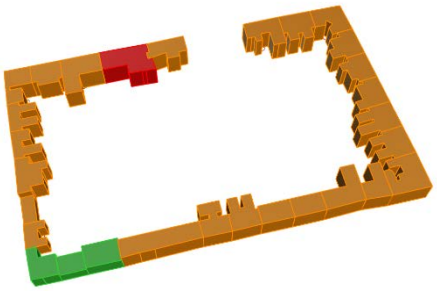
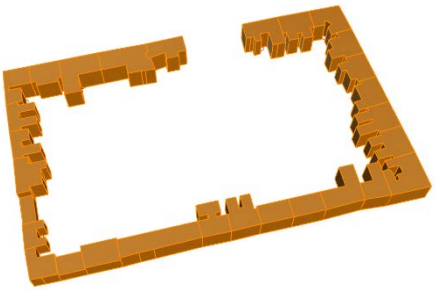
Vibor Cipan. (2021, siječanj 16). *Potresi u Zagrebu i Petrinji – uzroci i rasjedi uz prikaz i animaciju*.

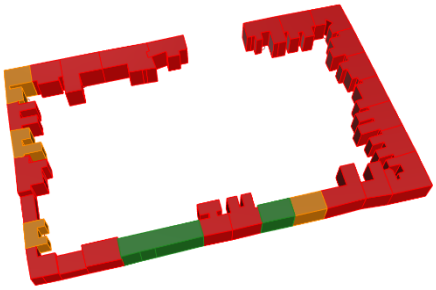
Bonevska T., Grlić M., Horvat M., Miholić L., & Martinić I. (2020). Zagrebački Potres 22.Ožujka 2020. *Geografski horizont*, *2*, 21–32.

Popis Slika

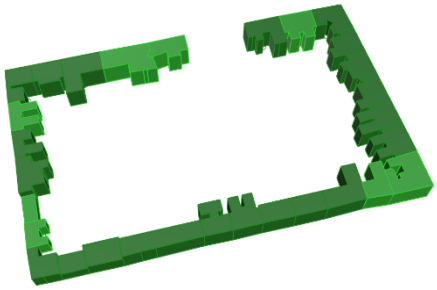
Popis tablica

Prilozi

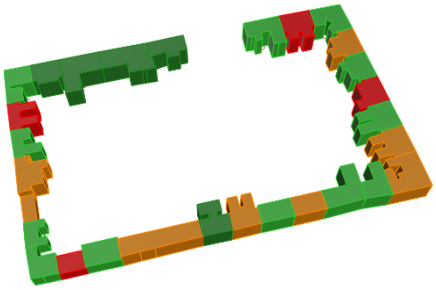
 Blok1 Parametar 1	 Blok1 Parametar 2
 Blok1 Parametar 2	 Blok1 Parametar 3
 Blok1 Parametar 4	 Blok1 Parametar 5
 Blok1 Parametar 6	 Blok1 Parametar 7



