Izrada aplikacije za optimalno prostorno raspoređivanje izvora svjetlosti

Kaučić, Ivan

Undergraduate thesis / Završni rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:941423

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-17**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Preddiplomski studij

Izrada aplikacije za optimalno prostorno raspoređivanje izvora svjetlosti

Završni rad

Ivan Kaučić

Osijek, 2020.

Sadržaj

1. UVOD 1

1.1. Zadatak završnog rada 1

2. TUNEL, DIJELOVI TUNELA I POSTOJEĆA RIJEŠENJA ZA POSTAVLJANJE RASVJETE U TUNELIMA..... 2

2.1. Ulazna zona 2

2.2. Zona praga..... 3

2.3. Prijelazna zona..... 3

2.4. Unutarnja zona..... 4

2.5. Izlazna zona..... 4

2.6. Postojeća programska rješenja za postavljanje osvjetljenja u tunelima……5

3. OPIS KORIŠTENIH TEHNOLOGIJA.....7

3.1. Povijest Qt-a.....7

3.2. Značajke Qt alata.....8

3.3. Razvojno okruženje Qt alata.....8

4. APLIKACIJA ZA IZRAČUN OSVJETLJENJA U TUNELU..... 11

4.1. Matematičke formule za izračun osvjetljenja 11

4.2. Koeficijent pada osvjetljenja.....14

4.3. Aproksimacija krivulje pada osvjetljenja pravcem..... 15

4.4. Algoritam postavljanja izvora osvjetljenja.....16

5. IZRADA APLIKACIJE U QT-U.....19

5.1. Dizajn projekta.....19

5.2. Korisničko sučelje..... 20

5.3. Implementacija funkcionalnosti 21

5.4. Prikaz funkcionalnosti..... 25

6. PROVJERA REZULTATA.....28

7. ZAKLJUČAK 34

LITERATURA..... 35

SAŽETAK..... 36

ABSTRACT.....37

ŽIVOTOPIS.....….....….38

1. UVOD

Prilikom gradnje tunela, važno je rasporediti rasvjetu unutar tunela. Ljudskom oku treba vrijeme prilagodbe kada ulazimo iz okoline velike osvijetljenosti u okolinu znatno manje osvijetljenosti. Stoga, tunelska rasvjeta pruža bolju prilagodbu ljudskoga oko na značajne promjene osvjetljenja. Ovaj rad je namijenjen pomaganju i olakšavanju određivanja položaja izvora osvjetljenja prilikom projektiranja rasvjete tunela.

Unutar tunela postoji više zona kroz koje se prolazi. Svaka ima svoje značajke i vrijednosti osvjetljenja koje mora postići i zadržati. Dijelovi tunela su: ulazna zona, zona praga, prijelazna ili tranzitna zona, unutarnja zona i izlazna zona. Najvažnija zona prilagodbe ljudskog oka u tunelu je tranzitna zona. Ovaj završni rad bavi se rješavanjem problema optimalnog stavljanja izvora rasvjete u tranzitnoj zoni. Postavljanje rasvjete ovisi o brzini kretanja vozila, vanjskom osvjetljenju, gustoći prometa i zaustavnom putu.

Tema o kojoj govori ovaj završni rad je obrađena i postoji programsko rješenje u obliku *Reluxa*. *Relux* je program pomoću kojega se mogu izračunati razmaci izvora osvjetljenja i osvijetljenost unutar cijelog tunela. Također, radi provjeru koja daje informaciju je li uspješno dosegnuta potrebna razina osvijetljenosti. U okviru ovog rada se predstavlja sličan program koji mjeri i ispisuje potrebne razine osvijetljenosti unutar tranzitne zone i razmake između izvora osvjetljenja.

1.1. Zadatak završnog rada

Potrebno je u Qt razvojnom okruženju napraviti aplikaciju koja će odrediti optimalan raspored izvor svjetlosti u tranzitnom dijelu tunela. Proračun za određivanje optimalnog rasporeda izvora svjetlosti provodi se na temelju podataka o brzini kretanja, vanjske osvijetljenosti, gustoće prometa i zaustavnog puta pri ulasku u tunel.

2. TUNEL, DIJELOVI TUNELA I POSTOJEĆA RIJEŠENJA ZA POSTAVLJANJE RASVJETE U TUNELIMA

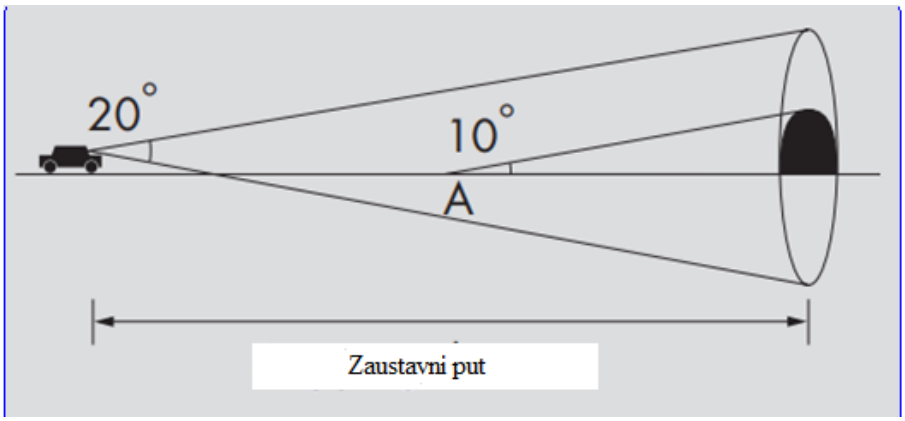
Tunel je podzemna građevina ispod površine terena koja osigurava prostor za različite namjene i s jednim ili oba kraja izlazi na površinu. Tunel definiraju širina, duljina, a i činjenica može li se njime putovati u oba smjera. Najčešće su visoki između dva do tri i pol metra, a po duljini ima kraćih (do 200 m), običnih (između 200 m i 500 m) i dugački (preko 500 m).

Duljina tunela je jako bitna zbog brzine vožnje prometnih vozila kroz njih, te tako i određuje koliko koja zona mora biti dugačka i kako se osvjetljenje mora ponašati u kojoj zoni.

2.1. Ulazna zona

Ulazna zona ne nalazi se unutar tunela, nego je to komad kolnika koji se proteže prije samog ulaska u tunel. Iz ove zone svaki vozač bi morao moći vidjeti unutrašnjost tunela, kako bi uspio i na vrijeme reagirao ako postoje nekakve prepreke, i pri tome smanjio brzinu vozila.

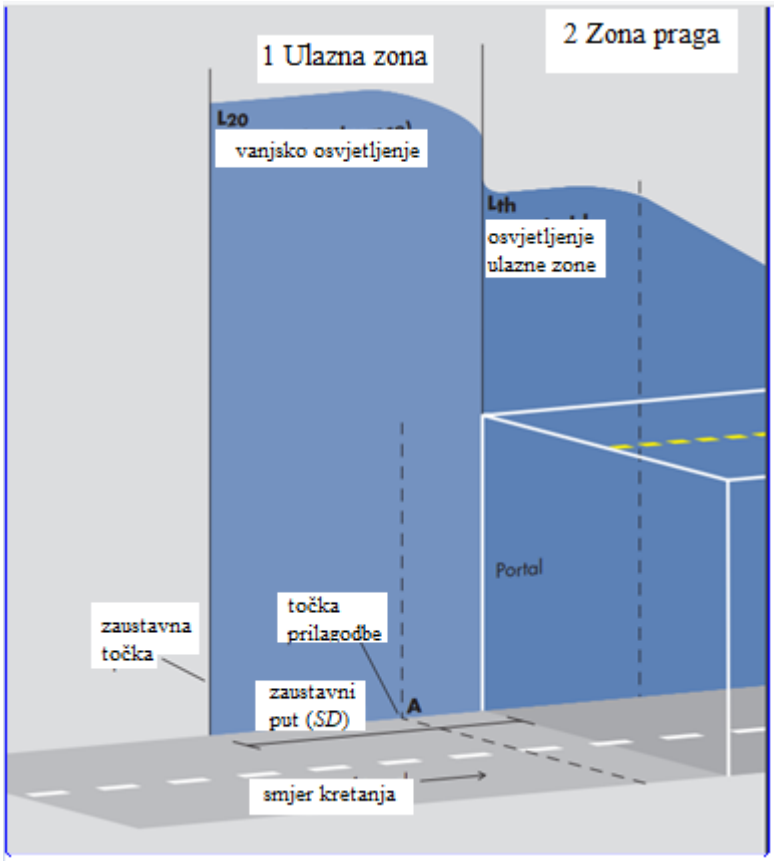
Nivo osvjetljenja u ovoj zoni, korišteno od strane CIE za izračun vizualne prilagodbe oka, računa se metodom L20. Ta metoda uzima u obzir prosječno osvjetljenje okoliša, neba, ceste i vidnog područja od 20 stupnjeva centriranog u ravnini vozačeva pogleda od početka ulazne zone. Metoda L20 se računa na daljini zaustavne udaljenosti od tunela. Zaustavna udaljenost je udaljenost unutar koje vozač sigurno i na vrijeme može reagirati u slučaju nepredviđenog i opasnog događaja (Slika 2.1.)[7].



Slika 2.1. Prikaz zaustavne udaljenosti[7].

2.2. Zona praga

Zona praga je prvi dio tunela, i najčešće je duljine zaustavnog puta vozila. U prvom dijelu te zone, osvjetljenje je povezano s vanjskim osvjetljenjem izmjerenim L20 metodom i stanjem prometa. Na kraju ove zone, osvjetljenje drastično pada na 40 posto početne vrijednosti. (Slika 2.2) [7].



Slika 2.2. Prikaz unutarnje zone.[7]

2.3. Prijelazna zona

Kroz cijelu duljinu prijelazne zone, osvjetljenje se postupno smanjuje na stupanj potreban za ulazak u unutarnju zonu. Smanjenje osvjetljenja ne smije pasti za više od trećine prijašnjeg osvjetljenja, jer stvara neugodnosti ljudskom oku prilikom prilagodbe.

Prijelazna zona završava kada osvjetljenje postaje tri puta manje od početnog osvjetljenja unutar prijelazne zone.

2.4. Unutarnja zona

Unutarnja zona je zona između prijelazne i izlazne zone. U pravilu je najdulja zona u tunelu. Razina osvjetljenja je određena brzinom kojom se smije voziti u tunelu i stanjem prometa.

Raspon osvjetljenja je različit i ovisi o gustoći prometa i brzini kretanja. (Tablica 2.3)[7].

Tablica 2.3. Raspon osvjetljenja unutarnje zone.[7]

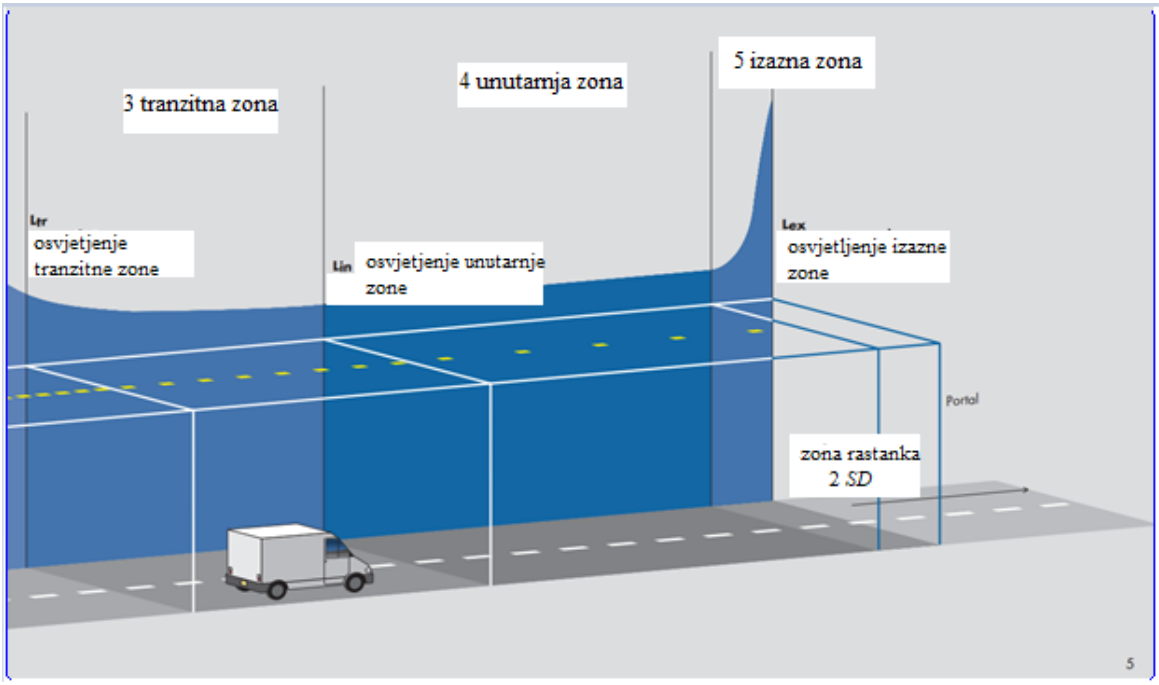
Potrebno osvjetljenje unutarnje zone	
Velegradska sredina, rijedak promet, male brzine (<70 km/h)	1.5 to 3cd/m²
Velegradska sredina, gust promet, veće brzine (>70 km/h)	2 to 6cd/m²
Autoput	4 to10cd/m²
Gradska sredina	4 to10cd/m²

2.5. Izlazna zona

Završni dio tunela je izlazna zona. Po danu, vidno polje vozača, koji se približava kraju, je pod utjecajem danje svjetlosti izvan tunela (Slika 2.4)[7].

