

3D-Model Pergol

Tehnična dokumentacija

(Projektna naloga pri predmetu Projektni seminar)

Hasan Mahmutagić, Boris Turudija

Opredelitev problema

Vsi smo bili v situaciji, ko smo želeli kupiti nekaj stvari za našo hišo in smo zato imeli največja pričakovanja, vendar ko gre za sam nakup in z namestitvijo le-teh na koncu nismo zadovoljni, ker ni tako, kot smo si zamislili. Najboljši primer tega je nova terasa za hišo, kjer želi naročnik svoj denar vložiti v novo teraso, delodajalec pa izpolniti želje kupca in zaslužiti. Težava nastane, ko kupec in naročnik nimata enakih pogledov, zato kupci zaradi teh razlogov opustijo nakup, prodajalci pa ostanejo brez prihodkov.

Rešitev tega problema vidimo v aplikaciji, ki bi omogočala lažjo komunikacijo med uporabniki in delodajalci. Takšna aplikacija bi uporabnikom omogočila izdelati 3d model po svojih željah glede na ponudbo materialov in možnosti prodajalca. Aplikacija bi uporabniku na podlagi 3d modela omogočila vizualni videz zelenega, prodajalcu pa bi to olajšala. vse tehnične specifikacije in vizualne želje naročnika, na podlagi tega bi v prvi vrsti dobili boljšo komunikacijo med naročnikom in prodajalcem, večje povpraševanje in s tem bi dobili zadovoljstvo na obeh straneh.

Kot smo rekli, želimo sistem, s katerim bosta zadovoljni obe strani. Sistem mora biti enostaven za končne uporabnike, da omogoča varčevanje vse podatke tako s strani kupca kot s strani prodajalca ter vse ostale funkcionalnosti, ki so pomembne za rešitev tega problema.

Študija izvedljivosti

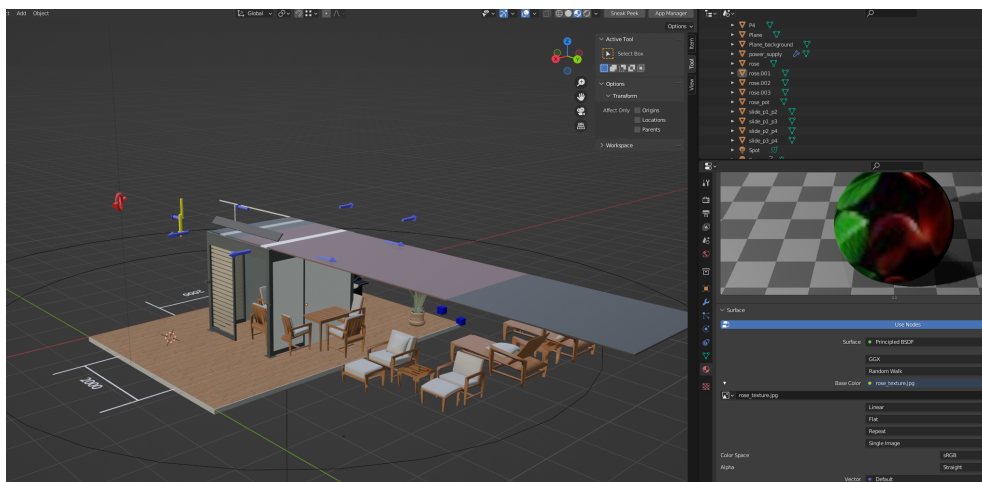
V tehničnem smislu smo morali narediti aplikacijo, ki se je morala povezati z obstoječo bazo podatkov (iz katere bi dobili vse parametre: širine, visine, materiale, barve, dodatne elemente...), in ki poleg same izdelave modela potem shranijo model v PDF dokumente. Specifična težava ki sva se srečala z njo je da v ni mogoče spremeniti velikost elementa

tako da določimo novo dolžino, ampak se element mora skalirati za določeno vrednost da bi se prejšnja velikost spremenila na novo.

