SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

mRanger

**PROJEKT U SKLOPU stem revolucija u zajednici**

Varaždin, 2018.

Sadržaj

[1. Uvod 1](#_Toc528788296)

[2. Instaliranje potrebnih programa 1](#_Toc528788297)

[3. Izrada prvog projekta u Android Studio 2](#_Toc528788298)

[3.1 Izrada novog projekta 2](#_Toc528788299)

[3.2 Prvi koraci u Android Studio alatu 6](#_Toc528788300)

[3.3 Pokretanje projekta 15](#_Toc528788301)

[4. Dodavanje novog zaslona, slike i gumba 18](#_Toc528788302)

[4.1 Dodavanje novog zaslona 18](#_Toc528788303)

[4.2 Dodavanje slike 20](#_Toc528788304)

[4.3 Dodavanje gumba 25](#_Toc528788305)

[4.4 Prijelaz s jednog zaslona na drugi 27](#_Toc528788306)

# Uvod

Cilj ovog dokumenta je upoznati mladež, čime su ciljana skupina učenici osnovnih i srednjih škola, kao i svi oni koji žele naučiti nešto više o robotici. U ovom dokumentu ćemo se potruditi što jednostavnije proći kroz proces izrade Android aplikacije koja će služiti za daljinsko upravljanje mBot Ranger robotom, na koji ćemo se povezati putem Bluetooth-a.

Aplikacija će imati sljedeće funkcionalnosti:

* Povezivanje na mBot Ranger putem Bluetooth-a
* Daljinsko upravljanje robotom
* Mogućnost podešavanja brzine kretanja
* Izbjegavanje sudara robota i prepreke
* Bilježenje temperature zraka u bazu podataka

# Instaliranje potrebnih programa

Našu aplikaciju ćemo izraditi u Android Studiu, a to je razvojno okruženje koje služi za kreiranje Android mobilnih aplikacija. Detaljni prikaz instalacije ćemo preskočiti, no postoje razni sadržaji na internetu koji prikazuju kako instalirati Android Studio (npr. [Instalacija Android Studia](https://www.youtube.com/watch?v=YaLAAdh9RBg)).

Android Studio se može preuzeti na: [Android Studio - Službena stranica](https://developer.android.com/studio/install)

# Izrada prvog projekta u Android Studio

U nastavku, sve upute ćemo se potruditi prikazati uz slikovne materijale i komentare kako bi što lakše predočili postupak izrade aplikacije. Na pojedinim slikama se mogu pronaći obojeni okviri i strelice kako bismo usmjerili vašu pozornost na bitne elemente.

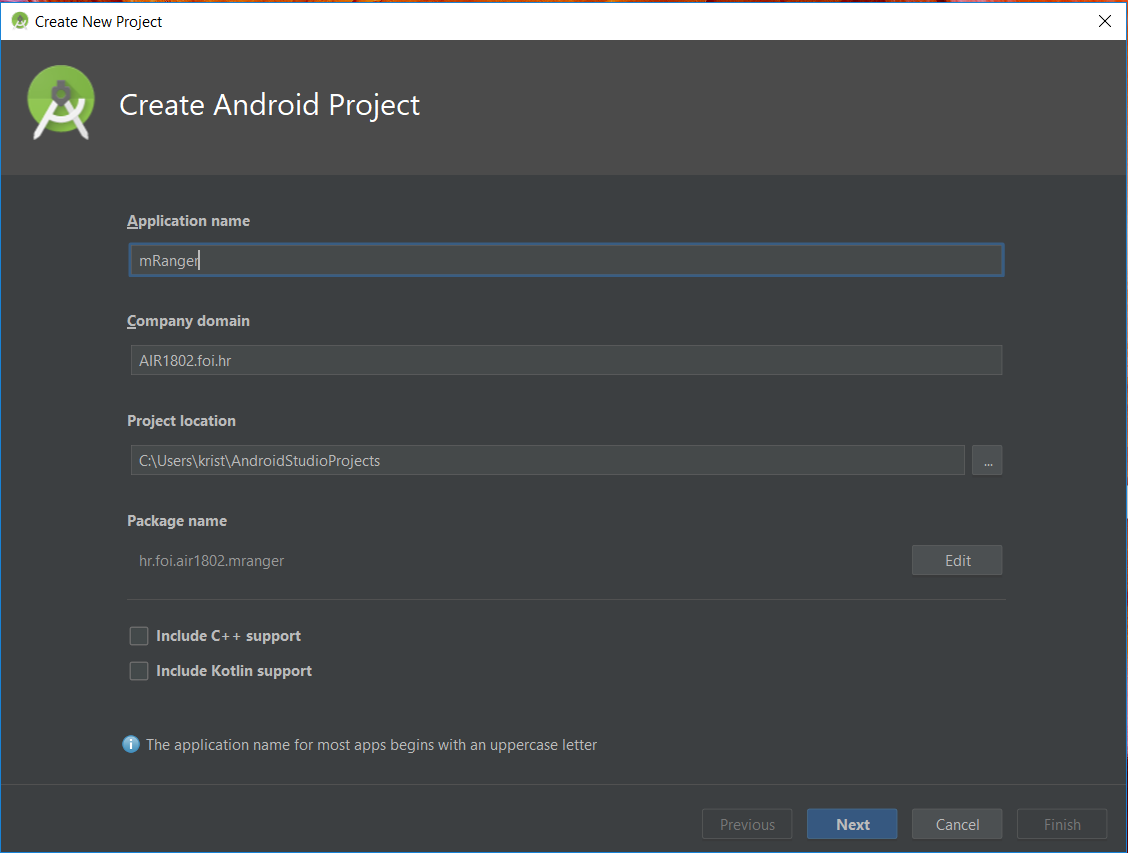
Želimo Vam puno sreće prilikom izrade vaše prve aplikacije za kontroliranje metalnog ljubimca. 😉

## 3.1 Izrada novog projekta

Slika 1. Početni prozor Android Studio

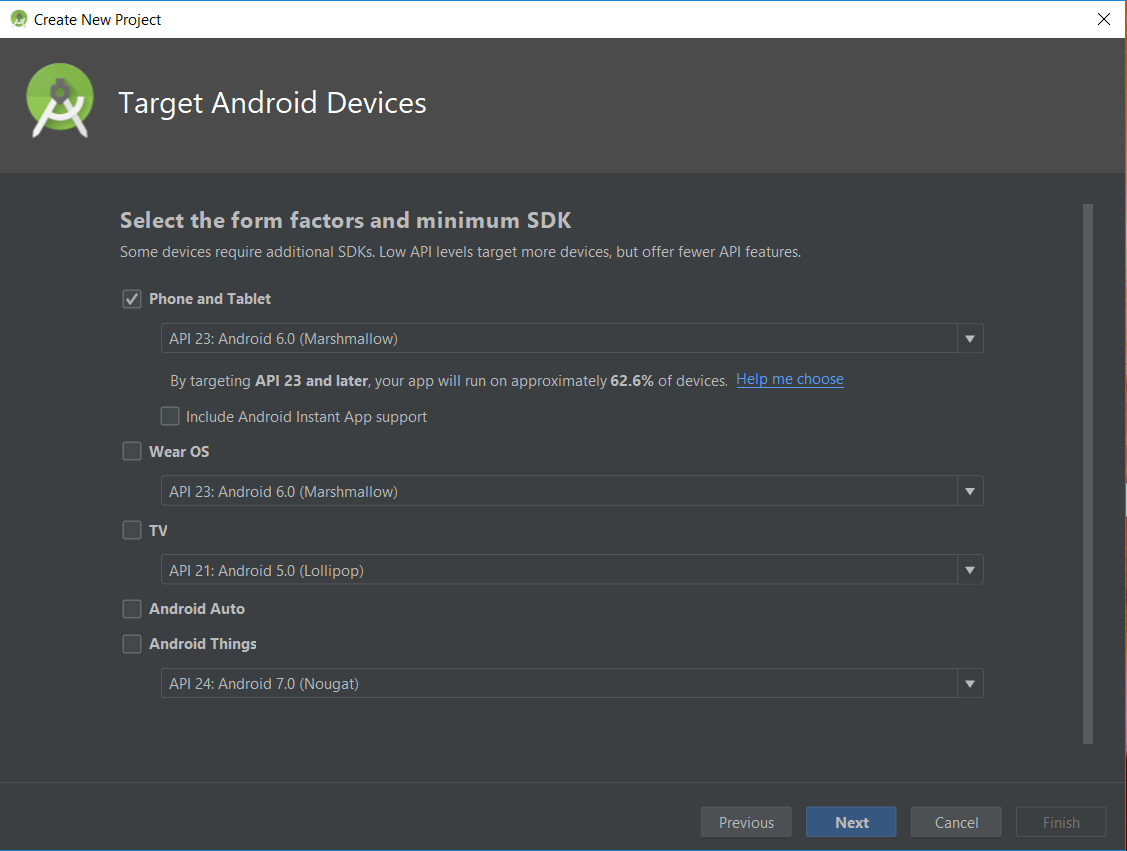
Prilikom pokretanja Android Studia otvara se prozor prikazan na slici iznad. Odaberemo prvu opciju iz prozora, *Start a new Android Studio project*, kojom ćemo započeti kreiranje novog projekta.

