

Dron.co	Verzija: 1.0
Tehnička dokumentacija	Datum: 29/01/26

Dron.co
Tehnička dokumentacija
Verzija 1.0

Studentski tim: Jakov Ramljak
Stjepan Džidžić
Leon Katić
Martin Golub
Matea Maračić

Nastavnik: Željka Mihajlović

Dron.co	Verzija: 1.0
Tehnička dokumentacija	Datum: 29/01/26

Sadržaj

1.	Opis razvijenog proizvoda	3
1.1	Tema	3
1.2	Tijek igre	3
2.	Tehničke značajke	5
2.1	Korišteni alati	5
2.2	Letenje drona	5
2.3	Nošenje paketa	7
	2.3.1 Paketi	7
	2.3.2 Sustav za nošenje paketa	7
	2.3.3 Dostava paketa	9
2.4	Baterija	10
2.5	Učinak vjetra	11
2.6	Proceduralno generiranje modela	12
	2.6.1 Geometrijski čvorovi	12
	2.6.2 Postupak izrade građevine	13
	2.6.3 Mreže čvorova za izradu građevina	14
2.7	Vizualni efekti i stil	17
2.8	Audio efekti	19
2.9	Dizajn razine igre	20
	2.9.1 Urbanističko planiranje i organizacija grada	20
	2.9.2 Sjedište kompanije i punionice	22
	2.9.3 Detalji i prepreke	24
2.10	Korisničko sučelje	25
3.	Upute za korištenje	26
3.1	Kako igrati	26
3.2	Kontrole	26
3.3	Izvršna datoteka i izvorni kod	26
4.	Literatura	27

Dron.co	Verzija: 1.0
Tehnička dokumentacija	Datum: 29/01/26

1. Opis razvijenog proizvoda

1.1 Tema

Tema projekta Dron.co igra je dostavljanja paketa dronom u urbanističkom gradskom okruženju. Tema igre inspirirana je sve češćom uporabom dronova u stvarnom svijetu, a taj se koncept ovom igrom približava igraču kroz ulogu dostavljača koji mora sigurno i učinkovito prenositi pakete do određenih lokacija na mapi.

Motivacija za odabir ove teme i koncepta proizlazi iz zanimanja za rad u 3D prostoru, izgradnju igrica, kao i želje za stjecanjem novog znanja u području računalne grafike i razvoja igara. Dodatna motivacija bila je izrada zabavne i zanimljive igre u kojoj se igrač mora orijentirati u prostor, precizno upravljati dronom te donositi brze odluke tijekom leta. Time igrač razvija iskustvo u prostornoj orijentaciji i koordinaciji, a istovremeno se upoznaje s osnovnim izazovima upravljanja dronom u pravom svijetu.

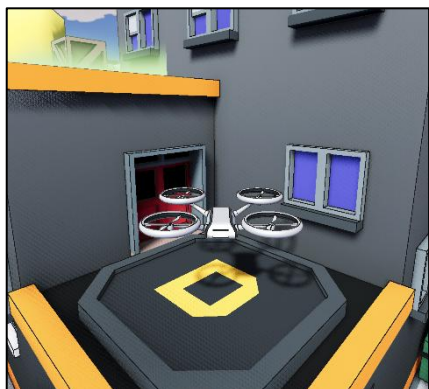
Igra je osmišljena tako da naglasak stavlja na kontroliranje drona, kretanje prostorom uz izbjegavanje prepreka, uz ograničenja vremena i konstantno pražnjenje baterije. Takav pristup igri daje jasno definirane ciljeve, ali i dovoljno izazova koje igru čine zanimljivijom. Tema je također izabrana zbog toga što omogućuje primjenu tehničkih rješenja, poput fizike, proceduralnog generiranja modela i dizajna razina igre, što je čini prikladnom za završni projekt.

1.2 Tijek igre

Dron.co interaktivna je dostavljačka igra u kojoj igrač preuzima ulogu jednog od dronova dostavne kompanije Dron.co čiji je zadatak raznositi pakete po gradu. Igra se fokusira na precizno upravljanje dronom, planiranje rute kroz ulice grada te upravljanjem razine baterije. Igrač mora balansirati brzinu i sigurnost kako bi uspješno obavio dostave unutar zadanog vremena.

Igra započinje tako što se dron pojavljuje u glavnom sjedištu dostavne kompanije. Na toj lokaciji automatski se stvara prvi paket koji igrač mora dostaviti. Nakon prikupljanja paketa, igrač ga mora što brže dostaviti na lokaciju koja je u gradu jasno označena snopom svjetlosti i strelicom iznad drona. Tijekom leta igrač upravlja dronom kroz gradske ulice, pazeći na prepreke i optimalnu rutu kako bi na vrijeme stigao do odredišta. Kada je paket uspješno dostavljen, broj dostavljenih paketa se povećava, a vremenski ograničeni brojač se produžuje. Istovremeno se na početnoj lokaciji ponovno stvara novi paket, zbog čega se igrač mora vratiti u sjedište kompanije i ponoviti cijeli postupak. Igra traje dokle god traje preostalo vrijeme na brojaču. Kada vrijeme istekne, igra se završava te se prikazuje završni rezultat, odnosno ukupan broj paketa koje je igrač uspio uspješno dostaviti tijekom igre.

Dron.co	Verzija: 1.0
Tehnička dokumentacija	Datum: 29/01/26



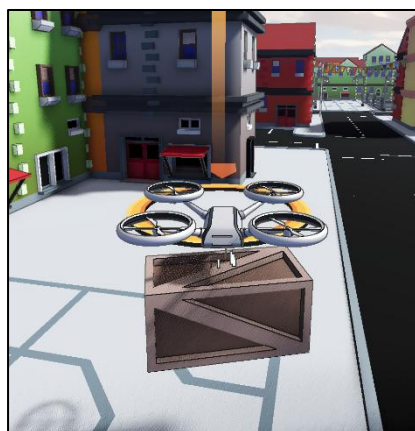
Slika 1: Početak igre na glavnoj postaji



Slika 2: Prikupljanje paketa



Slika 3: Nošenje paketa po gradu



Slika 4: Dostavljanje paketa na odredište

Dron.co	Verzija: 1.0
Tehnička dokumentacija	Datum: 29/01/26

2. Tehničke značajke

2.1 Korišteni alati

Za razvoj igre Dron.co korišteno je nekoliko alata koji su poslužili pri izradi 3D modela, scena, dizajna razine i obrade zvuka. Alat Godot Engine korišten je za glavni razvoj igre, kompletnu logiku igre, upravljanje fizikom, korisničko sučelje te integraciju vizualnih i audio efekata. Skripte su u Godot Engineu pisane u programskom jeziku GDScript, koji omogućuje jednostavno i organizirano pisanje koda.

Za izradu 3D modela korišten je program Blender. U Blenderu su modelirani svi objekti u igri, uključujući tlo, ceste, nogostup, zgrade, dron, pakete, detalje okoliša i ostalo. Dizajn same razne igre, unutar Godot Enginea, realiziran je koristeći alat GridMap koji omogućuje jednostavno slaganje modela unutar mreže kako bi se dijelovi grada smisljeno i ujednačeno spajali. Za izradu i prilagodbu materijala i vizualnih efekata korišteni su Godot Shaderi. Za obradu zvučnih efekata koristili su se zvučni zapisi preuzeti s vanjskih izvora, a koji su onda uvezeni u Godot. Upravljanje projektom i planiranje rada odrađeno je komunikacijom preko Discorda, objavljivanjem koda i napretka na Github te izradom Ganttovih dijagrama i WBS struktura s pomoću alata na internetu.

2.2 Letenje drona

Kretanje drona obavlja se manipuliranjem Godotovim varijablama stanja tipa *Vector3*, a to su *position*, *velocity* i *rotation*. Vektor *acceleration* postavljamo na nulvektor.

```

1 var input_dir := Input.get_vector("go_left", "go_right", "go_front", "go_back")
2 var direction := (DRONE.transform.basis * Vector3(input_dir.x, 0, input_dir.y)).normalized()
3 if direction != Vector3.ZERO:
4     acceleration.x = direction.x * SPEED * (WIND_SLOWDOWN if windy else 1.0)
5     acceleration.z = direction.z * SPEED * (WIND_SLOWDOWN if windy else 1.0)
6 else:
7     acceleration.x = 0.0
8     acceleration.z = 0.0

```

Slika 5: Upravljanje akceleracije drona po xz površini

Da bismo saznali smjer kretanja tipkama WASD ili desnim analognim *stickom* na kontroleru potrebno je koristiti metodu *Input.get_vector* koja nam vraća vektor dimenzije 2 ovisno o tome koje tipke su pritisnute te ga zatim pretvoriti u vektor dimenzije 3 s y komponentom postavljenom na 0. Da bi se ostvarilo kretanje u odnosu na relativne koordinate drona, a ne na globalne koordinate potrebno je pomnožiti s rotacijskom matricom drona. Tako se dobiva jedinični vektor *direction* pomoću kojeg možemo zadati akceleraciju drona x i z komponente. Ako je vektor *direction* nulvektor, akceleracije se postavljaju na 0.

```

1 var throttle_input := Input.get_axis("go_up", "go_down")
2 if abs(throttle_input) > 0.05:
3     acceleration.y = -throttle_input * SPEED
4 else:
5     acceleration.y = 0.0

