



# УНИВЕРЗИТЕТ У НОВОМ САДУ ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА



УНИВЕРЗИТЕТ У НОВОМ САДУ  
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА  
НОВИ САД  
Департман за рачунарство и аутоматику  
Одсек за рачунарску технику и рачунарске комуникације

## ЗАВРШНИ (BACHELOR) РАД

Кандидат: Виолета Чикош  
Број индекса: RA 75/2011

Тема рада: Једно решење 3D корисничког окружења за приказ података на контролној табли возила

Ментор рада: Иван Каштелан

Нови Сад, Јун, 2015



## КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, РБР:			
Идентификациони број, ИБР:			
Тип документације, ТД:	Монографска документација		
Тип записа, ТЗ:	Текстуални штампани материјал		
Врста рада, ВР:	Завршни (Bachelor) рад		
Аутор, АУ:	Виолета Чикош		
Ментор, МН:	Иван Каштелан		
Наслов рада, НР:	Једно решење 3D корисничког окружења за приказ података на контролној табли возила		
Језик публикације, ЈП:	Српски / латиница		
Језик извода, ЈИ:	Српски		
Земља публиковања, ЗП:	Република Србија		
Уже географско подручје, УГП:	Војводина		
Година, ГО:	2015		
Издавач, ИЗ:	Ауторски репринт		
Место и адреса, МА:	Нови Сад; трг Доситеја Обрадовића 6		
Физички опис рада, ФО: (поглавља/страница/цитата/табела/слика/графика/прилога)	7/23/0/3/10/0/0		
Научна област, НО:	Електротехника и рачунарство		
Научна дисциплина, НД:	Рачунарска техника		
Предметна одредница/Кључне речи, ПО:	OBD II, графичка корисничка спрела, рачунар у возилу, контролна табла		
УДК			
Чува се, ЧУ:	У библиотеци Факултета техничких наука, Нови Сад		
Важна напомена, ВН:			
Извод, ИЗ:	У овом раду је приказано једно решење 3D приказа података на контролној табли возила. Подаци који су приказани првенствено укључују брзину возила и број обртаја мотора, статус горива, статуса уља итд. Промене приказаних параметара су омогућени преко тастатуре.		
Датум прихватања теме, ДП:			
Датум одбране, ДО:			
Чланови комисије, КО:	Председник:	др Јелена Ковачевић	
	Члан:	др Небојша Ђевалица	Потпис ментора
	Члан, ментор:	др Иван Каштелан	



## KEY WORDS DOCUMENTATION

Accession number, <b>ANO:</b>			
Identification number, <b>INO:</b>			
Document type, <b>DT:</b>	Monographic publication		
Type of record, <b>TR:</b>	Textual printed material		
Contents code, <b>CC:</b>	Bachelor Thesis		
Author, <b>AU:</b>	<b>Violeta Cikos</b>		
Mentor, <b>MN:</b>	<b>Ivan Kastelan</b>		
Title, <b>TI:</b>	<b>One solution of 3D user interface for data display on the control panel of the vehicle</b>		
Language of text, <b>LT:</b>	Serbian		
Language of abstract, <b>LA:</b>	Serbian		
Country of publication, <b>CP:</b>	Republic of Serbia		
Locality of publication, <b>LP:</b>	Vojvodina		
Publication year, <b>PY:</b>			
Publisher, <b>PB:</b>	Author's reprint		
Publication place, <b>PP:</b>	Novi Sad, Dositeja Obradovica sq. 6		
Physical description, <b>PD:</b> (chapters/pages/ref./tables/pictures/graphs/appendices)	7/23/0/3/10/0/0		
Scientific field, <b>SF:</b>	Electrical Engineering		
Scientific discipline, <b>SD:</b>	Computer Engineering, Engineering of Computer Based Systems		
Subject/Key words, <b>S/KW:</b>	OBD II, graphical user interface, In-Vehilce Infotainment, instrument panel		
<b>UC</b>			
Holding data, <b>HD:</b>	The Library of Faculty of Technical Sciences, Novi Sad, Serbia		
Note, <b>N:</b>			
Abstract, <b>AB:</b>	<p>In this paper is presented one solution of 3D user interface for data display on the control panel in vehicle. Presented data primarily include vehicle speed and engine speed, fuel status, the status of oil and so on. Changes the displayed parameters are enabled with keyboard.</p>		
Accepted by the Scientific Board on, <b>ASB:</b>			
Defended on, <b>DE:</b>			
Defended Board, <b>DB:</b>	President:	Jelena Kovačević, PhD	
	Member:	Nebojša Pjevalica, PhD	Menthor's sign
	Member, Mentor:	Ivan Kaštelan, PhD	

## **Zahvalnost**

Zahvaljujem se Nenadu Jovanoviću, Tomislavu Maruni, Branimiru Kovačeviću, Marku Kovačeviću i Ivanu Kaštelanu na stručnoj pomoći i savetima pri izradi ovog rada.

Zahvalnost dugujem i svojim članovima porodice na pruženoj podršci u toku čitavog školovanja, a posebno deda Đuri na pruženoj finansijskoj podršci.



# УНИВЕРЗИТЕТ У НОВОМ САДУ

## ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА



### SADRŽAJ

1.	Uvod .....	1
2.	Teorijske osnove.....	3
2.1	Android operativni sistem.....	3
2.1.1	Korisnička sprega .....	3
2.2	OBD II sistem .....	4
2.3	Računarska grafika.....	4
2.3.1	Grafičko korisničko okruženje .....	5
2.3.2	3D računarska grafika.....	5
2.3.3	OpenGL ES .....	5
2.3.4	libGDX .....	6
2.4	C-More razvojno okruženje .....	6
2.4.1	Pozicija C-More razvojnog okruženja u Android aplikacijama .....	6
2.4.2	Arhitektura softvera.....	7
3.	Koncept rešenja .....	9
3.1	Analiza zadatka .....	9
3.2	Dizajn.....	10
4.	Implementacija .....	13
4.1	Dizajnersko rešenje.....	13
4.2	Modularnost .....	16
4.2.1	Modul za inicijalizaciju sveta .....	16
4.2.2	Modul za prikaz scene aktivne tokom vožnje .....	16
4.2.3	Modul za prikaz scene sa dodatnim opcijama.....	17
4.2.4	Modul sa funkcijama za animaciju scena .....	18
5.	Rezultati.....	20
6.	Zaključak .....	22
7.	Literatura .....	23

## SPISAK SLIKA

Slika 1 Prikaz savremene kontrolne table automobila.....	1
Slika 2. C-More pozicija u Android softveru .....	7
Slika 3. Glavni moduli C-More UI Frameworka.....	8
Slika 4. Skica uobičajenog izgleda kontrolne table .....	10
Slika 5. Skica uobičajenog izgleda kontrolne table .....	11
Slika 6. Skica prelaza sa jedne na drugu celinu .....	11
Slika 7. Skica prikaza druge celine .....	12
Slika 8. Prikaz nekih od modela koje sadrži aplikacija .....	14
Slika 9. Prikaz prve scene .....	15
Slika 10. Prikaz druge scene .....	15

## **SPISAK TABELA**

Tabela 1. Funkcije koje su definisane u klasi za animaciju scena.....	19
Tabela 2. Ispunjenošć zahteva.....	20
Tabela 3. Rezultati testiranja aplikacije .....	21

## SKRAĆENICE

API – *Application Programming Interface*, programska sprega

ECU – *Engine Control Unit*, centralni računar za upravljanje radom motora

GPU – *Graphics Processing Unit*, grafički processor

IVI – *In-Vehicle Infotainment*, računar u automobilu

OBD – *On-Board Diagnostics*, dijagnostika na uređaju (automobilu)

OpenGL ES – *Open Graphics Library for Embedded Systems*, otvorena grafička biblioteka za razvoj ugrađenih uređaja

PC – *Personal computers*, lični računar

## 1. Uvod

Razvojem elektronike tradicionalni analogni instrumenti koji su se koristili za prikaz podataka na kontrolnoj tabli vozača, zamenjeni su digitalnim instrumentima. Uz dalji razvoj elektronike kao i uz razvoj računara i grafičkog hardvera, digitalni instrumenti su zamenjeni LCD (engl. *liquid crystal display*) ekranima. Takođe, povećan je spisak informacija koje se prikazuju. Pored osnovnih informacija (brzina automobila, broj obrtaja motora, nivo goriva, temperatura motora itd.) savremene kontrolne table sadrže operativni sistem koji pružaja mogućnost: prikaza navigacije, izbor radio stanica, povezivanje na internet, povezivanje sa uređajima kao što su pametni telefoni itd. Primer jedne savremene kontrolne table može se videti na slici 1.