Time nam se otvara sljedeći prozor



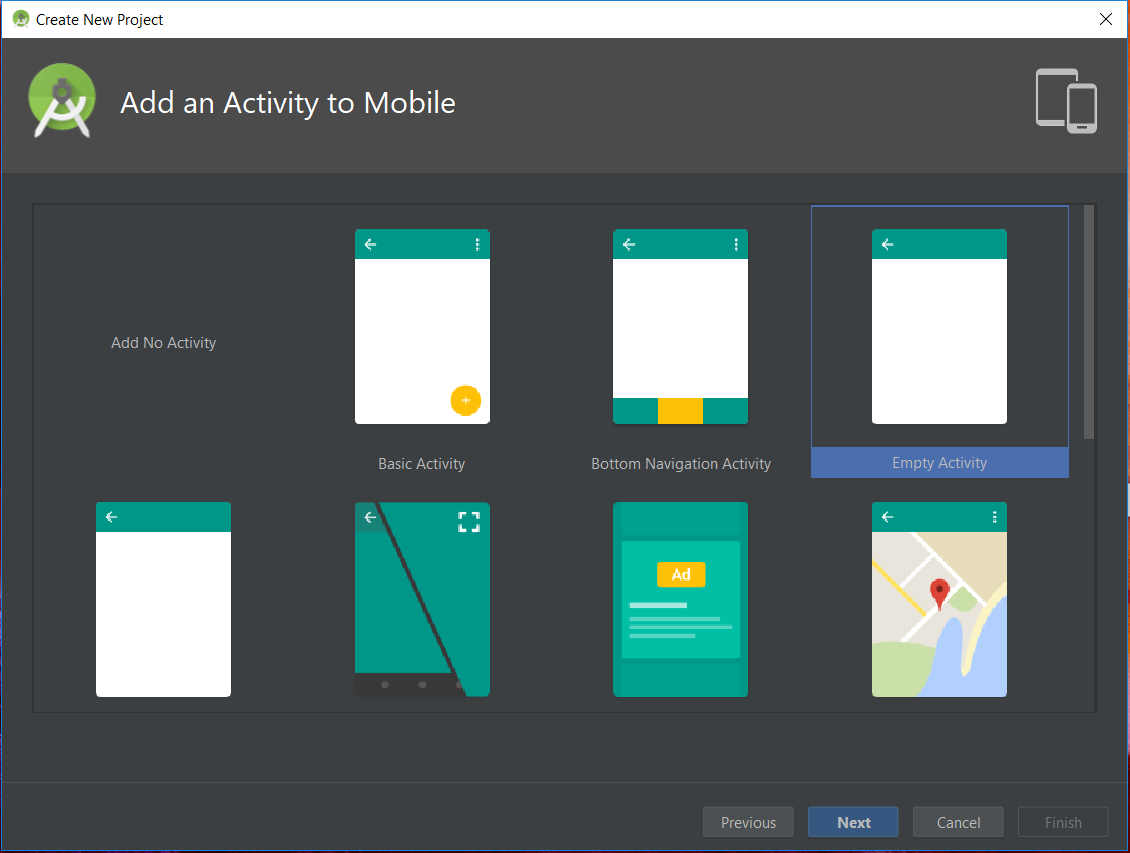
Slika 2. Imenovanje projekta

U ovom prozoru cilj je odrediti naziv projekta, podatke o tvrtki ako ona postoji (projekt će biti smješten u mapu *C:\Users\imeKorisnika\AndroidStudioProjects\tvrtka\mRanger).* Ostale opcije ostavljate takve kakve jesu i kliknete na *Next*.



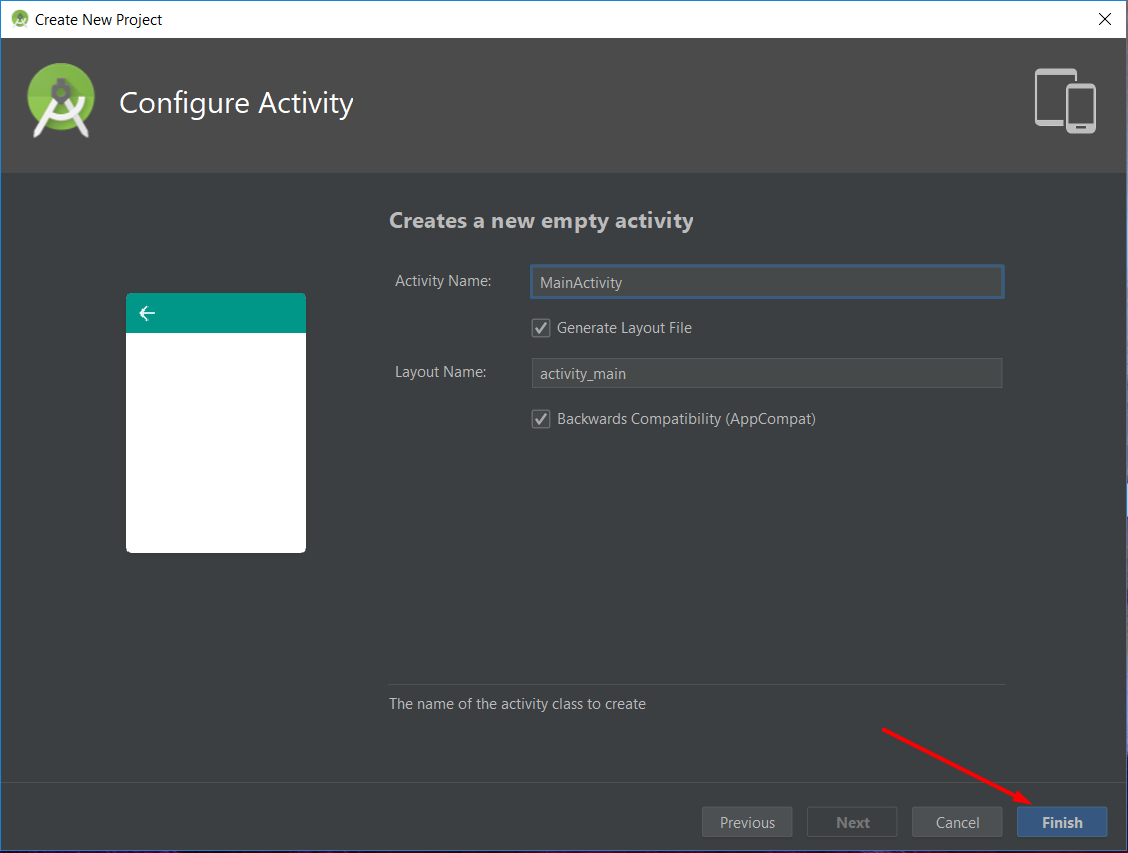
Slika 3. Odabir ciljanih uređaja

Nakon imenovanja projekta potrebno je odabrati vrste uređaje na kojima će se aplikacija izvoditi. U našem slučaju odabiremo *Phone and Tablet* opciju, te nakon toga biramo minimalnu verziju operacijskog sustava za android uređaje (minimalni zahtjevi sustava). Mi odabiremo predloženu opciju *API 23:Android 6.0 (Marshmallow)* prilikom čega dobivamo informaciju o postotku uređaja na kojemu će biti moguće koristiti aplikaciju. Zatim kliknemo *Next*.



Slika 4. Odabir početnog zaslona

U ovom prozoru imamo mogućnost jednog od predložaka za izgled početnog zaslona aplikacije. Odabiremo predloženi zaslon *Empty Activity* i kliknemo *Next*.



Slika 5. Naziv početnog zaslona

Na slici 5. je moguće odabrati željeno ime za početni zaslon, međutim ostavljamo sve kako je zadano i kliknemo na *Finish*.

## Slika na kojoj se prikazuje snimka zaslona, elektronički, monitor, zaslon Opis je generiran uz vrlo visoku pouzdanost3.2 Prvi koraci u Android Studio alatu

Slika 6. Sučelje Android Studio alata nakon otvaranja projekta

Sljedeća slika prikazuje sučelje Android Studio alata nakon otvaranja našeg novog izrađenog projekta. Na zaslonu nam se po početnim postavkama prikazuju dva taba (*activity\_main.xml* i *MainActivity.java*).

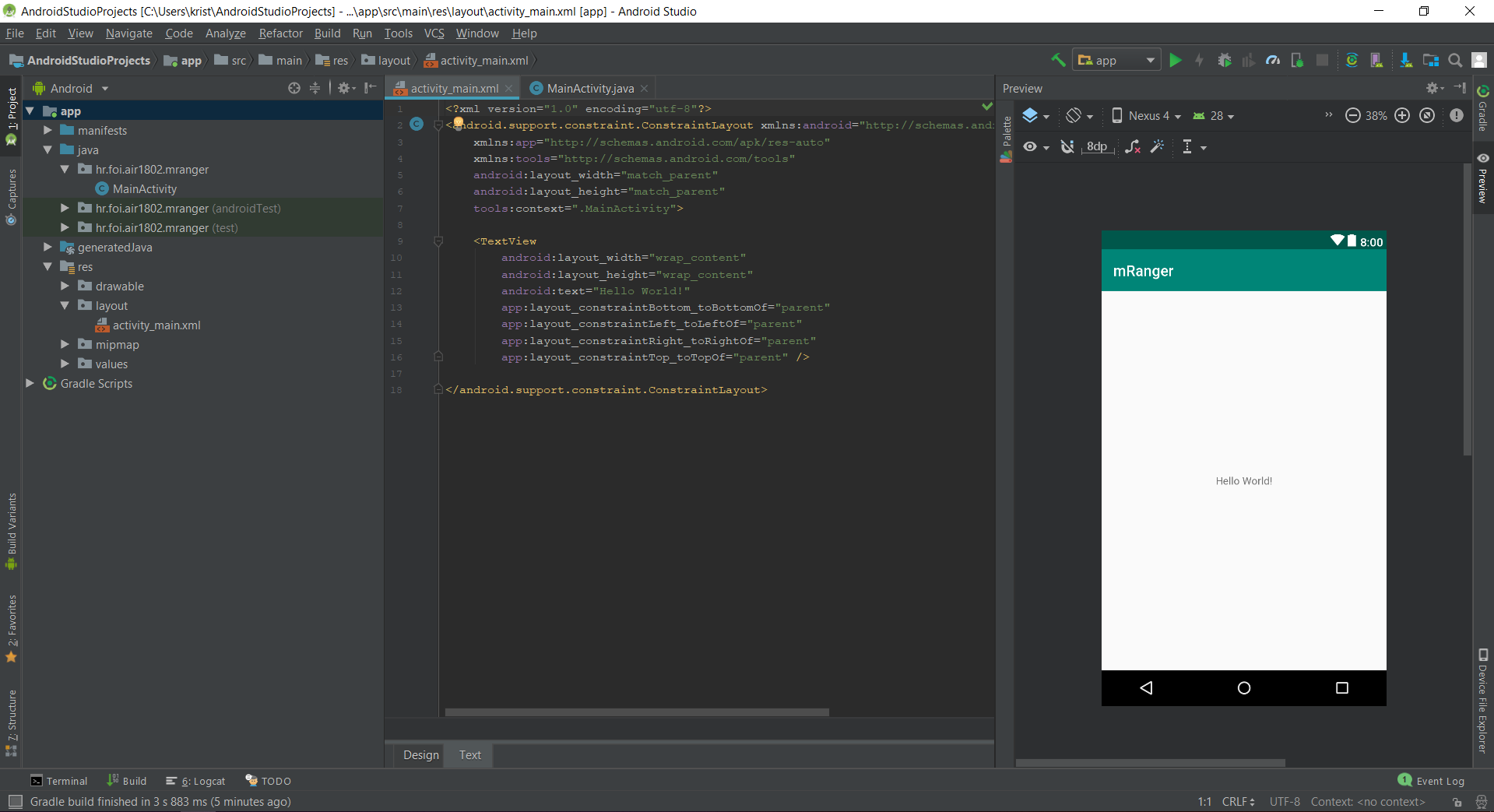
Slika na kojoj se prikazuje snimka zaslona, monitor, prijenosnik, sjedenje