```

Slika 6: Upravljanje vertikalne akceleracije

Za realizaciju vertikalnog kretanja koristi se metoda *Input.get_axis* koja nam vraća float od -1 do 1 te pomoću te vrijednosti se postavlja akceleracija u y smjeru. Naravno, ako je *throttle_input* malen vertikalne akceleracije nema da se ne bi za male pomake na lijevom analognom *sticku* promijenila akceleracija.

```

1 var look_input := Input.get_axis("rotate_left", "rotate_right")
2 if abs(look_input) > 0.05:
3     DRONE.rotate_y(-look_input * ROTATE_SENSITIVITY * delta)

```

Slika 7: Upravljanje rotacije drona

Dron.co	Verzija: 1.0
Tehnička dokumentacija	Datum: 29/01/26

Analogno vertikalnom kretanju ostvaruje se i rotacija lijevo-desno pomoću metode `rotate_y` koja rotira dron za neki kut u radijanima oko vertikalne osi. Argument metode se množi sa delta *timeom* da se dron okreće istom brzinom neovisno o FPS-ima igre.

```
if direction != Vector3.ZERO:
    > var local_dir := DRONE.global_transform.basis.inverse() * direction
    > var target_rot_x = local_dir.z * TILT_STRENGTH
    > var target_rot_z = -local_dir.x * TILT_STRENGTH

    > DRONE.rotation.x = lerp(DRONE.rotation.x, target_rot_x, TILT_SMOOTHNESS * delta)
    > DRONE.rotation.z = lerp(DRONE.rotation.z, target_rot_z, TILT_SMOOTHNESS * delta)
else:
    > DRONE.rotation.x = lerp(DRONE.rotation.x, 0.0, TILT_SMOOTHNESS * delta)
    > DRONE.rotation.z = lerp(DRONE.rotation.z, 0.0, TILT_SMOOTHNESS * delta)
```

Slika 8: Efekt nagiba

Nagib drona ne utječe na let i za njega se koristi efekt koji simulira nagib. Ostvaren je efekt gdje se kretanje i nagib osjeća prirodno i glatko. Prvo je potrebno provjeriti kreće li se uopće dron, ako ne onda se nagib vraća u početnu poziciju koja je paralelna s podom. Inače, ako se dron kreće onda lagano postavljamo nagib. Problem je što varijabla *direction* pokazuje u globalnom prostoru, a rotacija drona se izvodi u njegovom lokalnom prostoru. Ovo je riješeno tako što orijentaciju drona transformiramo u lokalni koordinatni prostor drona i pomnožimo s varijablom *direction* te dobijemo smjer nagiba koji je relativan dronu. Komponenta smjera nagiba x nam govori koliki je nagib lijevo ili desno, dok nam komponenta smjera z govori koliki je nagib naprijed ili nazad. Konačno, glatko se mijenja rotacija funkcijom linearne interpolacije (`lerp`), uzimajući u obzir broj slika u sekundi.

```
DRONE.velocity += acceleration * delta + wind_force
DRONE.velocity *= FRICTION
self.push_pushables(delta)
DRONE.move_and_slide()
```

Slika 9: Brzina drona

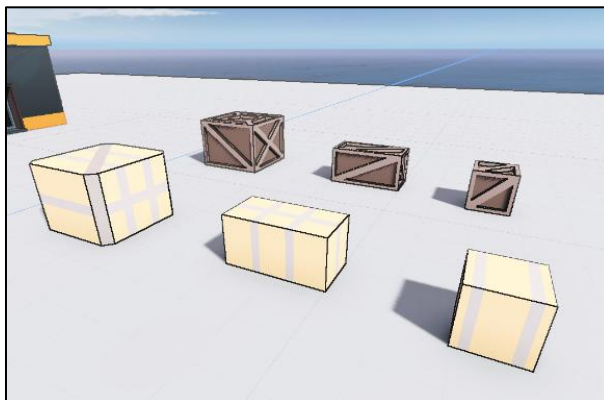
Na kraju obrade vektora *acceleration* ažurira se iznos vektora *DRONE.velocity* na način da se mu zbroji brzina *acceleration * delta* i *wind_force* (objašnjeno kasnije) i na kraju se brzina drona pomnoži faktorom trenja da bi se osiguralo da dron ima maksimalnu brzinu i da ima efekt usporavanja. Potrebno je pozvati ugrađenu metodu *DRONE.move_and_slide* da se ažurira pozicija drona.

Dron.co	Verzija: 1.0
Tehnička dokumentacija	Datum: 29/01/26

2.3 Nošenje paketa

2.3.1 Paketi

U igri postoje šest različitih vrsta paketa, pri čemu svaka veličina ima vlastite parametre poput mase, dimenzija i ponašanja u fizikalnom sustavu. Iako se paketi razlikuju po veličini i težini, svi oni koriste istu skriptu (Package3D), čime je osigurana konzistentna logika ponašanja. Razlike u teksturama su samo vizualne (bijeli paketi su isti kao smeđi).



Slika 10 Paketi

2.3.2 Sustav za nošenje paketa

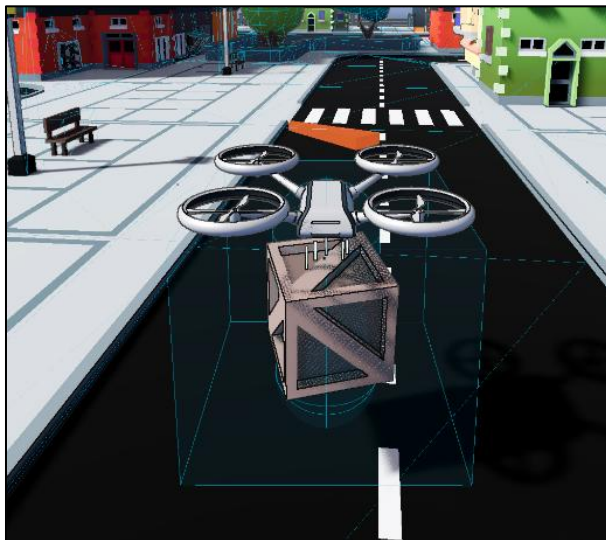
Hook node sustav na dronu služi za nošenje paketa. Implementiran je kao zasebna komponenta drona koja upravlja detekcijom, kačenjem i otpuštanjem paketa. Sustav je dizajniran tako da može nositi pakete različitih dimenzija, a svaka veličina ima drugačiji utjecaj na ponašanje drona.

Kuka se nalazi na donjoj strani drona i sastoji se od detekcijske zone (koja prati sudare s paketima), referentne točke za kačenje i fizičkog zgloba. Kada se dron približi paketu, paket ulazi u detekcijsko područje, čime se automatski provjerava je li objekt valjan za nošenje (ako je Package3D). Ako dron u tom trenutku ne nosi paket, aktivira se proces kačenja.

Prilikom kačenja paket se najprije poravnava s markerom koji definira točnu poziciju paketa u letu. Nakon toga se paket fizički povezuje s dronom s pomoću *Generic6DOFJoint3D* zgloba. Umjesto statičkog “lijepljenja” paketa za dron, koristi se fizički spoj koji omogućuje prirodno ponašanje tijekom leta. Linearne i kutne opruge zgloba drže paket blizu ravnotežne točke, dok *damping* smanjuje pretjerano njihanje i rotaciju. Tako paket djeluje stabilno, ali i dalje reagira na promjene smjera i brzine.

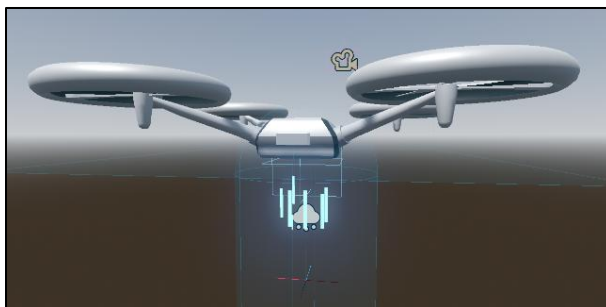
Kako paketi dolaze u šest različitih veličina, kuka se dinamički prilagođava svakom od njih. Prilikom kačenja mijenja se kolizija drona tako da obuhvati paket, čime se sprječavaju neželjeni sudari s okolinom. Također, tijekom nošenja paket je isključen iz određenih slojeva sudara kako se ne bi sudarao s dronom dok ga nosi.

Dron.co	Verzija: 1.0
Tehnička dokumentacija	Datum: 29/01/26



Slika 11 Kolizija drona dok nosi paket

Sustav uključuje i vizualnu povratnu informaciju za igrača. Kada je paket zakačen, aktiviraju se čestice koje daju do znanja da se nosi paket kako ne bi samo levitirao u zraku.



Slika 12 Vizualni efekt nošenja paketa

Igrač u bilo kojem trenutku može ručno otpustiti paket pritiskom na odgovarajuću tipku. U tom trenutku zglob se uklanja, paket se vraća u normalno fizikalno stanje, kolizije se resetiraju, a čestice se gasu.

