



УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА У
НОВОМ САДУ



Дејан Попов Тапавички

**Једна реализација графичке
корисничке спрете за репродукцију ТВ
садржаја на преносним уређајима
заснованим на Андроид платформи**

ДИПЛОМСКИ РАД
- Основне академске студије -

Нови Сад, јун 2013



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:	Дејан Попов Тапавички		
Ментор, МН:	др Јелена Ковачевић		
Наслов рада, НР:	Једна реализација графичке корисничке спреге за репродукцију ТВ садржаја на преносним уређајима заснованим на Андроид платформи		
Језик публикације, ЈП:	Српски / латиница		
Језик извода, ЈИ:	Српски		
Земља публиковања, ЗП:	Република Србија		
Уже географско подручје, УГП:	Војводина		
Година, ГО:	2013		
Издавач, ИЗ:	Ауторски репрингт		
Место и адреса, МА:	Нови Сад; трг Доситеја Обрадовића 6		
Физички опис рада, ФО: (поглавља/страница/цитата/табела/ слика/графика/прилога)	7/34/0/0/14/0/0		
Научна област, НО:	Електротехника и рачунарство		
Научна дисциплина, НД:	Рачунарска техника		
Предметна одредница/Кјуине речи, ПО:	ДТВ, Андроид, Графичка корисничка спрела, Искоришћење системских ресурса		
УДК			
Чува се, ЧУ:	У библиотеци Факултета техничких наука, Нови Сад		
Важна напомена, ВН:			
Извод, ИЗ:	У овом раду је приказана једна реализација пројектовања графичке корисничке спреге апликације за Андроид преносне уређаје који репродукују ТВ садржај послат од стране ДТВ пријемника. Описан је комплетан изглед и начин реализације компоненти графичке корисничке спреге.		
Датум прихватања теме, ДП:			
Датум одбране, ДО:			
Чланови комисије, КО:	Председник:	проф. др Никола Теслић	
	Члан:	mr Милан Савић	Потпис ментора
	Члан, ментор:	др Јелена Ковачевић	



KEY WORDS DOCUMENTATION

Accession number, ANO:		
Identification number, INO:		
Document type, DT:	Monographic publication	
Type of record, TR:	Textual printed material	
Contents code, CC:	Bachelor Thesis	
Author, AU:	Dejan Popov Tapavički	
Mentor, MN:	Jelena Kovačević, PhD	
Title, TI:	One graphical user interface realization for reproduction of the TV content on portable devices based on Android platform	
Language of text, LT:	Serbian	
Language of abstract, LA:	Serbian	
Country of publication, CP:	Republic of Serbia	
Locality of publication, LP:	Vojvodina	
Publication year, PY:	2013	
Publisher, PB:	Author's reprint	
Publication place, PP:	Novi Sad, Dositeja Obradovica sq. 6	
Physical description, PD: (chapters/pages/ref./tables/pictures/graphs/appendices)	7/34/0/0/14/0/0	
Scientific field, SF:	Electrical Engineering	
Scientific discipline, SD:	Computer Engineering, Engineering of Computer Based Systems	
Subject/Key words, S/KW:	DTV, Android, Graphical User Interface, CPU usage	
UC		
Holding data, HD:	The Library of Faculty of Technical Sciences, Novi Sad, Serbia	
Note, N:		
Abstract, AB:	This paper presents one solution of graphical user interface design for portable devices based on Android platform that displays reproduction of the TV content delivered from DTV device. The paper describes design and implementation of the components created and used in graphical user interface.	
Accepted by the Scientific Board on, ASB:		
Defended on, DE:		
Defended Board, DB:	President:	Nikola Teslić, PhD
	Member:	Milan Savić, MSc
	Member, Mentor:	Jelena Kovačević, PhD
		Menthor's sign

Zahvalnost

Zahvaljujem se institutu RT-RK na pruženoj mogućnosti za realizaciju ovog rada.

Takođe se zahvaljujem mentoru dr Jeleni Kovačević, stručnim saradnicima Nikoli Kuzmanoviću i Milanu Saviću, kao i kolegama iz AMUSE tima na savetima i stručnoj pomoći prilikom izrade ovog rada.

Na kraju, zahvaljujem se svojoj porodici na pruženoj podršci tokom mog školovanja.

SADRŽAJ

1.	Uvod.....	1
2.	Teorijske osnove	3
2.1	Digitalna televizija	3
2.2	Android platforma.....	4
2.3	Grafička korisnička sprega uređaja zasnovanim na Android platformi	5
2.3.1	Način realizacije grafičke korisničke sprege na Android platformi	7
3.	Koncept rešenja.....	9
3.1	Fragmeni	9
3.1.1	Životni ciklus fragmenata.....	10
3.2	Primena fragmenata u korisničkoj sprezi	11
3.3	Razlaganje grafičke korisničke sprege na manje celine	12
4.	Programsko rešenje.....	14
4.1	Tranzicija fragmenata u grafičkoj korisničkoj sprezi.....	14
4.2	Moduli, opis i funkcije	15
4.2.1	<i>DetailsMasterFragment</i>	15
4.2.2	<i>DLNAMediaExplorerFragmet</i>	17
4.2.3	<i>SettingsFragment</i>	17
4.2.4	<i>LiveTVFrgment</i>	18
4.2.4.1	Organizacija grafičkih elemenata modula za reprodukciju video zapisa	19
4.2.5	<i>GamesFragment</i>	19
4.2.6	<i>TouchPadFrgment</i>	20
5.	Ispitivanje i verifikacija	21
5.1	Programsko ispitivanje funkcionalnosti i navigacije	21

5.2	Direktno ispitivanje od strane krajnjeg korisnika	23
6.	Zaključak	24
7.	Literatura.....	26

SPISAK SLIKA

Slika 2.1 GUI arhitektura Android platforme	5
Slika 3.1 Životni ciklus fragmenata	10
Slika 3.2 Primeri upotrebe fragmenata na ekranima različitih dimenzija	11
Slika 3.3 Raspodela fragmenata u grafičkoj korisničkoj sprezi	12
Slika 3.4 Podela reprodukcije TV saržaja po modulima	13
Slika 4.1 Raspodela fragmenata sadržanih u grafičkoj sprezi	14
Slika 4.2 Izgled fragmenata za navigaciju kroz DLNA poslužioce	17
Slika 4.3 Izgled fragmenata za podešavanja	17
Slika 4.4 Lista dostupnih kanala za reprodukciju	18
Slika 4.5 Modul za reprodukciju video zapisa i reprodukovani kanal	18
Slika 4.6 Organizacija grafičkih elemenata modula za reprodukciju video zapisa	19
Slika 4.7 Izgled fragmenata za igrice	19
Slika 4.8 Izgled fragmenata koji reprezentuje kompjuterskog miša	20

SKRAĆENICE

DTV	- <i>Digital Television</i> , Digitalna televizija
GUI	- <i>Graphical User Interface</i> , Grafička korisnička sprega
OS	- <i>Operating System</i> , Operativni sistem
TV	- <i>Television</i> , Televizija
API	- <i>Application Programming Interface</i> , Programska sprega
UI	- <i>User Interface</i> , Korisnička sprega
XML	- <i>Extensive Markup Language</i> , Proširivi metajezik za označavanje tekstualnih dokumenata
RUI	- <i>Remote User Interface</i> , Udaljena korisnička sprega
DLNA	- <i>Digital Living Network Alliance</i>
SDK	- <i>Software Development Kit</i> , Okruženje za razvoj programske podrške

1. Uvod

U ovom radu je predstavljen jedan od načina realizacije grafičke korisničke sprege aplikacije (eng. *Graphical User Interface*) koja reproducuje TV sadržaj za prenosne uređaje zasnovane na Android platformi. Rešenje je realizovano za platforme sa Android programskom podrškom verzije 4.0 (eng. *Ice Cream Sandwich - ICS*) ili novije, ali sa naglaskom da bude nezavisno od fizičke arhitekture radi lakše prenosivosti. Ispitan je i verifikovan rad na MIPS, ARM i x86 fizičkim arhitekturama.