Opis je generiran uz visoku pouzdanost

Na slici broj 7. vidimo početnu strukturu mape našeg projekta. Bitne mape su *java* i *res. U java* mapi, u podmapi sa nazivom naše tvrtke i aplikacije, nalazi se programski kod našeg početnog zaslona kojeg smo prethodno nazvali *MainActivity*. U mapi *res* u podmapi *Layout* nalazi se *activity\_main.xml*  u kojem se uređuje grafički izgled početnog zaslona naše aplikacije.

Slika 7. **java** i **res** mape

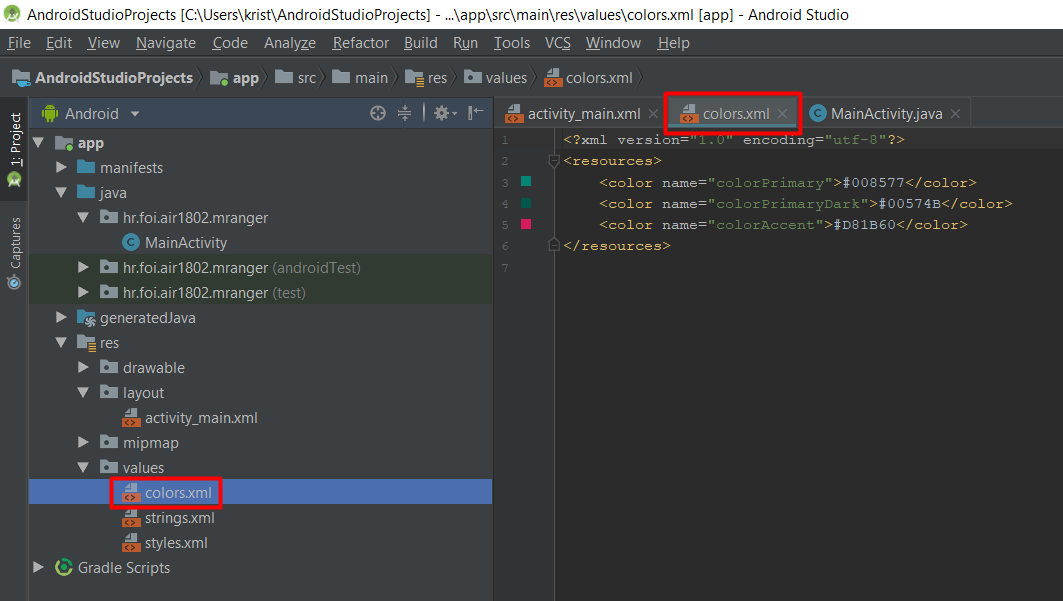
U dokumentu *MainActivity.java* nalazi se programski kod koji se izvršava na početnom zaslonu.



Slika 8. **XML** datoteka

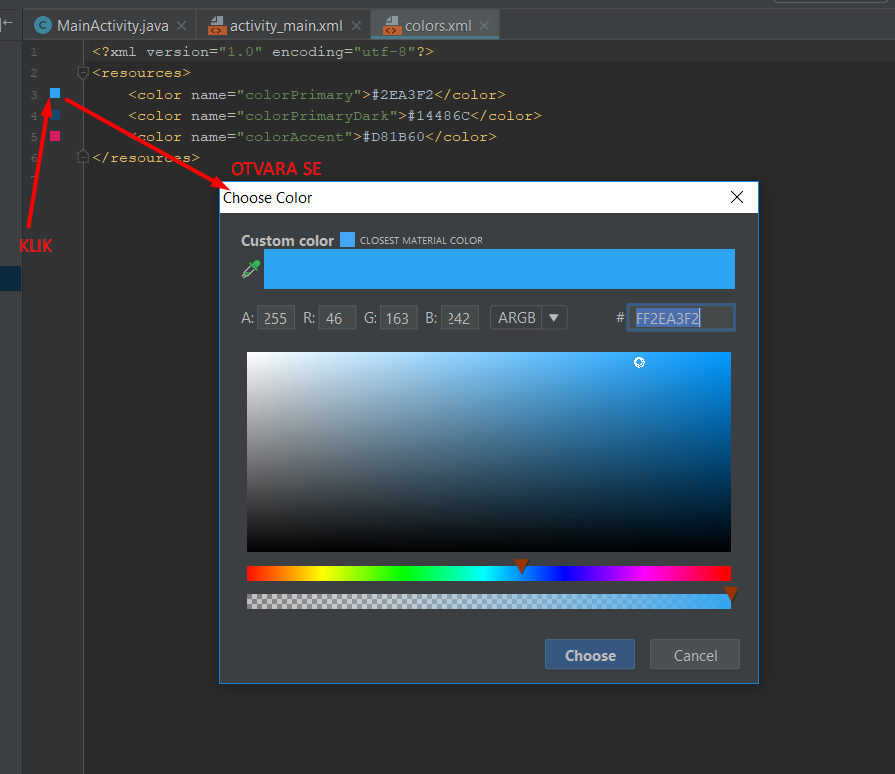
U dokumentu *activity\_main.xml* se nalazi programski kod u XML jeziku, koji definira grafički izgled našeg zaslona te podataka koje ćemo prikazivati na njemu. Po početnim postavkama Android Studio alata, na sredini zaslona se prikazuje tekst *Hello World!* Taj tekst možemo također uređivati u ovom XML dokumentu, no o tome ćemo reći nešto više ubrzo.

Predložene boje našeg početnog zaslona su nijanse zelene. Idemo to malo izmijeniti, ne želimo zelenu aplikaciju ! 😉

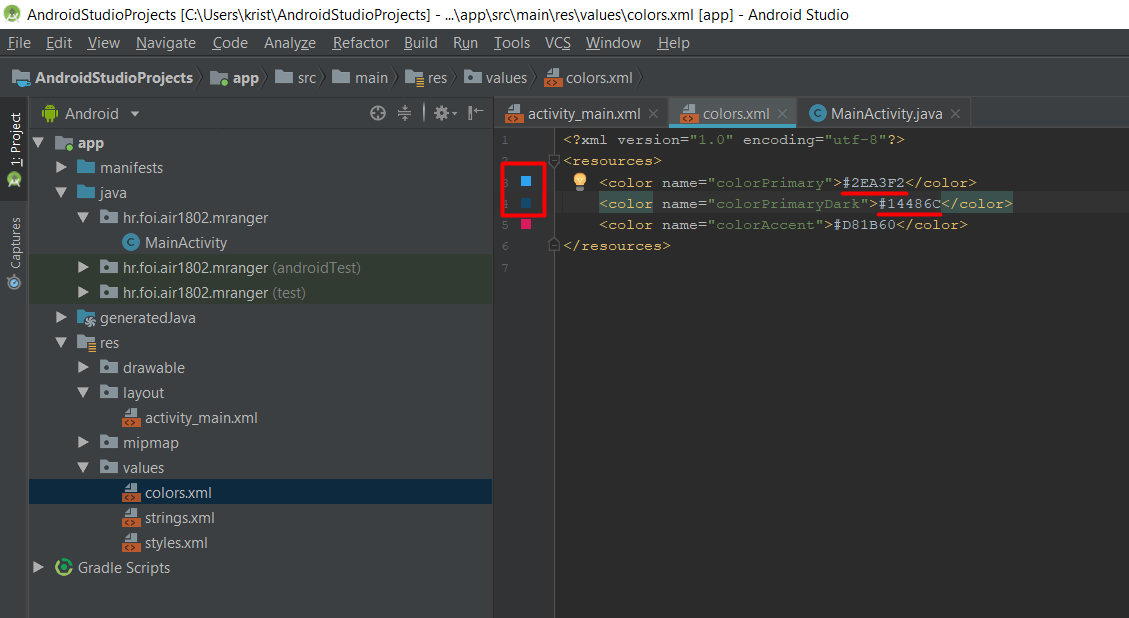


Slika 9. Promjena boja zaslona 1/2

U prethodno spomenutom **res** mapi, imamo mapu **values** koja sadrži datoteku **colors.xml** gdje možemo promijeniti boje zaslona naše aplikacije. Kao što možete vidjeti, boje zaslona su zapisane u heksadecimalnom zapisu.