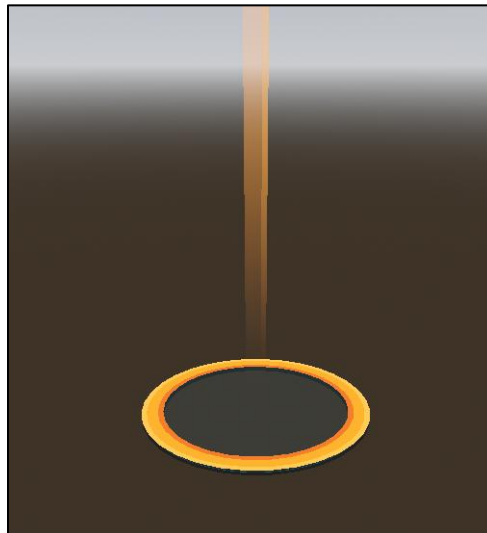
Dron.co	Verzija: 1.0
Tehnička dokumentacija	Datum: 29/01/26

2.3.3 Dostava paketa

Paketi se u igri stvaraju u glavnom sjedištu (engl. headquarters) dostavne kompanije, koje služi kao početna i povratna točka svake dostave. U svakom ciklusu igre generira se novi paket, a istovremeno se nasumično odabire lokacija u gradu na koju paket treba biti dostavljen. Ta lokacija se u svijetu igre označava kao dostavno područje.



Slika 13 Sjedište dostavne kompanije



Slika 14 Lokacija dostave

Nakon što igrač preuzme paket pomoću *hook* sustava, iznad drona se aktivira hologramska strelica koja pokazuje smjer prema lokaciji dostave. Strelica je prisutna sve dok paket nije dostavljen te se cijelo vrijeme rotira kako bi stalno pokazivala prema lokaciji dostave.



Slika 15 Hologramska strelica

Kada dron stigne do označene lokacije, igrač ispušta paket iznad dostavnog područja. Ako paket uđe u zonu dostave, odnosno dođe u koliziju s dostavnom lokacijom, dostava se bilježi kao uspješna, povećava se brojčasti dostavljenih paketa, paket i dostavna lokacija se uništavaju te se aktivira vizualni efekt konfeta. Nakon toga započinje novi ciklus stvaranjem sljedećeg paketa u sjedištu i odabirom nove nasumične lokacije u gradu.

Dron.co	Verzija: 1.0
Tehnička dokumentacija	Datum: 29/01/26

2.4 Baterija

Tijekom trajanja igre kontinuirano se prati razina baterije drona, koja se postupno prazni tijekom leta. Ako razina baterije dođe do nule, igra se završava. Kada baterija padne na 25 % ili manje, aktivira se vizualni efekt desaturacije cijele slike, čime se igraču jasno daje do znanja da je baterija pri kraju. Kako bi spriječio završetak igre, igrač mora posjetiti lokacije za punjenje koje su raspoređene po gradu. Kada dron dođe u kontakt s područjem punjenja, baterija se počinje automatski puniti. Tijekom punjenja prikazuje se vizualni efekt koji označava da je punjenje aktivno, a proces traje sve dok dron ostaje unutar područja punjenja.



Slika 16 Lokacija za punjenje



Slika 17 Animacija punjenja

Dron.co	Verzija: 1.0
Tehnička dokumentacija	Datum: 29/01/26

2.5 Učinak vjetra

Aspekt vjetra promatramo na dva načina, a to su efekt vjetra na gibanje drona i vizualni dio vjetra koji ostvarujemo simuliranjem čestica.

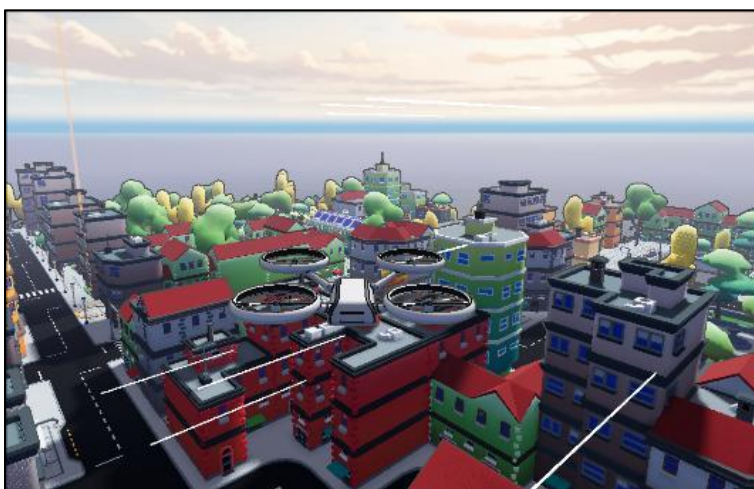
```
var wind_x = sin(wind_time * WIND_SPEED) * WIND_STRENGTH
var wind_z = cos(wind_time * WIND_SPEED * 0.7) * WIND_STRENGTH
wind_x += random.randf_range(-0.1, 0.1)
wind_z += random.randf_range(-0.1, 0.1)
wind_force = Vector3(wind_x, 0, wind_z) * (1.0 if windy else 0.0)
```

Slika 18: Vektor vjetra

```
acceleration.x = direction.x * SPEED * (WIND_SLOWDOWN if windy else 1.0)
acceleration.z = direction.z * SPEED * (WIND_SLOWDOWN if windy else 1.0)
```

Slika 19: Utjecaj vjetra na iznos akceleracije

Efekt vjetra na gibanje drona ostvaruje se korištenjem funkcija sinusa i kosinusa. Rezultat je da dron otprilike prati neki oblik elipse. Dakle, kako vrijeme *wind time* protječe, rotira se vektor vjetra. Također dodaje se mala slučajna vrijednost u komponente x i z vjetra za realističnije mijenjanje smjera vjetra. Akceleracija drona je nešto manja ako se dron nalazi u *bounding boxu* od vjetra što se provjerava preko varijable *windy* tipa bool. Također vektor *wind_force* ostaje kakav je ako je *windy true*, a ako je *windy false* postavlja se na nulvektor da nema efekta vjetra. Varijabla *windy* mijenja se pomoću signala.

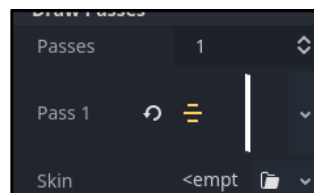


Slika 20: Čestice vjetra kad igrač uđe u područje vjetra

```
@export var FOLLOW_NODE: Node3D

func _process(_delta: float) -> void:
    > self.global_position.x = FOLLOW_NODE.global_position.x
    > self.global_position.z = FOLLOW_NODE.global_position.z
```

Slika 21: Praćenje drona sa vjetrom



Slika 22: Čestica vjetra

Za vizualni dio vjetra koristi se *Node GPUParticles3D*. Pomoću njega simulira se kretanje čestica vjetra. *Node* od vjetra prati x i z koordinatu da vizualni efekt vjetra bude uvijek prisutan vertikalno oko drona.

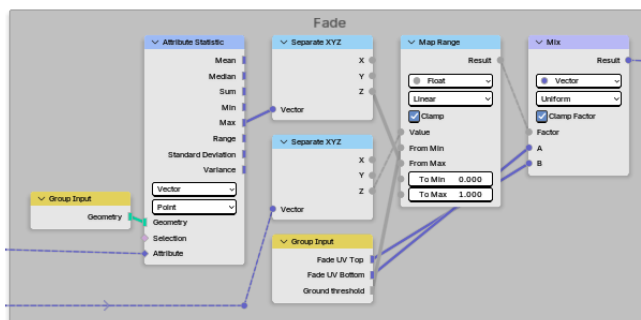
Dron.co	Verzija: 1.0
Tehnička dokumentacija	Datum: 29/01/26

2.6 Proceduralno generiranje modela

Proceduralno generiranje modela je način stvaranja 3D modela s pomoću skupa unaprijed definiranih pravila, algoritama ili matematičkih funkcija, umjesto ručnog oblikovanja svakog pojedinačnog elementa. Promjenom osnovnih parametara, dizajner može regenerirati cijeli model, što omogućuje vrlo učinkoviti iterativni rad, jednostavno skaliranje i postizanje uvjerljive prirodne raznolikosti u modelima. Proceduralno modeliranje se najviše koristi izradu izuzetno složenih i detaljnih objekata ili cijelih okolina (poput šuma, gradova ili planinskih lanaca) iz minimalnog broja ulaznih podataka. U ovom projektu je proceduralno generiranje korišteno za izradu modela građevina koje služe za izradu grada u kojem dron dostavlja pakete.

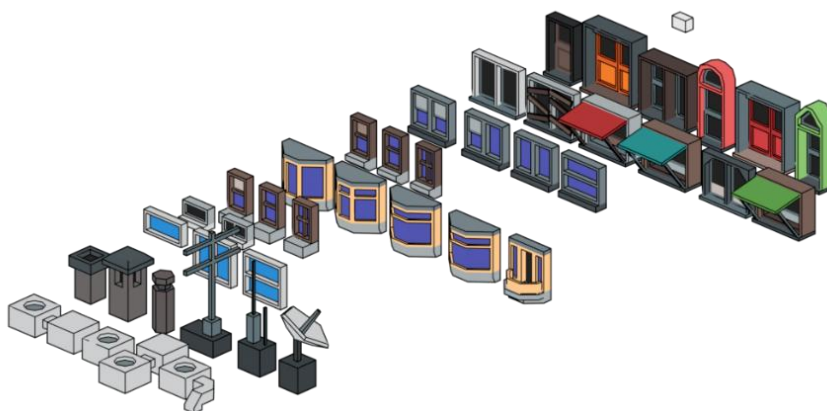
2.6.1 Geometrijski čvorovi

Korišteni alat za ostvarenje proceduralnog generiranja se zove *Geometry Nodes*, koji je dio programa *Blender*. Alat omogućuje generiranje i manipulaciju geometrije s pomoću vizualnog sučelja. Pravila i algoritmi za oblikovanje geometrije se definiraju s pomoću dijagrama funkcijskih čvorova (engl. *FBD - Function Block Diagram*). Svaki čvor (engl. *Node*) ima ulazne parametre, izvršava neku funkciju i vraća izlazne podatke. Čvorovi se međusobno spajaju poveznicama (engl. *Noodle*) i time se kontrolira tok podataka kroz algoritam. *Geometry Nodes* sadrži veliki broj gotovih funkcija namijenjenih za generiranje 3D modela, a i omogućuje korisniku da izradi svoje.