Ljudsko oko gotovo se u trenutku može prilagoditi na veći intenzitet svjetlosti, međutim nekada je potrebno staviti osvjetljenje jer bi kraj tunela s jedne strane mogao biti početak tunela s druge strane, pa i o tim stvarima treba voditi brigu.

Najveća duljina izlazne zone ne prelazi 50 metara, a razina osvjetljenja je pet puta veća od unutrašnje zone.



Slika 2.4. Prikaz tranzitne, unutarnje i izlazne zone[7].

2.6. Postojeća programska rješenja za postavljanje osvjetljenja u tunelima

Prvo takvo programsko rješenje je *Relux*. *Relux* je visoko učinkovit i intuitivan program za simulaciju umjetnog i dnevnog svijetla. Omogućuje brzu i učinkovitu podršku u svim pogledima, za velike i male projekte, bez puno okolišanja. Simulacije se odvijaju u stvarnom vremenu (engl. *Real time*). Pruža izračun apsolutnih vrijednosti, nacionalne i internacionalne standardi, kompatibilnost s CAD i BIM sistemima, ali i još mnogo drugih mogućnosti. Unutar samog programa razne nadogradnje su omogućila modeliranje i izgradnju tunela, zgrada i drugih stambenih objekata, kao i mjerenje energetske potrošnje[9].

Aplikacija ovog završnog rada je zapravo samo dio mogućnosti koje pruža *Relux*. U samom *Reluxu* moguće je dizajnirati tunel, opremiti ga izvorima osvjetljenja, postaviti vanjsko osvjetljenje i simulirati razine osvjetljenja u samom tunelu, dok se u aplikaciji završnog rada samo ispisuju razine osvjetljenja tranzitne zone tunela i razmaci između rasvjete, uz mogućnost postavljenja početnih parametara.

Drugo programsko rješenje je *Vivaldi*. *Vivaldi* je interaktivan dizajnerski alat koji pojednostavljuje dizajn i prezentaciju dinamičkih koncepata osvjetljenja. U *Vivaldiju,* slike, koje

su zasebno izračunate, uvezene su iz različitih kontrolnih grupa, i tako povećavaju preciznost vizuaizacije. Imamo mogućnost prikaza posljedica dobivenih prigušivanjem svijetla ili mijenjanjem boja u stvarnom vremenu i pohrane dinamičke vremenske skale ili integriranja sekvence dnevnoga svijetla. Radeći s tim mogućnostima, program uvijek uzima u obzir razine osvjetljenja i energetsku potrošnju. Slike iz *Vivaldija* mogu se automatski izvesti (engl. *Export*) iz *Reluxa*, ali mogu biti i prebačene s *Dialux Evo* programa i drugih programa za simulaciju kao HRD ili EXR slike[10].

U usporedbi s aplikacijom završnog rada, *Vivaldi* aplikacija nudi dinamičke promjene dizajna rasvjete ovisno o promjeni razine osvjetljenja. Iznimno korisna stvar za smanjenje energetske potrošnje. Aplikacija završnog rada, nakon upisivanja vrijednosti za vanjsko osvjetljenje, nema mogućnost dinamičke promjene rasvjete i razmaka između susjednih izvora osvjetljenja. Također za provjeru rezultata aplikacije završnog rada, u *Dialuxu* moramo dizajnirati vlastiti tunel, postaviti osvjetljenje, pokrenuti izračun osvjetljenja i usporediti rezultate, dok u *Vivaldiju* možemo prenijeti model tunela iz *Dialuxa* te će *Vivaldi* sam izračunati i usporediti rezultate.

3. OPIS KORIŠTENIH TEHNOLOGIJA

Tehnologija korištena za izgradnju ove aplikacije se temelji na Qt alatima, u koje se ubrajaju *Qt GUI*, *Qt QML*, *Qt Widgets*, *Qt Core* i programski jezik C++, koji je podržan unutar Qt alata. Svi korišteni elementi Qt alata su objašnjeni u nastavku.

3.1. Povijest Qt-A

U ljeto 1990. Haavard Nord i Eirik Chambe-Eng (izvorni *developer* Qt-ai predsjednik) radili su zajedno na primjeni baze podataka koja bi spremača slike ultrazvuka, napisanim u C++ jeziku, koji se mogao pokretati na Mac OS-u, Unix-u i Microsoft Windows-u. Započeli su izgradnju Qt-a 1991. Jedan od glavnih zahtjev ove aplikacije je bilo to da aplikacija ima grafičko korisničko sučelje (eng. *Grahpic user interface, GUI*), tj. da bude prilagođeno za ljudsku uporabu. Nord i Chambe-Eng dolaze na ideju stvaranja objektno orijentiranog sustava i zajedno stvaraju Qt softverski okvir (eng. *Framework*), višeplatformskog i objektno orijentiranog grafičkog korisničkog sučelja.[2]

1991. godine grade prve klase unutar Qt-a, a naredne, 1992. godine dolaze na ideju o signalima i utorima (eng. *Signals and losts*), koja će se ispostaviti da je jedna od glavnih paradigmi Qt-a. 1993. godine izgrađuju prvu grafičku jezgru (eng. *Kernel*) i prve *widgete* (eng. *Window gadget*).

Alat se zvao Qt jer je slovo Q izgledao privlačno Haavardu u *Emacs* tipu uređivaču teksta, a t je bilo nadahnuto od strane *Xt-a*, tj. *X toolkit-a*, gdje t označava *toolkit*.

Prve dvije verzije Qt-a imali su samo mogućnost biti na dva operacijska sustava: Qt/X11 za Unix i Qt/Windows za Microsoft Windows.[2]

Datum od velike značajnosti za Trolltech, tvrtka Nord-a i Chamb-Eng-a, bio je 20. svibnja 1995. Na taj dan na tržište stavljaju Qt verziju 0.90, prvu javnu verziju Qt-a koja se mogla koristiti unutar Windows i Unix okruženja te je bila dostupna pod dvije vrste licenci: komercijalnoj i besplatnoj, za razvoj open-source aplikacija (ovakva praksa licenciranja nastavlja se i danas). U ožujku 1996. Europska svemirska agencija kupuje deset komercijalnih licenca te tako postaje prva velika institucija koja koristi Qt za izradu grafičkih sučelja svojih aplikacija. U srpnju iste godine izlazi Qt 1.0, a do kraja godine Trolltech je prodao još 18 licenci kupcima unutar 8 zemalja[6].

3.2. Značajke Qt alata

Qt je korišten za izradu grafičko korisničkog sučelja i za pokretanje aplikacija koje se mogu koristiti na svim velikim *desktop* i mobilnim platformama, uključujući i neke ugradbene platforme. Većina aplikacija s grafičko korisničkim sučeljem napravljenim u Qt-u imaju dosta jednostavno sučelje, te je u tom slučaju Qt klasificiran i kao *widget toolkit*. Također se mogu i napraviti programi koji nemaju grafičko korisničko sučelje, kao npr. konzole za servere.

Qt podržava i različite prevoditelje (eng. *Compiler*), kao što su C++ prevoditelj, koji se koristi u *Visual Studio* razvojnom okruženju. Qt također donosi i *Qt Quick* , a ono uključuje i deklarativan skriptirajući jezik (eng. *Declarative scripting language*) zvan QML, koji omogućuje korištenje *JavaScript*-a za pružanje potrebne logike. Uz *Qt Quick,* započeo je brz razvoj za mobilne uređaje, gdje se logika može i dalje napisati jednostavnim kodom, koji postiže najbolje moguće rezultate.

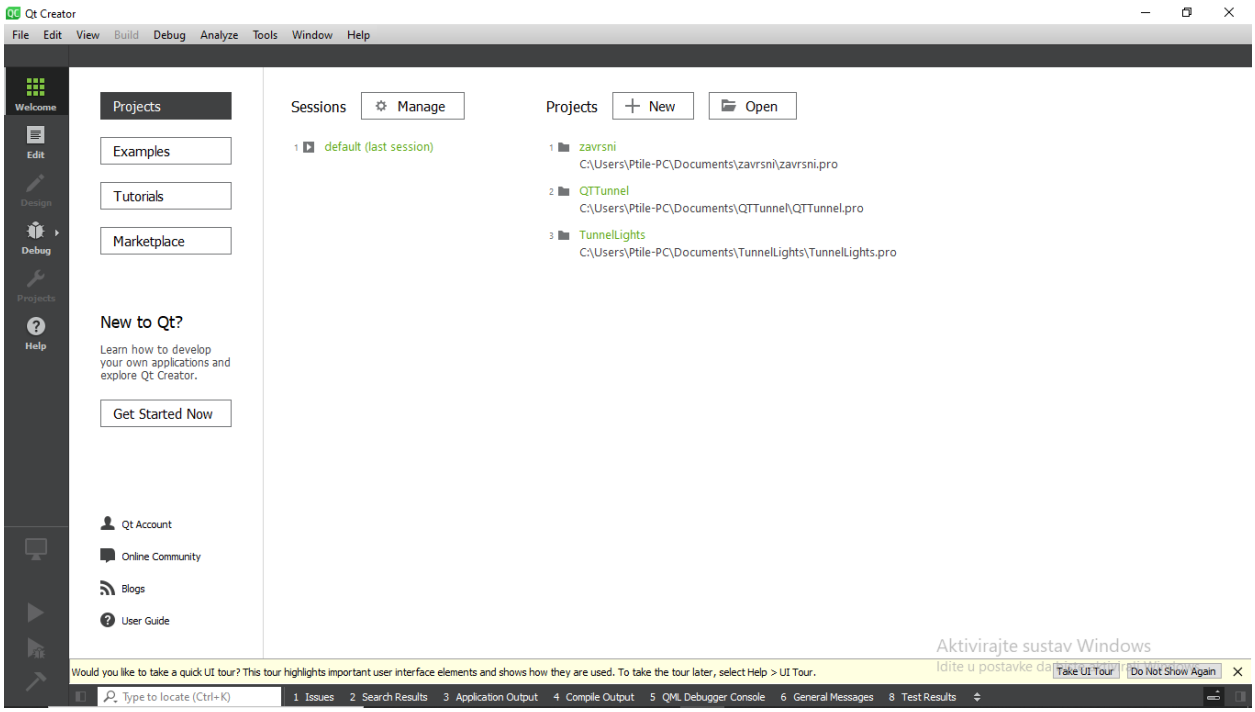
Qt Widgets sadrži klase za klasične *widget* aplikacije s grafičko korisničkim sučeljem i *QSceneGraph* klasama.[2]

Qt Core je jedini neophodan modul, sadrži klase koje koriste drugi moduli, uključujući sistem meta objekata (eng. *Meta-object system*), kontejnere, događajne sisteme (eng. *Event system*), I/O postrojenja (eng. *Input/output facilities*).[2]