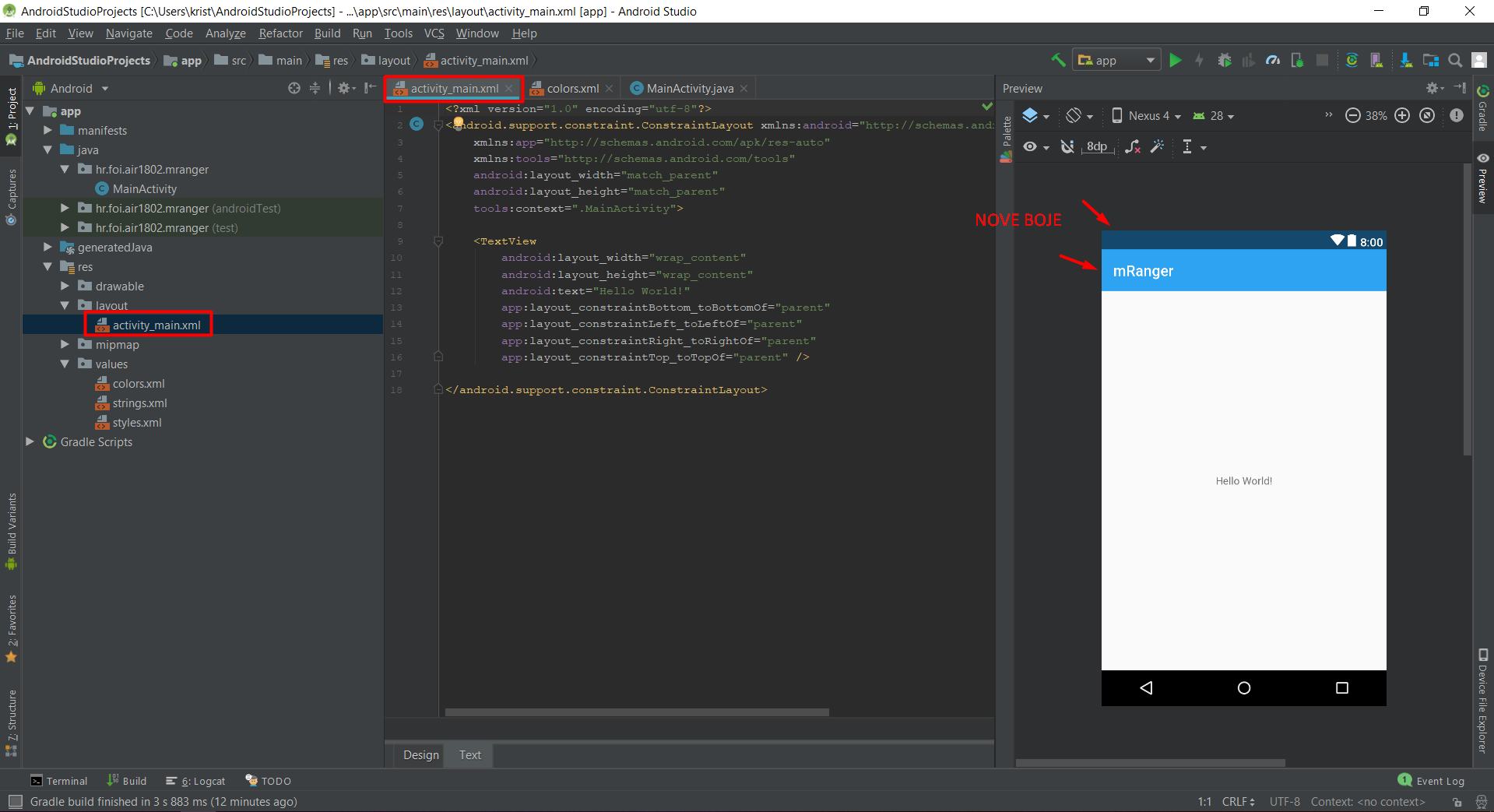
Klikom na kvadratić sa bojom pokraj linije koja sadrži heksadecimalni zapis boje, otvara nam se prozor gdje možemo odabrati željenu boju na vrlo jednostavan način, bez potrebe da znamo stvarni heksadecimalni kod boje.

Slika 10. Promjena boja zaslona 2/2



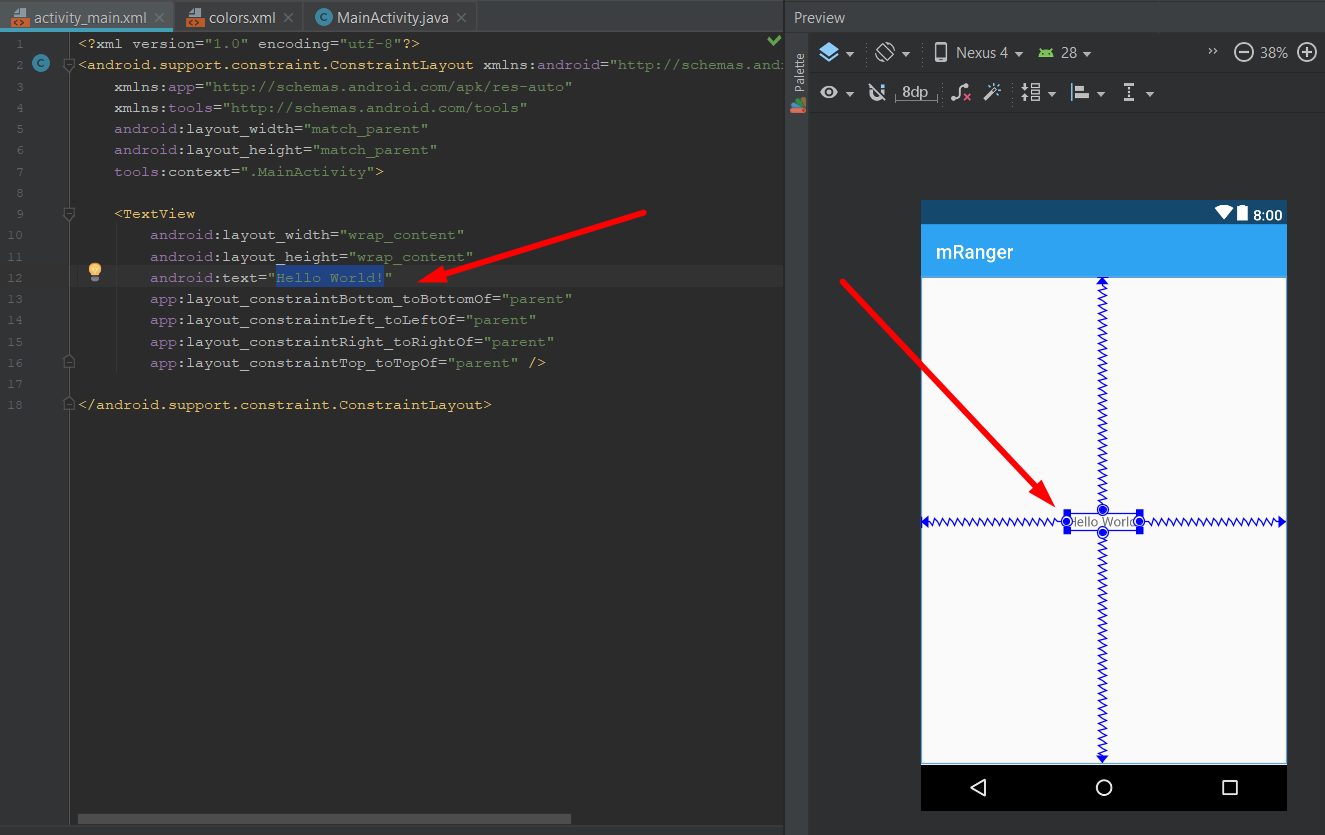
Slika 11. Promjena heksadecimalnog zapisa boje