Slika 23: Primjer mreže čvorova alata *Geometry Nodes*

Prije nego što se napravi algoritam, treba analizirati problem i pronaći neke elemente koji se mogu lagano programski ostvariti tako da se taj dio posla napravi automatski. Svaka građevina ima iste elemente (prozori, ulazna vrata, boja zida,...), stoga se oni mogu proceduralno generirati. Time je jedina razlika između međusobnih građevina njihov početni oblik i parametri algoritma. Generiranje modela građevine se svodi na automatsko postavljanje detalja na površinu. Ručno je napravljena kolekcija modela za dodavanje detalja građevinama (vidljivo na slici broj 24). Svaka vrsta detalja ima nekoliko sličnih alternativnih modela tako da postoji više raznolikosti među građevinama. Gledajući sliku ispod s lijevo na desno: uređaji za ventilaciju, dimnjaci, antene, nekoliko vrsta prozora, dućanski izlozi i ulazna vrata.

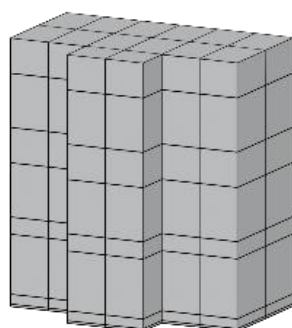


Slika 24: Ručno napravljeni detalji građevina

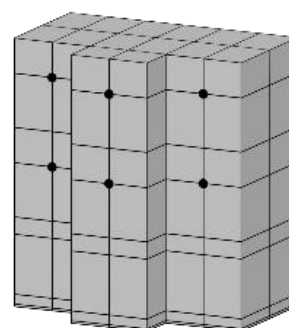
Dron.co	Verzija: 1.0
Tehnička dokumentacija	Datum: 29/01/26

2.6.2 Postupak izrade građevine

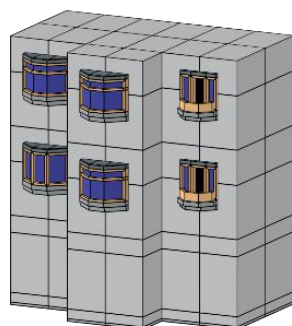
Postupak za izrade nove građevine se može podijeliti na pet koraka. Svi koraci su prikazani redom na slici broj 25. Izrada građevine započinje ručnim modeliranjem njezinog osnovnog oblika (Slika 25, Korak 1). Bitno je početnom obliku dodati dovoljno geometrije (točaka) za sljedeći korak. U sljedećem se koraku obavlja selekcija točaka na koje će se postaviti detalji građevine (Slika 25, Korak 2). Na temelju selekcije, algoritam postavlja detalje na površinu modela (Slika 25, Korak 3). Zatim se drugi i treći korak ponove za sve vrste detalja građevine (Slika 25, Korak 4). Konačno, u parametre algoritma se navedu dvije boje koje se iskoriste za stvaranje gradijenta na površini modela (Slika 25, Korak 5).



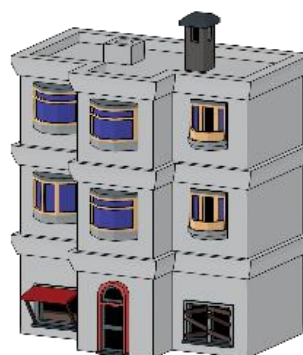
Korak 1



Korak 2



Korak 3



Korak 4



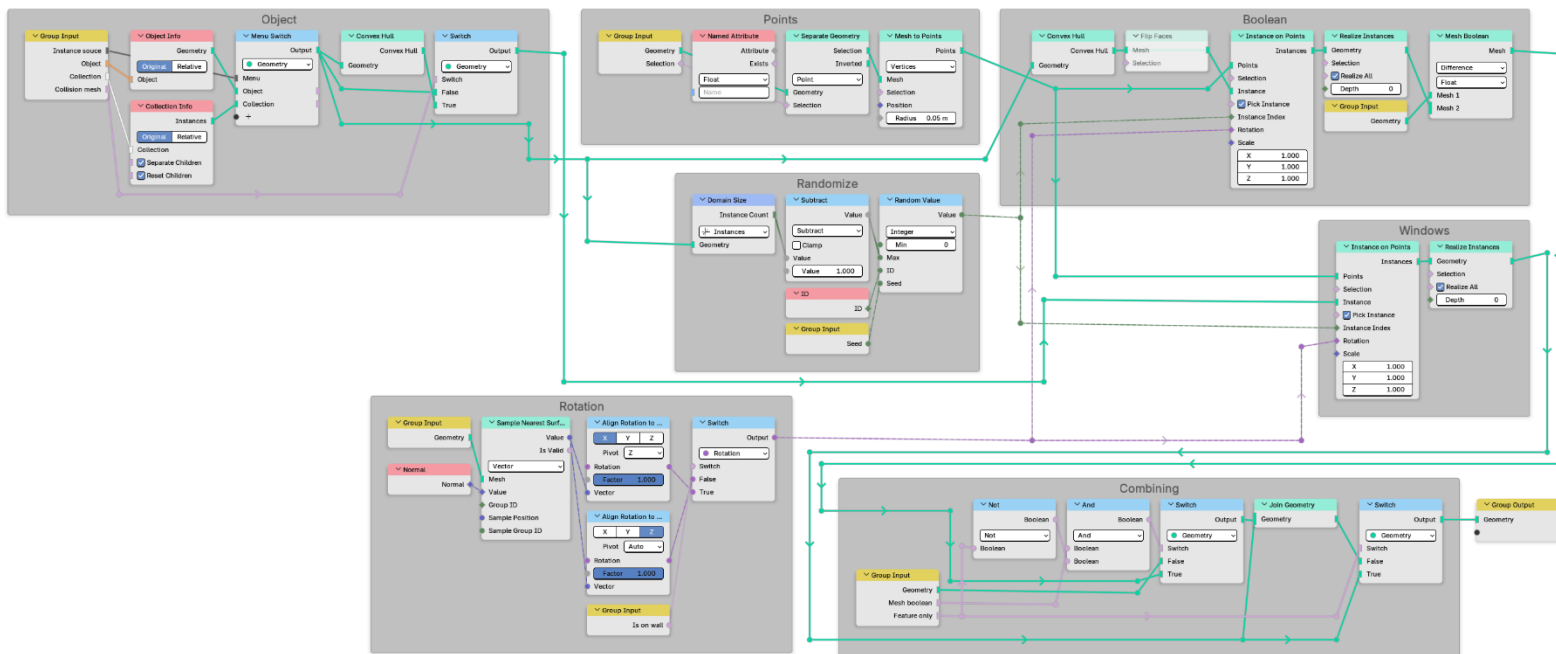
Korak 5

Slika 25: Postupak izrade i generiranja modela građevine

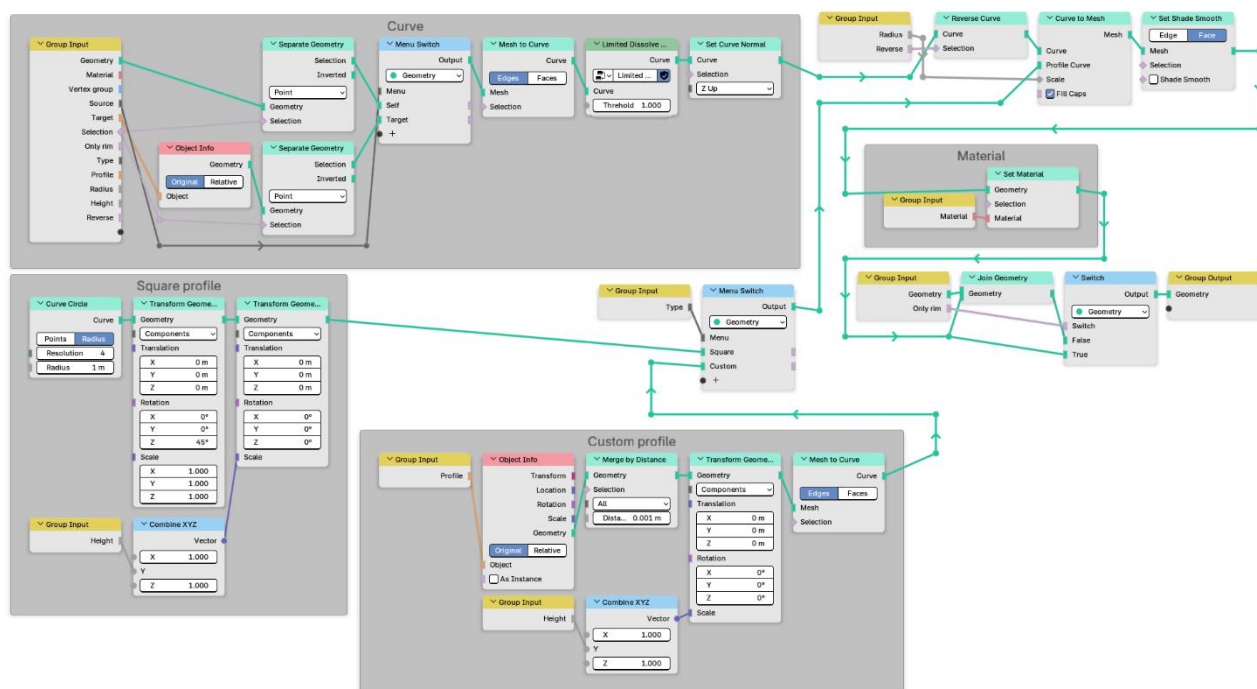
Dron.co	Verzija: 1.0
Tehnička dokumentacija	Datum: 29/01/26

2.6.3 Mreže čvorova za izradu građevina

Sljedeće slike prikazuju mreže čvorova koje implementiraju algoritam za generiranje građevina.