Slika 1 Prikaz savremene kontrolne table automobila

Postoji težnja da se informacije prikažu što realnije i da budu što približnije korisniku. Ovo je jedan od razloga zašto je u aplikacijama sve više zastupljen 3D svet. U automobilskoj industriji razvoj računarske grafike doneo je mnoge pogodnosti. Sve ovo omogućava

korisniku bolju preglednost i bolju dostupnost informacija kao i izbor informacija koje korisnik želi da prikaže.

U ovom radu je opisana realizacija aplikacije pod nazivom “ClusterUI”. Ona predstavlja jedno rešenje 3D korisničke sprege za prikaz podataka na kontrolnoj tabli automobila. Razvoj aplikacije je vršen na personalnom računaru (engl. *Personal Computer – PC*). Ciljna platforma je Android.

Rad se sastoji iz nekoliko poglavlja.

Prvo poglavlje ukratko objašava temu rada i opisuje njegov sadržaj.

Drugo poglavlje uvodi čitaoca u okruženje u kome je rešenje nastalo i navodi neke teorijske osnove koje će čitaocu olakšati razumevanje daljeg teksta. U ovom poglavlju su objašnjene osnove Android operativnog sistema, osnove računarske grafike, biblioteke koje se koriste za razvoj 3D aplikacija kao i C-More razvojno okruženje za razvoj grafičke korisničke sprege.

Treće poglavlje se sastoji od izlaganja zahteva koje rešenje treba da zadovolji i objašnjenja njihovog ispunjavanja.

Četvrto poglavlje predstavlja objašnjenje rešenja ovog zadatka. Uvodi čitaoca u programsko rešenje i njegovu implementaciju.

Peto poglavlje sadrži prikaz rezultata aplikacije.

Šesto poglavlje sadrži kratak pregled onoga što je do sada urađeno u aplikaciji. Priloženi su rezultati testiranja aplikacije kao i planovi za dalji razvoj aplikacije.

U sedmom poglavlju se nalazi spisak literature koja je korišćenja za izradu ovog rada.

## 2. Teorijske osnove

Radi boljeg razumevanja daljeg opisa rešenja, u ovom poglavlju je objašnjen sistem za dijagnostiku u uređaju (engl. *On-Board Diagnostic – OBD*), osnove kompjuterske grafike kao i C-More razvojno okruženje za razvoj grafičke korisničke sprege na kome je zasnovana ova aplikacija.

### 2.1 Android operativni sistem

Android operativni sistem je zasnovan na Linux jezgru koji se trenutno razvija od strane kompanije Google. Android je primarno namenjen za prenosive uređaje osetljive na dodir. Ekspanziju doživljava rasprostiranjem mobilnih uređaja kao što su pametni telefoni i tablet uređaji.

Android je trenutno najrasprostranjeniji operativni sistem za pametne telefone. Popularan je i među tehnološkim kompanijama koje zahtevaju gotove, jeftine, prilagodive i lage operativne sisteme za svoje visokotehnološke uređaje. Pored toga što je prvenstveno namenjen za pametne telefone i tablet uređaje, Android je našao primenu i kod televizora, igračkih konzoli, digitalnih kamera i druge elektronike. Takođe je našao i primenu i u računarima u automobilu (engl. *In-Vehicle Infotainment – IVI*).

#### 2.1.1 Korisnička sprega

Korisnička sprega (engl. *User Interface – UI*) je sve ono što korisnik može da vidi i sa čime može da rukuje. Kod Android operativnog sistema je zasnovana na direktnoj manipulaciji objektima na ekranu, korišćenjem ulaza u vidu dodira koji odgovaraju pokretima u realnom svetu ili pritiskom tastera. Dodatni hardver kao što su akcelerometar, žiroskop i senzor brzine koriste se za dodatne zahteve korisnika, kao što su na primer podešavanje orijentacije ekrana u zavisnosti od položaja uređaja.

## 2.2 OBD II sistem

OBD II sistem se u današnje vreme nalazi u većini automobila i kamiona. Tokom '70ih godina i početkom '80ih proizvođači su počeli da koriste elektroniku za kontrolisanje funkcija motora i dijagnostiku problema motora. Tokom godina OBD II sistem je postao sofisticiraniji. OBD II sistem, novi standard koji je predstavljen sredinom '90ih, pruža kontrolu motora kao i praćenje ostalih delova automobila i pomoćnih uređaja.

Proizvođači su počeli sa uključivanjem OBD II sistema u proizvodnju raznovrsnih modela u ranim 90-im godinama. Svi automobili proizvedeni od 1. januara 1994. imaju OBD II sistem.

Pre pojave OBD II sistema, automobili su imali konektore na različitim pozicijama u okviru kontrolne table i ispod haube. Svi automobili sada imaju konektor koji se nalazi u putničkom prostoru lako dostupnom sa sedišta vozača. Ovaj konektor se povezuje kablom alata za skeniranje koji su dostupni na PC-u.

OBD II signali su najčešće traženi kao odgovor na osvetljeni indikator kvara ili kao odgovor na probleme koji se pojavljuju u toku vožnje. Podaci koje pruža OBD II sistem ukazuju na specifičnu komponentu koja je u kvaru čime se štedi dosta vremena i troškova u odnosu na popravke u kojima se nagađa koja komponenta je u kvaru. Takođe, oni mogu dati korisne informacije o stanju korišćenog automobila[1].

## 2.3 Računarska grafika

Vizuelno računarstvo je računarstvo koje dozvoljava interakciju i kontrolisani rad rukujući sa vizuelnim slikama kao direktnim objektima ili kao objektima koji predstavljaju druge objekte koji sami po sebi nisu vizuelni. Vizuelne slike mogu biti: fotografije, 3D scene, video sekvene, blok dijagrami ili jednostavne ikonice.

Računarska grafika je polje vizuelnog računarstva gde se pomoću računara stvara slika. Ona predstavlja stvaranje, obradu, doradu, modelaciju i vizuelizaciju sintetičkih slika, kao i animacija u dve ili tri dimenzije uz pomoć računara i računarskih programa koji su namenjeni za tu svrhu. Računarska grafika danas ima široku primenu u nauci, industriji, arhitekturi i dizajnu. Računarska grafika se može podeliti u nekoliko polja: 3D izvođenje (engl. *render*) slike u realnom vremenu (koristi se u računarskim igrama), računarska animacija, hvatanje (engl. *capture*) i stvaranje videa, stvaranje specijalnih efekata, potpuno ili delimično računarski stvorenih filmova, obrada slike i modeliranje (koristi se u medicinske svrhe, kao i u arhitekturi). [2]

U ovom radu pojednostavlja virtuelni objekat definisan skupom tačaka koji se može posmatrati iz različitih pravaca, presecati, uobičavati i slično. On sadrži određeni skup

podataka. Ti podaci mogu sadržati podatke o geometriji, tački gledišta, teksturi i podatke o osvetljenju. Render je postupak stvaranja slike od nekog modela uz pomoć posebnog programa. Engleska reč *Render* se takođe koristi kod opisivanja procesa izračunavanja specijalnih efekata u obradi videa. Ovaj proces se koristi kod računara i računarskih igara, filmova, simulatora kao i u vizualizaciji dizajna.