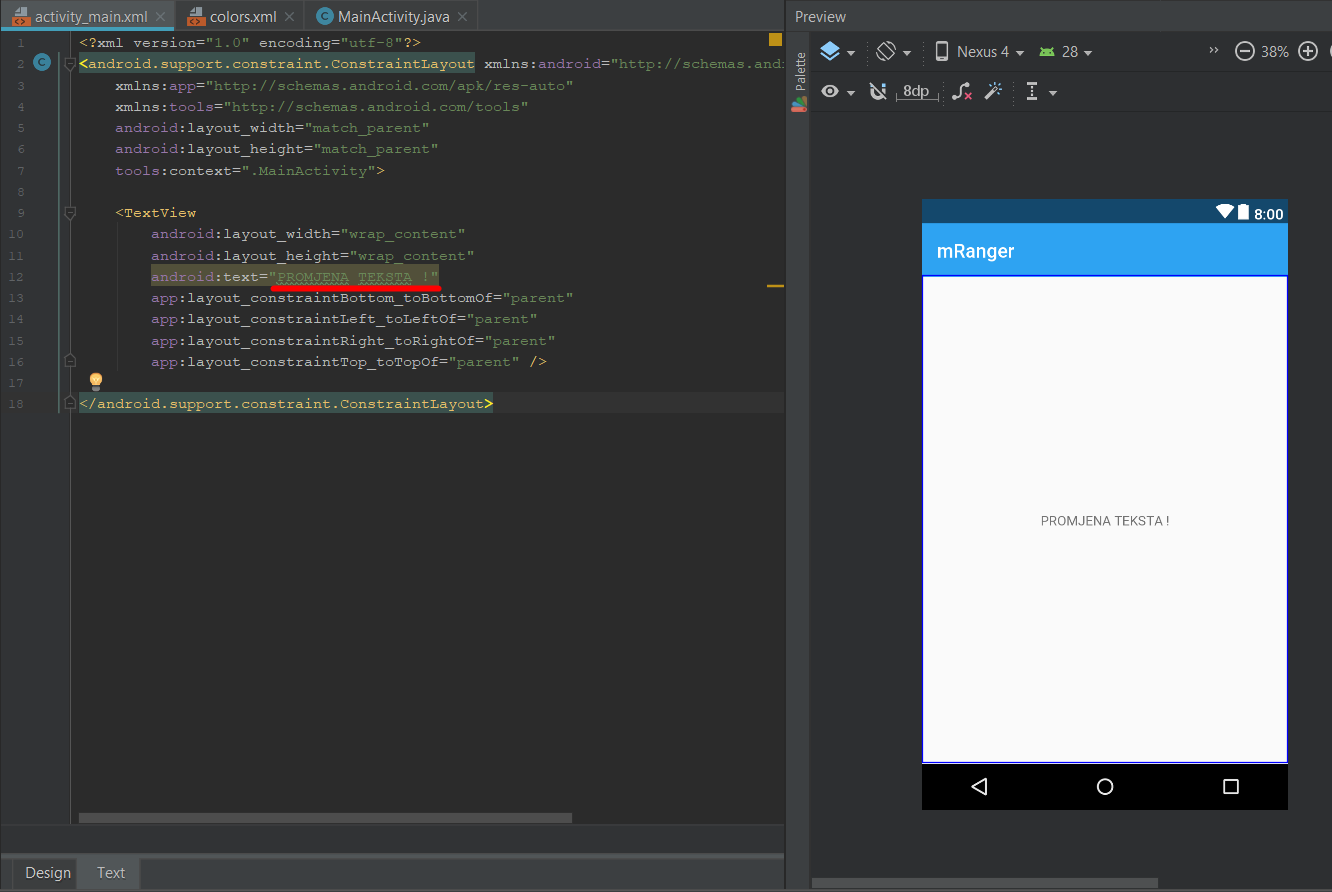
Nakon odabira željenih boja, možemo vidjeti kako se promijenio heksadecimalni kod u **colors.xml** dokumentu.

 Otvorimo li *activity\_main.xml,* možemo vidjeti prikaz našeg zaslona te uočiti da su se boje izmijenile.

Slika 12. Promjena boja zaslona

Također, idemo malo izmijeniti i ovaj tekst *Hello World!* Koji se sam od sebe stvorio u našoj aplikaciji prilikom kreiranja. Ako malo bolje promotrimo *activity\_main.xml* uočavamo da postoji redak gdje je taj tekst zapisan.





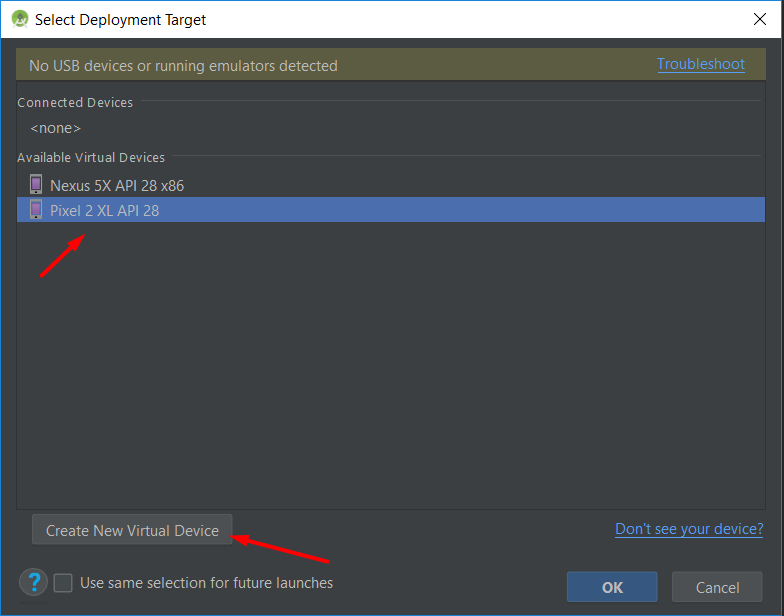
Slika 13. Tekst u XML-u

Slika 14. Promjena teksta

## 3.3 Pokretanje projekta

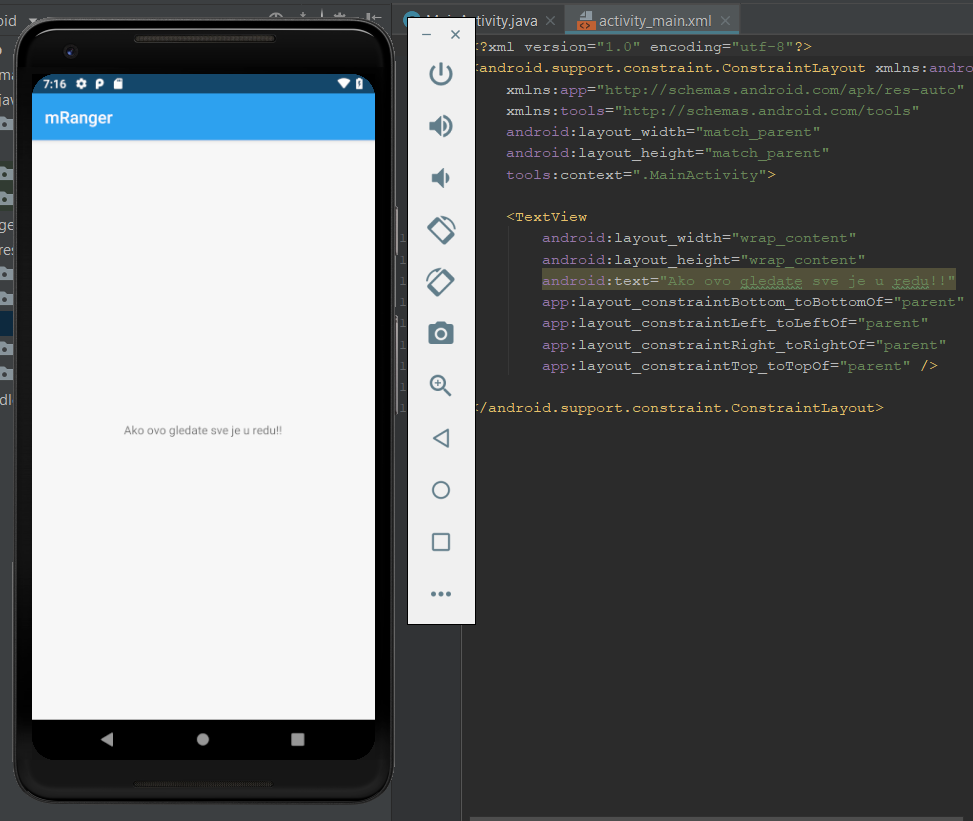
Slika 15. Gumb za pokretanje projekta

Kada poželimo vidjeti našu aplikaciju na djelu, u gornjem desnom dijelu alatne trake se nalazi gumb za pokretanje aplikacije.



Slika 16. Odabir virtualnog uređaja

Nakon toga prikazuje se zaslon za odabir virtualnog uređaja na kojem želimo pokrenuti aplikaciju. Možemo odabrati jedan od već ponuđenih ili pak dodati novi uređaj. Klik na *OK.*

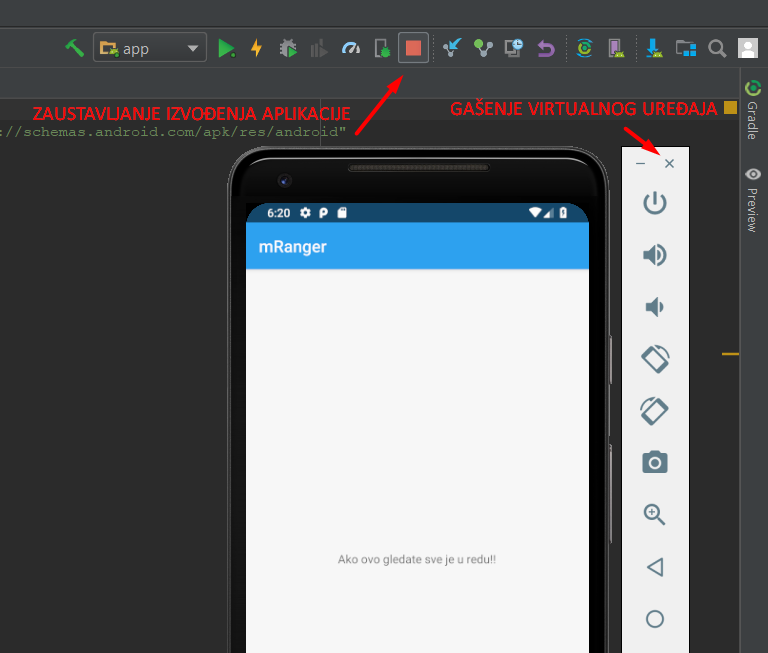
Prilikom pokretanja virtualnog uređaja automatski se pokrene i prikaže na zaslonu naša aplikacija.

Slika 17. Prikaz aplikacije na virtualnog uređaju uslijed pokretanja

Važno je napomenuti:

* Prilikom programiranja u Android Studio alatu, jednom kada pokrenemo virtualni uređaj nije potrebno gasiti, već samo minimiziramo dok radimo promjene u kodu.
* Ukoliko vaše računalo nema minimalno 8 GB RAM memorije, vrlo vjerojatno nećete moći koristiti značajku virtualnog uređaja, no tada imate opciju priključiti svoj pravi uređaj putem USB, te odabrati ga prilikom pokretanja kao na slici 16.