Izbrana ordoja in postopek izdelave

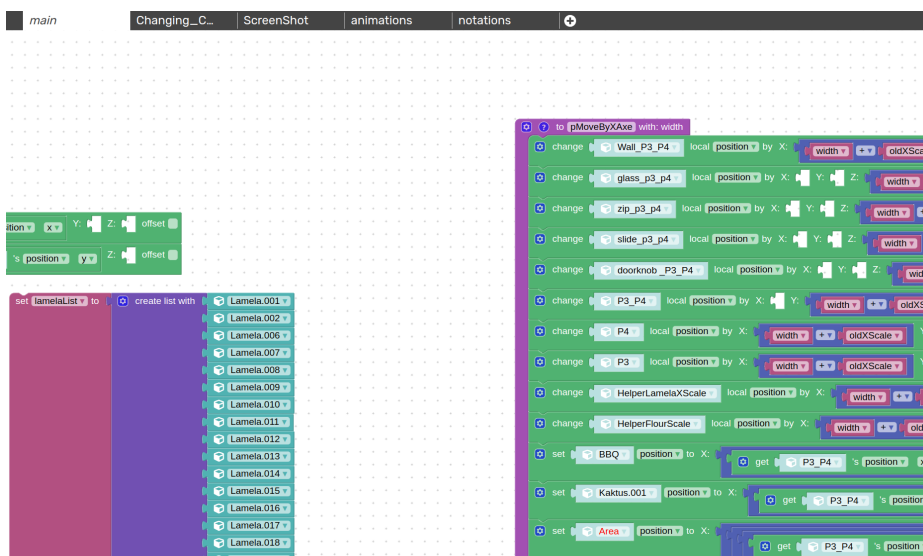
Prvobitni plan za izdelavo aplikacije je bil da se uproblja TypeScript z Angular orodjem.

Osnovna modela je izdelana (narisana) v Blenderju, programu za risanje animacij, visualnih efektov, 3d modelov... Dostop do Blenderja i več o samem programu se lahko dobi na [linko](#). [Primer](#) različnih modelov narisanih v Blenderju se lahko dobijo na linko.