### 2.3.1 Grafičko korisničko okruženje

Grafičko korisničko okruženje (engl. *Graphical user interface - GUI*) je softversko okruženje koje omogućava korisniku komunikaciju sa kompjuterom koristeći prethodno definisane funkcije. Najzaslužniji za njegov razvoj su razvoj Apple Macintosh i Microsoft Windows operativnih sistema. Grafičko korisničko okruženje je zamenilo okruženje komandne linije (engl. *Command line interface - CLI*) koje koristi samo tekst i rukovanje isključivo preko tastature. Osim tastature, u grafičko-korisničkom okruženju komande se zadaju i pomoću miša, pokazivačke ploče ili kugle za upravljanje. Programi, komande, datoteke, direktorijumi ili uređaji predstavljeni su pomoću malih slika ili simbola koje nazivamo ikonicama.[3]

### 2.3.2 3D računarska grafika

Razvojem grafičkih procesora došlo je do pojave 3D računarske grafike, osnovane na vektorskoj grafici. Vektorska grafika je način prikazivanja slike pomoću geometrijskih oblika kao što su tačke, linije, krive i poligoni. Princip je skoro isti, umesto čuvanja podataka o tačkama, linijama i krivim u 2D prostoru, računar ih čuva u tro-dimenzionalnom prostoru. 3D računarska grafika se bazira na smeštanju tačaka koje se nazivaju temena (engl. *vertex*), linija koje spajaju temena i strana koje se formiraju definišući formu poligona. Poligon je prostor formiran od najmanje tri temena. To su bili počeci, danas osim smeštanja 3D poligona u memoriju, grafički softver može izvršavati komplikovanije stvari kao što su senčenje ili dodavanje tekstura koji daju osećaj stvarnosti nekom računarski napravljenom objektu.

### 2.3.3 OpenGL ES

OpenGL, biblioteka podržana od strane različitih jezika i koja je raspoloživa na različitim platformama, je programska sprega (engl. *Application Programming Interface - API*) koji služi za renderovanje dvo-dimenzionalne i tro-dimenzionalne vektorske grafike. API se obično koristi za interakciju sa GPU da bi se postioglo hardverski –ubrzano renderovanje.

OpenGL Embedded Systems (OpenGL ES) je podsistem OpenGL-a. On je dizajniran za ugrađene (engl. *embedded*) sisteme kao što su pametni telefoni, tablet uređaji, igračke konzole

i lični digitalni asistenti. API je raspoloživ na različitim platformama i podržan od strane različitih jezika. Android razvojno okruženje za razvoj grafičke korisničke sprege pruža mnoštvo standardnog alata za stvaranje atraktivnih, funkcionalnih grafičkih korisničkih sprega. Ukoliko je potrebno više kontrole prilikom iscrtavanja na ekran ili prilikom postavljanja objekata u 3D grafici, standardni alati nisu dovoljni. OpenGL ES API obezbeđen od strane Android framework-a nudi set alata za prikaz grafike [4][5].

### 2.3.4 libGDX

libGDX je razvojno aplikativno okruženje za razvoj grafičke korisničke sprege raspoloživo na različitim platformama koje se koristi za razvoj igrica. Pisan je u Java programskom jeziku uz dodatak C i C++ komponenata. Ovo omogućava razvoj PC i mobilnih igrica koristeći istu bazu koda. Podržava Windows, Linux, Mac OS X, Android, iOS i web pretraživače sa WebGL podrškom [6].

## 2.4 C-More razvojno okruženje

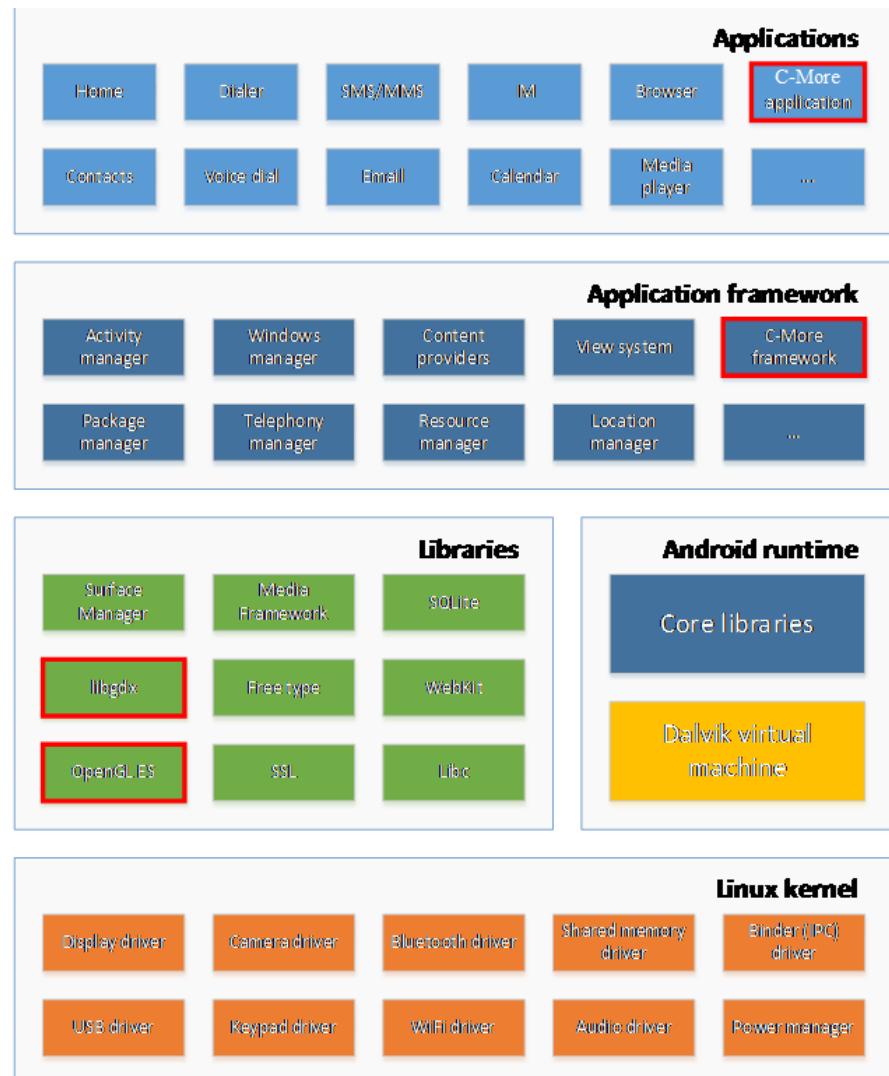
C-more razvojno korisničko okruženje za razvoj grafičke korisničke sprege je 3D Java razvojno okruženje za razvoj grafičke korisničke sprege. Okruženje je raspoloživo na različitim platformama bazirano na OpenGL standardu, koja radi na Windows, Linux, Mac, OS X, Android, iOS platformama kao i na pregledaćima Interneta koji podržavaju WebGL. C-More razvojno okruženje je razvijano na istraživačko-razvojnem institutu RT-RK.