Prilikom zaustavljanja izvođenja aplikacije (pogledati sliku 18. ispod) ne gasimo virtualni uređaj, već zaustavljamo samo izvođenje aplikacije pritiskom na *Stop* znak koji se pojavi nakon pokretanja u gorenjem desnom dijelu alatne trake.



Slika 18. Prekid izvođenja aplikacije

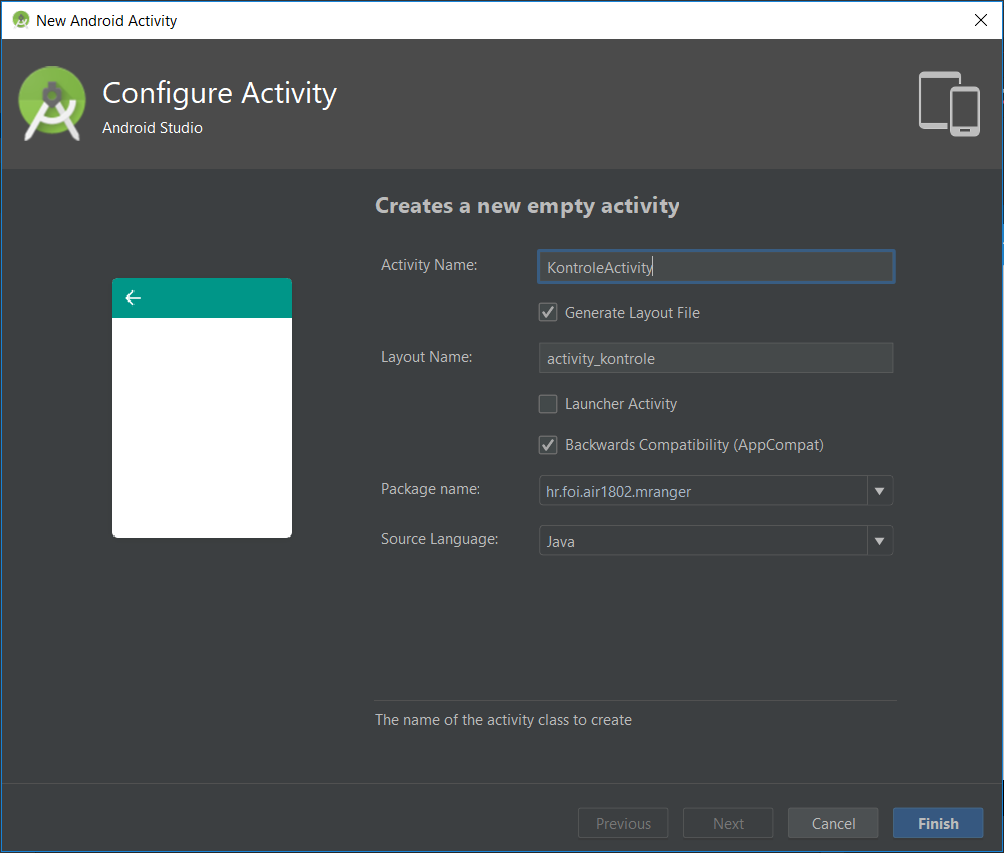
# Dodavanje novog zaslona, slike i gumba

U prethodnom poglavlju stvorili smo i uredili početni zaslon sa pripadajućim datotekama, *MainActivity.java* i *activity\_main.xml*. Pošto se najprije moramo spojiti Bluetooth-om na robota, taj početni zaslon će nam služiti za spajanje mobilnog uređaja na robot putem Bluetooth-a. Isto tako nakon spajanja, želimo imati zaslon putem kojeg ćemo upravljati robotom, tako da ćemo sada dodati novi zaslon u naš projekt.

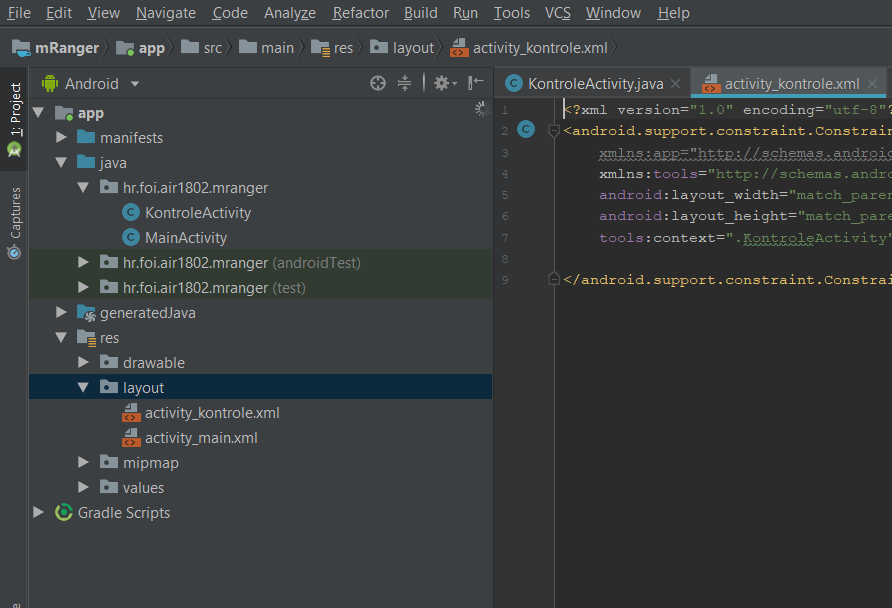
## 4.1 Dodavanje novog zaslona

Slika 19. Kreiranje novog zaslona (nove aktivnosti)

Slika 20. Imenovanje novog zaslona (nove aktivnosti)



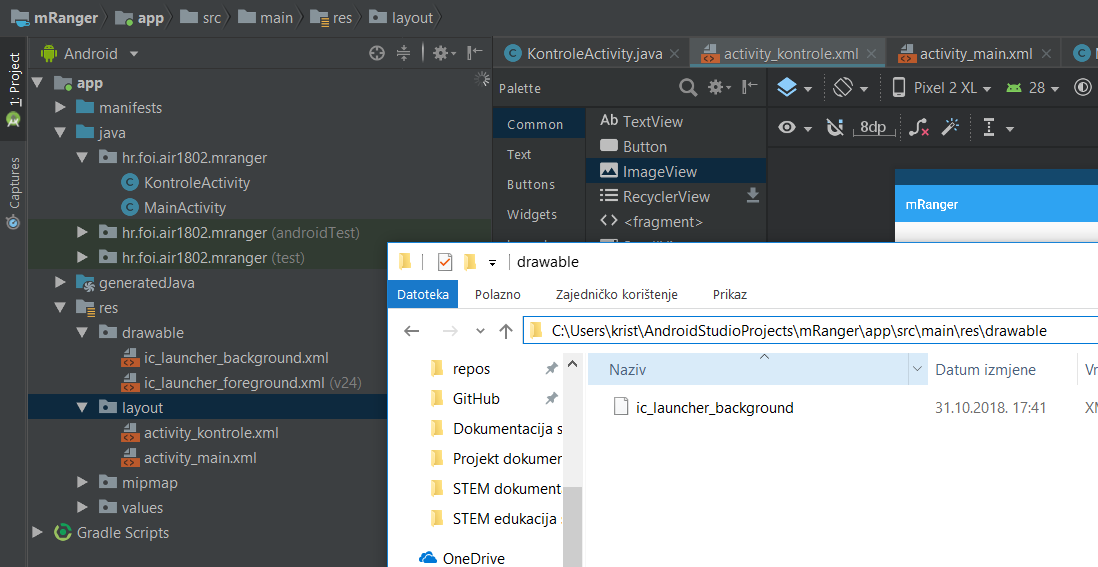
Novokreirani zaslon će sadržavati kontrole (joystick) za upravljanje mBot uređajem te ga stoga nazivamo *KontroleActivity*. Razlog takvome nazivu je nekakva praksa da programeri u androidu različite zaslone smatraju i nazivaju aktivnostima.

 Uz svaki zaslon dolaze dvoje datoteke, jedna za *layout* u kojoj se uređuje izgled. Druga je *Java klasa* u kojoj se nalazi pripadajući kod koji će se izvršavati za taj zaslon.