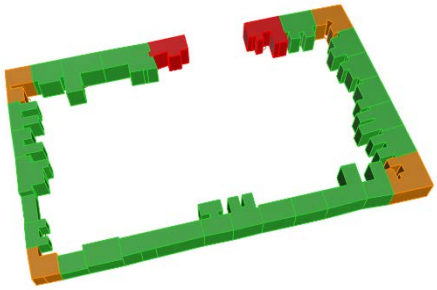
Blok1 Parametar 9



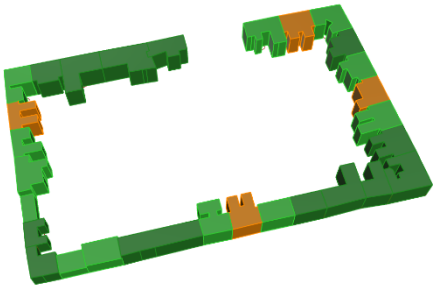
Blok1 Parametar 10



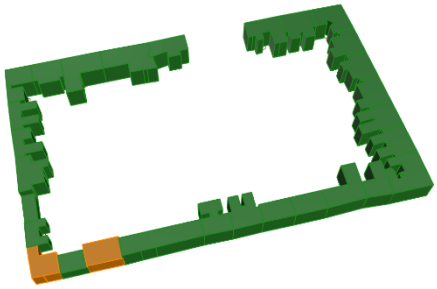
Blok1 Parametar 11



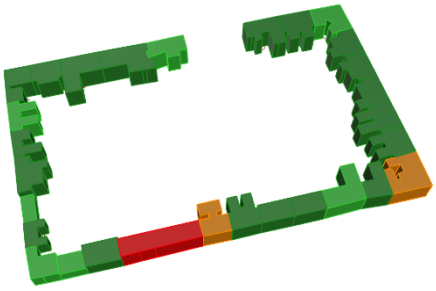
Blok1 Parametar 12



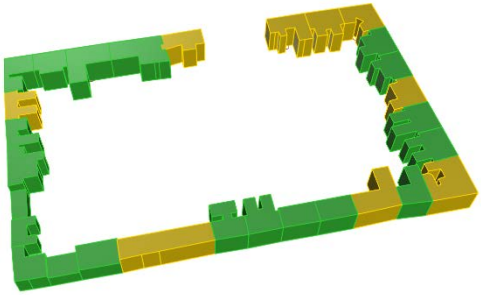
Blok1 Parametar 13



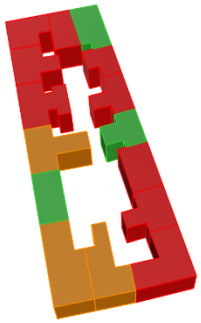
Blok1 Parametar 14



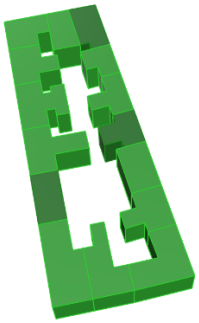
Blok1 Parametar 15



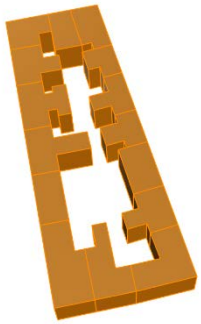
Blok1 Ukupno



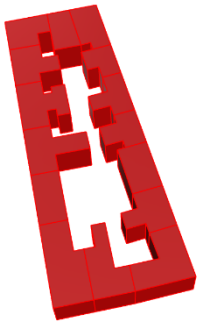
Blok2 Parametar 1



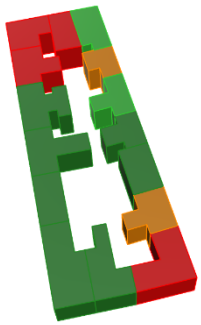
Blok2 Parametar 2



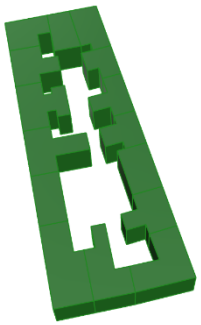
Blok2 Parametar 3



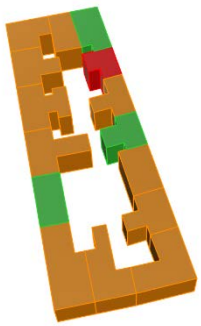
Blok2 Parametar 4



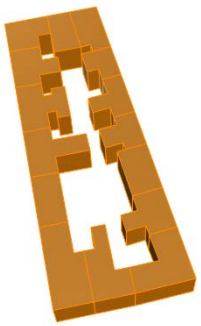