### 2.4.1 Pozicija C-More razvojnog okruženja u Android aplikacijama

C-More softver je integriran u Android-u na različitim nivoima:

- Aplikacioni nivo: sadrži aplikacije napisane na vrhu Android API-ja.
- Aplikaciono okruženje: sadrži Java biblioteke koje podržavaju Android aplikacije. Ovaj nivo sadrži C-More razvojno okruženje.
- Nativne biblioteke: sadrže nativne biblioteke koje podržavaju Android aplikaciono okruženje. Ovaj nivo sadrži libGDX i OpenGL ES (standardna Android biblioteka) koje koristi C-More razvojno okruženje.

Grafički prikaz pozicije C-More razvojnog okruženja u Android softveru se nalazi na slici 2.



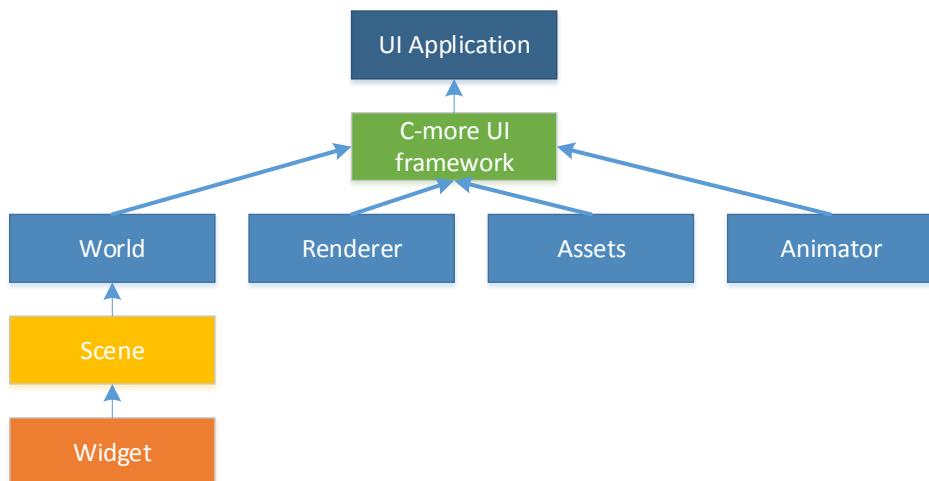
Slika 2. C-More pozicija u Android softveru

#### 2.4.2 Arhitektura softvera

C-More razvojno okruženje se koristi za razvijanje 3D UI aplikacija na raznim platformama kao što su PC i Android. Prednosti C-More razvojnog okruženja su te da programeri mogu da naprave veoma napredne 3D UI aplikacije pisane u Javi, koristeći sve prednosti OpenGL-a koje omogućavaju sjajne performanse.

C-More UI razvojno okruženje za razvoj grafičke strukture je usko povezan sa libGDX razvojnim okruženjem za razvoj grafičke strukture koji je podržan od strane različitih platformi i približava API za Android platformu.

Na sledećoj slici su prikazani glavni moduli C-More razvojnog okruženja za razvoj grafičke korisničke sprege.



Slika 3. Glavni moduli C-More UI Frameworka

UI aplikacije se razvijaju u Javi za ciljanu platformu. Aplikacije za grafičko renderovanje koriste API koji je izведен iz modula C-More grafičkog okruženja.

C-More Framework sadrži četiri glavna modula:

- Svet modul (engl. *World module*): ovaj modul je mesto gde su postavljene sve UI aplikacione komponente. Svet je napravljen od scena u 3D svetu od različitih grafičkih elemenata. Scene mogu biti predstavljene kao nezavisni grafičke celine unutar aplikacije. Ovaj modul sadrži kameru koja omogućava korisniku da se kreće kroz 3D svet.
  - Scene mogu da sadrže različite grafičke elemente (slika, tekst, lista, dugme) i oni mogu biti postavljeni relativno svuda unutar 3D sveta. Ovo omogućava programerima da kreiraju napredne i interesantne efekte.
- Modul za iscrtavanje (engl. *Renderer module*): ovaj modul integriše niz faza renderovanja. Ovaj modul je odgovoran za sve što se pojavljuje na ekranima uređaja.
- Modul za resurse (engl. *Assets module*) je odgovoran za rukovođenje svim neophodnim resursima aplikacije (na primer tekstura zvuk i 3D modeli). Modul za resurse efikasno koristi memoriju što je veoma bitno kada se radi o uređajima sa nedostatkom memorije i procesne snage.
- Animator je odgovoran za animacije objekata. Animator može da animira poziciju, rotaciju, factor skaliranja, transparentnost kao i ručno napravljene animacije [7][8].

## 3. Koncept rešenja

U ovom poglavlju predstavljena je analiza problem i uslovi koji aplikacija treba da zadovolji. Izloženi su glavni zahtevi koje dizajn rešenja treba da zadovolji i predstavljene su skice rešenja.

### 3.1 Analiza zadatka

Rešenje koje opisano u ovom radu trebalo je da predstavi podatke koji se nalaze na kontrolnoj tabli automobila i koji su neophodni za vožnju. Neke od neophodnih informacija koje je potrebno prikazati su: brzina automobila, broj obrtaja motora, status goriva, temperatura motora itd. Ove podatke potrebno je prikazati u 3D prostoru. U ovom radu podaci će biti simulirani. Implementacija je takva da odabrani podaci za prikaz ilustruju dizajn. Na primer, kazaljka za brzinu nije uvek na zadatoj brzini jer se ovim ne postiže prikaz rada aplikacije. Kazaljka se u aplikaciji pomera od početne brzine do krajnje brzine za koje su date konačne vrednosti.

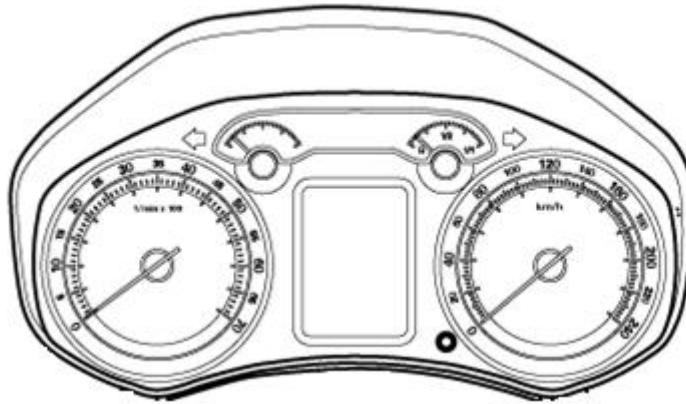
Tokom izrade rada treba se voditi sledećim zahtevima:

- Vozaču se ne sme odvlačiti pažnja vizuelnim efektima tokom vožnje. Treba da postoji opcija za isključivanje bilo koje dinamičke informacije.
- Informacije treba predstaviti na vreme i dati im prioritet po potrebi.
- Izbegavati duge i iskprekidane interakcije kao i vremenski kritične zahteve dobavljanja informacija. Omogućiti da se nastavi prekinuti zahtev.
- Omogućiti da odgovori budu jasni, predvijljivi i dosledni i da pružaju zatraženu informaciju i poruku ukoliko je odgovor neuspešno poslat[9].