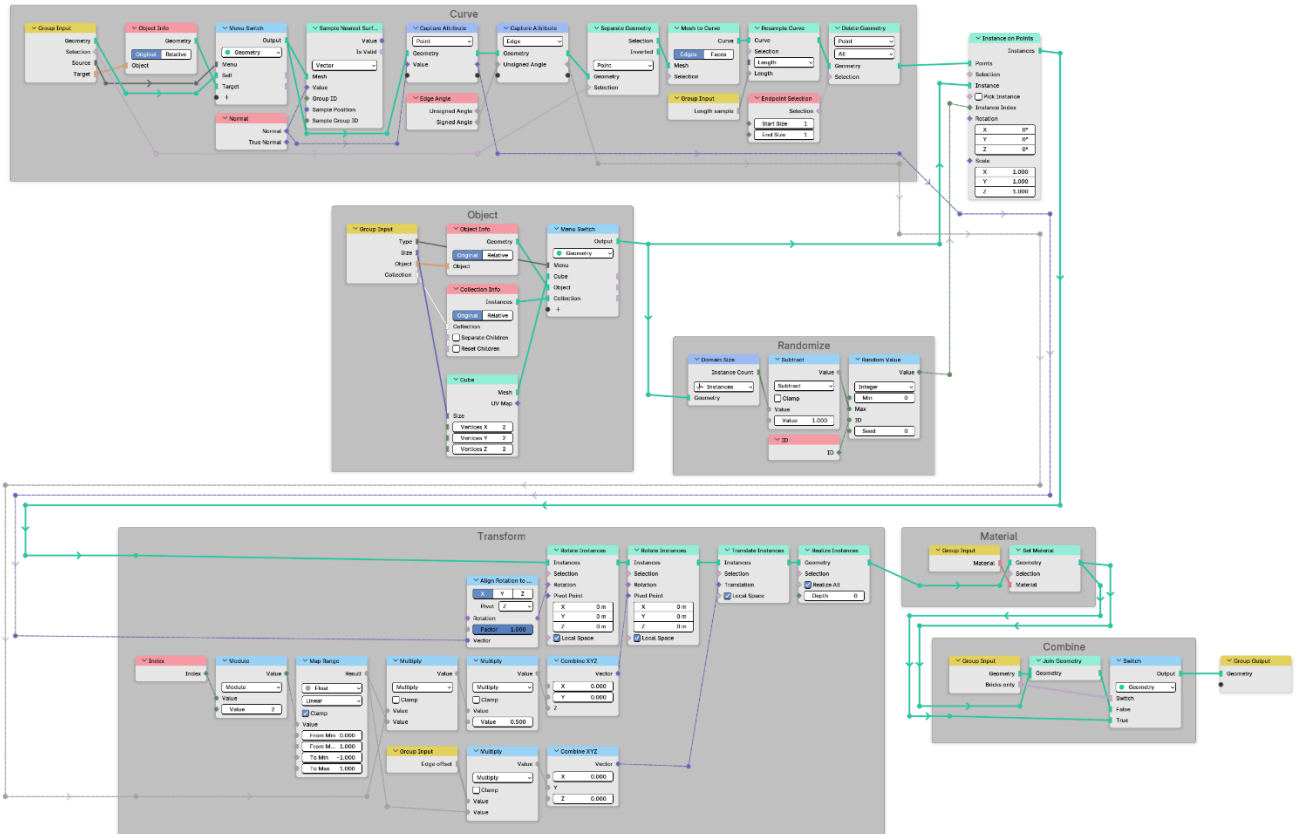


Slika 26: Mreža čvorova za generiranje detalja grada na zidu (prozori, vrata, ...)

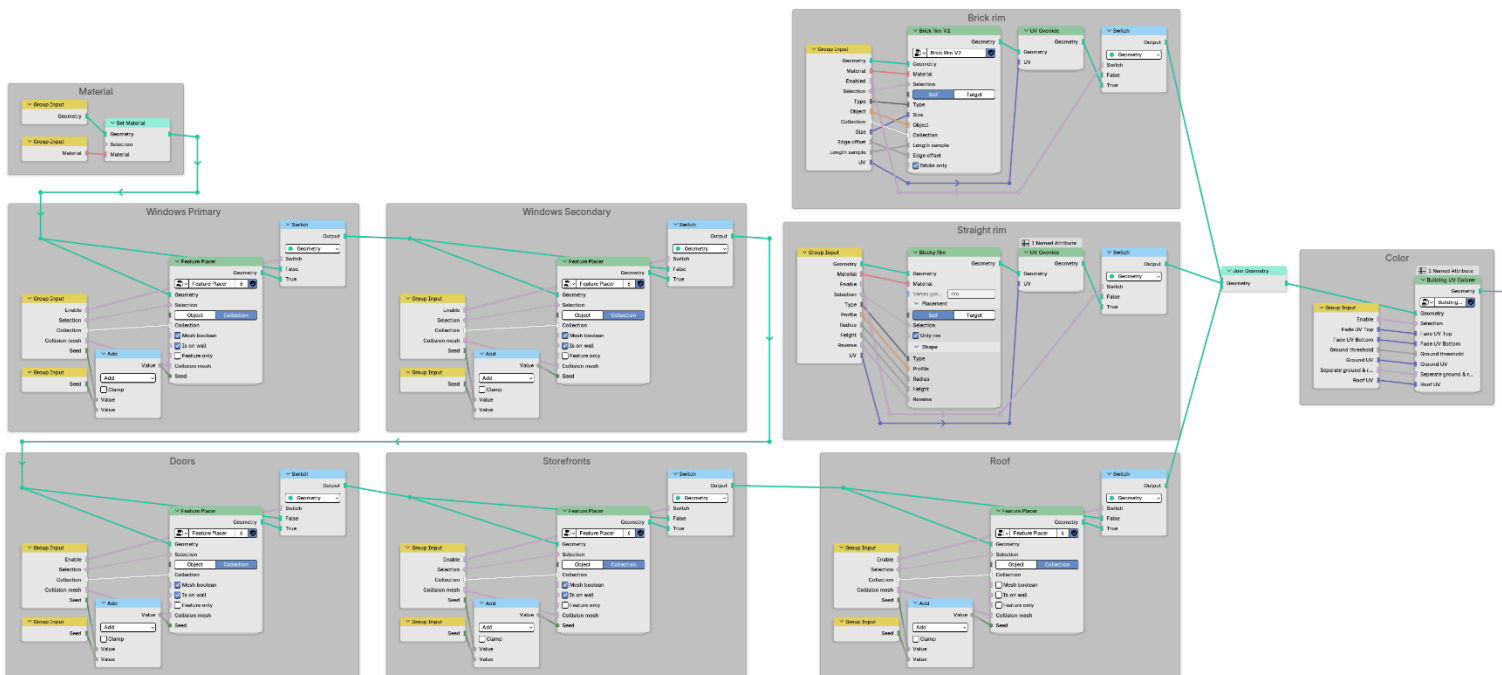


Slika 27: Mreža čvorova za generiranje horizontalnih okvira

Dron.co	Verzija: 1.0
Tehnička dokumentacija	Datum: 29/01/26



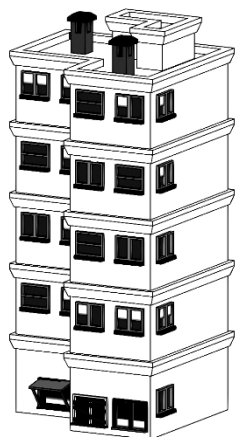
Slika 28: Mreža čvorova za generiranje cigla



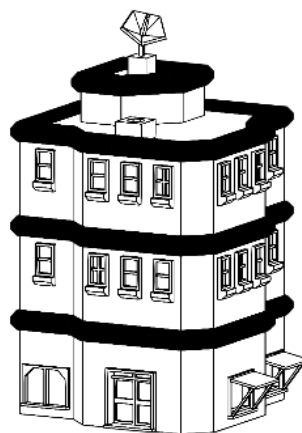
Slika 29: Mreža čvorova za spajanje svih dijelova u jedan model

Dron.co	Verzija: 1.0
Tehnička dokumentacija	Datum: 29/01/26

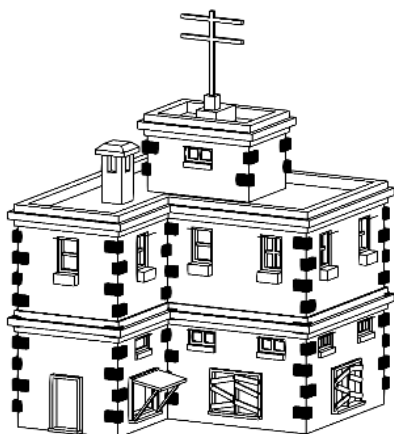
Iz prije navedenih slika čvorova se generiraju pojedini elementi građevine. Zadnje navedena mreža čvorova koristi sve druge mreže tako da ih spoji u jedan model koji se može lagano prenijeti iz *Blendera* u *Godot* za korištenje u projektu. Rezultati tih mreža su prikazani u slijedećim slikama crnom bojom i obojano za konačan rezultat.