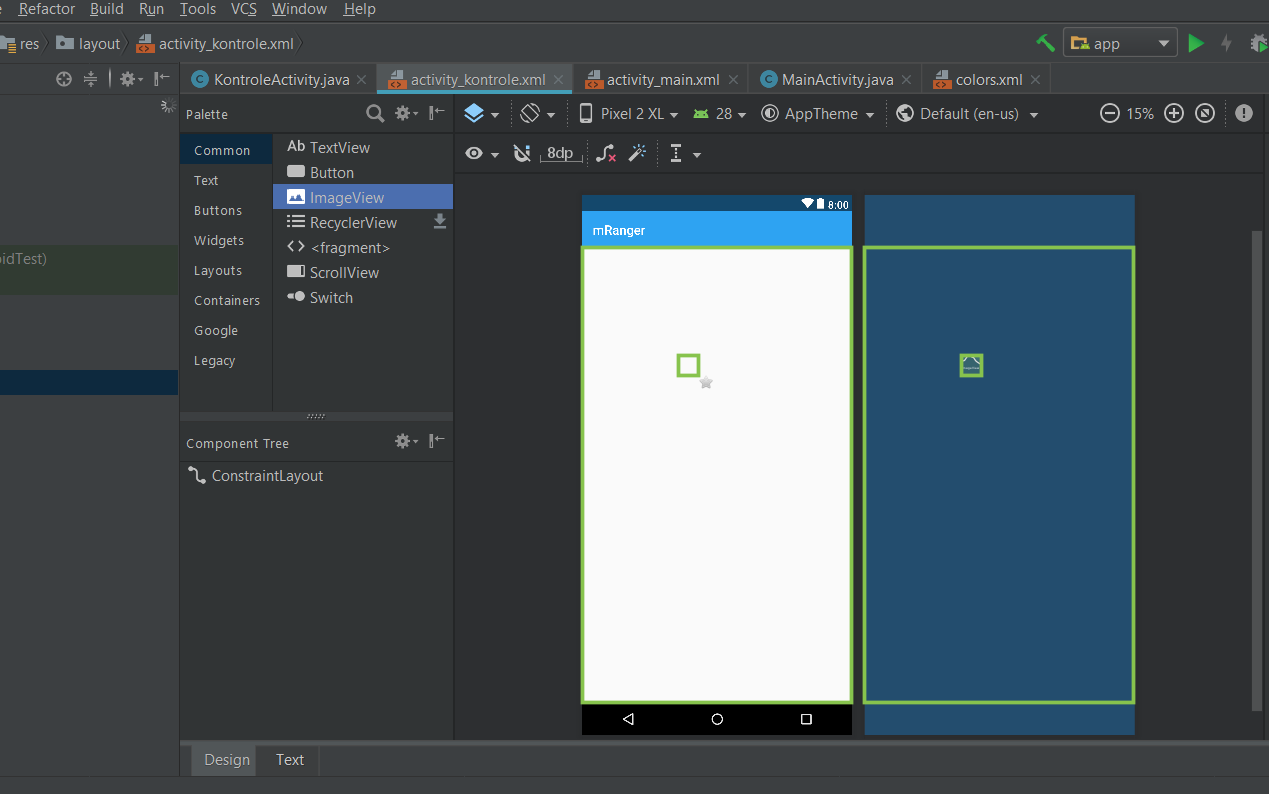
Slika 21. Stvoren novi zaslon

## 4.2 Dodavanje slike

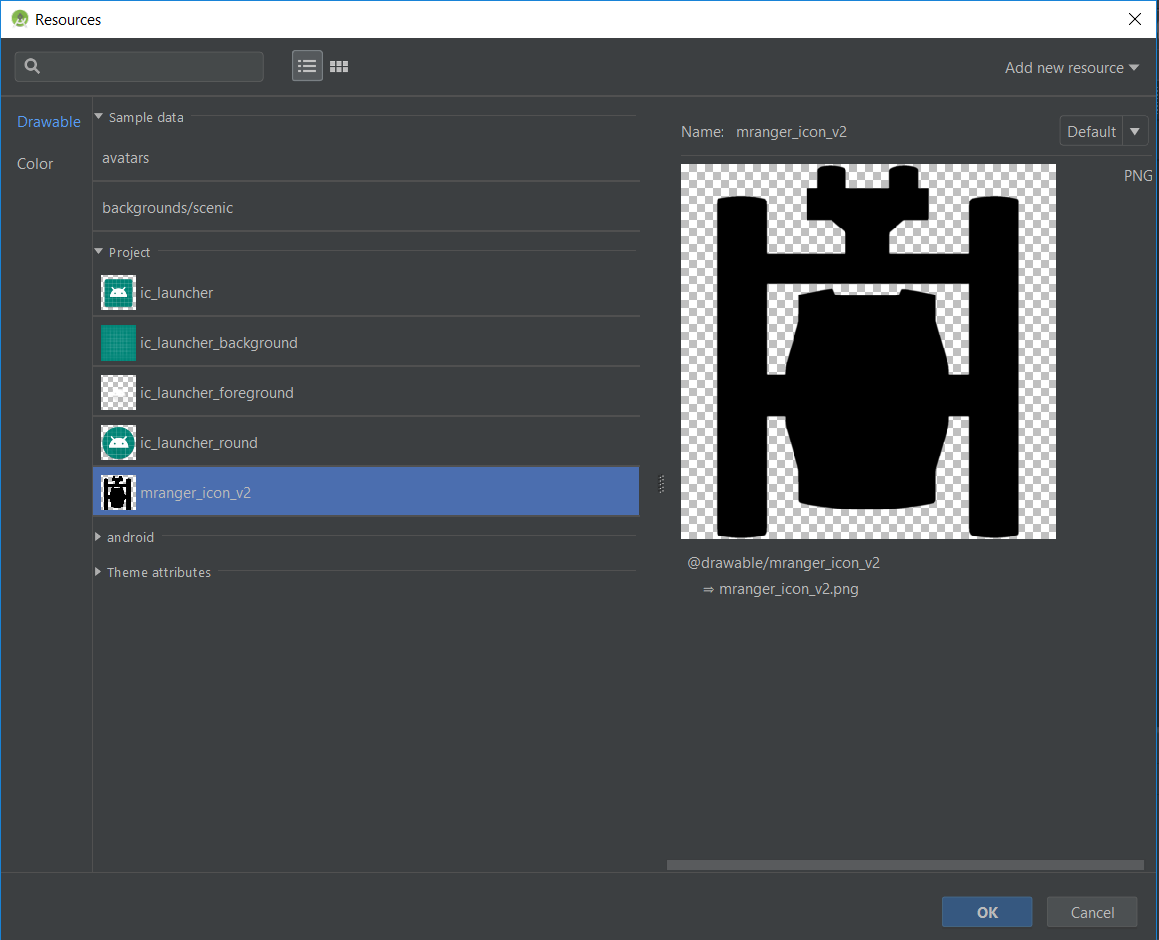
U sjedećem poglavlju pokazati ćemo kako se slika dodaje na zaslon, te koji su sve koraci potrebi za to. Kako bi razlikovali ovaj novi zaslon, na njega ćemo staviti sliku koju ćemo poslije iskoristiti. Kako bi slike dodavali na zaslon u Android Studio-u, te slike moraju biti u posebnom projektnoj mapi.

 Na slici se također nalazi putanja do spomenute mape (mapa *drawable*) u koju prije dodavanja slike je potrebno postaviti željenu sliku. Ime slike mora obavezno biti napisano malim slovima, te ako se radi o više riječi, one moraju biti odvojene donjom crtom („\_“).