### 3.2 Dizajn

Glavni problem realizacije ovog rešenja jeste dizajn. Raspored modela u prostoru i sam izgled modela nigde nije specificiran. Potrebno je osmisliti izgled modela tako da korisnik ima osećaj treće dimenzije i u stanju mirovanja modela. Ne treba ih suviše komplikovati jer je potrebno da oni pruže jasan prikaz informacije. Glavni zahtev je da najkorisnije informacije potrebne korisniku trebaju biti lako uočljive pa samim tim modele treba rasporediti u prostoru prvenstveno po ovom zahtevu. Boje modela igraju veliku ulogu. Veoma je bitno da vozači ne čitaju informacije sa kontrolne table sa velikim naporom. Ukoliko su modeli suviše svetli ili tamni i nema puno kontrasta, teško je pročitati podatke koje nam aplikacija pruža.

Tradicionalni izgled kontrolne table automobila podrazumeva prikaz: sata za brzinu motora, sata za broj obrtaja motora, statusnu traku za prikaz nivoa goriva, statusnu traku za nivo ulja što se može videti na slici 4. Takođe, na tabli se nalaze lampice za obaveštenja kao i lampice za žmigavce. Vozači su navikli na ovakav izgled kontrolne table automobila koji je dovoljno jasan. Ovo je razlog zašto je aplikacija rađena po ugledu na njega.



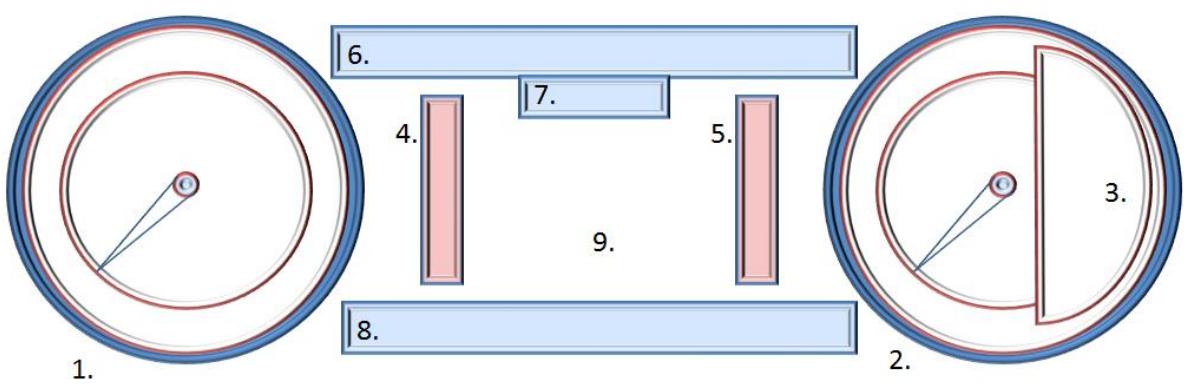
Slika 4. Skica uobičajenog izgleda kontrolne table

Proučavanjem već postojećih rešenja za 3D prikaz kontrolne table automobila i sadržaja informacija koje pružaju, uočeno je da pored osnovnih informacija postoje dodatne opcije kao što su prikaz navigacije, prikaz i izbor radio stanica, prikaz informacija sa dodatno priključenog uređaja kao što je pametan telefon itd. Pored ovog proširenja u postojećim rešenjima za 3D prikaz tabele postoji mogućnost izmene sadržaja informacija koji će se prikazati kao i njihov raspored. Ovo proučavanje je samo nametnulo ideju da se rešenje koje je u ovom radu opisano proširi tako što će postojati izbor dodatnih opcija.

Ideja je da aplikacija sadrži dve celine. Dizajn prve celine rađen je po ugledu na tradicionalni izgled kontrolne table. Skica prve celine se može videti na slici 5.

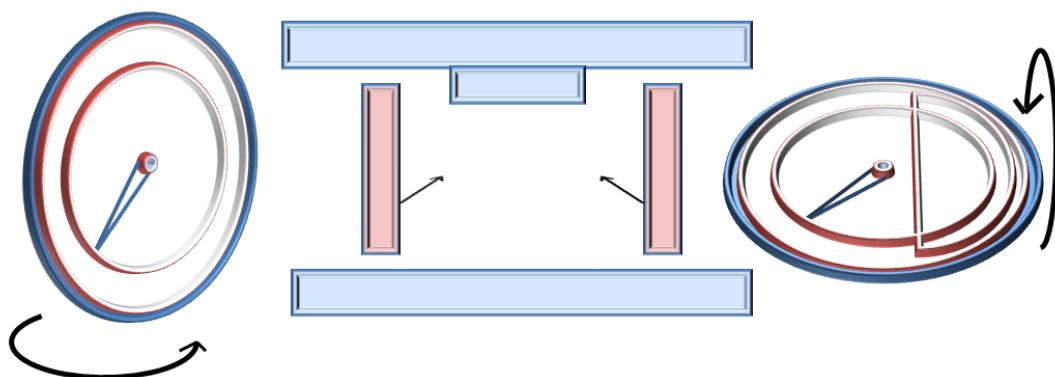
Prva celina treba da sadrži prikaz:

1. Sat za prikaz brzine automobila
2. Sat za prikaz broja obrtaja motora
3. Statusne trake za prikaz nivoa goriva
4. Statusne trake za prikaz nivoa ulja u motoru
5. Statusne trake za prikaz temperature motora
6. Polje za prikaz datuma, pređenog puta i temperature
7. Polje za prikaz vremena
8. Polje za prikaz obaveštenja koja se uključuju u toku vožnje
9. Prostor za prikaz navigacije i signala za skretanje



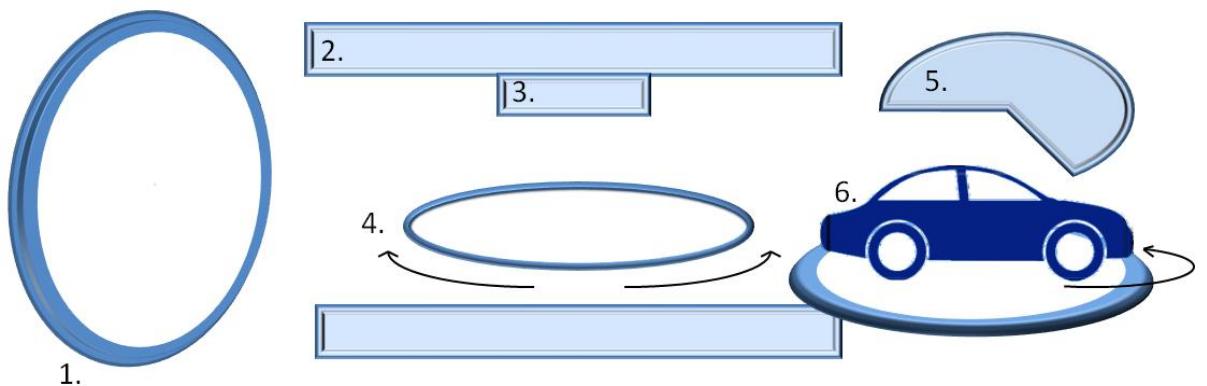
Slika 5. Skica uobičajenog izgleda kontrolne table

Prelaz sa jedne na drugu celinu treba osmisiliti tako da korisniku dočara postojanje treće dimenzije. Skica prelaza se može videti na slici 6. Prelaz treba da postane aktivan onda kada korisnik to zatraži. Takodje, on se ne sme desiti ukoliko je vozilo u pokretu jer druga celina sadži samo prikaz dodatnih opcija.



Slika 6. Skica prelaza sa jedne na drugu celinu

Druga celina sadrži prikaz dodatnih opcija i mogućnost njihovog uključivanja i isključivanja. Skica druge celine se može pogledati na slici 7. Treba omogućiti promenu prikaza dodatne opcije onda kada korisnik to zatraži kao i promenu vrednosti dodatne opcije. Pored dodatnih opcija u ovoj celini treba da postoji prikaz opštih informacija kao što su vreme, datum, temperature i pređeni put. Promena dodatne opcije treba vizuelno da se prikaže na modelu automobila.