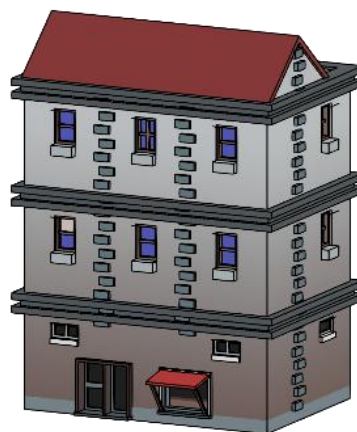
Slika 30: Detalji zida (označeno crno)



Slika 31: Horizontalni okviri (označeno crno)



Slika 32: Detalji cigla (označeno crno)

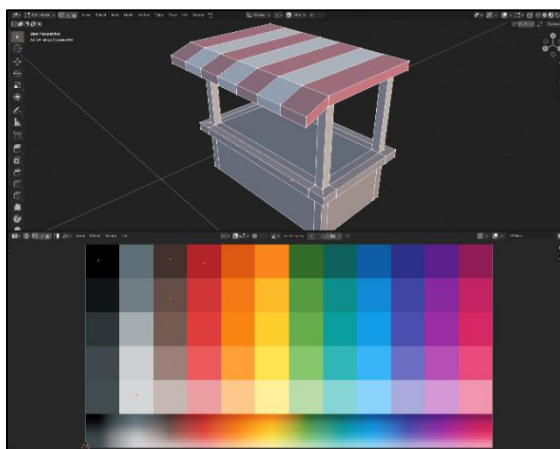


Slika 33: Svi dijelovi građevine spojeni u jedan model

Dron.co	Verzija: 1.0
Tehnička dokumentacija	Datum: 29/01/26

2.7 Vizualni efekti i stil

Za bojanje površina svih modela je korištena metoda zvanom *palette mapping*. Umjesto stvaranja više različitih materijala za različite vrste površina, *palette mapping* koristi samo jedan materijal s jednom teksturom. Ta tekstura se sastoji od osnovnih boja koje se koriste za bojanje modela i često se naziva *color palette*. Površine se zatim oboje mijenjanjem UV koordinata točaka poligona. UV koordinate su u računalnoj grafici dvodimenzionalne vrijednosti koje određuju na koji način se preslikava dvodimenzionalna slika na trodimenzionalnu površinu. U slučaju *palette mapping*, da bi se obojila neka površina specifičnom bojom iz teksture, naprave se sljedeći koraci. Prvo se izabrana boja pronađe na teksturi boja. Zatim se izabere površina kojoj se želi dodijeliti boju. U *UV editoru* u *Blenderu* se rubne točke te površine skaliraju s faktorom nula tako da svaka točka ima iste UV vrijednosti. Na kraju se točke površine pomaknu iznad željene boje. Prednost *palette mapping* u odnosu na standardno teksturiranje modela su jednostavnost (brzo uređivanje boja), performanse (postoji samo jedan glavni materijal i jedna tekstura) i stilistički efekt koji daje. Najveći nedostatak je manjak fleksibilnosti zbog toga što se koristi samo jedna tekstura.



Slika 34: Bojanje modela korištenjem metode *Palette mapping*

Kao posljedica korištenja navedene metode bojanja modela, površine su velikom većinom jednoboje. Radi postizanja malo više detalja na površinama i boljeg osjećaja dubine, dodan je efekt kontura. Efekt je postignut u *Godotu* s pomoću grafičkog programa sjenčara (engl. *shader*) koji se izvodi za svaki piksel iscrtane slike scene prije nego što se prikaže na ekranu. Programski kod koji se koristi za program je naveden u literaturi i koristi operator *Sobel* nad slikom dubine da se dobiju konture oblika.



Slika 35: Scena bez kontura



Slika 36: Scena s konturama

Dron.co	Verzija: 1.0
Tehnička dokumentacija	Datum: 29/01/26

Za vizualno naglašavanje kritičnih stanja u igri koriste se dva postprocesna efekta: efekt šuma i efekt desaturacije. Efekt šuma implementiran je s pomoću sjenčara koji generira dinamički, pikselizirani šum baziran na vremenu, čime se postiže dojam nestabilnosti i tehničkih smetnji u prikazu slike. Šum se prikazuje kao poluprozirni sloj preko ekrana.

Efekt desaturacije koristi ugrađenu *saturation* varijablu unutar *Environment nodea* te ne zahtijeva poseban sjenčar. Zasićenost boja se linearno smanjuje ovisno o stanju baterije drona, pri čemu se pri niskim vrijednostima baterije slika postupno pretvara u gotovo crno-bijeli prikaz.

Prilikom uspješne dostave paketa aktivira se vizualni efekt konfeta koji služi kao jasna povratna informacija igraču. Konfeti ostavljaju efekt traga u različitim bojama dok se kreću.



Slika 37: Efekt šuma



Slika 38: Efekt desaturacije



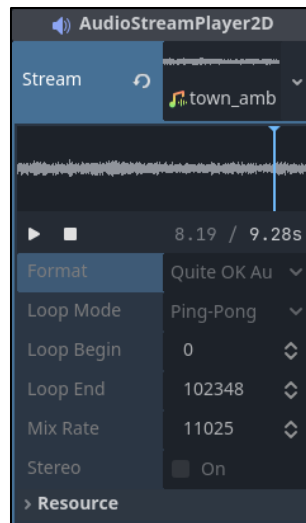
Slika 39: Pojava konfeta nakon dostavljanja paketa

Dron.co	Verzija: 1.0
Tehnička dokumentacija	Datum: 29/01/26

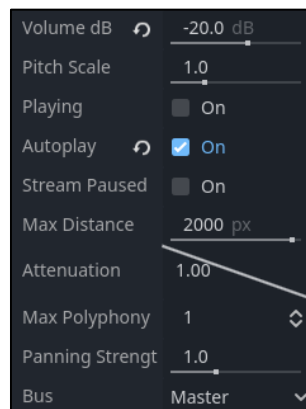
2.8 Audio efekti

Zvučni efekti služe za poboljšanje korisničkog iskustva i pružanje zvučne povratne informacije tijekom interakcije sa sustavima igre. Sljedeći sustavi koriste zvučne efekte: gumbi korisničkog sučelja, pozadinski šum, vjetar, propeleri drona, punjenje baterije drona, ukopčavanje i iskopčavanje paketa te dostava paketa.

Zvučne efekte koji mogu trajati neodređeno vrijeme potrebno je podesiti tako da se ne primijeti trenutak ponovnog pokretanja zvuka. Primjeri takvih zvučnih efekata su pozadinski šum i zvuk propelera, pri čemu je potrebno omogućiti neprimjetno ponavljanje zvučnog zapisa duljine nekoliko sekundi. To se postiže postavljanjem opcije *Loop Mode* na *Forward*, *Backward* ili *Ping-Pong*, ovisno o načinu na koji je zvučna datoteka dizajnirana. U praksi se pokazalo da opcija *Ping-Pong* daje najbolje rezultate za ponavljajuće zvukove korištene u ovom projektu.