Problem koji je uočen pri projektovanju grafičke korisničke sprege Android aplikacije je velika zauzetost sistemskih resursa koji onemogućavaju tačan rad aplikacije. Veliki broj prozora, različitih ekrana zauzimaju procesorsko vreme i memoriju, pogotovo ako se radi o grafici u visokoj rezoluciji.

Akcenat je stavljen na sledeće elemente:

- Prenosni uredaj treba da oponaša funkcionalnosti naprednog daljinskog upravljača (površina za pomeraj miša (eng. *TouchPad*), reprodukcija TV sadržaja (eng. *LiveStreaming*), daljinska kontrola...)
- Navigacija kroz aplikaciju i interakcija sa korisnikom mora biti jednostavna (ne sme otežavati korisniku da lako i brzo, u bilo kom trenutku, pronađe željenu stavku)
- Upotreba novih tehnika dostupnih sa Android programskom podrškom verzije 3.0, kako bi se dobilo na poboljšanju korisničkog okruženja i smanjenju iskorišćenosti sistemskih resursa
- Način optimizacije koda kao i način iscrtavanja grafičkih elemenata

Ovaj rad je sačinjen od sedam poglavlja.

Prvo poglavlje predstavlja kratak opis problema i argumente na koje se mora obratiti pažnja pri implementaciji rešenja. Drugo poglavlje opisuje osnove digitalne televizije, Android operativnog sistema kao i novije tehnologije dostupne u oblasti projektovanja grafičke korisničke sprege Android aplikacija. Treće poglavlje opisuje koncept rešenja gde je počevši od uvoda o načinu korišćenja novih tehnologija i opšte slike rešenja, razrađeno do nivoa detalja kako bi trebalo rešenje da izgleda. Četvrto poglavlje je konkretizacija koncepta i detaljan opis rešenja. Peto poglavlje predstavlja načine ispitivanja i verifikacije ispravnosti Android aplikacije, odnosno uspešnosti projektovanja grafičke korisničke sprege u njoj. Šesto poglavlje daje kratak osvrt na ono što ovaj rad predstavlja i mogućnosti daljeg razvoja i unapređenja. Poslednje poglavlje daje spisak korišćene literature prilikom izrade ovog rada.

2. Teorijske osnove

Poglavlje opisuje teorijske osnove potrebne za razumevanje problema koji se javljaju pri projektovanju grafičke korisničke sprege Android aplikacije kao i probleme pri reprodukciji TV sadržaja na prenosne uređaje zasnovane na Android platformi.

2.1 Digitalna televizija

Televizija je najrasprostranjeniji i najpopularniji kućni uređaj koji informiše, obrazuje i zabavlja ljude.

Reč *televizija* se prvi put koristi 1900. godine. Prvi digitalni sistemi visoke rezolucije prikazani su 1990. godine, dok je prva demonstracija digitalnog televizijskog prenosa održana 1995. godine.

Osnovna karakteristika digitalne televizije predstavlja prenos slike i zvuka sa dodatnim informacijama u digitalnom formatu. Digitalni prenos obezbeđuje bolji kvalitet. Slika i zvuk se prilikom snimanja pretvaraju u digitalnu formu postupcima odmeravanja, kvantovanja, kodovanja i u takvom obliku se prenose kroz medijum prenosa, odnosno kanal. Prednosti ovakvog načina prenosa i zapisa signala su mnogobrojni:

- teoretski, signal se može preneti na neograničenu udaljenost
- postoji mogućnost kompresije signala, a samim tim i bolje iskorišćenje resursa
- digitalno primljena slika teži savršenosti jer je sistem otporan na interferenciju tokom prenosa, impulsne smetnje i refleksije koje su karakteristične za analognu televiziju.
- detekcija i korekcija grešaka

2.2 Android platforma

Android je zasnovan na Linux jezgru. Prvenstveno je bio namenjen samo mobilnim telefonima, a danas je platforma za ugrađene sisteme (uredjaje) kao što su:

- Mobilni telefoni
- Tablet uređaji
- DTV prijemnici (eng. *Set-Top Box*)
- TV

Android se sastoji od četiri glavna sloja:

- **Linux jezgro** (eng. *Linux kernel*) – omogućava komunikaciju na nivou fizičke arhitekture, umrežavanje (eng. *networking*), upravljanje procesima i memorijom.
- **Biblioteke** (eng. *Libraries*) – koriste se za razvoj aplikacija (SQLite, WebKit, OpenGL). U okviru ovog sloja se nalazi Androidova podrška u realnom vremenu (eng. *Android Runtime*), koji sadrži standardne Java biblioteke prilagođene Android platformi i Dalvik virtualnu mašinu zaduženu za pokretanje aplikacija višeg nivoa napisanih u Java programskom jeziku.[1]
- **Okruženje za razvoj aplikacija** (eng. *Application Framework*) – programska sprega (eng. *Application Programming Interface - API*) koju sve Android aplikacije koriste za pristup najnižem nivou arhitekture. Ovaj sloj je napisan u programskom jeziku Java.
- **Aplikacije** (eng. *Applications*) – krajnje Android aplikacije koje su vidljive korisniku i koje korisnik direktno koristi. Aplikacije su pisane u programskom jeziku Java.

Android je besplatan (eng. *Open Source*), projektovan i zasnovan na Linux platformi za ugrađene sisteme, pa se sve više proizvođača DTV uređaja odlučuje za realizaciju rešenja programske podrške [2] zasnovane na ovom operativnom sistemu, koji pored toga nudi veliki broj aplikacija koje su dostupne za preuzimanje [3]. Kombinacija digitalne televizije i Android operativnog sistema menja pojam gledanja televizije sa dosadašnje pasivne u interaktivnu, dvosmernu komunikaciju, tako da ovaj spoj omogućava istovremeno gledanje DTV

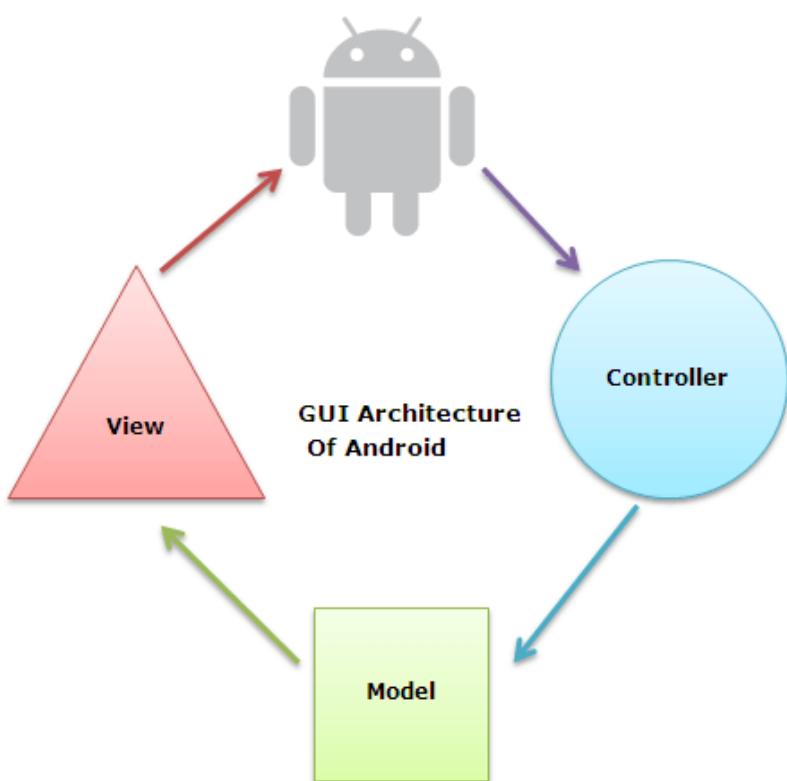
sadržaja, krstarenje Internetom, gledanje fotografija, korišćenje socijalnih mreža i sl. na samom TV-u i/ili na Android prenosnom uređaju.