3.3. Razvojno okruženje Qt alata

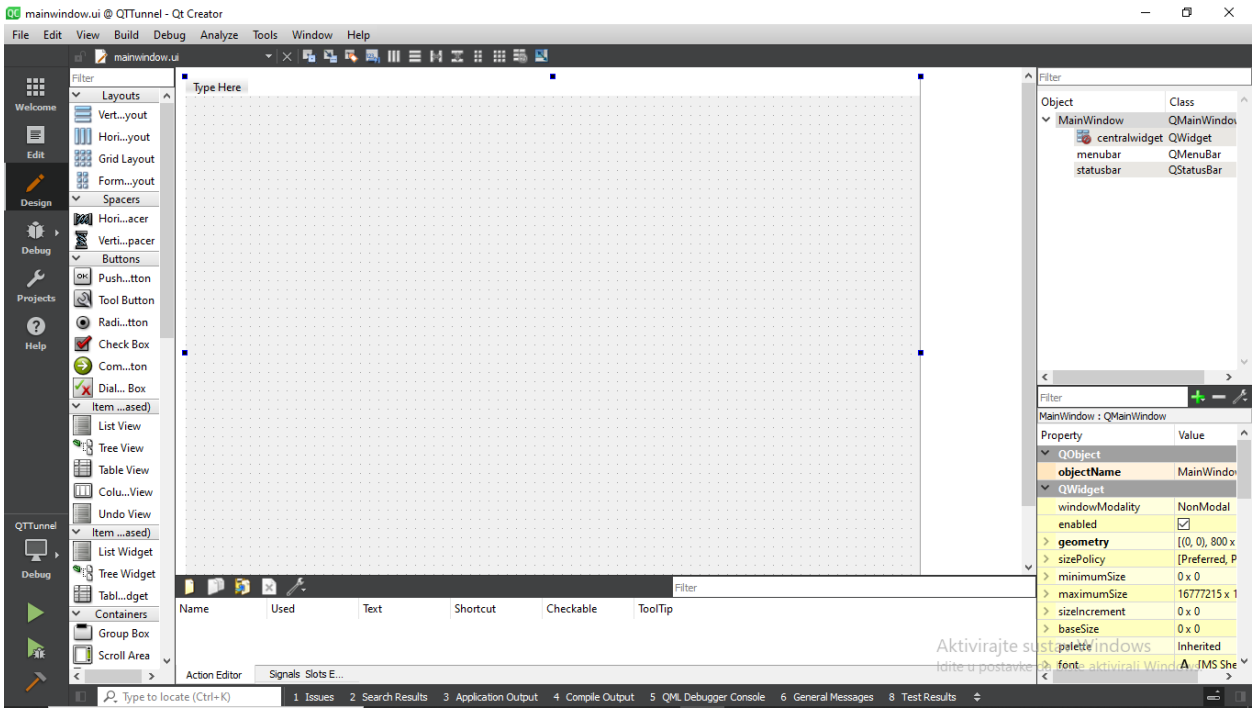
Unutar Qt aplikacije nalazi se integrirano razvojno okruženje zvano Qt Creator, a služi za razvoj aplikacija i softverski razvoj. Prilikom razvoja aplikacija unutar Qt-a je ugrađen Qt uređivač koda (eng. *Editor*) koji uvelike pomaže prilikom kodiranja, jer su mu neke od funkcionalnosti pregledavanje sintaktičke greške, automatsko nadopunjavanje koda i korištenje tekstualnih makro naredbi. Uređivaču koda se pristupa klikom na karticu Edit koja se nalazi na navigatorskoj alatnoj traci, a nalazi se na lijevoj strani Qt Creator sučelja (Slika 3.1).

Unutar Qt Creatora lako se može pristupiti svim projektima, dizajniran je sa što boljom preglednosti za korisnika. Koraci izrade početnog dijela aplikacije su potpuno objašnjeni.



Slika 3.1. Razvojno okruženje unutar Qt Creator-a.

Uz Qt Creator, postoji još jedan alat koji se zove Qt Designer. To je vizualni uređivač pomoću kojeg se izrađuje grafičko korisničko sučelje. Qt Designeru pristupamo preko Design kartice unutar Qt Creatora. Unutar Design kartice postoje razni *widgeti* koji se koriste za izradu grafičkog sučelja, kao npr. *Push Button*, *Check Box*, *Radio Button* i sl. Stavljanje tih *widgeta* unutar grafičkog sučelja (Slika 3.2.) se postiže takozvanom radnjom povuci i spusti (eng. *Drag and drop*). Nakon označavanja jednog od *widget* elemenata unutar grafičko korisničkog sučelja, s desne donje strane dobijamo pregled svih elemenata koji se mogu mijenjati unutar tog *widgeta*. U te elemente pripadaju ime objekta, visina, i širina objekta, boja, može li se na taj objekt kliknuti itd. Raspored *widgeta* se može mijenajti ručno, povlačenjem tog *wdigeta* do željene lokacije, ali postoji i opcija na alatnoj traci *widgeta* gdje se nalazi više tipova okvira za raspored (eng. *Layout*).



Slika 3.2. Qt Designer s alatnom trakom za *widgete*.

4. APLIKACIJA ZA IZRAČUN OSVJETLJENJA U TUNELU

Aplikacija za optimalno prostorno raspoređivanje svijetlosti pružat će:

- Unos ulaznih podataka važnih za izračun, ostale elemente koji su potrebni, a nisu na popisu ulaznih podataka aplikacija sama izračunava,
- Ispis osvjetljenja u tranzitnom dijelu tunela svakih deset metara,
- Ispis ključnih podataka kao što su duljina tranzitne zone, vrijednost osvijetljenosti zone praga, vrijednost osvijetljenosti početka tranzitne zone, vrijednost osvijetljenosti unutarnje zone,
- Ispis udaljenosti između dva susjedna izvora osvjetljenja.

Prije izrade aplikacije moraju se odrediti formule koje se koriste za dobivanje traženih veličina u aplikaciji. Sve formule vezane uz osvjetljenje u tunelu i duljine zona tunela su zadane u CIE pravilniku o tunelskoj rasvjeti [8].

4.1. Izračun osvjetljenja

Za osvijetljenost unutar tunela važni su parametri vanjske osvijetljenosti (*Lv*), brzina kretanja vozila (*v*), zaustavni put (*SD*) i gustoća prometa (*g*). Iz njih su dobivene sve ostale važne vrijednosti vezane uz tunel.

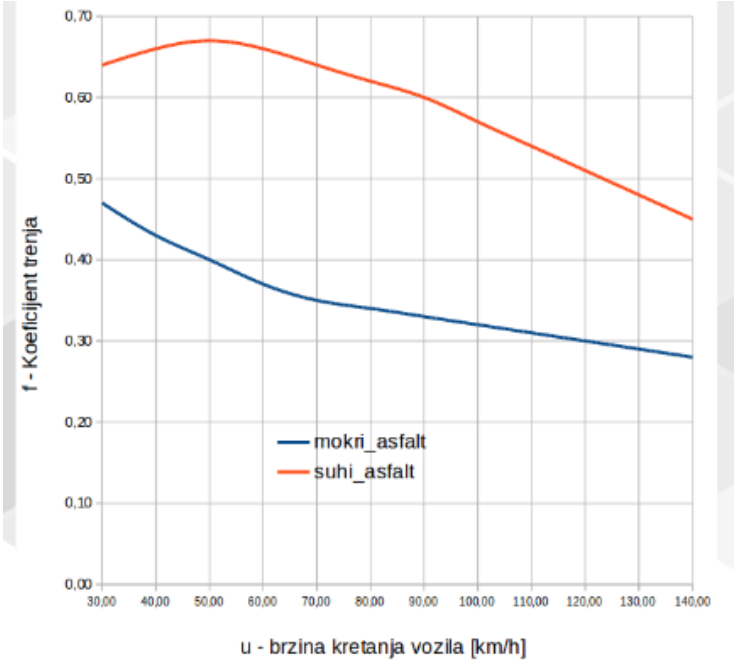
Prije svega treba odrediti duljinu zaustavnog puta (SD) prema (4-1)[1], koja se iskazuje u metrima. Ona kazuje koliko će dugačka biti ulazna zona :

$$SD = ut_0 + \frac{u^2}{2g(f \pm s)} \tag{4-1}$$

gdje je:

- *SD* – duljina zaustavnog puta iskazana u metrima,
- u – brzina kretanja vozila iskazana u metrima po sekundi,
- t_0 – vrijeme reakcije vozača iskazano u sekundama i uvijek je jedna sekunda,
- f – koeficijent trenja mokrog kolnika,
- s – nagib kolnika iskazan u postotku.

Krivulja ovisnosti koeficijenta trenja i brzine kretanja prikazana je na slici 4.1.



Slika 4.1. Graf koeficijenta trenja mokrog kolnika u ovisnosti o brzini kretanja.

Prva takva vrijednost koja se treba izračunati nakon zaustavnog puta je osvjetljenje zone praga (L_{th}) (4-2) koje se računa[8]:

$$L_{th} = kL_v \tag{4-2}$$

gdje je:

- L_{th} – osvjetljenje zone praga u kandelama po metru kvadratnom,
- k – koeficijent pada osvjetljenja između vanjskog osvjetljenja i osvjetljenja zone praga,
- L_v – vanjsko osvjetljenje izraženo u kandelama po metru kvadratnom.

Koeficijent pada osvjetljenja (k) ovisi o brzini vožnje iskazanima u km/h. Iz tablice 4.2 vidi se da koeficijent raste s porastom brzine vožnje. [1]

Tablica 4.2. Ovisnost koeficijenta *k* o brzini[1].

Brzina (km/h)	*k*
<= 60	0,05
80	0,06
90	0,07
100	0,08
110	0,09
120	0,10

Iduća vrijednost koju treba izračunati je osvjetljenje na početku tranzitne zone (L_{tr1})(4-3) koja se računa[8]:

$$L_{tr1} = 0.4L_{th} \tag{4-3}$$

gdje je:

- L_{tr1} – osvjetljenje početka tranzitne zone i izražava se u kandelama po metru kvadratnom,
- L_{th} – osvjetljenje zone praga i izražava se u kandelama po metru kvadratnome

Osvjetljenje unutarnje zone tunela (L_{in}) ovisi o gustoći prometa i brzini vožnje, a može se očitati pomoću tablice, gdje klase tri i četiri označuju visoku i nisku gustoću prometa (Tablica 4.3.)[4].

	Brzina kretanja vozila		
	km/h	km/h	km/h
Klasa	**<=60**	**80-100**	**>120**
4	**3**	**6**	**10**
3	**2**	**4**	**6**

Tablica 4.3. Ovisnost unutarnje osvjetljenja.[4]

4.2. Koeficijent pada osvjetljenja

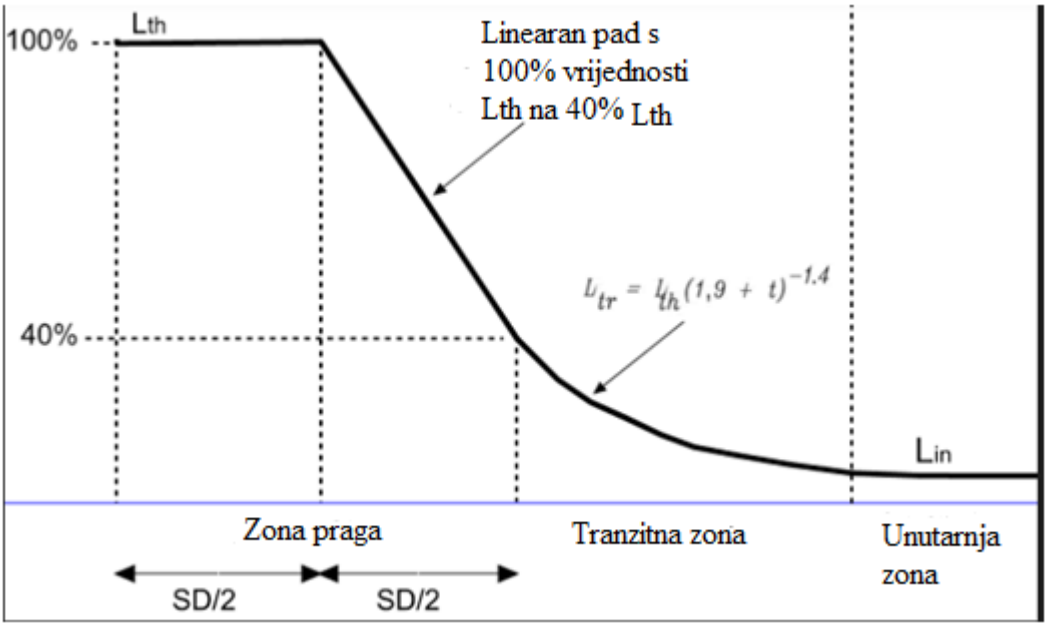
Koeficijent pada osvjetljenja k_{pad} je jako bitan u algoritmu postavljanja izvora osvjetljenja jer govori o tome kako će se mijenjati osvijetljenost u odnosu na prijeđeni put. Ova formula ne postoji ni u jednoj literaturi, ona je eksperimentalno sama izvedena od strane autor ovog završnog rada. Koeficijent k_{pad} se računa izrazom (4-4):

$$k_{pad} = \frac{\Delta L}{l_{tr}} = \frac{L_{in} - L_{tr1}}{l_{tr}} \tag{4-4}$$

gdje je:

- k_{pad} – količnik razlike osvjetljenja unutarnje zone i osvjetljenja početka tranzitne zone i duljine tranzitne zone izražena u kandelama po metru,
- ΔL – razlika osvjetljenja unutarnje zone i osvjetljenja početka tranzitne zone izražena u kandelama po metru kvadratnom
- l_{tr} – duljina tranzitne zone izražena u metrima,

Duljina tranzitne zone se mora izraziti iz funkcije krivulje osvijetljenosti tranzitne zone (Slika 4.4.)[3].