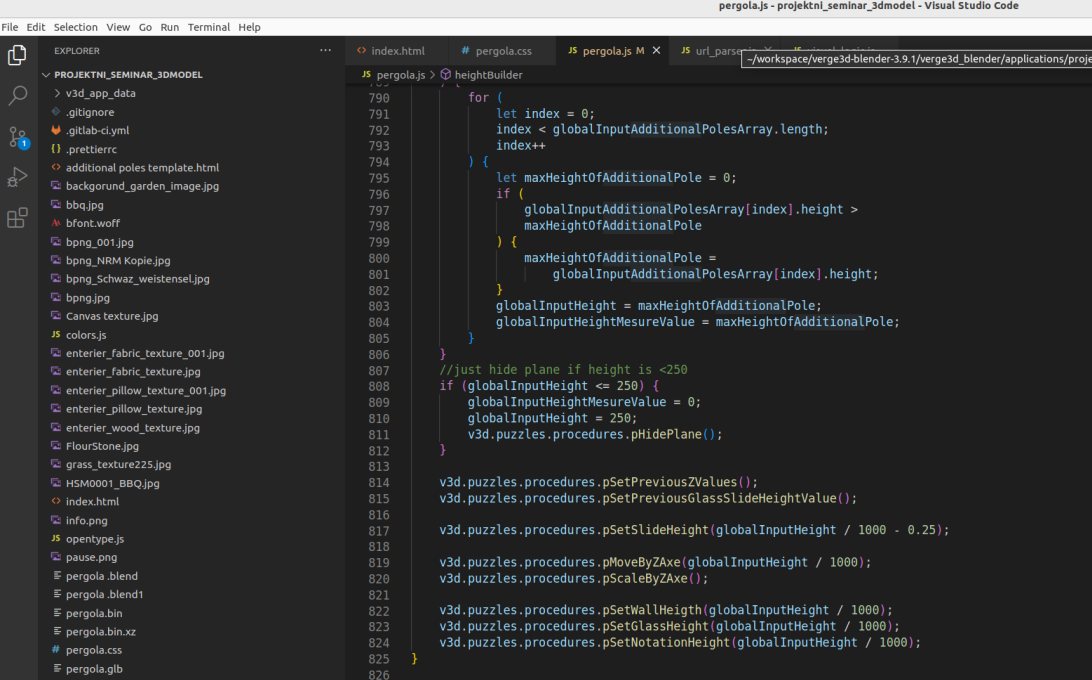
Slika 1: .blend file

Potem je potrebno Blender raširiti z Verge3d-jem, ki je orodje za renerovanje, kreiranje in spememebo 3d modelov u realnom vremenom. Za nas je bilo idealno za lažjo manipulacijo ozirom komunikacijo med .blend fajlom in samim JavaScript-om. Vec o Verge3d-ju se lahko dobi na [linko](#). Soft8soft [primer](#) uporabe Verge3d-a lahko vidimo na linku.



Slika 2: Verge3d visual logic

Posledično smo imali težave, oziroma dodatne komplikacije pri povezavi TypeScript-a z Verge3D-jem. Kljub tem da za sam projekt ni bilo pomembno sva se odločila funkcionalnosti izdelati v čistemu JavaScript-u
Uporabniški interfejs je narajen v HTML-u, z dodatkom Angular Material-sa, ter CSS-a.



```
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
}

for (
  let index = 0;
  index < globalInputAdditionalPolesArray.length;
  index++
) {
  let maxHeightOfAdditionalPole = 0;
  if (
    globalInputAdditionalPolesArray[index].height >
    maxHeightOfAdditionalPole
  ) {
    maxHeightOfAdditionalPole =
      globalInputAdditionalPolesArray[index].height;
  }
  globalInputHeight = maxHeightOfAdditionalPole;
  globalInputHeightMeasureValue = maxHeightOfAdditionalPole;
}

//just hide plane if height is <250
if (globalInputHeight <= 250) {
  globalInputHeightMeasureValue = 0;
  globalInputHeight = 250;
  v3d.puzzles.procedures.pHidePlane();
}

v3d.puzzles.procedures.pSetPreviousZValues();
v3d.puzzles.procedures.pSetPreviousGlassSlideHeightValue();

v3d.puzzles.procedures.pSetSlideHeight(globalInputHeight / 1000 - 0.25);

v3d.puzzles.procedures.pMoveByZAxis(globalInputHeight / 1000);
v3d.puzzles.procedures.pScaleByZAxis();

v3d.puzzles.procedures.pSetWallHeight(globalInputHeight / 1000);
v3d.puzzles.procedures.pSetGlassHeight(globalInputHeight / 1000);
v3d.puzzles.procedures.pSetNotationHeight(globalInputHeight / 1000);
```

Slika 3: Del .js kode

Celi projekt je bilo potrebno še povezati z obstoječim portalom narajenim v Angular-ju. Ter še potem avtomatski generirane posnetke ekrana poslati v bazo.

Projekt je postavljen v DigitalOcan sever kateri skrbi za dostop in izdelavo modela v vsakemu momentu. Dostop do projekta se lahko dobi preko [linka](#).

S pregledom zakona lahko ugotovimo, da naša prihodnja uporaba ne presega meja katerega koli pravila/zakona Republike Slovenije in EU.

Vsi potrebni podatki, ki jih mora uporabnik posredovati, so v skladu z zakonodajo, kot tudi varovani v skladu z zakonom o varstvu podatkov.

Ekonomsko je projekt zelo izvedljiv in stroškovno učinkovit, ker ne potrebujemo posebne opreme, tj. dodatna naložba v opremo, razen serverja ki ga je podjetje že imelo.

Projekt je zaradi svoje preprostosti zelo sprejemljiv za uporabo, predvsem zato, ker ne zahteva novih prilagoditev ali pa novih znanj, ki bi bila potrebna za uporabo aplikacije.

Funkcijske in ne funkcijske zahteve novega sistema

Uporaba sistema je namenjena samo za ponudnika. Se pravi da končni izdelek ki pride do stranke ne riše sama stranka ampak prodajlec. Za to smo se odločili ker je izdelava enega takega produkta dosti kompleksna in smo skbeli da stranke same ne bi niti mogle izdelati realan model. Potrebno je poznavati same produkte, kaj oni nudijo, njihove omejitve.