Na osnovu analize tržišta [4] popularnost Android operativnog sistema raste i nastaviće sa tendencijom rasta. Takođe, digitalna televizija krajnjem korisniku može da ponudi različite interaktivne servise kao što su:

- Medijski servisi
- Internet
- Reprodukovanje TV sadržaja na Android prenosne uređaje
- i mnoge druge

2.3 Grafička korisnička sprega uređaja zasnovanim na Android platformi

Android okruženje dodaje još jednu grafičku korisničku spregu (eng. *Graphical User Interface - GUI*) alat u Java GUI ekosistem, pored AWT, Swing, SWT, LWUIT. Kao i kod navedenih alata, Android GUI alat je vođen događajima (eng. *Event-Driven*) i zasnovan je na jednoj niti. Android UI okvir je, kao i drugi Java UI okviri, organizovan kao Model – Pogled – Kontroler obrazac (eng. *Model-View-Controller - MVC*).



Slika 2.1 GUI arhitektura Android platforme

Android UI nam daje strukturu i alate za pravljenje kontrolera koji obrađuje ulazne događaje, poput pritiska na tastere i dodira ekrana, i pogled (eng. *View*) koji obrađuje i prikazuje grafičke informacije na ekranu. [5]

Model je srž aplikacije, odnosno šta aplikacija zapravo radi. Može predstavljati muzičku bazu podataka na prenosnom uređaju i programsku podršku koja reprodukuje taj sadržaj ili listu koja sadrži kontakte i programsku podršku koja vrši slanje poruka.

Kontroler je deo aplikacije koji reaguje na spoljne aktivnosti: pritisak tastera, dodir ekrana, dolazni poziv i druge. Realizuje se kao red događaja (eng. *Event Queue*). Svaka eksterna akcija je predstavljena kao jedinstveni događaj u redu. Posle izvršenja, svaki događaj se uklanja iz reda i dostavlja na dalju obradu.

Pogled je vizuelizacija modela. Predstavlja deo aplikacije odgovoran za obradu i prikaz ekrana, slanja zvuka sagovornicima, generisanje povratne informacije...

Grafička sprega u Android operativnom sistemu je definisana vizuelnim komponentama:

- dijalog prozorima
- vizuelnim kontrolama
- 2d grafikom
- ostalim multimedijalnim elementima za efikasnost i upotrebljivost

Grafička korisnička sprega se sastoji od više:

- formi
- prozora
- ekrana

a svaki prozor se kontroliše uz pomoć Aktivnost (engl. *Activity*) objekta koji ima kompleksni životni vek. [5] Svim objektima u UI koji se nalaze u aktivnosti može se pristupiti tek kad je ona napravljena, a postoje sve dok je aktivnost živa. Loša strana ovoga je što se sve odvija u jednoprocesnom režimu rada. Napredak Android programske podrške predstavlja tehnologije za razvoj grafičke korisničke sprege sa kojima je moguće razdeliti jednu veliku celinu u više zasebnih elemenata. To znači da svaki gradivni element možemo izvršavati u odvojenom procesu koji je i dalje vezan za glavni proces. Jedno od rešenja se može prikazati upotrebom fragmenata. [6]

2.3.1 Način realizacije grafičke korisničke sprege na Android platformi

Kod izrade Android aplikacija postoje dva načina realizacije grafičke korisničke sprege: deklarativan i proceduralan.

Deklarativan načina se koristi za opisivanje statičke korisničke sprege, odnosno statičkih grafičkih elemenata koji se jednom postavljaju na ekran i ne menjaju se dokle god je aplikacija živa. Upotreba statičke grafičke korisničke sprege je najzastupljenija. Bazira se na [6] XML jeziku za kodiranje koji se može lako pročitati i razumeti. Korišćenje XML jezika za opis statičke grafičke sprege ima još jednu prednost u dizajniranju, a to je grafičko kodovanje ekrana. Kod statičke grafičke sprege postoje dva tipa pisanja XML koda:

- ručno u XML datoteci gde se svaki vizuelni element koji se treba prikazati mora deklarisati u okviru XML čvora
- grafički dizajn koji sam generiše XML kod na osnovu onoga što je grafički podešeno od strane programera

Prednosti statike kao rešenja za opis grafičke korisničke sprege su:

- XML datoteka može biti izmenjena uz pomoć običnog tekstu editora i njegova sintaksa je veoma jednostavna u odnosu na Java programski jezik
- Vrlo lako se može izmeniti dizajn u XML jeziku bez rekomplajiranja izvornog koda
- XML datoteke se nalaze u kompresovanoj binarnoj datoteci i može se izvršiti nezavisno od komplajiranja izvornog koda

Proceduralan način opisuje ono što deklarativan ne može, a to je dinamički deo korisničke sprege, pokretljivi grafički elementi koji se dodaju i brišu po zahtevu korisnika. Koristi se programski jezik Java za opisivanje korisničke sprege. Kod ovakvog načina implementiranja svaki objekat korisničke sprege mora biti deklarisan i definisan u kodu kako bi mogao da se koristi, odnosno da se pojavi na ekranu korisnika.

Kada je reč o pregleđnosti koda statički način je u ozbiljnoj prednosti jer se sam dizajn odvaja od logike aplikacije što odgovara većini programera. Nema gomilanja koda i mešanja dizajna sa logikom u istim klasama i sl. Korišćenjem statičkog načina kodovanja dobija se na čitljivosti koda, većoj razumljivosti logike i lakoj izmeni dizajna, ako je potrebno.

Statička sprega ima svoje prednosti koje doprinose brzini otvaranja i prikazivanja Android aplikacija dok dinamička sprega prihvata i obrađuje zahteve od strane korisnika. Sama ideja

spajanja ove dve sprege u jednu aplikaciju, odnosno korišćenje obe sprege u prikazu jednog prozora na ekranu je doprinelo razvoju zahtevnih Android aplikacija. Statičku spregu i njene opcije treba koristiti koliko je god moguće kako bi se dobilo na brzini prikazivanja, a dinamičku da bi se obradili zahtevi korisnika u realnom vremenu.[7]

3. Koncept rešenja

Poglavlje predstavlja idejni projekat realizacije grafičke korisničke sprege. Opisan je način na koji je postignuto razlaganje ekrana kao jedne celine i upotreba novih tehnologija dostupnih sa Android programskom podrškom verzije 3.0.

3.1 Fragmenti

Fragmenti predstavljaju način za deljenje aktivnosti u ponovo upotrebljive grafičke komponente, svaka sa svojim životnim vekom i izgledom.