Blok2 Parametar 5



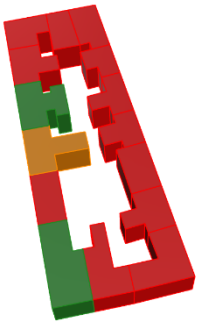
Blok2 Parametar 6



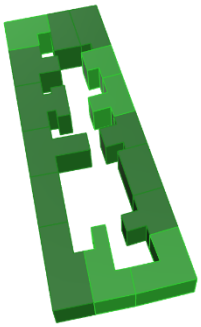
Blok2 Parametar 7



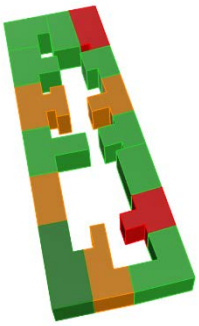
Blok2 Parametar 8



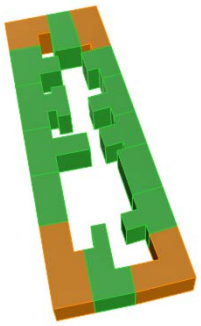
Blok2 Parametar 9



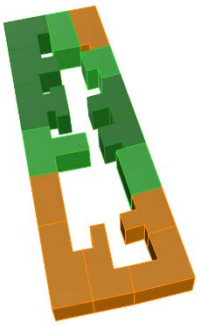
Blok2 Parametar 10



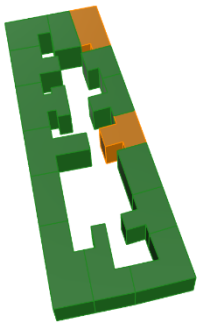
Blok2 Parametar 11



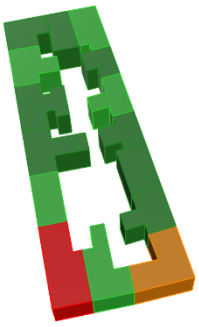
Blok2 Parametar 12



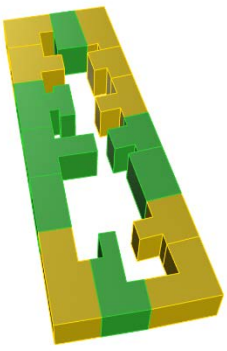
Blok2 Parametar 13



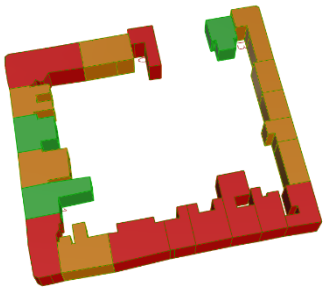
Blok2 Parametar 14



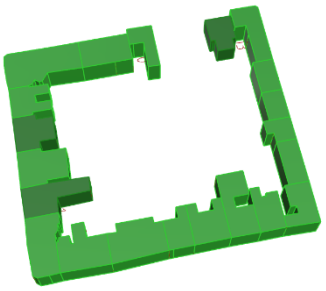
Blok2 Parametar 15



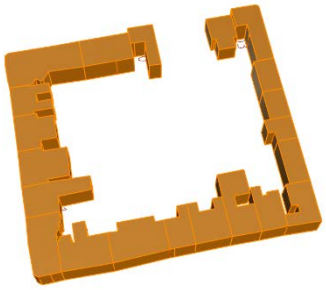
Blok2 Ukupno



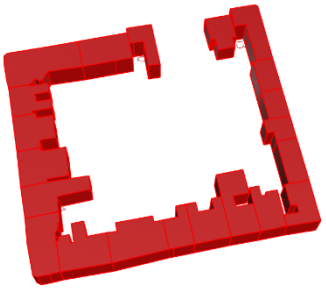
Blok3 Parametar 1