Pomembno kazati da v samo aplikacijo ni nobenega logiranja, ali shranjivanja podatkov. To so funkcionalnosti za katere skrbi portal.

Funkcijske zahteve ki jih bo morala omoguciti za ponudnika:

- Osnovna funkcionalnost sistema je da prebere vhodne parametre iz portala in z njim konstruira 3D model Pergole, po tem naredi posnetke zaslona iz tri različna kota.

Ki se potem v portali vpisejo v pdf in shranijo v bazo.

Obvezni parametri ki se morajo upoštevati so: dolžina, širina, barve. Potem so tukaj se dodatne možnosti inštalacije stebrov, sten, stekla, zipa, ter sami parametri dodatnih elementov...

-Upravljanje modela: Ponudnik mora imati možnost manipulacije karakteristika modela. To pomeni da ima možnost spremembe osnovnih parametrov dobijenih iz portala, samim tim spremembo modela.

- Aplikacija omogoča interackcijo, ozirma vizualno manipulacijo same pergole. Kaj pomeni da uporabnik pegolo lahko rotira, prestavlja, doda pozadinsko slikco, spremeni pozicijo sonca/sence ne glede na njene parametre. Na koncu potem lahko naredi dodatni screenshot po svoji želji..

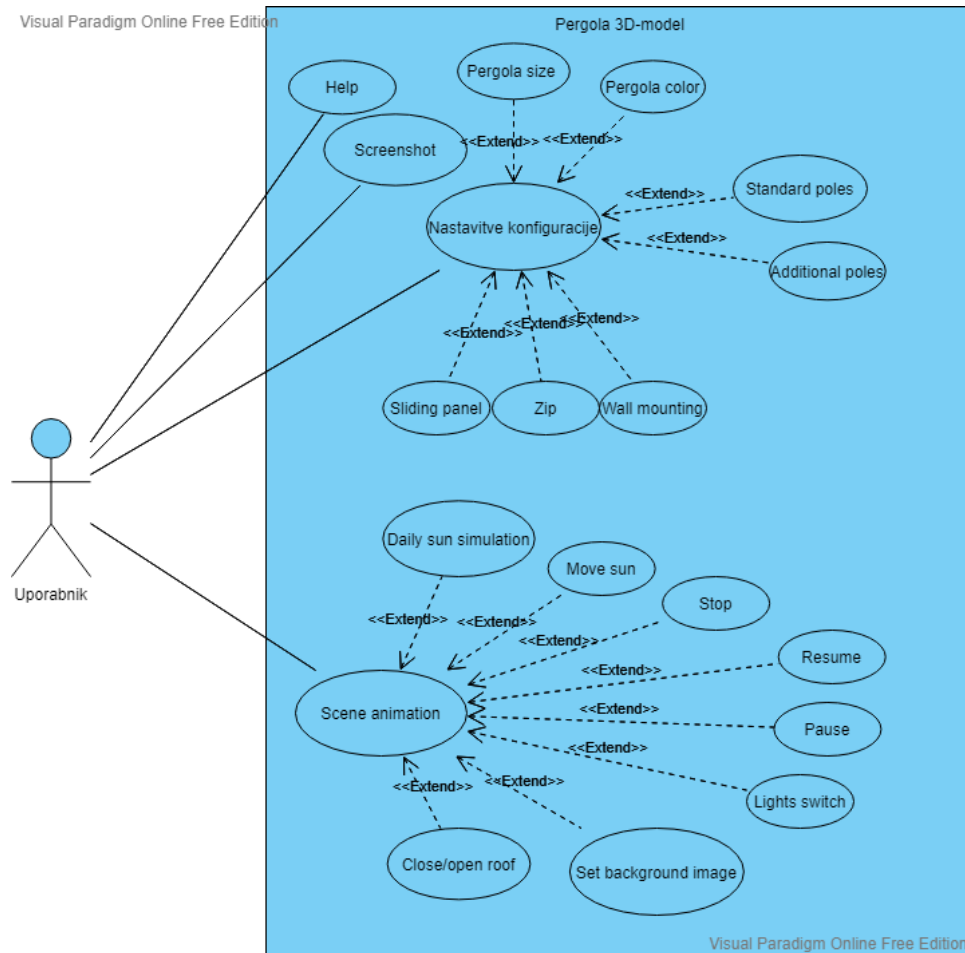
- Uporabnik tudi ima možnost spreminjanja vseh vhodnih parametrov.