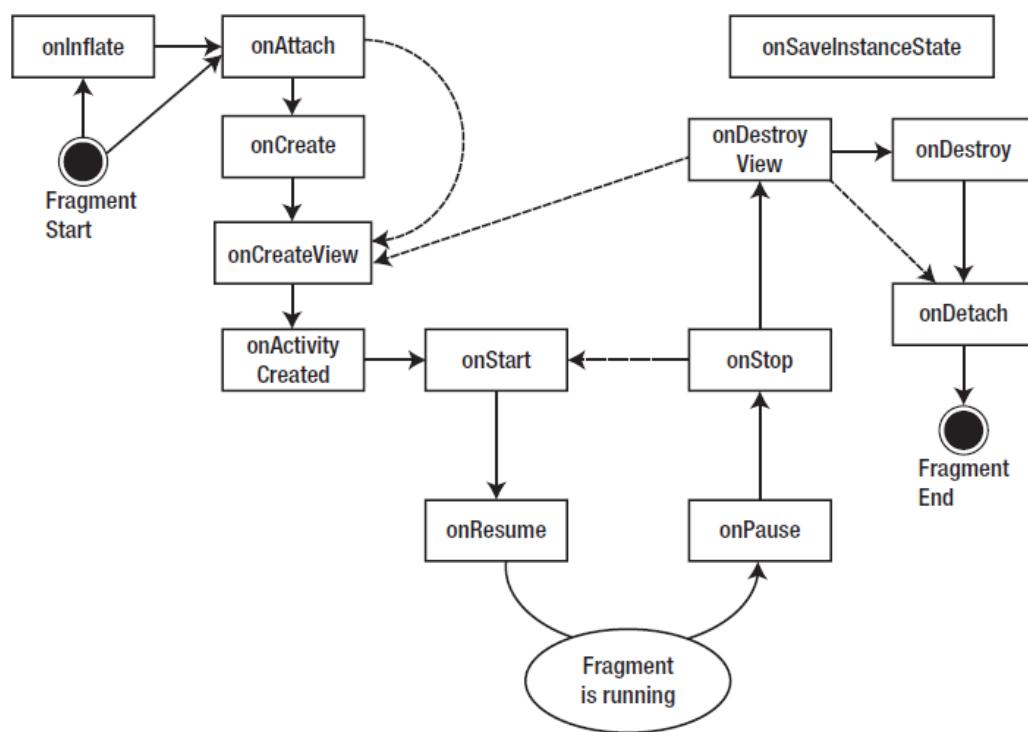
Prednost fragmenata je lakoća sa kojom se mogu kreirati dinamički i fleksibilni UI dizajni koji se mogu upotrebiti za veliki opseg veličina ekrana, od pametnih telefona sa malim, do tablet uređaja sa velikim ekranom. Kod tablet uređaja fragmenti imaju značajnu ulogu, jer ekran tablet uređaja više podseća na ekran personalnog računara nego na ekran pametnog telefona, ako se uzme u obzir veličina i multi-panel aplikacije. Takođe upotreba fragmenata je poželjna ako je dozvoljeno menjanje orijentacije položaja ekrana. Kada je dozvoljeno menjanje orijentacije u aplikaciji, pri svakoj promeni položaja (sa horizontalnog u vertikalni ili obrnuto) mora se resetovati izgled aplikacije (obrisati svi postojeći grafički elementi i napraviti se ponovo u drugom položaju). Ovaj proces traži vreme za izvršavanje koje jednostavno ne sme biti veliko kako bi krajnji korisnik imao utisak da aplikacija tečno radi. Uz upotrebu fragmenata ovaj problem se prevazilazi tako što je sada nepotrebno resetovanje jer fragment ne traži menjanje jedne velike celine, celog ekrana, već samo delova ekrana što traje znatno kraće i ne utiče na rad aplikacije.

Svaki fragment predstavlja nezavisan modul koji je tesno povezan sa aktivnosti u kojoj se nalazi. Dok je aktivnost u stanju rada, može se manipulisati nezavisno svakim fragmentom, kao

što je dodavanje, uklanjanje ili transakcija između fragmenata. Pri izvršavanju transakcije, fragment se može dodati u pozadinski stek kako bi mu se kasnije moglo pristupiti povratkom na isti. Ovaj postupak se izvršava pod nadležnosti *FragmentManager-a* povezanog sa aktivnosti. Iz pozadinskog steka se fragmenti mogu pozivati pritiskom na *Back* dugme. Svaki fragment sadrži informacije o sebi (identifikator i oznaku) kako bi kasnije *FragmentManager* znao na koji fragment da se vrati, odnosno na koju se poziciju iz pozadinskog steka treba vratiti.

3.1.1 Životni ciklus fragmenta

Životni ciklus fragmenta je dosta složeniji od životnog ciklusa aktivnosti i mora se обратити pažnja kada je određena funkcionalnost moguća. Sa slike 3.1 se može videti da je fragment strogo zavisan od aktivnosti u kojoj se nalazi i da prolazi kroz nekoliko koraka, dok bi se kod aktivnosti ovaj broj koraka znatno smanjio.

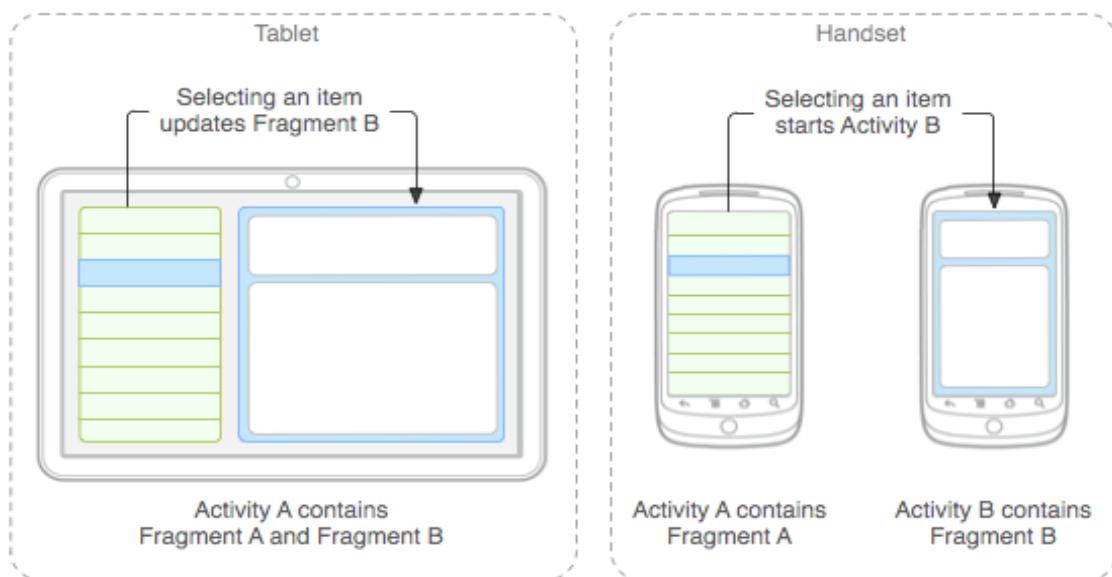


Slika 3.1 Životni ciklus fragmenta

Na samom početku fragment je kreiran. Sada postoji kao objekat u memoriji. Prvo što se dešava je inicijalizacija fragment objekta. Ovo je redovan postupak ako se fragment pravi iz već sačuvanog stanja. Sa kljentske strane, dobijanje fragment objekta se vrši pozivanjem funkcije *newInstance()* koja vraća fragment sa argumentima. Ostale funkcije navedene na slici se bave funkcionalnostima vezanim za fragment.

3.2 Primena fragmenata u korisničkoj sprezi

Fragment je prvenstveno napravljen da bi aplikacije bile skalabilnije između pametnih telefona i tablet uređaja. [8]



Slika 3.2 Primeri upotrebe fragmenata na ekranima različitih dimenzija

Na slici 3.2 su prikazani primeri upotrebe fragmenata na uređajima sa ekranima različitih dimenzija. Konkretno, na slici, su prikazani pametan telefon i tablet uređaj. Primer sa tablet uređajem pokazuje najbolji primer upotrebe fragmenata u korisničkoj sprezi. U ranijim verzijama Androida, pre verzije 3.0 i objavljuvanja fragmenata, realizacija ovakvog grafičkog prikaza je bila kompleksna, a kod je bio teško održiv. Svaki član liste sa primera prikazanog na tablet uređaju ima svoj grafički prikaz u fragmentu B, što znači da imamo veliku modularnost kada upotrebom fragmenata, te svaki fragment možemo po želji menjati. Fragmenti nam omogućavaju da za sekciju B koristimo različite Android XML datoteke za grafički prikaz. Obrada svakog dela treba da se obavlja u samom fragmentu, tako da gubimo problem pristupa grafičkim elementima i da li ti elementi postoje.

Na primer, ukoliko imamo 10 članova liste i ako svaki pojedinačni prikaz člana u sekciji B ima po 50 grafičkih elemenata, to znači da sa fragmentima obrađujemo odjednom 50 elemenata, dok bez njih, obrađujemo istovremeno čak 500. Iz ovoga se može lako zaključiti zbog čega dobijamo na jednostavnosti kreiranja aplikacija upotrebom fragmenata.