Slika 4.4. Jednadžba tranzitne zone [3].

Iz te jednadžbe treba se iskazati varijabla *t* (4-5), koja označava vrijeme u sekundama[3]:

$$t = \left(\frac{L_{in}}{L_{th}}\right)^{-0.714} - 1.9 \quad (4\text{-}5)$$

gdje je:

- t – vrijeme prolaska kroz tranzitnu zonu iskazana u sekundama,
- L_{in} – osvjetljenje unutarnje zone iskazane u kandelama po metru kvadratnom,
- L_{th} – osvjetljenje zone praga iskazane u kandelama po metru kvadratnom.

Nakon što je dobivena vrijednost *t*, množi se brzinom kretanja vozila *v*, kako bi se dobio iznos duljine tranzitne zone (4-6) :

$$l_{tr} = tv \quad (4\text{-}6)$$

gdje je:

- l_{tr} – duljina tranzitne zone iskazana u metrima,
- t – vrijeme prolaska korz tranzitnu zonu iskazana u sekundama,
- v – brzina kretanja vozila iskazana u metrima po sekundi.

4.3. Aproksimacija krivulje pada osvjetljenja pravcem

Jednadžba osvijetljenosti tranzitne zone je krivulja višeg reda (Slika 4.4.). Kako bi lakše mjerili i očitavali osvijetljenost u tranzitnoj zoni tunela i kako bi osigurali da osvjetljenje uvijek bude veće , aproksimirat ćemo krivulju pravcem koji će imati jednadžbu (4-7) :

$$L - L_{tr1} = \frac{L_{in} - L_{tr1}}{l_{tr} + SD - SD}(l - SD) \quad (4\text{-}7)$$

Ako izraz ispred zagrade u gornjoj jednadžbi zamijenimo varijablom *K*, i dodatno sredimo jednadžbu, dobijemo novu jednadžbu (4-8):

$$L = Kl - KSD + L_{tr1} \quad (4\text{-}8)$$

Te nam ovaj pravac govori kakav će iznos osvjetljenja biti u kojem metru tranzitne zone. Bitno je za napomenuti da će varijabla K uvijek biti negativnog predznaka i da pravac mora biti padajući. Rezultati ove jednadžbe pravca bit će okarakterizirani mjernom jedinicom cd/m^2, tj. rezultati će se odnositi na luminaciju unutar tranzitne zone. Prilikom provjere unutar programa

za izračun osvijetljenosti, kao što je *Dilaux*, bit će potrebno postići jednaku ili veću vrijednost od one zadane funkcijom pravca unutar ovog poglavlja.

4.4. Algoritam postavljanja izvora svjetlosti

Pošto je već izveden koeficijent pada osvjetljenja u tranzitnoj zoni tunela, sada treba taj koeficijent povezati s izvorima osvjetljenja. Važno je bilo pronaći način postavljanja osvjetljenja unutar tranzitne zone. U obzir su išle dvije metode. Prva metoda se temeljila na padu osvijetljenosti po razinama, gdje pad u osvjetljenju ne smije biti veći od trećine prijašnje vrijednosti. U prvoj metodi, unutar svake razine, razmak između izvora osvjetljenja je jednak. Druga metoda se temeljila na postupnom povećavanju razmaka između izvora osvjetljenja do kraja tranzitne zone tunela. Prva metoda se nije pokazuje se kao dobar pristup nakon simulacije tunelske rasvjete unutar *Dialuxa*. Vrijednosti osvjetljenja unutar tranzitne zone su imale velika odstupanja u usporedbi s rezultatima koje je dala aproksimacijska krivulja osvjetljenja tranzitne zone. Kao bolji postupak postavljanja rasvjete se pokazala druga metoda. Funkcija (4-9) koja je najbolje opisala postupno povećanje razmaka između redova rasvjete je eksponencijalna funkcija, koja povezuje koeficijent pada osvjetljenja i informaciju o kojem se redu rasvjete radi :

$$\Delta l_{izvor} = i^{k_{pada}} \tag{4-9}$$

gdje je:

- Δl_{izvor} – razmak između dva reda osvijetljena iskazan u metrima,
- i – redni broj reda osvjetljenja,
- k_{pada} – koeficijent dobiven formulom (4-4) iskazan u kandelama po metru,

varijabla i predstavlja o kojem se redu svjetala radi npr. *i*=1 i *i*=2 označava da se radi o prvom i drugom redu svjetala. Koliko svjetala ima u redu ovisi o tome koliko ima prometnih traka unutar tunela. Za dvije prometne trake bio bi dobar odabir imati tri izvora osvjetljenja, na sredini svake prometne trake po jedno i jedno na liniji koja razdvaja te dvije prometne trake, iako to nije obaveza nego prijedlog. Za tri prometne trake u redu će biti pet izvora osvjetljenja. Taj će broj izvora rasvjete u unutarnjem dijelu tunela prijeći na onaj broj koliko ima prometnih traka, dakle ako ima tri prometne trake, u unutarnjoj zoni će biti tri izvora osvjetljenja, u svakom prometnom

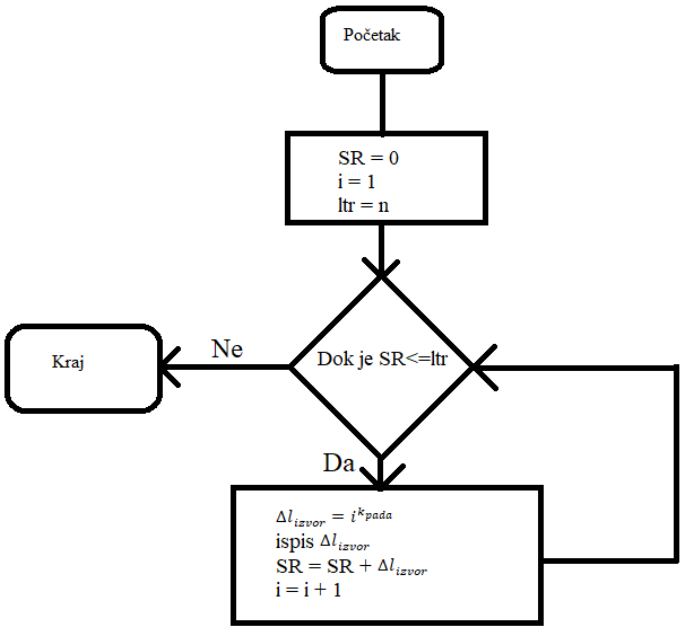
traku po jedno u redu. Unutar programa *Dialux*, pomoću primljenih podataka o udaljenosti redova svjetala i osvjetljenju koje mora postojati u određenim dijelovima tunela, uzeti će se rasvjeta koja odgovara i zadovoljava razine osvjetljenja.

Potrebno je znati kada treba stati s postavljanjem izvora osvjetljenja u tranzitnoj zoni, za to će pomoći petlja koja će zbrajati razmake između redova svjetala dok god zbroj razmaka ne prelazi duljinu tranzitne zone l_{tr}. (Slika 4.5.)

```
i=1;
SumaRazmaka = 0;
while (SumaRazmaka <= l_tr ){
Δl_izvor = i^k_pada
SumaRazmaka = SumaRazmaka + Δl_izvor;
i++;
}
```

Slika 4.5. Algoritam postavljanja izvora osvjetljenja.

gdje je *SumaRazmaka* ukupan zbroj svih razmaka između trenutnog i slijedećeg reda izvora osvjetljenja. Za pojednostavljenje koda, ali i samog algoritma slika 4.6. pokazuje tijek radnje algoritma.



Slika 4.6. Tijek rada algoritma.

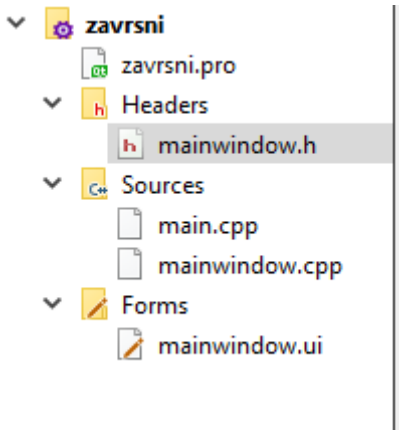
Unutar slike 4.6. *SR* predstavlja sumu udaljenosti između dva susjedna reda osvjetljenja, *i* predstavlja brojač koji kazuje o kojem se redu osvjetljenja radi i počinje od prvog reda, *ltr* je varijabla koja predstavlja već izračunatu duljinu tranzitne zone s vrijednošću n.

5. IZRADA APLIKACIJE U QT-U

Za izradu aplikacije potrebno je znanje C++ jezika, u što se ubraja služenje objektima i metodama te poznavanje sintakse. Qt je objektno orijentirano razvojno okruženje u kojemu je svaki *widget* novi objekt. Važno je stoga znati da se metode koje objekti imaju pozivaju pokazivačima (->).

5.1. Dizajn projekta

Aplikacija je napravljena pomoću Qt alata. Napravljen je u Qt Creatoru, a pomoć u grafičko korisničkom sučelju je pružio Qt Designer. Aplikacija ima tri glavne mape, a to su zaglavlja (eng. *Headers*), izvori (eng. *Sources*) i izgledi ili forme (eng. *Forms*) (Slika 5.1.).



Slika 5.1. Datoteke unutar aplikacije.