Slika 40: Primjer postavki učitanoog zvuka koji se ponavlja

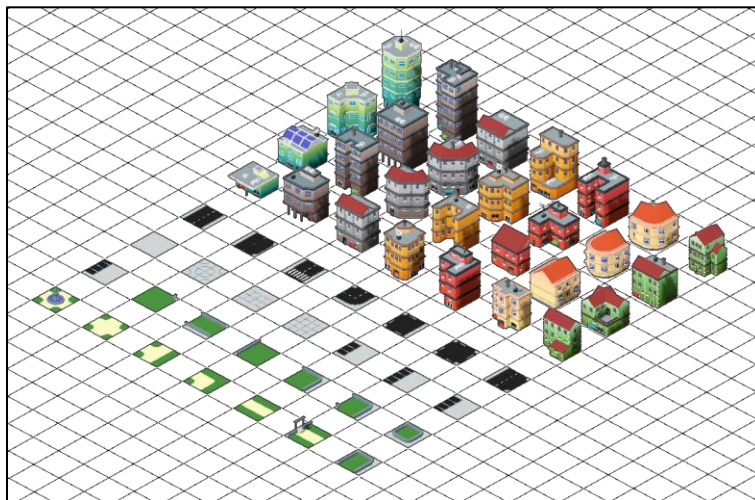


Slika 41: Primjer postavki zvučnog objekta pozadinskog zvuka u Godot Inspectoru

Dron.co	Verzija: 1.0
Tehnička dokumentacija	Datum: 29/01/26

2.9 Dizajn razine igre

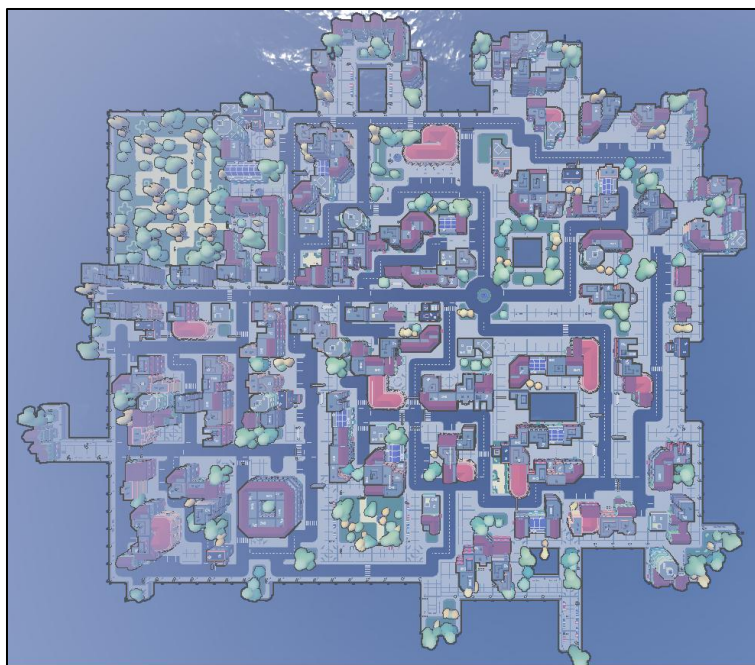
Razina igre u projektu Dron.co stilizirani je grad. Za izgradnju grada je korišten alat *GridMap* unutar *Godota*. *GridMap* služi za efikasno popunjavanje velikih prostora ponavljajućim modularnim dijelovima. Glavne prednosti korištenja alata u odnosu na klasično ručno postavljanje modela u scenu su znatno brža izgradnja i iteracija razina, automatsko poravnanje elemenata u pravokutnu mrežu i bolja optimizacija. Oblici modela građevina i cesta čine ovaj alat idealnim za izgradnju i dizajniranje grada.



Slika 42: Prikaz modela na postavljenih u GridMap alatu

2.9.1 Urbanističko planiranje i organizacija grada

Grad u projektu Dron.co nije statična kopnena struktura, već je osmišljen kao plutajući kompleks koji se sastoji od niza kvartova, cesta, parkova, trgova, čak i kulturnih objekata. Činjenica da grad pluta na vodi pridodaje njegovom jedinstvenom vizualnom dizajnu. Tlocrt grada prikazan je na slici 43.

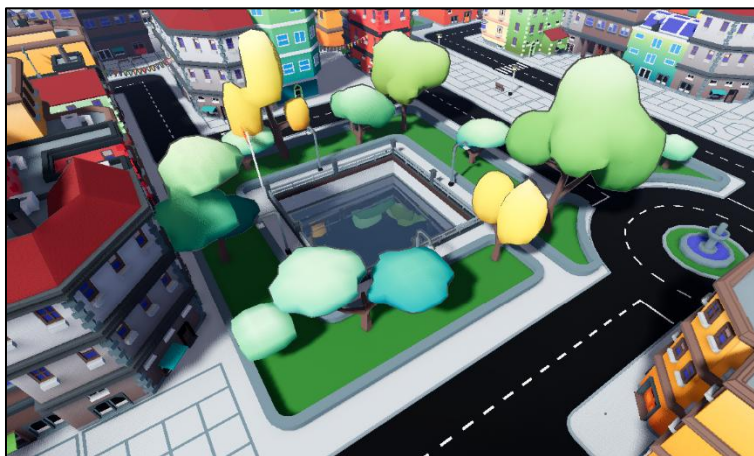


Slika 44: Tlocrt grada slikan odozgo

Dron.co	Verzija: 1.0
Tehnička dokumentacija	Datum: 29/01/26

Dizajn grada u projektu Dron.co osmišljen je temeljeći se na urbanističkoj logici modernih gradova uz naglasak stavljen na složen raspored zgrada i detalja zbog što zanimljivijeg igračeg iskustva. S obzirom na to da igrač kontrolira dron kroz dizajnirani grad te se mora snalaziti u prostoru, ali ne olako, grad je dizajniran sa što više moguće prepreka, bez da mu je narušena estetika.

Osnovni raspored grada prati kvadratni oblik, ali ta se geometrija razbija dodavanjem poluotoka i izbočenih dijelova kvartova. Ovakve se promjene u pravilnosti tlocrta grada odražavaju na zanimljiviji dizajn razina igre i više mogućnosti otežavanja igre. Uvedene su prvenstveno zbog postizanja vizualne zanimljivosti scene te razbijanja monotonije koju grid sustav može stvoriti, čime je postignuta raznolikost cijelog dizajna.



Slika 45: Prikaz raznolikog plana grada, detalj parka s "jezerom"

Unutar samo strukture, građevine su smisleno podijeljene u kvartove koji su povezani sustavom cesta, nogostupa, pješačkih prijelaza. Prilikom dizajniranja naglasak je stavljen i na prepoznatljivost dijelova grada od kojih je nekoliko tematski obilježeno samo jednim stilom građevina, na primjer zelena četvrt (Slika 45) ili lokacijom velikih parkova, dok je ostatak grada povezan na smislen i vizualno privlačan način. Na ovaj način, ostvaruje se bogato vizualno okruženje, a igrač može steći osjećaj snalaženja u gradu nakon nekoliko dostavljenih paketa, ali ne dovoljno da igru učini manje izazovnom.



Slika 46: Zelena četvrt

Dron.co	Verzija: 1.0
Tehnička dokumentacija	Datum: 29/01/26

Osim stambenih elemenata grada, u tlocrt su integrirana dva parka te nekoliko trgova i zona za odmaranje, druženje. Cilj dizajniranja grada, osim dizajniranja prikladne razine za dostavljanje paketa u igri, bio je i osmisliti urbanističko središte namijenjeno ljudima. Kazalište je u tlocrt dodano kako bi pridonijelo raznolikosti tlocrta te jedinstvenosti razine.



Slika 47: Kazalište

2.9.2 Sjedište kompanije i punionice

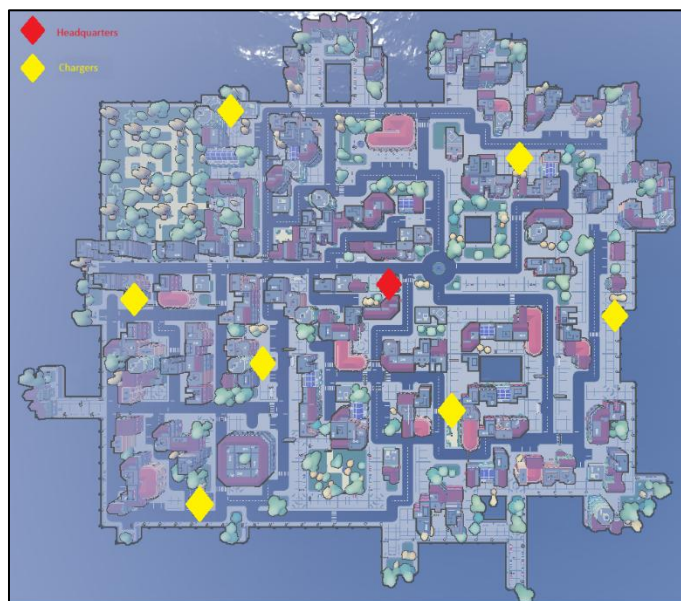
Pri završetku kreiranja rasporeda građevina i cesta razine igre, u obzir se moraju uzeti zgrade točki interesa: sjedište kompanije *Headquarters* i punionice *Chargers*.



Slika 48: Tlocrt grada bez Headquarters i Chargers objekata

Dron.co	Verzija: 1.0
Tehnička dokumentacija	Datum: 29/01/26

Headquarters objekt morao je biti smješten u sredini tlocrta grada kako bi se ostvarila konzistentnost u razini igre. Lokacije *Chargers* objekata određene su po udaljenosti od *Headquarters* zgrade te po odgovarajućoj međusobnoj udaljenost. Cilj je bio postaviti dovoljan broj objekata za punjenje kako se ne bi prouzročio osjećaj frustracije u igračima pri pražnjenju baterije. Određene lokacije objekata točki interesa prikazane su na Slici 48.



Slika 49: Tlocrt grada s lokacijama *Headquarters* i *Chargers*



Slika 50: Lokacija *Headquarters*

Kombinacijom različitih tipova građevina i pažljivim planiranjem zelenih površina unutar modularne mreže, postignuta je visoka razina detaljnosti koja doprinosi uvjerljivosti i atmosferi cijelog projekta. Daljnja ideja za razvitak igre mogla bi sadržavati dizajniranje više razina drugačijih težina kako bi ostvarili raznolikost i razbili monotoniju jednog grada.

Dron.co	Verzija: 1.0
Tehnička dokumentacija	Datum: 29/01/26

2.9.3 Detalji i prepreke

U planu grada, osim građevina, cesta, parkova i pješačkih zona, nalaze se i brojni detalji (Slika 50). Detalji su uvedeni kako bi se gradu dodao osjećaj prirodnosti i živosti te dodao više dimenzija osim samo cesta i zgrada. Kreirani u programu *Blender*, preneseni su u *Godot* te su također integrirani u *GridMap* sustav kao entiteti.