U istom primeru se može primetiti da i pri menjanju izgleda sekcije B, opet imamo isti slučaj 50 elemata sa korišćenjem fragmenata, odnosno čak 500 bez njih, jer sve što treba da uradimo je da zamenimo fragment i nastavimo da radimo sa elementima drugog fragmenta, dok bez njih u jednoj XML datoteci moramo eksplisitno da pazimo sa kojim elementima radimo, odnosno koje elemente prikazujemo, a koje ne.

Sad vidimo i koliko potencijalno veliko čuvanje memorije imamo. Ne moramo da zauzimamo memoriju za sve elemente odjednom, već samo za deo elemenata koji koristimo u fragmentu. Samim tim, ne moramo da brinemo hoće li ili neće sakupljač otpadaka (eng. *Garbage Collector*) osloboditi deo memorije koji nam treba.

Takođe možemo imati i nevidljive fragmente, koji ili čekaju da budu prikazani, ili uopšte nemaju grafičke elemente već služe kao radnici u Aktivnosti. [9]

3.3 Razlaganje grafičke korisničke sprege na manje celine

Koristeći fragmente moguće je razložiti grafičku korisničku spregu na manje gradivne celine kako bi se dobilo na uštedi procesorskog vremena, sistemskih resursa i modularnosti koda.

Grafička korisnička sprega se može posmatrati kao jedna velika celina koju sačinjavaju fragmenti. Fragmenti omogućavaju da prethodna, kompleksna i nepraktična rešenja za razvoj grafičke korisničke sprege pojednostavimo i ubrzamo rad aplikacije.

Na slici 3.3 je predstavljena raspodela fragmenata u grafičkoj korisničkoj sprezi. Ovi fragmenti popunjavaju ekran i vrše međusobnu tranziciju kada grafička sprega izda zahtev za to.



Slika 3.3 Raspodela fragmenata u grafičkoj korisničkoj sprezi

Levi deo ekrana je statički postavljen fragment sa određenim elementima. Desni i veći deo ekrana predstavlja deo grafičke sprege koji se menja u zavisnosti od odabira fragmenta za

prikazivanje od strane korisnika. Tako u desnom delu postoji nekoliko fragmenata koji vrše međusobnu tranziciju.

Moduli od značaja za reprodukovanje TV sadržaja se mogu podeliti u 3 grupe:

- Reprodukcija uživo (eng. *LiveTV*)

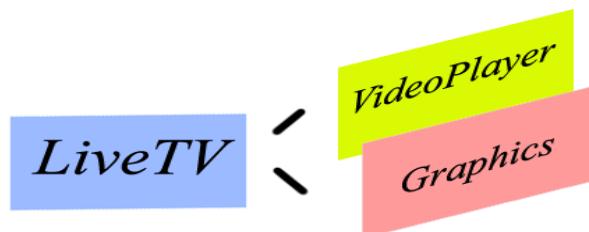
Glavni modul koji sadrži spisak kanala spremnih za reprodukovanje. Nalazi se u desnom delu ekrana gde se vrši odabir željenog kanala za reprodukovanje.

- Modul za reprodukciju video zapisa (eng. *VideoPlayer*)

Modul koji sarži komponentu za prikazivanje video sadržaja. Nalazi se preko celog ekrana, dok će grafičke komponente modula biti prikazane preko njega.

- Grafički modul reprodukcije video zapisa

Modul koji služi za isertavanje sadržaja elemenata u grafičkoj korisničkoj sprezi koji su potrebni za navigaciju kroz reprodukciju video zapisa. Nalazi se preko celog ekrana i može se ukloniti klikom miša na za to predviđeno mesto.



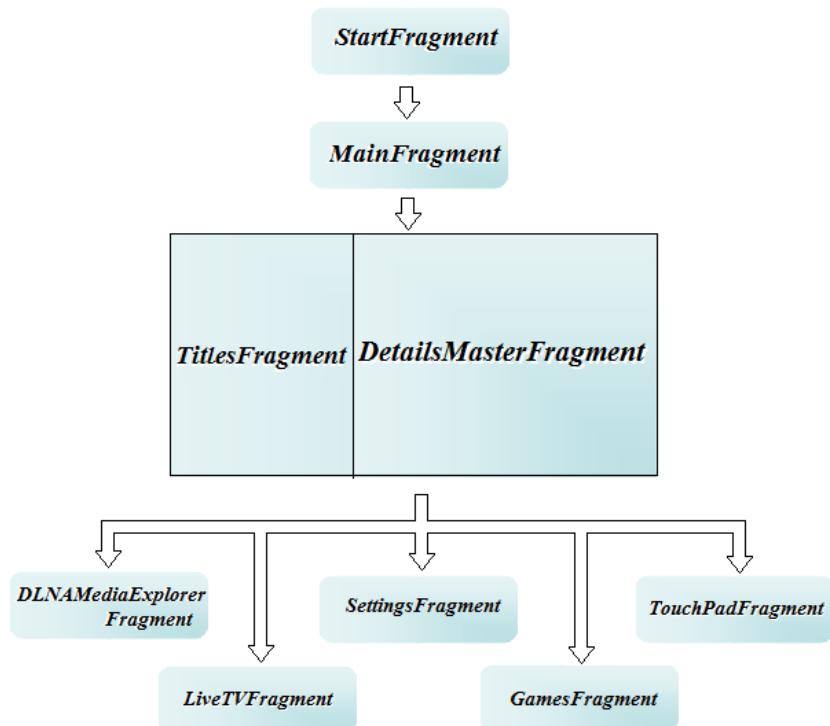
Slika 3.4 Podela reprodukcije TV saržaja po modulima

4. Programsko rešenje

U ovom poglavlju su prikazani detalji realizacije rešenja. Programsko rešenje se sastoji od 2 modula koji ispunjavaju ceo ekran i realizovana su u programskom jeziku Java uz Android programsku podršku. Prvi modul se sastoji od statičke liste fragmenata, a drugi modul sačinjavaju dinamički fragmenti koji vrše međusobno smenjivanje.

4.1 Tranzicija fragmenata u grafičkoj korisničkoj sprezi

Blok dijagram tranzicije predstavlja promenu fragmenata odnosno njihovo pojavljivanje na ekran ili menjanje sa drugim koji se nalazi na istom mestu.



Slika 4.1 Raspodela fragmenata sadržanih u grafičkoj sprezi

Svi fragmenti proizilaze iz *StartFragmenta* odnosno svaki fragment ima iste osobine kao i početni fragment. Ceo ekran se sastoji iz dva dela koja popunjavaju fragmenti. *TitlesFragment* popunjava levu stranu ekrana i predstavlja listu koja sarži spisak fragmenata koji se menjaju. Na desnoj strani ekrana se nalaze ostali fragmenti izvedeni iz *DetailsFragment*, njegovi potomci:

- *DLNAMediaExplorerFragment*
- *SettingsFragment*
- *LiveTVFragment*
- *GamesFragment*
- *TouchPadFragment*

Svi ovi fragmenti se nalaze na desnoj strani ekrana ali je samo jedan vidljiv u određenom trenutku. Izabiranjem po listi sa leve strane se vrši tranzicija pomenutih fragmenata

4.2 Moduli, opis i funkcije

U ovom poglavlju su opisani programski moduli, kao i njihove funkcije potrebne za implementaciju ovog rešenja.