Unutar zaglavlja se nalaze privatni utori koji su zapravo pritisci gumbova na grafičko korisničkom sučelju aplikacije (Slika 5.2.).

```
private slots:
    void button_pressed();
    void on_calculateBtn_clicked();
    void on_pushButton_clicked();
    void on_jednadzbaBtn_clicked();
```

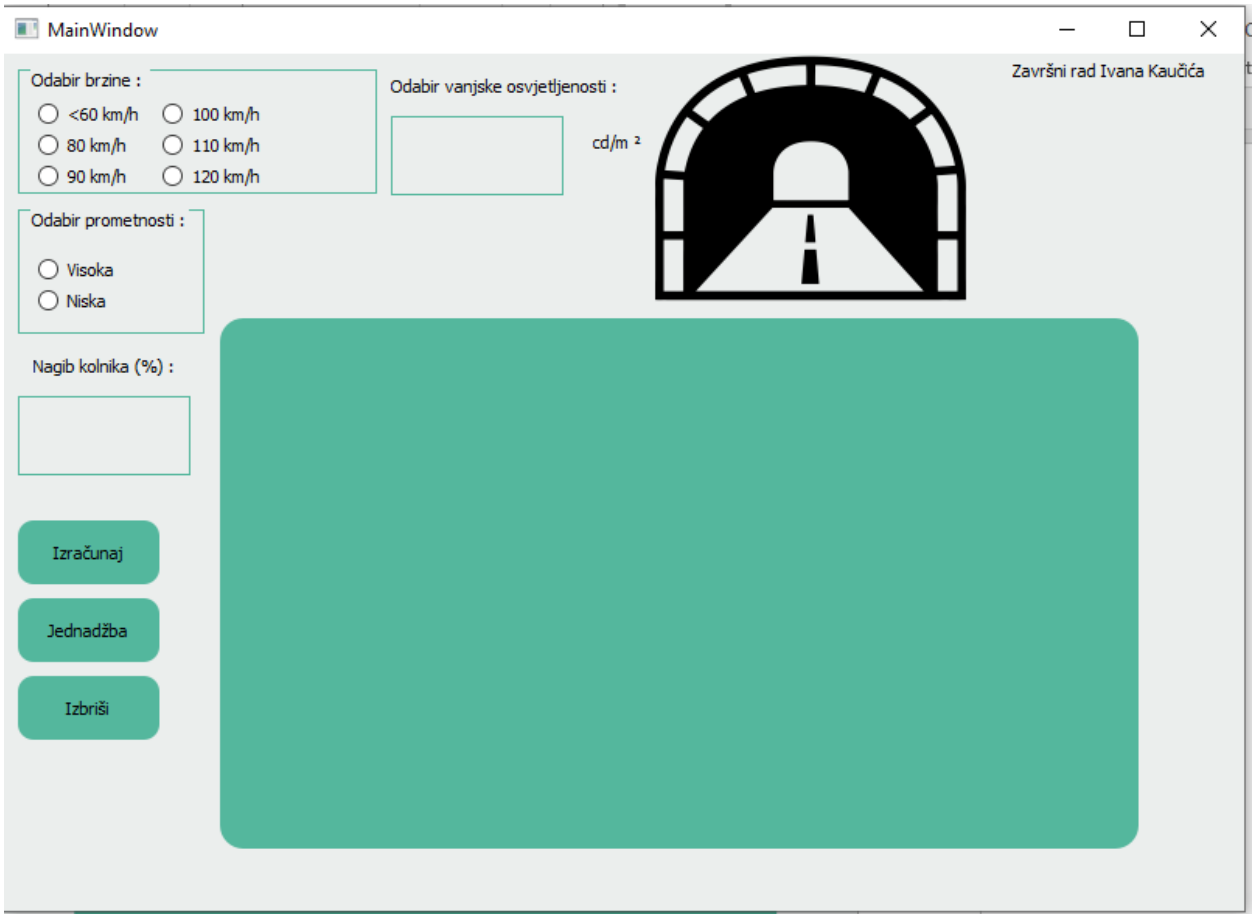
Slika 5.2. Akcije pritiska gumba

Unutar mape Izvori nalazi se sav kod i funkcionalnosti koji dozvoljavaju očekivani rad aplikacije. Ima dvije datoteke, a to su main.cpp i mainwindow.cpp. Prva datoteka održava rad aplikacije do isključivanja, a druga datoteka sadrži sve funkcionalnosti koje se pozivaju iz *widgeta*.

Mapa Forme sadrži dizajn grafičko korisničkog sučelja koji je korisnik osmislio, u njoj se fizički vide pozicije svih *widgeta*.

5.2. Korisničko sučelje

Korisničko sučelje ove aplikacije sastoji se od jednog *main windowa* u kojemu su raspoređeni svi *widgeti*. Dva *QGroupBoxa*, gdje se može odabrati samo jedan *QRadioButtona*. U *QGroupBoxovima* su sadržane specifične vrijednosti prilikom dizajna tunela koje kada se označe daju vrijednost elemenata tunela, kao npr. označavanjem brzine kretanja u tunelu dobije se određeni koeficijent k, koji služi za računanje osvjetljenosti ulazne zone tunela (L_{th}). Tu se nalaze i dva *widgeta* zvana *QTextEdit* u koji se upisuje vanjska osvjetljenost (L_v) i nagib kolnika. U aplikaciji postoje tri *QPushButton widgeta* gdje svaki ima svoju funkcionalnost interakcije s *widgetom* u kojemu se spremaju rezultati svih računjanja unutar aplikacije, a to je *QList widget*. Tu su i dva *QLable widgeta* od kojih je u jednom otisnut kreator aplikacije, a drugi *QLable* je pretvoren u mjesto gdje se stavlja logo unutar korisničkog sučelja. (Slika 5.3.)



Slika 5.3. Izgled grafičko korisničkog sučelja aplikacije.

5.3. Implementacija funkcionalnosti

Najvažnije implementacije funkcionalnosti unutar ove aplikacije su tri *widgeta QPushButtona*.

Prvi gumb na kojemu je natpis „Izračunaj“ zadužen je za izračunavanje i ispis najvažnijih podataka vezanih uz tunel i tranzitnu zonu tunela, a to su osvjetljenje početka ulazne zone, osvjetljenje početka tranzitne zone, osvjetljenje unutarnje zone, duljina tranzitne zone, jednadžba pada osvjetljenja i postavljanje udaljenosti između izvora osvjetljenja. Prva zadaća gumba „Izračunaj“ je pokupiti sve važne ulazne podatke, kao što su brzina kretanja vozila, vanjska osvjetljenost, zaustavni put i gustoća prometovanja. Slika 5.4. prikazuje dohvaćanje vrijednosti brzine i koeficijenta k iz *QGroupBoxa* u kojemu se nalaze *QRadioButtoni* za brzinu vožnje. Također prikazuje i povezivanje *QGroupBoxova* za gustoću prometovanja i duljinu zaustavnog puta s osvjetljenjem unutarnje zone tunela. Provjera je li neki *QRadioButton* označen ili ne

provodi se jednom if petljom unutar koje je metoda za *isChecked()* koja provjerava je li *QRadioButton* označen ili ne.

```
if(ui->sixtyRB->isChecked()){brzina=60;koefK=0.05;f=0.47;}
if(ui->eightyRB->isChecked()){brzina=80;koefK=0.06;f=0.34;}
if(ui->ninetyRB->isChecked()){brzina=90;koefK=0.07;f=0.32;}
if(ui->hundredRB->isChecked()){brzina=100;koefK=0.08;f=0.31;}
if(ui->hundredtenRB->isChecked()){brzina=110;koefK=0.09;f=0.305;}
if(ui->hundredtwentyRB->isChecked()){brzina=120;koefK=0.1;f=0.3;}
if(ui->highRB->isChecked() && brzina<=60){Lin=2;}
if(ui->highRB->isChecked() && brzina<=80 && brzina>=100){Lin=4;}
if(ui->highRB->isChecked() && brzina<=100 && brzina>=120){Lin=6;}
if(ui->lowRB->isChecked() && brzina<=60){Lin=3;}
if(ui->lowRB->isChecked() && brzina<=80 && brzina>=100){Lin=6;}
if(ui->lowRB->isChecked() && brzina<=100 && brzina>=120){Lin=10;}
```

Slika 5.4. Povezivanje *radio buttona* s potrebnim vrijednostima za tunel.

Svi matematički proračuni koji su bili izvedeni u proračunima u četvrtom poglavlju se moraju implementirati u gumbu za „Izračunaj“. Prilikom ispisivanja rezultata u *Qlist* važno je dobivene rezultate prebaciti u *QString* vrijednosti pomoću koji se onda lako može izvršiti ispis rezultata unutar *QLista*, uz pomoć funkcije unutar objekta *QLista* koji glasi *addItem(),* argument koji se šalje pomoću *addItem()* funkcije može biti *QString* . Pošto se većinom radi o decimalnim vrijednostima traženih rezultata, dobra je praksa stavljati varijable u kojima će se spremati rezultati u tip *double*. Dobivene rezultate lako je prebaciti u *QString* tip pomoću funkcije koju nasljeđuje *QString* klasa, a zove se *number*. Kako ne bi dobili kao rezultat znanstvene zapise brojeva, unutar *number* funkcije dodajemo argumente varijable koja ima vrijednost broja, slova g koji mijenja zapis broja kako on ne bi bio znanstveni i broj koji kazuje na koliko decimala prikazati taj broj. (Slika 5.5.)

```
vanjOsvjetljenje = (ui->inputTE->toPlainText());
vanjOsvjetljenost = vanjOsvjetljenje.toDouble();

Lth = koefK*vanjOsvjetljenost;
Ltrstart = 0.4*Lth;

result2 = QString::number(Lth,'g',15);
result22 = ("Početak ulazne zone ima: " + result2 + " cd/m^2");
ui->resultsLW->addItem(result22);
result = QString::number(Ltrstart,'g',15);
result1 = ("Početak tranzitne zone ima: " + result + " cd/m^2");
ui->resultsLW->addItem(result1);
result3=QString::number(Lin,'g',15);
result33= ("Unutarnja zona ima: " + result3 + " cd/m^2");
ui->resultsLW->addItem(result33);

vrijeme = pow((Lin/Lth),-0.714286)-1.9;
duljinaTr = (vrijeme*(brzina/3.6))*0.8;
resultDuljina = QString::number(duljinaTr,'g',10);
resultDuljina1 = ("Duljina tranzitne zone je: " + resultDuljina + " m");
ui->resultsLW->addItem(resultDuljina1);
koefPadaSvjetlosti = ((Ltrstart-Lin)/duljinaTr);


L1 = Ltrstart;
L2 = Lin;
m2 = duljinaTr;
koefequation=(L2-L1)/(m2-m1);
koefquationA = koefequation;
koefquationB = -koefequation*m1 + L1;

koefQA = QString::number(koefquationA,'g',10);
koefQB = QString::number(koefquationB,'g',10);
koefQAB = ("Jednadžba pada osvjetljenosti u odnosu na prijeđeni put(m)u tranzitnoj zoni
ui->resultsLW->addItem(koefQAB);
```

Slika 5.5. Koraci unutar „Izračunaj“ gumba za računanje i ispis vrijednosti.

Na kraju koda unutar „Izračunaj“ gumba tj. unutar *clicked()* metode (naziv za funkcije unutar klase) gumba nalazi se algoritam, koji je i smisao ovog završnog rada, korištene su sve iste sintakse kao i u prethodnim linijama koda, osim što sada postoji i petlja *while*. Unutar petlje se zbrajaju razmaci između redova rasvjete u varijablu *udaljenostZbroj*, sve do onog trenutka kada udaljenost postaje veća ili jednaka od varijable *duljinaTr* koja predstavlja duljinu tranzitne zone. Prilikom svakog prolaska kroz petlju povećava se i, koji predstavlja o kojem se redu svjetala radi. Unutar svakog prolaska petlje računa se i novi razmak između dva reda svjetala koji se sprema unutar varijable *udaljenostSvjetiljki*. Na kraju, petlja također služi i za ispis svih udaljenosti između redova svjetala. Udaljenosti između svjetala spremaju se unutar *QLista* nazvanim *resultLW* (Slika 5.6.).

```
while(udaljenostZbroj<=duljinaTr)
{
    brojSvjetiljki++;
    udaljenostSvjetiljki=pow((brojSvjetiljki-0.5),koefPadaSvjetlosti);
    resultUdaljenost = QString::number(udaljenostSvjetiljki,'g',15);
    resultUdaljenost1 = ("Udaljenost iduće svjetiljke od zadnje je: " + resultUdaljenost + " m");
    ui->resultsLW->addItem(resultUdaljenost1);
    itemNum = itemNum+1;
    udaljenostZbroj = udaljenostZbroj + udaljenostSvjetiljki;
}
```

Slika 5.6. Programersko rješenje postavljanja udaljenosti izvora osvjetljenja u tranzitnoj zoni.