Slika 51: Detalji grada

Detalji obuhvaćaju sve od ulične rasvjetе i semafora do tegla s cvijećem, stanica za bus, fontanu i grmlje, a ključni su za postizanje realističnog dojma naseljenog grada. Svaki je element modeliran u stiliziranom stilu, kao i grad, čime se osigurava konzistentnost u stilu igrice.

Prirodni elementi razbijaju stroge linije arhitekture građevina te su zbog te svrhe oni najbitniji detalji igrice. Drveće postavljeno između zgrada, na parkirališta, oko šetališta, sva imaju svoju svrhu te pružaju prirodan osjet gradu što ga čini realističnim i boljim od neprirodnog, beživotnog okruženja. Stabla i grmlje nisu postavljani samo kao ukras, već služe i kao prirodne prepreke koje igrač mora izbjegavati tijekom leta, čime dizajn okoliša izravno utječe na mehaniku igranja i navigaciju drona. Živost grada dodatno je naglašena elementima koji dokazuju ljudsku prisutnost i aktivnost, poput trga sa šarenim štandovima, fontana, klupa i zastavica (Slika 51). Ovi elementi postavljani su u pješačke zone i nogostupe kako bi se naglasila namjena tih prostora, ali i kako bi pružili zabavne i vizualno atraktivne prepreke za dron u razini.



Slika 52: Zastavice kao prepreka i ukras

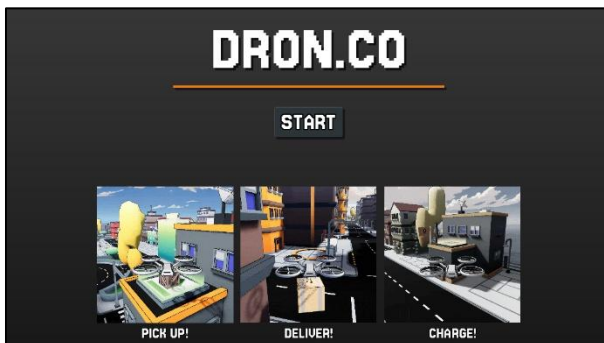


Slika 53: Reklame (Billboards) kao detalj i prepreka

Dron.co	Verzija: 1.0
Tehnička dokumentacija	Datum: 29/01/26

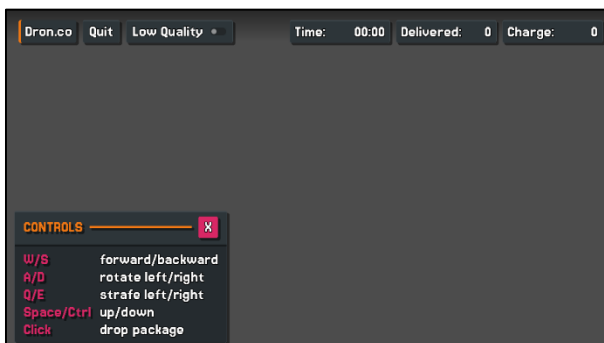
2.10 Korisničko sučelje

Korisničko sučelje u Dron.co je ostvareno s pomoću gotovih alata u *Godot* zvanim *Control nodes*. Dijeli se u tri glavna dijela: početni izbornik, sučelje tijekom igre i izbornik kraja igre. Sučelje služi za pojednostavljeno upravljanje aplikacijom i za davanje informacija korisniku. Početni izbornik (Slika 53) sadrži naslov projekta, gumb za početak i tri animirane slike koje ukratko objasne kako igrati.



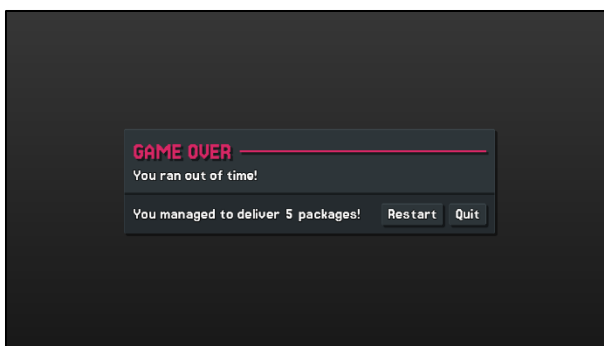
Slika 54: Početni izbornik

Sučelje tijekom igre (*engl. Heads Up Display - HUD*) ima tri interaktivna elemenata: zatvarajući izbornik za opis kontrola, gumb za izlaz iz aplikacije i gumb za odabir grafičke kvalitete. Opcija za odabir grafičke kvalitete je dodana za slabija računala koja ne mogu dovoljno brzo prikazivati vizualne efekte. Odabirom „niska kvaliteta“ se smanji kvaliteta sijena, isključi se efekt kontura i isključe se refleksije na površini vode. Nadalje, sučelje ima još tri neinteraktivna elemenata koji prikazuju ključne informacije za tijek igre: preostalo vrijeme, broj dostavljenih paketa i preostali naboj (*battery charge*) u bateriji drona.



Slika 55: Izbornik tijekom igre

Zadnje važno sučelje je izbornik kraja igre (Slika 56). Taj izbornik se pojavi kada dron na neki način umre, istekne svo vrijeme ili se izgubi paket. Sučelje prikazuje poruku zašto je igra završila, broj dostavljenih paketa tijekom runde, gumb za resetiranje i gumb za izlaz iz aplikacije.



Slika 56: Izbornik kraja igre

Dron.co	Verzija: 1.0
Tehnička dokumentacija	Datum: 29/01/26

3. Upute za korištenje

3.1 Kako igrati

Igrač preuzima kontrolu nad dronom i njegov je glavni zadatak preuzimati pakete u glavnom sjedištu dostavne kompanije te ih dostavljati na označene lokacije u gradu. Paket se preuzima tako da se dron pozicionira iznad njega, pri čemu se zakači paket. Nakon preuzimanja, igrač se mora uputiti prema lokaciji dostave, koju može pratiti pomoću hologramske strelice iznad sebe. Tijekom leta igrač mora pažljivo upravljati dronom, izbjegavati prepreke u gradu i voditi računa o visini leta, jer prevelika visina može negativno utjecati na stabilnost drona i paketa. Posebnu pozornost potrebno je posvetiti razini baterije, koja se neprestano troši. Ako baterija padne na kritičnu razinu, aktiviraju se vizualni efekti koji upozoravaju igrača, a ako se potpuno isprazni, igra završava. Kako bi to spriječio, igrač mora redovito koristiti stanice za punjenje raspoređene po gradu. Kada dron stigne do označene lokacije, igrač ispušta paket. Ako paket uđe u dostavno područje, dostava se smatra uspješnom i povećava se broj dostavljenih paketa. Nakon toga se u sjedištu stvara novi paket te započinje sljedeći krug igre. Cilj igre je u zadanom vremenu dostaviti što veći broj paketa.

3.2 Kontrole

Igra omogućuje kontroliranje drona preko tipkovnice ili kontrolera.

Tipkovnica

- Kretanje naprijed: **W**
- Kretanje unatrag: **S**
- Rotacija lijevo: **A**
- Rotacija desno: **D**
- Kretanje lijevo: **Q**
- Kretanje desno: **E**
- Kretanje prema gore: **Space**
- Kretanje prema dolje: **Ctrl**
- Ispuštanje paketa: **Klik lijeve tipke miša**
- Izlaz iz igre: **Escape**

Kontroler

- Rotacija lijevo / desno: **Lijeva analogna palica lijevo / desno**
- Kretanje gore / dolje: **Lijeva analogna palica gore / dolje**
- Kretanje lijevo / desno: **Desna analogna palica lijevo / desno**
- Kretanje naprijed / unatrag: **Desna analogna palica gore / dolje**
- Ispuštanje paketa: **Donja akcijska tipka (A / X)**

3.3 Izvršna datoteka i izvorni kod

Izvršna datoteka Dron.co se može preuzeti s interneta preko navedene poveznice:

<https://github.com/j-ramljak/Dron.co/releases>.

Izvorni kod rada je dostupan na GitHub preko slijede poveznice:

<https://github.com/j-ramljak/Dron.co>.

Dron.co	Verzija: 1.0
Tehnička dokumentacija	Datum: 29/01/26

4. Literatura

- [1] Juan L. & Ariel M., Godot Engine documentation, 2014, <https://docs.godotengine.org/en/stable/> [28.1.2026.]
- [2] Peter Höglund, Shaders – Godot Shaders, 2026, <https://godotshaders.com/> [28.1.2026.]
- [3] Knut Sveidqvist, Tyler Long, Mermaid Live Editor, 2017, https://mermaid.ai/play?utm_source=mermaid_live_editor#pako:eNqrVkrOT0lVslJSqgUAFW4DVg [28.1.2026.]
- [4] Curtis Tsang, Work Breakdown Structure Software, 2001, *Visual Paradigm*, <https://online.visual-paradigm.com/diagrams/features/work-breakdown-structure-software/> [28.1.2026.]
- [5] LoudFlameLava, Sobel Outline Postprocess Shader, 1.9.2024, *Godot Shaders*, <https://godotshaders.com/shader/sobel-outline-shader/> [28.1.2026.]
- [6] Team FreeStylized, Skybox 22, 2025, https://freestylied.com/skybox/sky_22/ [28.1.2026.]
- [7] Malido, Absorption Based Stylized Water, 18.10.2024, *Godot Shaders*, <https://godotshaders.com/shader/absorption-based-stylized-water/> [28.1.2026.]
- [8] Todd Carpenter, Royalty-free sound effects for download, 2025, <https://pixabay.com/sound-effects/> [28.1.2026.]