4.2.1 *DetailsMasterFragment*

Funkcije za rad sa fragmentima:

```
public void onInflate(Activity activity, AttributeSet attrs, Bundle bundle) {}
```

Funkcija koja se poziva kada se pozove funkcija *setContentView()* iz prvobitnog prozora. Ovo prosleđuje karakteristike fragmenta *AttributeSet*. *onInflate()* funkcija se koristi za čitanje ovih atributa i čuvanje istih za kasniju upotrebu. U ovom delu životnog ciklusa fragmenta je suviše rano da se išta uradi sa UI dizajnom jer fragment još nije asociran sa aktivnosti. Sledeći korak predstavlja ovu vezu.

```
public void onAttach(Activity myActivity) {}
```

Poziva se kada se fragment prvi put poveže sa svojom aktivnošću. Ovom funkcijom se dobija referenca aktivnosti na korišćenje koja može da se koristi za informacije o aktivnosti kao i druge operacije vezane za ovaj spoj. Posle ovog poziva je nemoguće

pozvati ponovno postavljanje argumenata jer se ono može postaviti samo pri inicijalizaciji na samom početku.

```
public View onCreateView(LayoutInflater li, ViewGroup vg, Bundle bundle) {}
```

Funkcija vraća pogled kao povratnu vrednost. Ako je ova vrednost prazna (eng. *NULL*) znači da se ovaj fragment neće pojaviti na ekranu.

```
public void onActivityCreated(Bundle bundle) {}
```

Poziva se kada se fragment prvi put napravi. Mesto gde korisnik prvi put može videti na ekranu šta se zapravo dešava sa fragmentom. Poziva se kada se obavi funkcija *onCreate()*. Pri pozivu ove funkcije sigurno su svi fragmenti povezani sa svojim aktivnostima.

```
public void onSaveInstanceState(Bundle bundle) {}
```

Poziva se kada se zatraži od fragmenta da sačuva svoje dinamičko stanje, kako bi se kasnije moglo rekonstruisati u novoj instanci procesa.

```
public void onDestroyView() {}
```

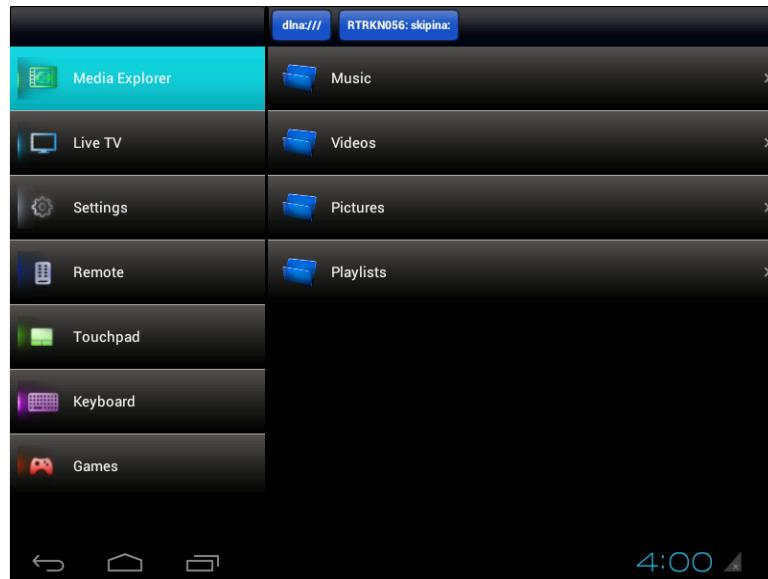
Poziva se kada fragment treba da se ubije i sačuva njegovo stanje za kasniju upotrebu. Poziv se može obaviti samo nakon izvršenja funkcije *onCreateView()* i odvezivanja fragmenta od aktivnosti. Može se pozvati bez obzira da li je funkcija *onCreateView()* vratila pogled ili *NULL* kao povratnu vrednost.

```
public void onDetach() {}
```

Poslednja funkcija u životnom ciklusu fragmenta. Poziva se kada se fragment odvezuje od svoje aktivnosti i oslobađa zauzete attribute. Poziv se obavlja nakon uspešno obavljenje funkcije *onDestroy()* kada fragment više nije u upotrebi.

4.2.2 ***DLNAMediaExplorerFragment***

Fragment za navigaciju kroz sve dostupne DLNA poslužioce (eng. *Servers*). Korisnik može da izabere da pregleda slike sa poslužiocu, sluša muziku ili gleda video zapis koji se nalaze na poslužiocu.

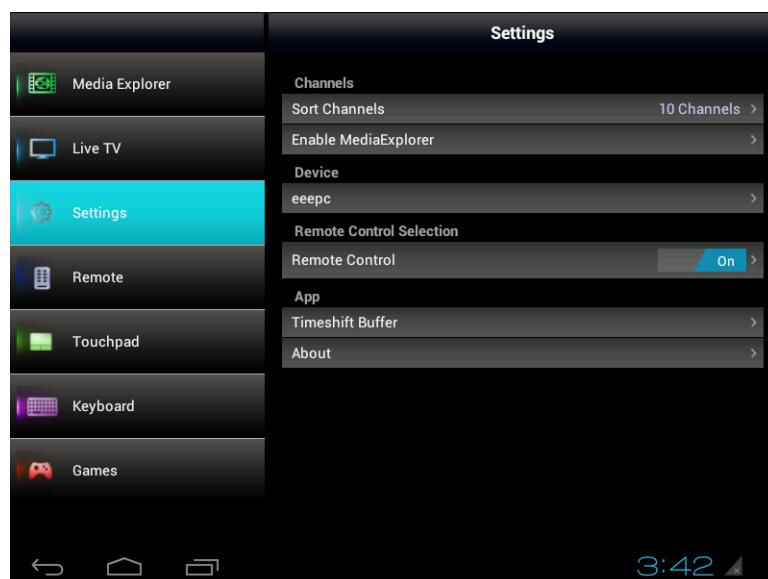


Slika 4.2 Izgled fragmenta za navigaciju kroz DLNA poslužioce

4.2.3 ***SettingsFragment***

Klasa koja opisuje opcije i podešavanja dostupna u aplikaciji:

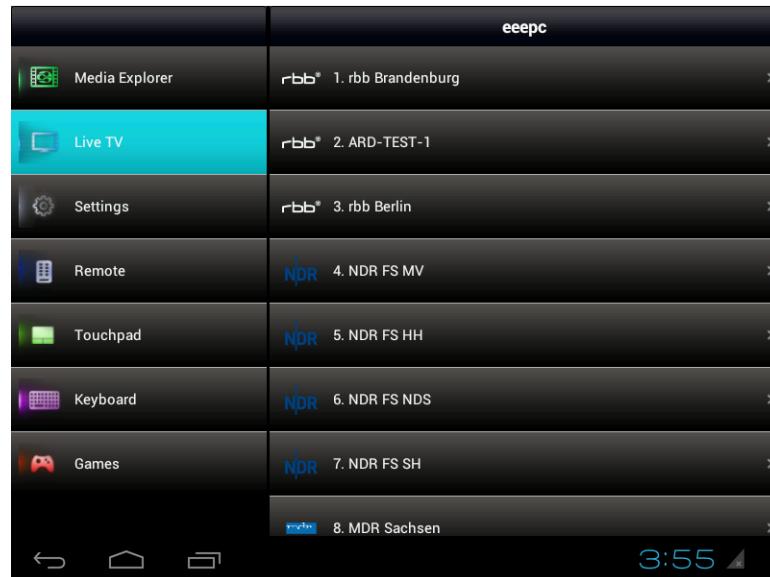
- Odabir protokola komunikacije
- Pregled dostupnih DLNA poslužioce



Slika 4.3 Izgled fragmenta za podešavanja

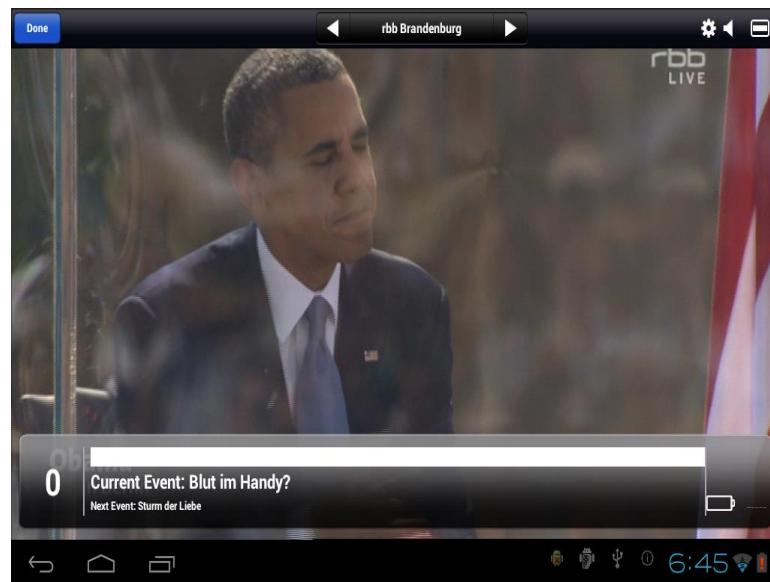
4.2.4 LiveTVFragment

Fragment za reprodukciju TV sadržaja koji korisniku nudi listu kanala dostupnih za gledanje odnosno spremni za reprodukciju od strane poslužioca.