Drugi gumb tj. *clicked()* metoda uzima aproksimirani pravac u potpoglavlju 4.3. i računa osvijetljene unutar tranzitne zone svakih prijeđenih deset metara, sve do kraja tranzitne zone, te ispisuje rezultat u *QList* objektu u kojemu je i „Izračunaj“ gumb upisivao svoje rezultate. Unutar ove metode *clicked()* također postoji *while* petlja koja se ponavlja sve dok varijabla distance, koja predstavlja položaj unutar tranzitne zone, ne bude veća ili jednaka varijabli *duljinaTr*, u kojoj je spremljena duljina tranzitne zone. (Slika 5.7.)

```
vanjOsvjetljenje = (ui->inputTE->toPlainText());
vanjOsvjetljenost = vanjOsvjetljenje.toDouble();

Lth = koefK*vanjOsvjetljenost;
Ltrstart = 0.4*Lth;

vrijeme = pow((Lin/Lth),-0.714286)-1.9;
duljinaTr = (vrijeme*(brzina/3.6)*0.8);

L1 = Ltrstart;
L2 = Lin;
m2 = duljinaTr;
koefequation=(L2-L1)/(m2+m1-m1);
koefquationA = koefequation;
koefquationB = -koefequation*m1 + L1;
distance = 0;
udaljenost = m1;


while(distance<=duljinaTr){
    equationResult = koefquationA*udaljenost + koefquationB;
    length = QString::number(udaljenost,'g',10);
    result1 = QString::number(equationResult,'g',10);
    result2 = ("Na udaljenosti " + length + " m osvjetljenje je :" +  result1 + " cd/m^2");
     ui->resultsLW->addItem(result2);
    udaljenost = udaljenost + 10;
    distance = distance +10;
}
```

Slika 5.7. Kod za ispis rezultata osvijetljenosti unutar tranzitne zone.

Zadnji gumb, tj. *clicked()* metoda služi za brisanje ispisa unutar *QList* objekta. Metoda koja briše sve elemente liste poziva se kroz objekt napravljen iz klase *QList* i zove se *clear()* (Slika 5.8.).

```
void MainWindow::on_pushButton_clicked()
{
    ui->resultsLW->clear();
}
```

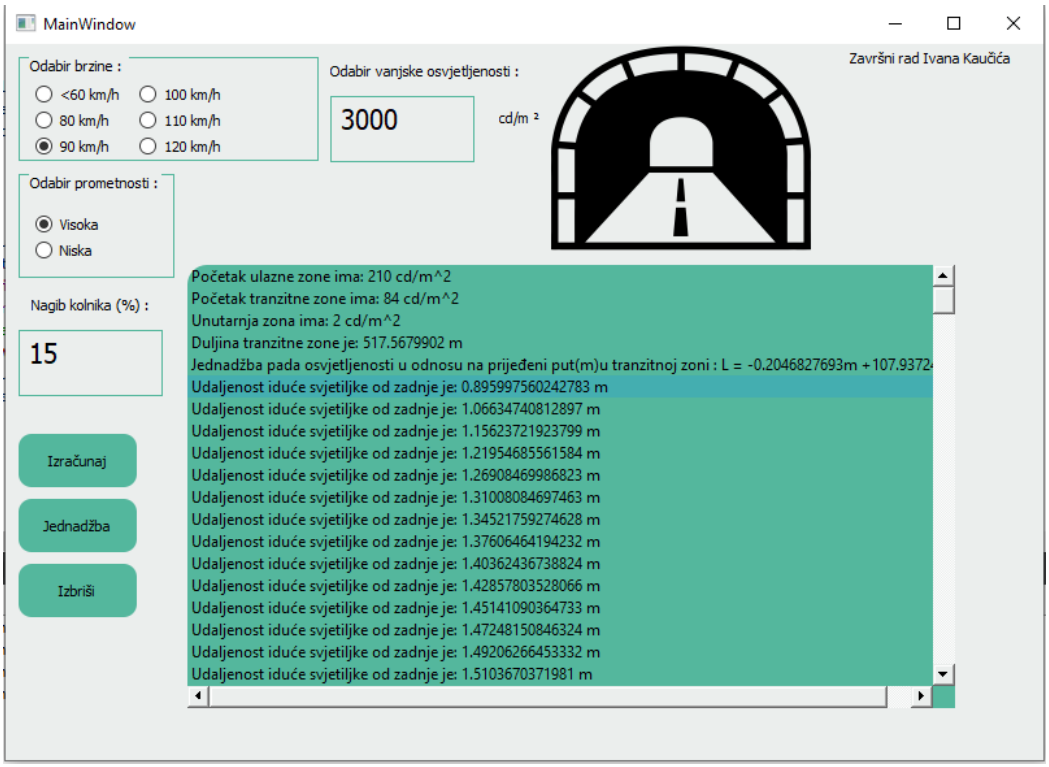
Slika 5.8. *Clicked()* metoda za brisanje elemenata *Qliste.*

5.4. Prikaz funkcionalnosti

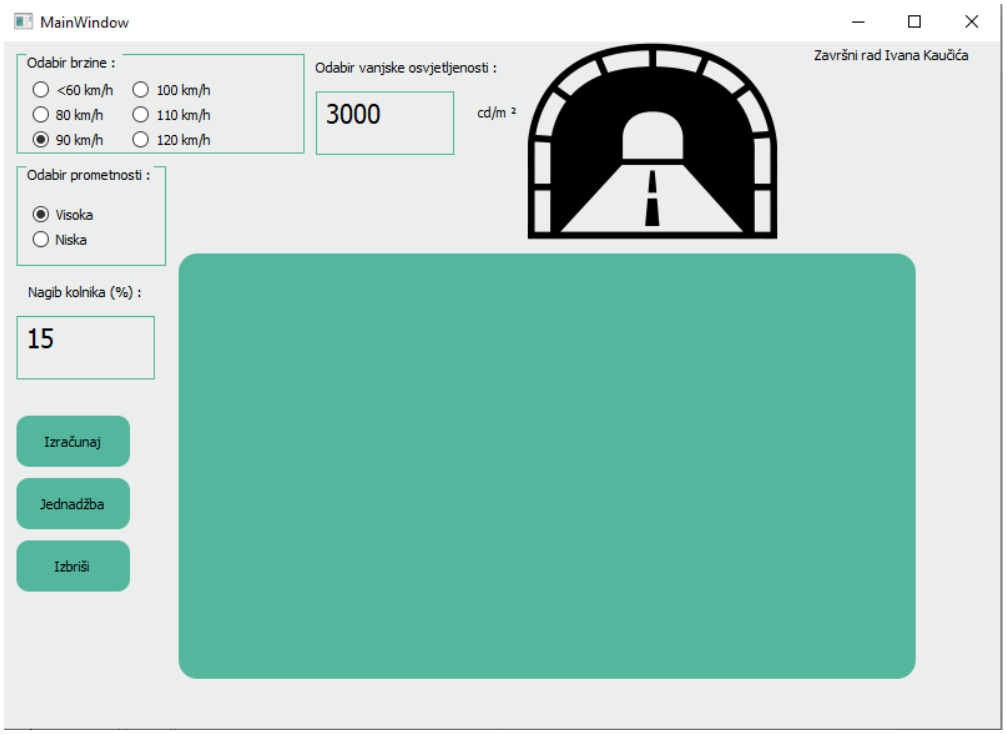
Funkcionalnost aplikacije se provjerava tako što pokrećemo kompletnu aplikaciju unutar Qt Creatora pritiskom na gumb *Run*, također postoji i opcija pokretanja aplikacije uz *debugging* koji je koristan za dodavanje točaka loma (eng. *Breakpoint*) na određenim linijama koda, ako dolazi do curenja memorije (eng. *Memory leak*) u nekim dijelovima aplikacije. Iznad gumba *Run* postoji i gumb čiji je zadatak napraviti datoteku u kojoj se nalazi *executable file* (program.exe) koji će biti kompatibilan sa svim operativnim sustavima za koje je aplikacija namjenjena. (Slika 5.9.)

Slika 5.9. Gumbovi za pokretanje aplikacije.

Nadopunjavajući aplikaciju parametrima potrebnim za računanje optimalnog razmještaja izvora osvjetljenja unutar tranzitne zone, aplikacija izbaci rezultate unutar *QList widgeta* pritiskom na gumb „Izračunaj“. (Slika 5.10.) Zatim odradimo brisanje sadržaja gumbom „Izbriši“ (Slika 5.11.), te možemo provjeriti osvjetljenje unutar tranzitne zone pritiskom na gumb „Jednadžba“ (Slika 5.12.).



Slika 5.10. Pritisak gumba „Izračunaj“ nakon popunjavanja parametara.



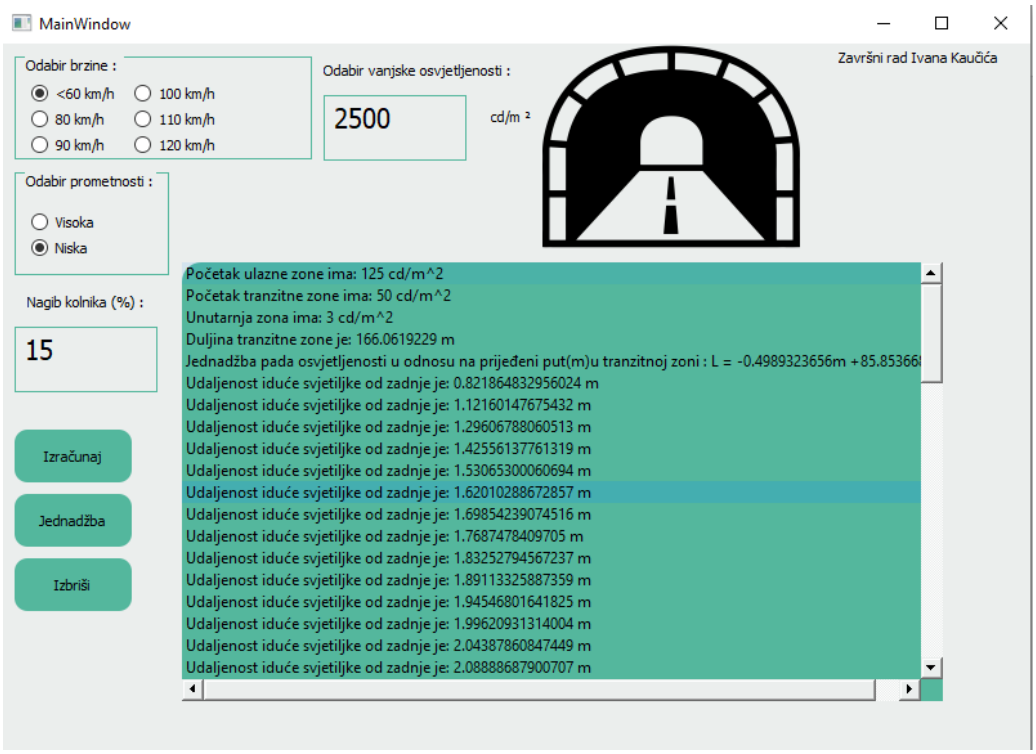
Slika 5.11. Aplikacija nakon pritiska na gumb „Izbriši“.

Slika 5.12. Ispis nakon pritiska na gumb „Jednadžba“.

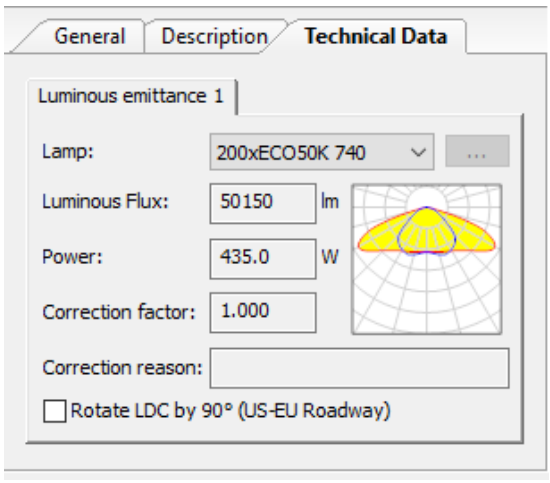
6. PROVJERA REZULTATA

Za provjeru rezultata koristi se najčešće, jer je najpopularniji i najpouzdaniji program za mjerenje osvjetljenja, *Dialux*. Pomoću njega se mjere i postavljaju rasvjete i osvjetljenja, što u uredskim zgradama, što u kućama, pa čak i ulične rasvjete i tuneli. Unutar samog programa postoje razni oblici pomoću kojih se mogu dizajnirati oblici kao što su kuće, zgrade i tuneli.