Slika 7. Skica prikaza druge celine

Opis skice prikaza druge celine:

1. Naziv celine
2. Polje za prikaz datuma, pređenog puta i temperature
3. Polje za prikaz vremena
4. Polje za prikaz naziva dodatne opcije
5. Polje za prikaz vrednosti dodatne opcije
6. Model automobila

## 4. Implementacija

U narednim poglavljima sledi obašnjenje ispunjenja uslova koji su izloženi u konceptu rešenja. Povezanost aplikacije sa simulatorom nije implementirana. Rad aplikacije je simuliran zadatim vrednostima i animacijama radi prikaza boljih performansi aplikacije. Pokretanje animacija se vrši pritiskom odgovarajućih tastera na tastaturi. Ovo se vrši radi simulacije rada aplikacije dok bi se tokom korišćenja aplikacije u automobilu to preslikalo na hardverske tastere i neke događaje u automobilu.

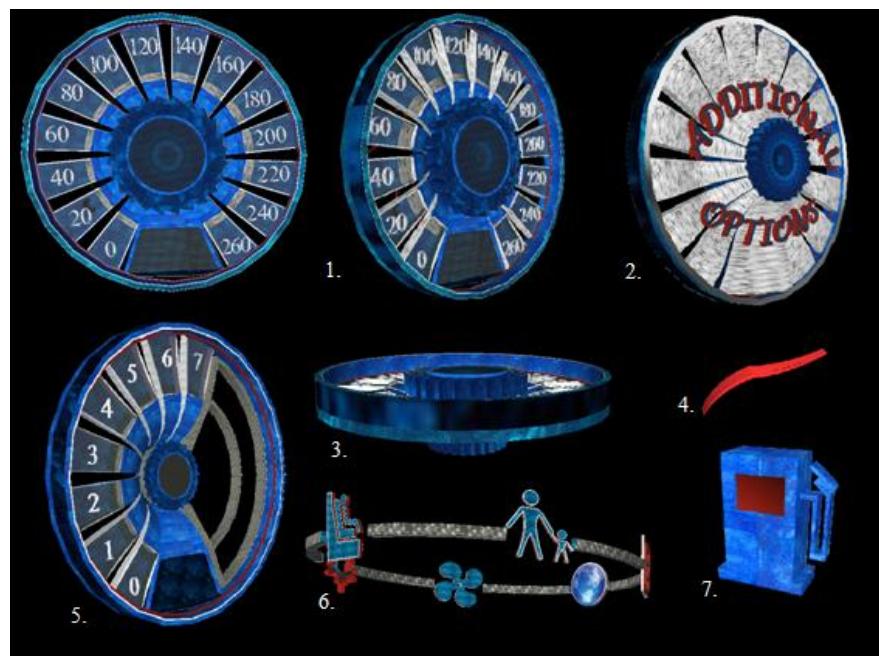
### 4.1 Dizajnersko rešenje

Modeli su crtani u programu za kompjutersko modelovanje SketchUp. Po završetku iscrtavanja modela, model se izvozi, prevodi u odgovarajući format datoteke i kao takav se učitava u aplikaciju. Koristeći C-More razvojno okruženje za razvoj grafičke korisničke sprege modeli se pozicioniraju u tro3D prostoru. Potrebno je sve modele pozicionirati na svoje mesto. Nad modelima se vrše razne animacije: pomeranja modela u prostoru, rotiranja i podešavanja transparentnosti.

Uz težnju da se ispune prethodno navedeni zahtevi, kreirani su modeli sata za prikaz brzine i broja obrtaja motora, strelica koja prikazuje vrednost na satu, okvir kao i ikonice za identifikaciju statusnih traka. Neki od modela koje sadrži aplikacija su prikazani na slici 8. Na ovoj slici se nalazi prikaz:

1. Prednje strane sata za prikaz brzine
2. Zadnje strane sata za prikaz brzine
3. Bočne strane sata za prikaz brzine. Sat za prikaz broja obrtaja motora sa bočne strane isto izgleda.
4. Kazaljke za prikaz vrednosti na satu

5. Prednje strane sata za prikaz broja obrtaja motora. U njemu se nalazi i statusna traka za prikaz nivoa goriva
6. Prstena sa ikonicama dodatnih informacija
7. Ikonica za prikaz nivoa goriva



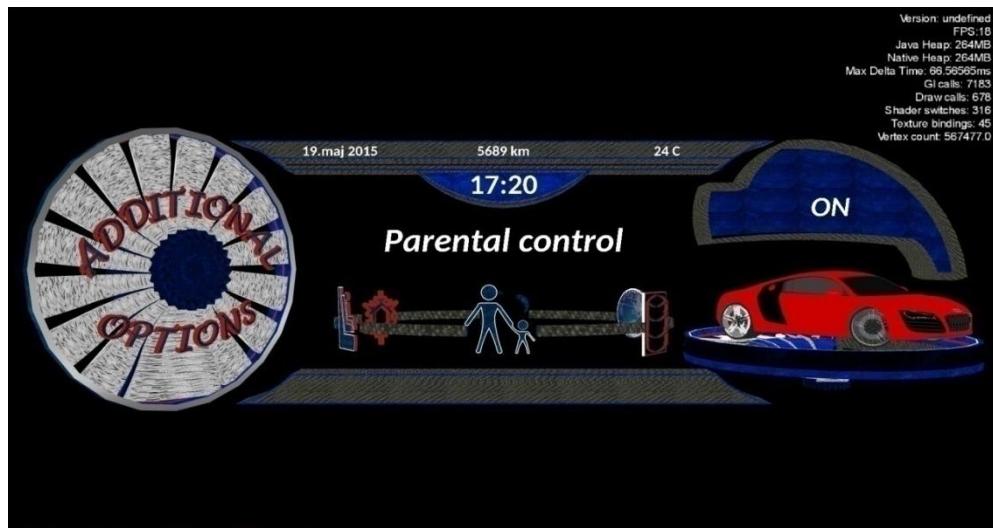
Slika 8. Prikaz nekih od modela koje sadrži aplikacija

Aplikacija sadrži dve scene (deo C-More razvojnog grafičkog okruženja). Ovim scenama su prikazane celine na koje aplikacija treba da se podeli. Dizajnersko rešenje prve scene ispunilo je većinu zahteva. Na slici 9 se nalazi prikaz prve scene. Zbog komplikovane implementacije nije omogućeno da korisnik prilagođava sadržaj prikazanih informacija po svojoj želji kao ni to da se menja izgled scene. Prikazane su sve tražene informacije. Najveći akcenat je stavljen na prikaz brzine i broja obrtaja motora. Velika važnost je data i statusnim trakama za prikaz nivoa goriva, toplove motora i nivoa ulja u motoru. Ikonice za obaveštenja u toku vožnje istaknuta su crvenim okvirom što ih čini lako primetljivim. Pri prikazu scene, ikonice za oznaku statusnih traka se rotiraju oko svoje ose. Ova animacija se može zaustaviti pritiskom na taster "S". Ovim je ispunjen zahtev da korisnik može u bilo kom trenutku zaustaviti animacije modela zbog odvlačenja pažnje. Njihovo ponovno pokretanje omogućeno je pritiskom tastera "P". Pokretanje simulacije promene nivoa statusnih traka vrši se pritiskom na taster "enter". Slike obaveštenja pojavljuju se pri pokretanju animacija za statusne trake. Pritiskom leve strelice na tastaturi uključuje se levi žmigavac, odnosto desne strelice desni žmigavac.