Slika 4.4 Lista dostupnih kanala za reprodukciju

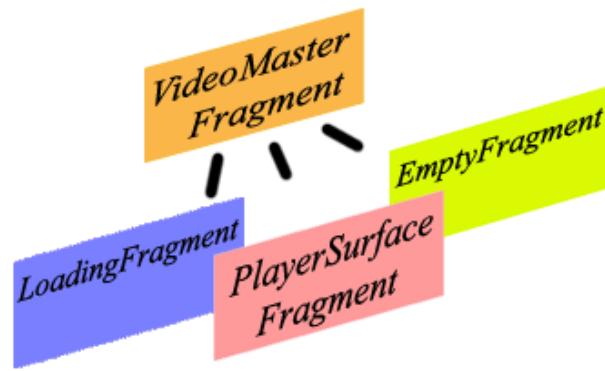
Posle odabira željenog kanala pojavljuje se ekran sa modulom za reprodukciju video zapisa i pušta se odabrani video. Takođe se pojavljuje i grafika, sastavni deo ovog modula, koja pomaže u interakciji sa korisnikom.



Slika 4.5 Modul za reprodukciju video zapisa i reprodukovani kanal

4.2.4.1 Organizacija grafičkih elemenata modula za reprodukciju video zapisa

Ekran se sastoji iz 3 sloja koja su istovremeno prisutna na ekranu. Korisnik vidi onaj sloj koji mu je najbliži odnosno onaj koji je iznad ostalih slojeva.

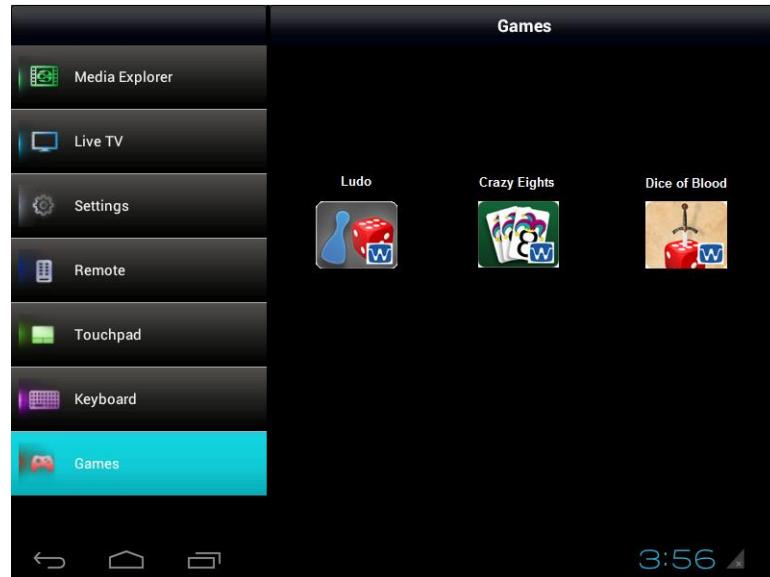


Slika 4.6 Organizacija grafičkih elemenata modula za reprodukciju video zapisa

LoadingFragment – fragment koji opisuje prelazak sa jednog kanala na drugi na kom se pojavljuje progres, logo i ime kanala na koji se prebacuje

PlayerSurfaceFragment – fragment koji služi za iscrtavanje grafičkih elemenata modula (ime trenutnog kanala, status mreže, status baterije, opcije za pojačavanje zvuka...)

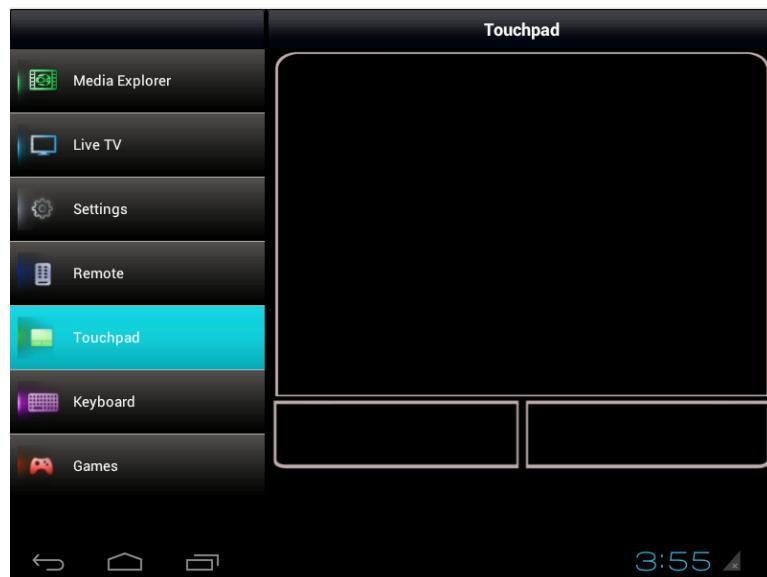
4.2.5 GamesFragment



Slika 4.7 Izgled fragmenta za igrice

Predstavlja se lista dostupnih igrica koje su optimizovane za više ekrana istovremeno.

4.2.6 *TouchPadFragment*



Slika 4.8 Izgled fragmenta koji reprezentuje kompjuterskog miša

Predstavlja se površina po kojoj se pomeranjem prsta reprezentuje pomeraj na TV ekranu. Može poslužiti ako je potrebno postaviti fokus na određeno mesto na ekranu koje običnim daljinskim upravljačem nije moguće.

5. Ispitivanje i verifikacija

Ispitivanje se sastoji iz dve faze. Prva faza obuhvata programsko ispitivanje funkcionalnosti i navigacije, koja su napisana u Java programskom jeziku, dok druga faza sadrži direktno ispitivanje od strane krajnjih korisnika.

5.1 Programsко испитивање функционалности и навигације

Ispitivanje i verifikacija navigacije i interakcije sa korisnikom u Android aplikaciji se svodi na oponašanje upotrebe aplikacije od strane korisnika. Pri tome trebalo je ispitati da li se svaka komponenta, vidljiva na ekranu, može obeležiti i izabrati, da li je svaki pritisak dugmeta ispravno tretiran odgovarajućom reakcijom, odnosno kakve su posledice nasumičnog pritiskanja dugmića. Za potrebe ispitivanja i verifikacije u Android aplikaciji je napravljen poseban modul koji je sačinjen od niza funkcija grupisanih po vrsti testova. Tako prepoznajemo sledeće grupe testova:

- Grupa testova koja ispituje da li se svaka stavka modula može obeležiti
- Grupa testova oponašanja kontrolisanih pritisaka dugmića na daljinskom upravljaču
- Grupa testova oponašanja nekontrolisanih pritisaka dugmića na daljinskom upravljaču

Sva ispitivanja su izvršena korišćenjem Android komponente *Instrumentation*, koja dozvoljava pravljenje događaja koji bi se dogodio pritiskom određenog dugmeta na daljinskom upravljaču, i prosleđivanja ovog događaja aplikaciji na tretiranje.

Prva grupa testova ispituje da li se svaka stavka modula može obeležiti. Za potrebe ovog ispitivanja, sve komponente modula su povezane sa osluškivačem (eng. *Listener*) koji reaguje kada je ova komponenta obeležena. Pošto je u pitanju kontrolisan test gde se unapred znaju

događaji koji će biti prosleđeni aplikaciji na tretiranje, lako se može izračunati koja komponenta će koliko puta biti obeležena. Na kraju izvršavanja ove grupe testova prikupljeni su rezultati od osluškivača povezanih sa određenim komponentama i upoređivanjem vrednosti zaključeno je da je ova grupa testova uspešno izvršena.