Nefunkcijske zahteve novega sistema

1. Sistem mora biti sposoben obdelati do 1000 oddaj na minuto.
2. Sistem bo moral omogociti rezultate 1000 iskanja na minuto
3. Sistem moram omogociti pohranu podatkov in zastito
4. Sistem mora podpirati dva tipa uporabnikov: ponudnik in izvajalec
5. Sistem mora zagotvaljati 24/7 delovanje s procentom časa aktivnega delovanja (up-time) minimalno 95% .
6. Sistem naj bo namestljiv kot spletna aplikacija. Namestitev in konfiguracijo je potrebno omogočiti preko avtomatiziranih namestitvenih skript.
7. Sistem mora vključevati dokumentacijo za namestitev, vzdrževanje in integracijo sistema.

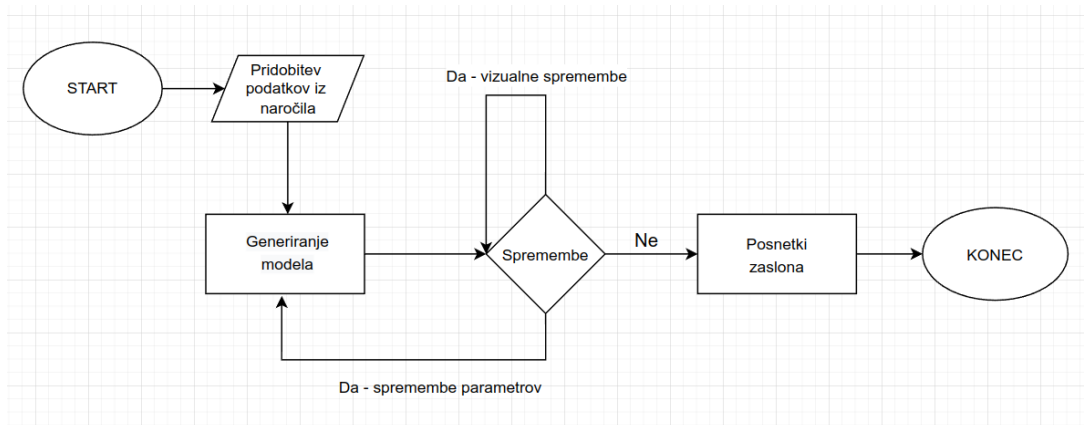
Uporabniški diagram opisuje možnosti aplikacije ki jih ponuja uporabniku. Osnovne nastavitve, oziroma osnovni model se konstruira sam, odvisno od parametrov doblih iz portala potem uporabnik lahko spremeni produkt po svoji želji, kot se lahko vidi iz slike 5.