Slika 9. Prikaz prve scene

Druga scena sadrži prikaz dodatnih opcija koji je napravljen na osnovu proučavanja već postojećih rešenja. Zbog kompleksnosti implementacije dodatnih opcija kao što je npr. izbor radio stanica, dodatne opcije su birane na osnovu njihove jednostavne funkcionalnosti tj. postoji samo mogućnost njihovog uključivanja i isključivanja. Dodatne opcije su: uključivanje roditeljske kontrole, interneta, čuvanje informacija u vezi događaja u vozilu u bazu podataka, podešavanje jačine i položaja klime i prikaz indikatora dečijeg sedišta. Izbor prikaza dodatne opcije se vrši pritiskom na levu i desnu strelicu na tastaturi. Promena vrednosti dodatne opcije se vrši pritiskom na taster enter. Jedna od ideja rešenja je bila ta da su dodatne informacije povezane sa modelom automobila koji je prikazan na ovoj sceni. Zbog trenutne ograničenosti C-More razvojnog grafičkog okruženja za razvoj grafičke korisničke sprege i komplikovanosti implementacije korišćenjem trenutnih mogućnosti C-More razvojnog okruženja, ovo nije izvršeno. Na sledećoj sliци se može videti izgled druge scene.



Slika 10. Prikaz druge scene

## 4.2 Modularnost

Kako bi aplikacija za prikaz kontrolne table automobila mogla biti nadograđena ili izmenjena, potrebno je izdeliti njegovu funkcionalnost na module. Realizovana aplikacija je podeljena u četri modula:

- Modul za inicijalizaciju sveta
- Modul za prikaz scene aktivne tokom vožnje
- Modul za prikaz scene sa dodatnim opcijama
- Modul sa funkcijama za animaciju scena

### 4.2.1 Modul za inicijalizaciju sveta

Svet predstavlja svet modul C-More grafičkog okruženja za razvoj grafičke korisničke sprege. Pri inicijalizaciji sveta kamera se postavlja na određenu udaljenost i zadaje se veličina vidnog polja kamere. U svetu su registrovane scene koje sadrži aplikacija. Veličina prikaza scene se zadaje prilikom njihove inicijalizacije u ovom modulu kao i osluškivač koji osluškuje da li je pritisnut taster space. Na scene je postavljena animacija za transparentnost (engl. *Alpha animation*) grafičkih elemenata. Ona je podržana od strane C-More grafičkog okruženja, ali nije podržana za modele. Animacija za transparentnost se pokreće prilikom promene scena odnosno prilikom pritiska na taster space. Pri inicijalizaciji sveta potrebno je staviti koja će scena biti aktivna pri prikazivanju sveta. U ovom rešenju to je scena za prikaz informacija u toku vožnje. Ovde se vrši učitavanje fonta potrebnog za prikaz teksta na scenama.

### 4.2.2 Modul za prikaz scene aktivne tokom vožnje

U ovom modulu se nalaze sve funkcije i pozivi funkcija potrebni za prikaz scene koja je aktivna u toku vožnje. Učitavaju se svi resursi (slike, modeli, fontovi) koji se nalaze u modulu za resurse C-More grafičkog okruženja. Svaka slika, model ili tekst se smešta u poseban grafički element. U daljem izvršavanju koda sve animacije se izvršavaju nad tim grafičkim elementima. Pozivaju se funkcije za učitavanje grafičkih elemenata modela, teksta i slika. Ove funkcije za učitavanje sadrže smeštanje resursa u grafički element, postavljanje veličine grafičkog elementa, njegovo registrovanje na scenu i zadavanje koordinata položaja grafičkog elementa u tro3D prostoru. Sve grafičke elemente je potrebno registrovati na scenu pozivom funkcije *registerChild* gde se kao parametar prosleđuje naziv grafičkog elementa. Grafički model ne može biti prikazan ukoliko nije registrovan na scenu. U ovom modulu se pokreće programska nit koja periodično dobavlja vreme i oscežava prikaz.

U ovom modulu nalaze se još funkcije:

- *setKeyListener()* – Služi za osluškivanje pritiska tastera na tastaturi.
  - Taster *space* – Promena scene.
  - Taster leve ili desne strelice – Uključivanje levog ili desnog žmigavca.
  - Taster enter – Pokretanje animacije promene vrednosti statusnih traka.
  - Taster S – Zaustavljanje animacije rotiranja ikonica statusnih traka.
  - Taster P – Pokretanje animacije rotiranja ikonica statusnih traka.
- *fadeIn()* – Sadrži sve pozive potrebne za animaciju grafičkih elemenata prilikom prikaza scene. Pozivaju se funkcije iz klase *SceneAnimations*.
- *fadeOut()* – Sadrži pozive funkcija potrebne za animaciju grafičkih elemenata prilikom sklanjanja scene. Kao i u prethodno navedenoj funkciji i ovde se pozivaju funkcije iz klase *SceneAnimations*.
- *setSpeedAndRpm()* – Pozivom ove funkcije prikazuje se tražena brzina i broj obrtaja motora. Kazaljke se rotiraju i postavljaju na određenu poziciju u prostoru koja je preračunata na osnovu tražene brzine, odnosno broja obrtaja motora.

#### **4.2.3 Modul za prikaz scene sa dodatnim opcijama**

Prikaz scene sa dodatnim informacijama se vrši na isti način kao i prethodno definisani modul. Sadrži pozive funkcija za učitavanje resursa i njihovo smeštanje u grafičke elemente. Sve grafički elementi su registrovani na scenu. Pored funkcija učitavanja resursa i njihovog smeštanja u grafičke elemente postoje funkcije:

- *setKeyListener()* – Služi za osluškivanje pritiska tastera na tastaturi.
  - Taster “space” – Promena scene.
  - Taster leve ili desne strelice – Promena prikaza naziva i vrednosti dodatne opcije.
  - Taster enter – Prmena vrednosti prikazane dodatne opcije.
  - Taster S – Zaustavljanje animacije rotiranja automobila.
  - Taster P – Pokretanje animacije rotiranja automobila.
- *show()* - Sadrži sve pozive potrebne za animaciju grafičkih elemenata prilikom prikaza scene. Pozivaju se funkcije iz klase *SceneAnimations*.
- *fadeOut()* – Sadrži pozive funkcija potrebne za animaciju grafičkih elemenata prilikom sklanjanja scene. Kao i u prethodno navedenoj funkciji i ovde se pozivaju funkcije iz klase *SceneAnimations*

- *leftKey()* – Sadrži logiku za rotaciju modela koji predstavlja ikonice dodatnih opcija i prikaz leve ikonice u odnosu na već prikazanu ikonicu. Svaka ikonica je povezana sa grafičkim elementom teksta koji prikazuje naziv dodatne opcije i sa grafičkim elementom teksta koji prikazuje vrednost dodatne opcije.
- *rightKey()* – Kao i prethodno objašnjena funkcija ova funkcija sadrži logiku za prikaz desne dodatne opcije u odnosu na već prikazanu.
- *enterKey()* – Sadrži logiku za promenu vrednosti izabrane dodatne opcije.

#### 4.2.4 Modul sa funkcijama za animaciju scena

Klasa *SceneAnimations* sadrži funkcije za animaciju modela, slike i teksta. Da bi se izvršila neka animacija na grafičkom elementu potrebno je napraviti novu animaciju, zadati dužinu njenog trajanja, broj ponavlja i reći da li se ona ponavlja itd. Napravljenu animaciju je zatim potrebno postaviti na određeni grafički element i pokrenuti je. U aplikaciji većina grafičkih elemenata ima iste animacije i zato nema potrebe praviti za svaki element posebnu animaciju. Na ovaj način potreno je pozvati funkciju sa traženom animacijom kojoj se prosleđuje ime grafičkog elementa, vreme trajanja animacije i odgovor da li je potrebno da se animacija ponavlja.