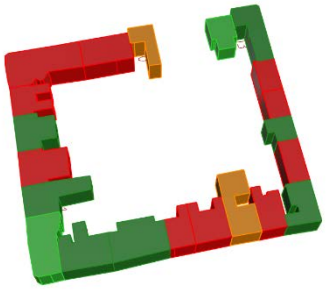
Blok3 Parametar 2



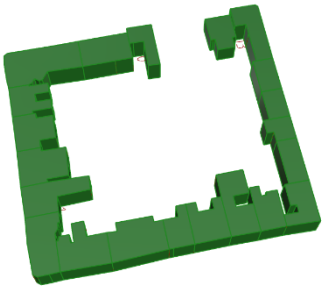
Blok3 Parametar 3



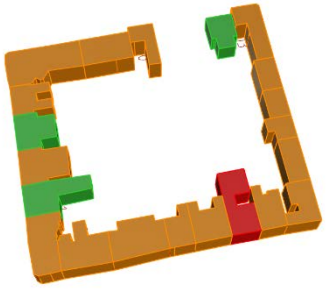
Blok3 Parametar 4



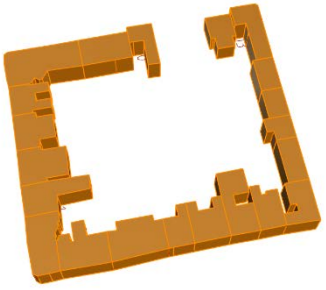
Blok3 Parametar 5



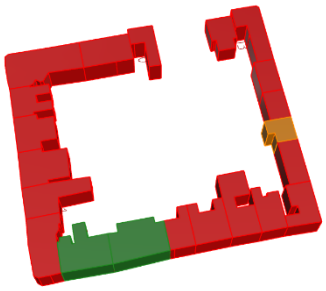
Blok3 Parametar 6



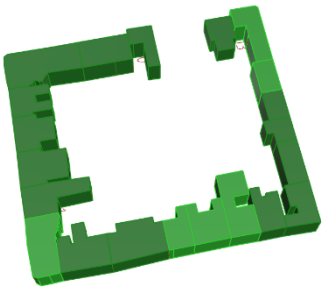
Blok3 Parametar 7



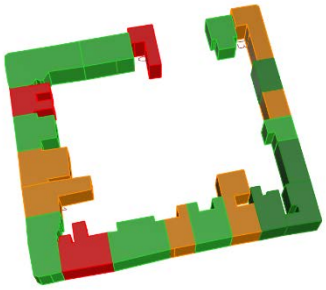
Blok3 Parametar 8



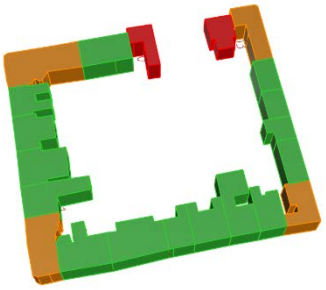
Blok3 Parametar 9



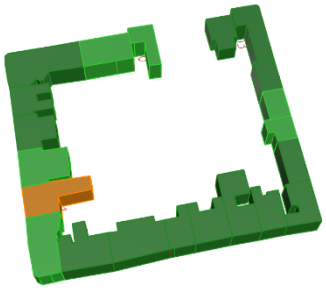
Blok3 Parametar 10



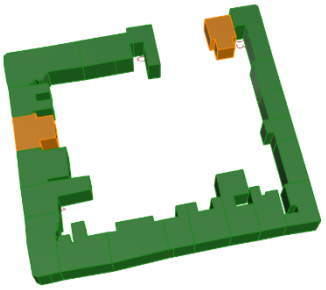
Blok3 Parametar 11



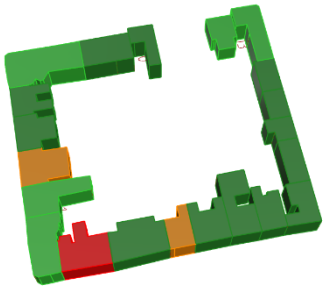
Blok3 Parametar 12



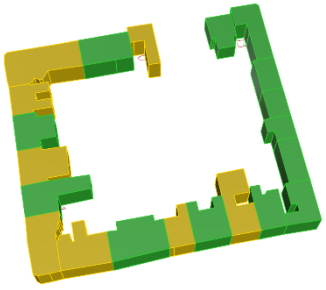
Blok3 Parametar 13



Blok3 Parametar 14



Blok3 Parametar 15



Blok3 Ukupno