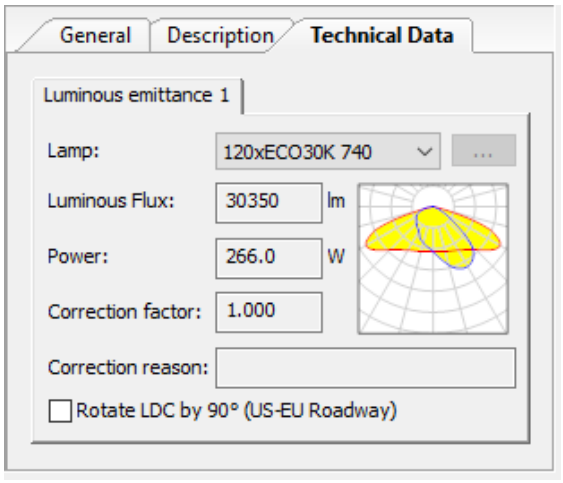
Unutar *Dialux-a* provjerit će se jedan slučaj (Slika 6.1.) tunela s dvije prometne trake koji je izrađen unutar aplikacije napravljene kroz ovaj rad. Međutim prije nego što se počne s izračunom osvjetljenja, prvo se mora izabrati vrsta rasvjete koja će se stavljati unutar tunela i tranzitne zone. Dobar izbor za slučaj (Slika 6.1.) biti će svjetiljke PHILIPS - TotalTunnel® FLOWSTAR BGB300 DTS / 50150lm 435W (Slika 6.2.), koja će osvjetljavati ulaznu zonu do potrebne razine. Unutar tranzitne zone koristit će se PHILIPS - TotalTunnel® FLOWSTAR + remote drivers BGB300 DTS / 30500lm 285W (Slika 6.3.). U unutarnjoj zoni, gdje su u svakom redu, svjetala po dvije svjetiljke, koristit će se PHILIPS - TotalTunnel® FLOWSTAR + remote drivers BGB312 DTS-WB / 8200lm 68W. (Slika 6.4.)



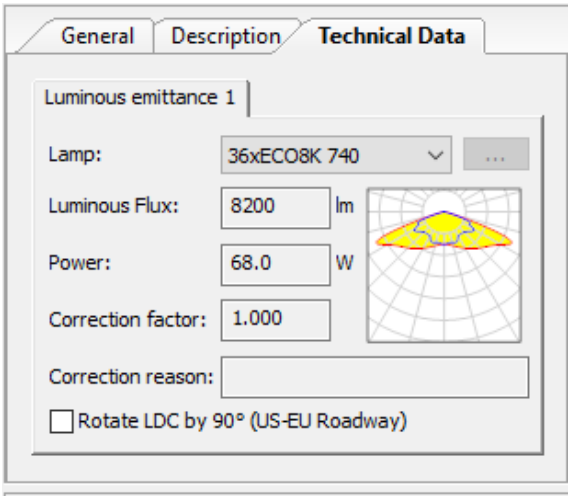
Slika 6.1. Rezultat računanja unutar aplikacije.



Slika 6.2. Svjetiljka za ulaznu zonu.

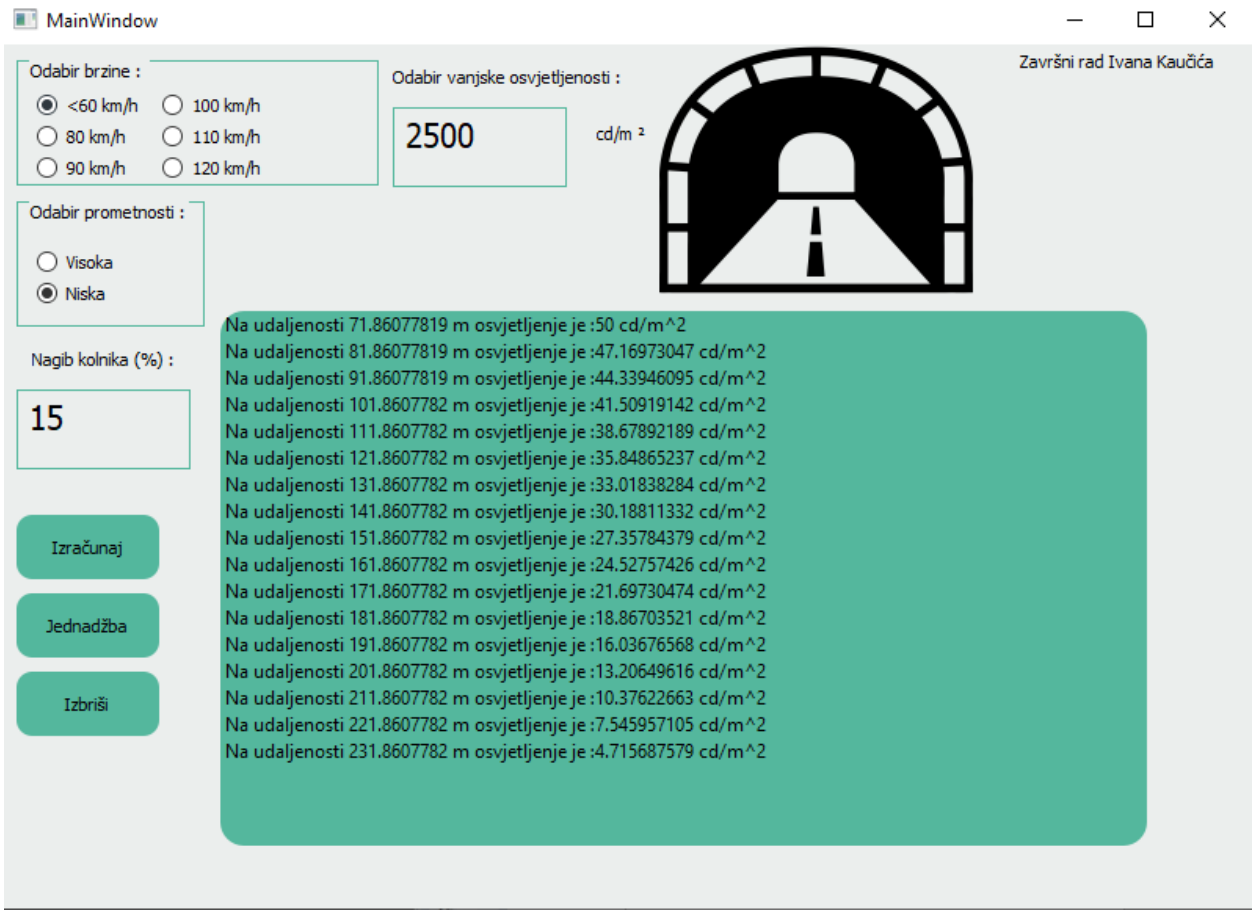


Slika 6.3. Svjetiljka za tranzitnu zonu.



Slika 6.4. Svjetiljka za unutarnju zonu.

Uz ovaj rezultat korištenja aplikacije, moramo još i odrediti padanje osvjetljenja unutar tranzitne zone, koje se dobiva jednostavno klikom na „Jednadžba“ unutar aplikacije. (Slika 6.4.)



Slika 6.4. Prikaz pada osvjetljenja unutar tranzitne zone.

Nakon odrađenog dijela unutar aplikacije, u programu *Dialux* radimo model tunela i postavljamo izvore rasvjete kao što je aplikacija izračunala. Nakon izrađenih svih elemenata unutar tunela i postavljanja rasvjete, pokrećemo akciju za računanje osvjetljenja unutar *Dialuxa*. Taj proces će trajati nekoliko minuta, te nakon toga možemo unutar *3-D viewa* „Šetati“ tunelom te ćemo na alatnoj traci ispod 3-D prikaza tunela, uz koordinate gdje se nalazimo imati i prikaz izračunate osvijetljenosti u *Dialuxu*. Stoga za bolji pregled rezultata i njihovog značenja pomoći će tablica 6.1.

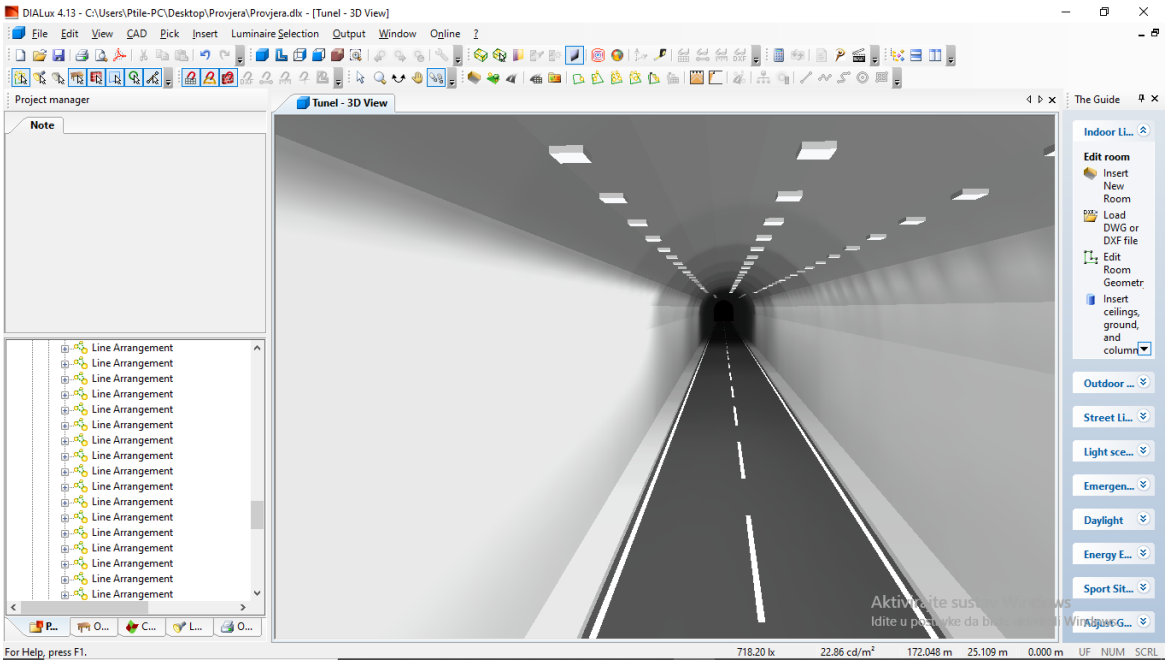
Tablica 6.1. Rezultati traženih i dobivenih vrijednosti osvjetljenja.

Udaljenost (m)	Potrebna razina osvjetljenja (cd/m^2)	Dobivena razina osvjetljenja (cd/m^2)	Razlika u razinama osvjetljenja (cd/m^2)	Razlika u razinama osvjetljenja (%)
72	50	68.72	18.72	37.44
82	47.17	63.50	16.33	34.62
92	44.34	56.22	11.88	26.80
102	41.51	43.13	1.62	3.90
112	38.68	41.54	2.86	7.39
122	35.85	38.40	2.55	7.11
132	33.02	36.28	3.26	9.85
142	30.19	32.44	2.25	7.45
152	27.36	28.61	1.25	4.57
162	24.53	26.80	2.27	9.25
172	21.70	22.86	1.16	5.35
182	18.87	21.05	2.18	11.6
192	16.04	18.02	1.98	12.34
202	13.21	15.34	2.13	16.12
212	10.38	10.63	0.25	2.41
222	7.55	8.21	0.66	8.74
232	4.72	4.95	0.23	4.87

Prvi stupac u tablici 6.1. predstavlja udaljenost unutar tunela (m). Drugi stupac prikazuje vrijednosti osvjetljenja izračunate aproksimiranim pravcem tranzitne zone, greška ovog pravca je veća što je veća razlika između osvjetljenja početka tranzitne zone i osvjetljenja unutarnje zone tunela. U trećem stupcu se nalaze vrijednosti osvjetljenja izračunate kroz model tunelske rasvjete u *Dialuxu*. Četvrti stupac prikazuje razliku izračunatog osvjetljenja trećeg i drugog stupca. Peti stupac dijeli dobivene razlike osvjetljenja u četvrtom stupcu s osvjetljenjem zadanim aproksimiranim pravcem u drugom stupcu i rezultat se množi sa sto kako bi se dobio postotak pogreške.