Diagram primerov uporabe



Slika 4: Diagram primerov uporabe

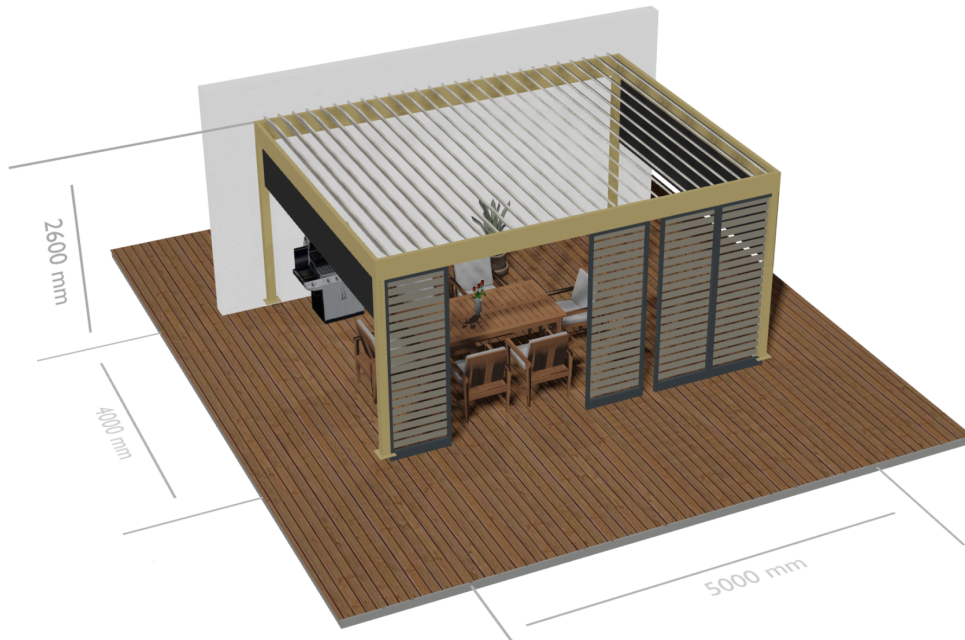
Detaljni diagram podatkovnega toka (Flow chart Diagram):



Slika 5 Diagram poteka

Prikaz 3D modela:

Model generiran iz portala brez sprememb kot končni izdelek se lahko vidi na sliki psodaj



Slika 6. Osnivni model

V naslednji sliki vidimo 3D model izdelan iz parametrov porta ki je spremenjen (način sprememb se lahko vidi v uporabniški dokumentaciji), tako da so vsi dodatni elementi zbrisani ter dolžina spremenjena.

Za se boljši občutek kako bi model izgledal v naravi je dodana slika ozadja.



Slika 7. Model z sliko ozadja

Opis dela:

Zaradi direkne komunikacije z podjetjem, Boris je nekako delal plan in diktiral plan projekta. Delo sva poskusila razdeliti na enake dele, sicer to v praksi ni možno pa se nisva niti obremenjevala preveč. Smo nekako gledali naredimo osnovno podelitev da bi Boris delal backend/funkcionalnosti, dokler bi Hasan poskrbel za izgled aplikacije in osnovno postavitve dokumentacije.

Seznam osameznih del:

Skupaj sva:

- Določili tehnologije
- Določili najboljše pozicije menijev, glede uporabniških iskusnj
- Testirali/spreminjali posamezne funkcionalnosti (kot je slajder ki spreminja kot lamel na strehi, sva morala testirati/spreminjati skupaj po urejanje frontend-a ker vrednosti vplivajo na to kako izgleda)
- Testirali končne izdelke
- Naredili predstavitev

Boris je:

- Narisal 3d model, vse elemente (stebre, lamele, zipe, stekla) ki se pol spreminjajo z kodo
- Naredil skripto za spremembe elementov, pozicijo, velikost, rotacijo (da bi se lahko dobil model kot uporabnik zeli)...
- Dodal UI funkcionalnosti za input polja, kateri vplivajo na spremembe oziroma prikaz posameznih elementov oziroma koncnega izdelka
Na primer za osnovni (samo striri stebra in streha) potrebno je skalirati stebere po visini za določeno vrednost odvisno od input polja "višina", potem se dva stebra morata premakniti po dolžini in elementi vmes skalirati za razliko med novo in prejšnjo dolžino. Potem proces ponovi za širino
- Sprotno testiral
- Naredil kompresijo celega fajla (blender nam ponuja to možnost kompresije in potem ko se celi fajl naloži se odpre in uporablja naprej)
- Dopolnil dokumentacijo

Hasan je:

- Poskrbel za izgled elementov (posamezni elementi nimajo samo barvo, pac so določene teksture namesto barv).
- Izbral slike/teksture
- Zmanjšal velikosti slik (zaradi velikine fajla, ni potrebno imeti slike 4k za posamezne elemente ker vplivajo na hitrost nalaganja) in uredil elemente. Primer: vrtnica - na celoma je majhna in je pomembno da ne igleda cisto zelene pa crvene barve, ampak da je bolj "živa" kje nam teksture pomagajo.
- Poskrbel za izgled svih input polja, otiroma celega UI-a.
- Sprotno testiral
- Naredil strukturo dokumentacije