Funkcije koje su definisane u ovoj klasi su priložene u tabeli 1.

Funkcije	Objašnjenje
<i>AlphaAnimationIn()</i>	Funkcija za smanjivanje transparentnosti od nevidljivog do skroz vidljivog grafičkog elementa. Postoje posebne definicije ove funkcije za tekst, model i sliku.
<i>AlphaAnimationOut()</i>	Funkcija za povećavanje transparentnosti od vidljivog do skroz nevidljivog grafičkog elementa. Postoje posebne definicije ove funkcije za tekst, model i sliku.
<i>relativeRotateModel()</i>	Funkcija za relativno rotiranje modela. Pored modela, vremena trajanja i odgovora da li se animacija ponavlja, prosleđuju se vrednosti za koje model treba da se rotira po x, y i z osi. Relativno rotiranje znači da će se model rotirati za zadatu vrednost. Na primer, ako se model nalazi na poziciji gde je vrednost x jednaka 30 i zadamo u ovoj funkciji da je x jednak 60, model će se rotirati do pozicije gde je vrednost x jednaka 90.
<i>absoluteRotateModel()</i>	Funcija za apsolutno rotiranje modela. Pored modela, vremena trajanja i odgovora da li se animacija ponavlja, prosleđuju se vrednosti za koje model treba da se rotira po x, y i z osi. Apsolutno rotiranje znači da će se model rotirati do zadate vrednosti. Na primer, ako se model nalazi na poziciji gde je vrednost x jednaka 30 i zadamo u ovoj funkciji da je x jednak 60, model će se rotirati do pozicije gde je vrednost x jednaka 60.
<i>moveModel()</i>	Funkcija za pomeranje modela u prostoru. Ovoj funkciji se prosleđuje naziv modela, vreme trajanja animacije, vrednost koordinata željene pozicije modela i odgovor da li je se animacija ponavlja.

Tabela 1. Funkcije koje su definisane u klasi za animaciju scena

## 5. Rezultati

Kako je rešenje ovog zadatka prikaz 3D korisničke sprege za kontrolnu tablu automobila, rezultati se mogu predstaviti u vidu zadovoljavanja zahteva koji su predloženi u konceptu rešenja. Zahtev, ispunjenost zahteva i objašnjenje priloženo je u tabeli 2.

Zahtev	Ispunjeno zahteva Da / Ne	Objašnjenje
<b>Vozaču se ne sme odvlačiti pažnja vizualnim efektima tokom vožnje.</b>	Da	Postoji opcija isključivanja dinamičkih vizuelnih efekta.
<b>Informacija mora biti predstavljena na vreme i treba im dati prioritet po potrebi.</b>	Da	Sve informacije se prikazuju po potrebi. Najveći prioritet je dat prikazu brzine i broja obrtaja motora.
<b>Izbegavati duge i isprekidane interakcije i vremenski kritične zahteve.</b>	Da	Odziv na zahtev prikaza informacije je minimalan.
<b>Aplikacija treba da odgovori na jasan, predvidiv i dosledan način i da pruži zatraženu informaciju.</b>	Da	Aplikacija odgovara na sve zahteve pružajući zatraženu informaciju.

Tabela 2. Ispunjeno zahteva

U tabeli 3. je priložen prikaz informacija potrošnje video memorije čije je merenje izvršeno alatom TechPowerUp GPU – Z. Priložen je i prosečni broj slika po sekundi (engl. *frames per second - FPS*) kao i potrošnja grafičke memorije. Konfiguracija računara na kojoj je aplikacija testirana:

- Procesor (CPU) – Intel(R) Core(TM) i5-2400 CPU @ 3.10 GHz
- Grafički procesor (GPU) – Intel HD Graphics 2000
- Količina raspoložive video memorije – 2 GB
- Količina raspoložive RAM memorije – 6 GB

Potrošnja memorije (RAM memorije)	5 MB
Potrošnja video memorije	228 MB
Prosečni FPS	55

Tabela 3. Rezultati testiranja aplikacije

Aplikacija je testirana i na Nexus 9 tablet uređaju. Broj prosečnih slika po sekundi je 60 dok je potrošnja memorije 12MB.

Ukoliko bi postojala aplikacija koja sadrži iste resurse kao i aplikacija “ClusterUI” pri čijoj bi se izradi koristilo drugo razvojno korisničko okruženje za razvoj grafičke korisničke sprege, rezultati testiranja aplikacija bi bili uporedivi. Zbog nepostojanosti ovakve aplikacije, priloženi rezultati testiranja su neuporedivi.

## 6. Zaključak

U ovom radu je predstavljeno jedno rešenje 3D prikaza korisničke sprege za kontrolnu tablu vozila. Aplikacija je razvijena na PC-u, ali je prebačena i na tablet uređaj sa Android operativnim sistemom gde je povezana sa OBD II simulatorom koji simulira rad automobila. Ovim razvojem je aplikacija dobila mogućnost primene u IVI sistemima.

Dalji ciljevi razvoja aplikacije:

- Povećavanje modularnosti rešenja – Obezbedilo bi se lakša izmena i nadogradnja aplikacije.
- Poboljšanje dizajna i prikaza informacija – Razvojem dizajna bi se korisniku pružila mogućnost menjanja sadržaja prikazanih informacija. Takođe, jedan od ciljeva je i povezivanje modela automobila sa podacima
- Postojanje više primera izgleda scene koja je aktivna u toku vožnje – Na primer, ukoliko je korisniku potreban veći prikaz navigacije da bude omogućeno da je navigacija prikazana preko većeg dela ekrana, a sat za prikaz brzine i broja obrtaja motora bude smanjen.
- Optimizacija grafičkih resursa koji se koriste u aplikaciji – Došlo bi do poboljšanja preformansi i smanjenja potrošnje video memorije ukoliko bi se broj tačaka u nekim modelima smanjio, ukoliko bi se smanjila veličina slika itd.

## 7. Literatura

- [1] [www.obdii.com](http://www.obdii.com)
- [2] Predavanje sa predmeta Digitalne multimedije 2, Visoka škola elektrotehnike i računarstva strukovnih studija, Beograd.  
[www.viser.edu.rs/download.php?id=18199](http://www.viser.edu.rs/download.php?id=18199), učitano 6.05.2015.
- [3] [www.linfo.org/gui.html](http://www.linfo.org/gui.html), učitano 10.05.2015.
- [4] <https://www.khronos.org/opengles/>, učitano 12.05.2015.
- [5] <http://developer.android.com/guide/topics/graphics/opengl.html>, učitano 12.05.2015.
- [6] <https://github.com/libgdx/libgdx/wiki/Introduction>, učitano 12.05.2015.
- [7] c-more\_ui\_framework\_0.2.0(Bogeyman) documentation: C-More UI framework top level architecture
- [8] Đorđe Kovačević, Violeta Vukobat, Stefan Pejić and Ištvan Papp, “*Building Blocks for GUI – Novel Solution for Composite Component Management*”, 2<sup>nd</sup> International Conference on Electrical, Electronic and Computing Engineering IcETRAN 2015, Silver Lake, Serbia, June 8-11 2015
- [9] Jan Sonnenberg, “*Service and User Interface Transfer from Nomadic Devices to Car Infotainment Systems*”, Technische Universität Braunschweig, Institute for Communications Technology, Proceedings of the Second International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI 2010), November 11-12, 2010, Pittsburgh, Pennsylvania, USA