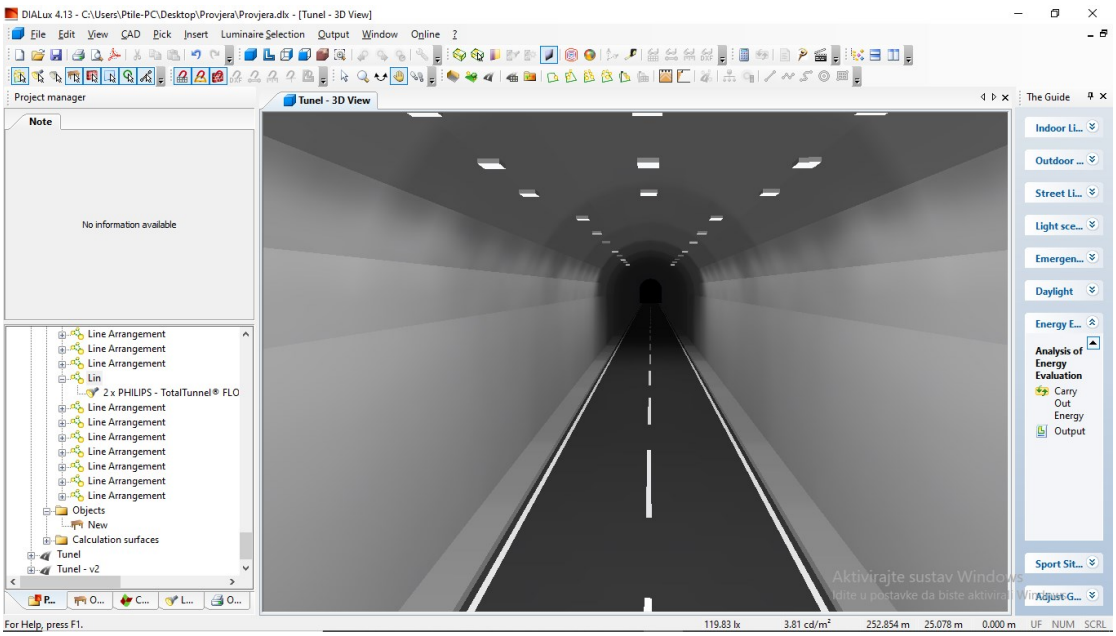
Iz tablice se može očitati da je najviša razina pogrešne osvijetljenosti na početku tranzitne zone, gdje razina pogreške drastično opada prolaskom kroz tranzitnu zonu.

Za prikaz tunela i osvjetljenja unutar njega, izabrana je slika 6.5. gdje na udaljenosti od sto sedamdeset dva metara možemo očitati osvijetljenost.



Slika 6.5. Osvjetljenje na 172m

Iz priložene slike u *Dialuxu* se vidi da je unutar simulacije tunela osvjetljenje na sto sedamdeset i drugom metru tunela 1.16 cd/m^2 različita od očekivane, što nije uopće veliko odstupanje. Uz to valja pogledati i simulirani rezultat početka unutrašnje zone tunela unutar *Dialuxa*. (Slika 6.6.)



Slika 6.6. Osvjetljenje na početku unutarnje zone.

Iz prikaza unutarnje zone može se vidjeti da je osvjetljenje unutarnje zone 3.85 cd/m^2, što je za 0.85 cd/m^2 različito od pretpostavljene vrijednosti, što opet, nije veliko niti kritično odstupanje za ljudsko oko.

7. ZAKLJUČAK

Tema ovog završnog rada je izrada aplikaciju u Qt programskom alatu, koja računa optimalan raspored izvora osvjetljenja unutar tranzitne zone tunela. Problem u dizajnu osvjetljenja unutar tunela je prilagodba ljudskog oka na prijelaz iz prostora visoke osvijetljenosti u prostor puno niže osvijetljenosti. Stoga postoje pravila kojih se treba pridržavati prilikom dizajniranja tunelske rasvjete. Jedno od pravila je da duljina zone praga tunela mora biti duljine zaustavnog puta, koji opet ovisi o brzini kretanja vozila unutar tunela. Iduće pravilo je da vrijednost osvjetljenja prilikom ulaza u tranzitnu zonu mora biti četrdeset posto vrijednosti osvjetljenja u zoni praga. Osvjetljenje u tranzitnoj zoni je opisano krivuljom zadanom CIE88 standardom, koja je padajuća i završava kada se dosegne željena vrijednost osvjetljenja. Unutar prvog dijela aplikacije, uzeta je krivulja osvjetljenja tranzitne zone i aproksimirana je pravcem, koji govori kako osvjetljenje pada prolaskom kroz tranzitnu zonu i koje vrijednosti mora imati na određenoj udaljenosti. Aplikacija ispisuje vrijednosti osvjetljenja za svakih 10 metara unutar tranzitne zone. Drugi dio aplikacije koristi izračunati koeficijent pada osvjetljenja, koji se dobije dijeljenjem razlike krajnjeg i početnog osvjetljenja u tranzitnoj zoni s duljinom tranzitne zone. Taj koeficijent se koristi u formuli za određivanje udaljenosti susjednih redova rasvjete. U ovom dijelu aplikacije ispisuju se razmaci između susjednih redova rasvjete. Za izračun svih podataka u aplikaciji moraju se unijeti neophodni podaci za proračun. Ulazni podaci su brzina kretanja vozila, gustoća prometa, vanjsko osvjetljenje i nagib kolnika. Pomoću gore objašnjena dva dijela aplikacije prikupljen je skup informacija potreban za projektiranje osvjetljenja unutar tranzitne zone koristeći program Dialux. U Dialuxu su korištene svjetiljke koje će što točnije ispuniti uvjete zadane u aplikaciji. Sami moramo birati i isprobavati svjetiljke jer se ne može iskazati izravna veza između karakteristike svjetiljki i osvjetljenja u aplikaciji.

Aplikacija se može poboljšati tako što bi se izravno mogla povezati karakteristika svjetiljki s osvjetljenjem i pomoću te veze točnije i pouzdanije odrediti razmak između susjednih svjetiljki, a da je u granicama koje propisuje aproksimirani pravac osvjetljenja tranzitne zone.

LITERATURA

[1] Davor Beck, Damir Blažević, Hrvoje Dragovan - Projektiranje dnevne rasvjete tunela - analiza zahtjeva, izrada modela i izračun [23.9.2020.]

[2] Wikipedia , Qt tools, dostupno na: https://en.wikipedia.org/wiki/Qt_(software) [23.9.2020.]

[3] Ransen's Lighting And Color Blog, dostupno na: https://ransenlightingandcolor.blogspot.com/2017/01/cie88-2004-calculation-of-tunnel.html [23.9.2020.]

[4] Huo-Yen Liu, Design criteria for tunnel lightning, 2005. [23.9.2020.]

[5] Službena wikipedija Qt stranice, dostupna na: https://www.qt.io/product/develop-embedded-devices [23.9.2020.]

[6] Robert Veseli, Izrada aplikacije koristeći Qt skup programskih alata – diplomski rad, 2017. godine, dostupno na: https://repozitorij.etfos.hr/islandora/object/etfos%3A1460/datastream/PDF/view [23.9.2020.]

[7] Thorn, Tunnel Lightning: http://www.thornlighting.com/download/TunnelINT.pdf [23.9.2020.]

[8] CIE Tehnical report, Guide for the lighting of road tunnels and underpasses 2nd edition [23.9.2020.]

[9] Reulux Desktop aplikacija, dostupna na: https://reluxnet.relux.com/en/relux-desktop.html [30.9.2020.]

[10] Vivaldi program za dizajniranje osvjetljenja, dostupan na: https://www.zumtobel.com/com-en/vivaldi.html [30.9.2020.]

SAŽETAK

Cilj završnog rada bio je napraviti aplikaciju pomoću Qt alata, koja računa optimalan razmak između izvora osvjetljenja unutar tranzitne zone tunela. Zbog bioloških karakteristika, ljudsko oko ima slabiju moć prilagodbe kada ulazi s prostora visoke osvijetljenosti u prostor puno niže osvijetljenosti. Stoga se tunel sastoji od pet zona, koje pomažu prilagodbi ljudskog oka. Tih pet zona su: ulazna, zona praga, tranzitna, unutarnja i izlazna. Programi za izračune osvjetljenja u tunelima postoje, ali nisu ograničeni samo za tunelsku rasvjetu. Koriste se za dizajniranje osvjetljenja u kućama, stambenim zgradama, vanjskog osvjetljenja i sl. Neki od tih programa su *Relux*, *Dialux* i *Vivaldi*. Qt alat je korišten za izradu grafičko korisničkog sučelja i za pokretanje aplikacija koje se mogu koristiti na svim velikim *desktop* i mobilnim platformama, uključujući i neke ugradbene platforme. Aplikacija završnog rada pruža unos ulaznih podataka važnih za izračun, ispis razina osvjetljenja u tranzitnoj zoni i ispis udaljenosti susjednih izvora osvjetljenja. Potrebne vrijednosti koje se moraju iskazati formulama i izračunati su: zaustavna udaljenost, osvjetljenje zone praga, osvjetljenje unutarnje zone, osvjetljenje početka tranzitne zone, duljina tranzitne zone, koeficijent pada osvjetljenja, aproksimirani pravac krivulje osvjetljenja tranzitne zone i algoritam za postavljanje rasvjete u tranzitnoj zoni. Algoritam povezuje koeficijent pada osvjetljenja i redni broj reda osvjetljenja te se rasvjeta postavlja sve dok suma razmaka nije veća od duljine tranzitne zone. Aplikacija u grafičko korisničkom sučelju sadrži tri gumba koja omogućuju računanje i ispis osvijetljenosti u tranzitnom dijelu tunela, ispis razmaka između susjednih izvora osvjetljenja i brisanje svih izračunatih ispisa. Rezultati izrađene aplikacije provjeravaju se na način da ih uspoređujemo s izračunom pomoću modela tunela u *Dialuxu*.

Ključne riječi: osvijetljenost, tuneli, tranzitna zona, Qt alati, prilagodba ljudskog oka

ABSTRACT

Creating application for optimal placement of light sources

The goal of this Bachelor's thesis was to create an application for optimal placement of light sources inside of transizion zone using Qt tools. Becouse of biological properties, human eye has a slightly weaker power of adjustment when exiting space with high luminance and entering space with low luminance. Therefore, tunnels have five zones that help human eye to adapt. Those five zones are: entry zone, threshold zone, tranzition zone, inner zone and exit zone. Applications that calculate luminance exist, but they are not limited to tunnel lighting only. They are used for designing luminance in houses, apartment building, outdoor lighting etc. Some of those applications are Relux, Dialux and Vivaldi. Qt tools are used for creating graphic user interfaces and running applications that can be used on all the biggest dektop and mobile platforms. The application of this Bachelor's thesis provides all of the important input entries used for caluculations, outuput of luminance in tranzition zone and distance of neighbouring lighting sources. The needed values and expressions that are described with formulas and calculations are: stopping distance, luminance of threshold zone, luminance of inner zone, luminance of starting point of tranzition zone, length of tranzition zone, coefficient of luminance drop, approximated line of curve that describes luminance inside transition zone and algorithm for lighting placement inside of tranzition zone. Algorithm connects coefficinet of luminance drop with ordinal number of row of lighting. The lighting is placed until the sum of interspaces, of neighbouring lighting sources, is greater than the length of transition zone. This application contains three buttons inside of it's graphis user interface. The buttons allow you to calculate and output luminance of transition zone, distance between neighbouring lighting sources, and deleting whole lists of these results. Results of this application can be checked and compared with a tunnel lighting simulation in Dialux.

Key words: luminance, tunnels, transition zone, Qt tools, human eye adaptation

ŽIVOTOPIS

Ivan Kaučić rođen je 4. siječnja 1998. godine u Zagrebu. Svoje osnovnoškolsko obrazovanje započinje 2004. godine u Osnovnoj školi Davorin Trstenjak u Čađavici. 2006. godine, zajedno s obitelji, seli u Slatinu i tamo nastavlja svoje osnovnoškolsko obrazovanje do 2012. godine u Osnovnoj školi Josip Kozarac. 2012. upisuje Opću gimnaziju u Srednjoj školi Marka Marulića u Slatini koju završava 2016. godine. Iste godine, upisuje se na Fakultet elektrotehnike, računarstva i informacijskih tehnologija u Osijeku na sveučilišni preddiplomski studij računarstva.