Druga grupa testova se bavi ispitivanjem da li pritisak dugmeta na daljinskom upravljaču uzrokuje nekom upravljačkom akcijom. Prilikom izvršavanja ove grupe testova, pored osluškivača koji reaguju na odabir i obeležavanje određene komponente modula, od bitnog značaja je bilo postavljanje i osluškivača na module svih nivoa, koji reaguju kada je određeni modul prikazan na ekranu, odnosno sklonjen. Na ovaj način ispitano je da li se pritiskom određenih tastera prikazuju određeni moduli nižeg nivoa, odnosno da li su ti moduli funkcionalni. Prikupljanjem podataka od osluškivača i upoređivanjem sa događajima koji su prosleđeni aplikaciji za tretiranje, zaključeno je da je ova grupa testova uspešno izvršena i da je funkcionalnost modula ispravna.

Grupa testova koja se bavi oponašanjem nekontrolisanih pritisaka dugmića pravi nasumične događaje i prosleđuje ih aplikaciji na tretiranje. Na ovaj način, testirani su zlonamerni pokušaji zbijivanja aplikacije i dovođenja do fatalnih posledica po istu. U slučaju osluškivača koji reaguje, kada je pritisnuto dugme, projektantu aplikacije se nudi mogućnost da tretira pritiske dugmića na način na koji to njemu odgovara. Tako se npr. pritiskom dugmeta liste kanala, korisniku na ekranu prikazuje modul za reprodukciju kanala. Ukoliko je pritisnuto dugme prevod, korisniku će biti prikazan modul za prevod, dok za sve ostale dugmiće postoji mogućnost da projektant odredi da se oni ne tretiraju. Na ovaj način će samo događaji koji imaju smisla, odnosno koji su tretirani od strane projektanta prilikom pisanja aplikacije, rezultovati nekom upravljačkom akcijom i prikazom na ekranu. Ovu činjenicu su i potvrdili rezultati izvršavanja testova iz ove grupe, koji su pokazali da su samo neki prosleđeni događaji bili tretirani određenom upravljačkom akcijom, dok su ostali bili ignorisani i nisu rezultovali nikakvom akcijom.

Izvršavanjem više puta nasumično odabranih grupa testova i ispitivanjem dobijenih rezultata, zaključeno je da navigacija i interakcija sa korisnikom u Android aplikaciji ispravno radi.

5.2 Direktno ispitivanje od strane krajnjeg korisnika

U okviru ovog ispitivanja okupljena je grupa ljudi različitog uzrasta koja do tад nije imala nikakvog iskustva u radu sa Android aplikacijom.

Prilikom ispitivanja, primećeno je da je starijim ljudima bilo teže da se snađu prilikom korišćenja grafičke korisničke sprege aplikacije, odnosno trebalo im je više vremena da „nauče“ da koriste aplikaciju, za razliku od mlađih „računarsko-obrazovanih“ ljudi, koji su sa lakoćom uspevali da pronađu željenu stavku.

Većina ljudi je bila zadovoljna načinom na koji je projektovana grafička korisnička sprega, a takođe su se i složili da su grafičke komponente dobro predstavljene i lako prepoznatljive.

6. Zaključak

U ovom radu opisano je jedno rešenje projektovanja grafičke korisničke sprege aplikacije za reproducovanje TV sadržaja za prenosne uređaje bazirane na Android platformi..

Prilikom projektovanja grafičke korisničke sprege za Android aplikaciju, zadatku je zadržati izgled aplikacije što jednostavnijim kako bi se korisnici svih uzrasta snašli u svakodnevnom korišćenju. Takođe je bitno iskoristiti novije tehnologije dostupne sa Android programskom podrškom verzije 3.0, koje su potrebne radi uštete sistemskih resursa i brzine rada čime se dobija tečniji rad aplikacije.

Programskim ispitivanjem Android aplikacije sa očekivanim rezultatima zaključeno je da je kretanje, odnosno navigacija kroz aplikaciju ostvarena ispravno, a da se grafičke komponente ponašaju u skladu sa njihovom realizacijom. Takođe, ispitivanje sa nasumično zadatim kodovima komandi je prošlo uspešno, što potvrđuje da je Android aplikacija u potpunosti iskontrolisana i projektovana tako da može da obradi pritisak bilo kog dugmeta sa Android prenosnog uređaja.

Dalja unapređenja ovakve aplikacije bi bile usmerene ka iskorišćenju funkcionalnosti digitalne televizije na Android prenosnom uređaju kao što je elektronski programski vodič (eng. *EPG – Electronic Programming Guide*). Razvoj ove funkcionalnosti ide u smeru kontrole gledanog sadržaja preko *EPG-a* kao što je pregled celokupnog sadržaja (eng. *Full EPG*) na celom ekranu ili snimanja zakazanog po želji korisnika (eng. *Schedule Recording*). Zakazano snimanje predstavlja koncept koji se oslanja na elektronski vodič i na osnovu njegovog sadržaja korisniku se daje mogućnost zakazivanja snimanja u budućnosti kada počinje određena emisija.

Dalji razvoj projektovanja grafičke korisničke sprege predstavlja standardizaciju komunikacije između DTV prijemnika i Android prenosnog uređaja korišćenjem RVU

protokola. Ovaj protokol se zasniva na udaljenoj sprezi (eng. *Remote User Interface – RUI*) za iscrtavanje grafičkih elemenata na strani poslužioca (eng. *Server*) i DLNA (eng. *Digital Living Network Aliance*) standardu za komunikaciju kroz koji se ova grafička sprega šalje do korisnika (eng. *Client*). Time se dobija na univerzalnosti aplikacije odnosno korisničkog uređaja, kao i nezavisnost od fizičke arhitekture na kojoj se grafička sprega prikazuje.

7. Literatura

- [1] “Programming Android: Java programming for the new generation of mobile devices”, Zigurd R.Mednieks, Laird Dornin, G. Blake Meike, Masumi Nakamura, September 2012 Second Edition
- [2] Vladimir Kovačević, Miroslav Popović: *Sistemska podrška u relanom vremenu*, Univerzitet u Novom Sadu, Fakultet Tehničkih Nauka, 2002
- [3] Sajt za preuzimanje Android aplikacija <https://play.google.com/store?hl=en>
- [4] Sajt za naučna istraživanja o pametnim telefonima
<http://www.gartner.com/newsroom/id/2482816>
- [5] “Programming Android: Java programming for the new generation of mobile devices”, Zigurd R.Mednieks, Laird Dornin, G. Blake Meike, Masumi Nakamura, September 2012 Second Edition
- [6] N. Šoškić, D. Popov Tapavički, Nikola Kuzmanović, *Member, IEEE*, Milan Savić, Darko VIdaković, “Jedno rešenje razlaganja grafičke korisničke sprege na uređajima zasnovanim na Android operativnom sistemu radi poboljšanja performansi”, 57. Konferencija za elektroniku, telekomunikacije, računarstvo, automatiku i nuklearnu tehniku (ETRAN), Jun 2013.
- [7] Dejan Popov Tapavički, Nenad Šoškić, Aleksandar Oroz, Đorđe Golubović, Miodrag Temerinac, *Member, IEEE*, “Optimizacija grafičke korisničke sprege Java aplikacija za uređaje bazirane na Android operativnom sistemu”, 57. Konferencija za elektroniku, telekomunikacije, računarstvo, automatiku i nuklearnu tehniku (ETRAN), Jun 2013.
- [8] “Smashing Android UI”, Juhani Lektimaki, 2013
- [9] Sajt za podršku Android razvoja (bolje razumevanje tehnologije upotrebe fragmenata)
<http://stackoverflow.com/questions/10609268/difference-between-fragment-and-fragmentactivity>