Slika 22. Dodavanje slike u mapu projekta

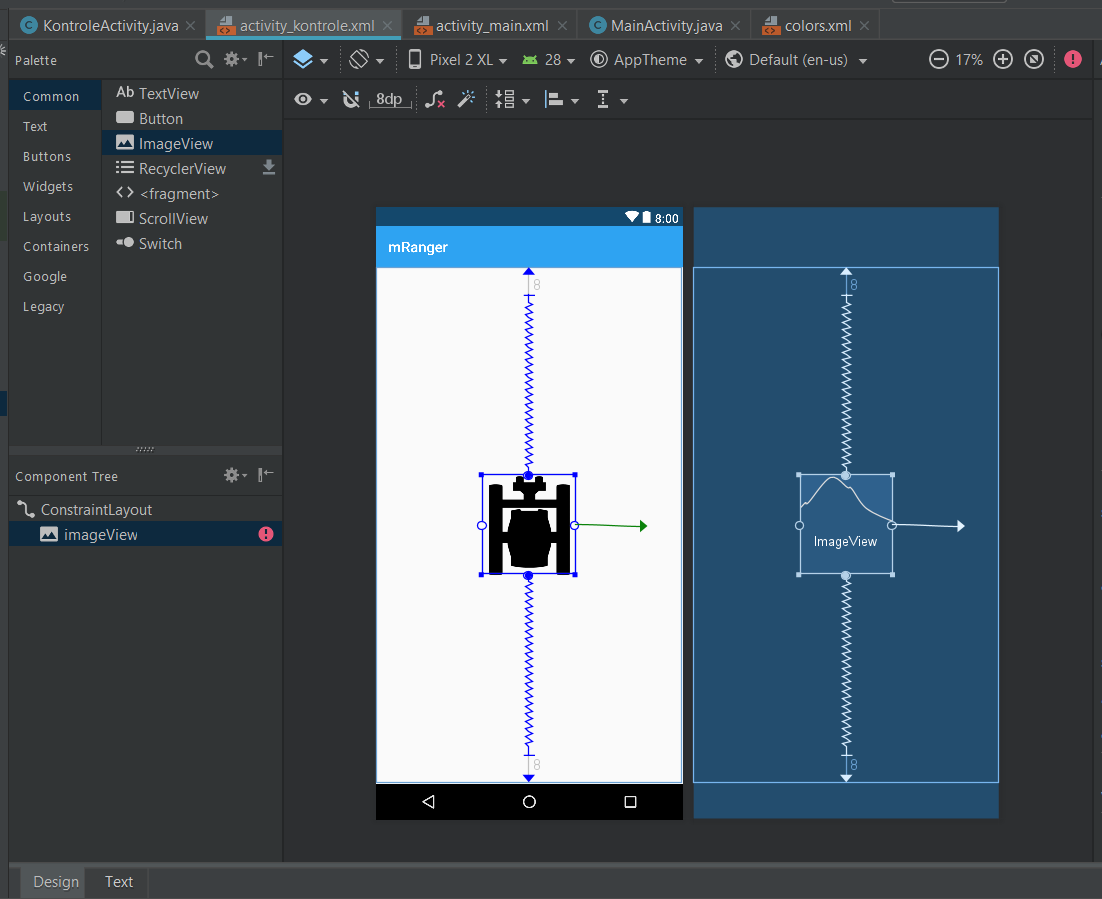


Slika 23. Dodavanje slike na zaslon

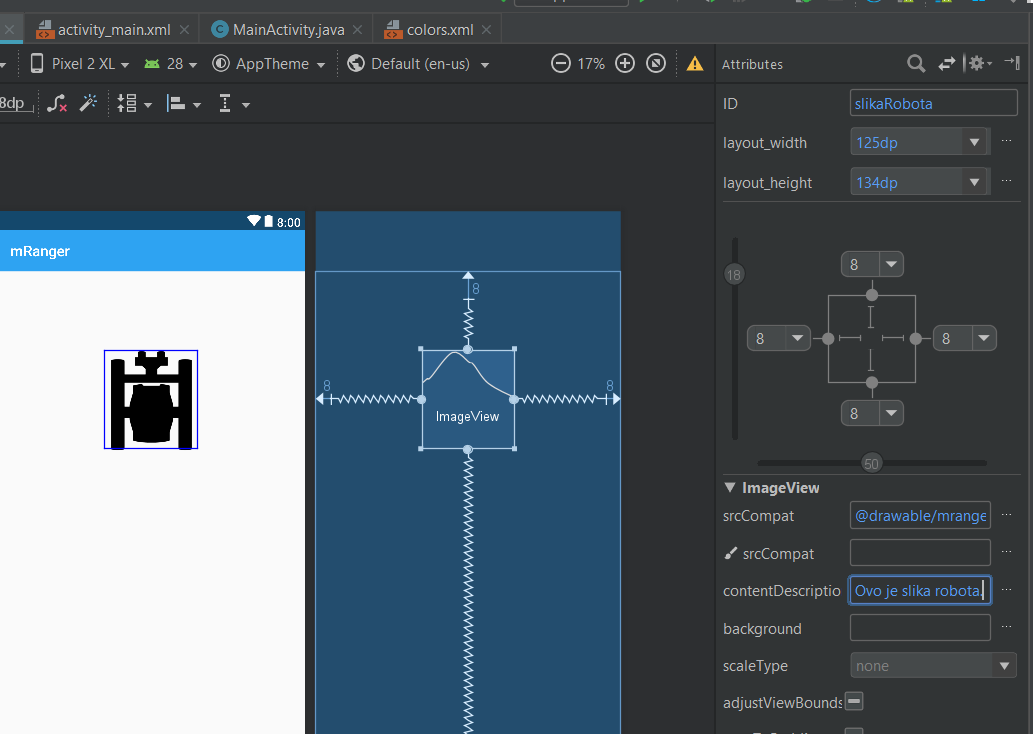
 Kako bi dodali sliku na zaslon, unutar *Design* taba dodajemo kontrolu *ImageView* koja predstavlja okvir unutar koje će se slika prikazivati. Kontrola se na zaslon dodaje po principu povuci i spusti (*eng. drag and drop*).

Slika 24. Odabir slike koja će se prikazat

Nakon ispuštanja kontrole, otvara nam se sljedeći prozor gdje biramo sliku koja će se prikazati u okviru. Tu ćemo na odabir imati neke predložene slike od Android Studio-a, kao i one koje smo dodali u *drawable* mapu.

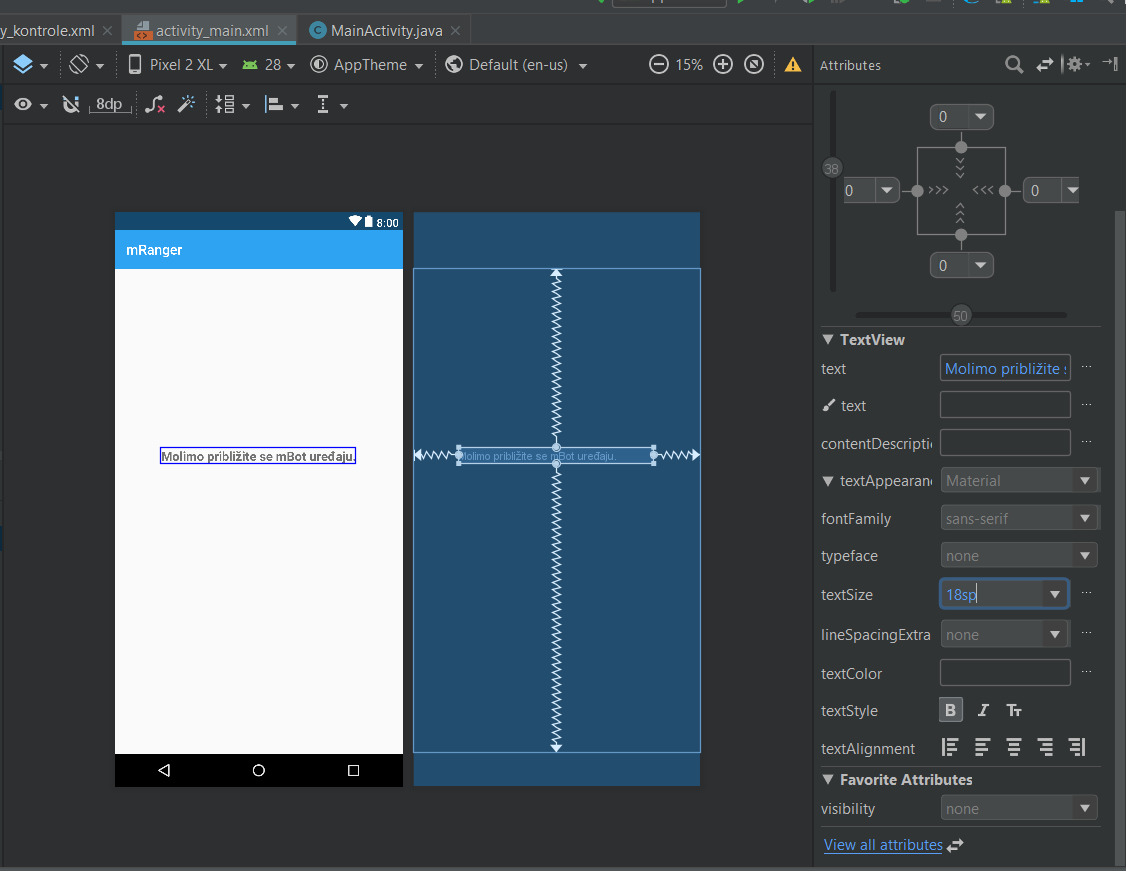
Kako bi se naša slika nalazila na željenom mjestu, a moramo uzeti u obzir da postoje zasloni različitih veličina, potrebno je sliku „usidriti“ sa svih njezinih strana okvira. Sliku usidravamo povlačenjem crte od točke sa sredine njezinog okvira pa sve do krajnjih rubova zaslona i/ili drugih kontrola.

Slika 25. Smještaj slike na zaslonu

 Novo dodanoj kontroli koja sadrži sliku dodajemo *id* kako bi je mogliprepoznati, te dajemo opis i ostale atribute po želji (proučite).

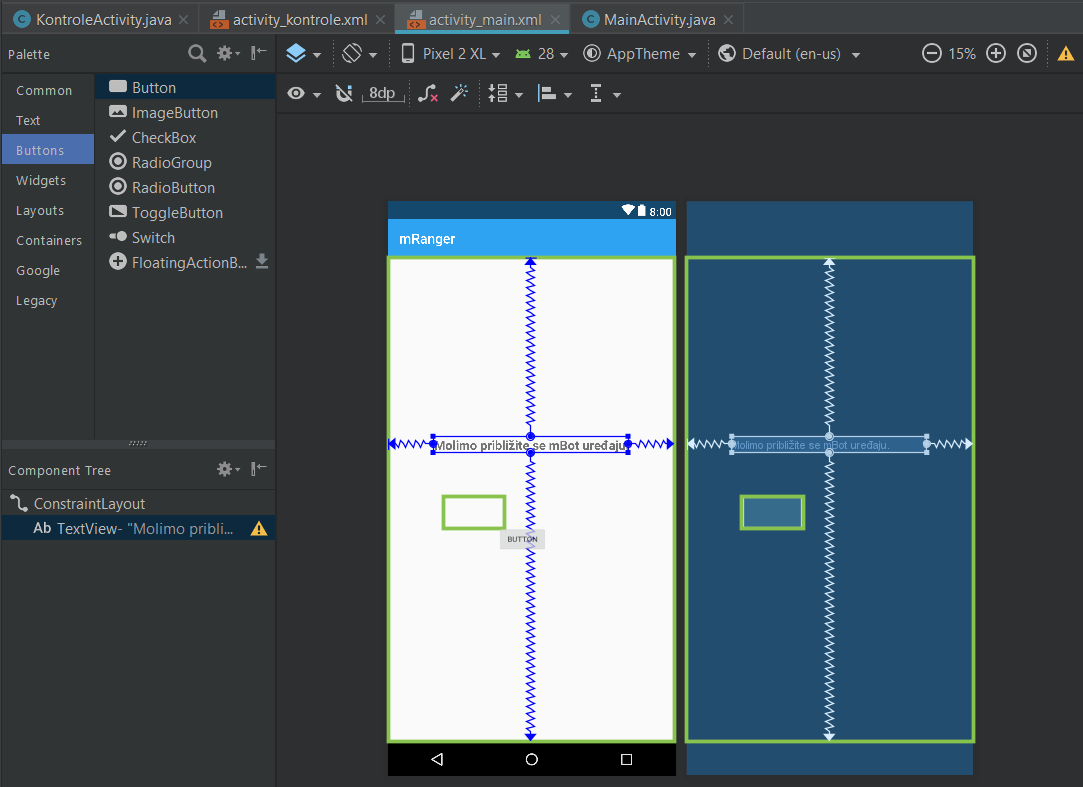
Slika 26. Dodavanje id-a i opisa slike

## 4.3 Dodavanje gumba



Slika 27. Sidrenje tekstualnog okvira

Iz istog razloga kao i kod slike sidrimo kontrolu tekstualnog okvira na željenu poziciju na zaslonu. Pošto je ovo naš početni zaslon, on se prvi otvara i na njemu se nalazi aktivnost povezivanja robota putem Bluetooth-a. Kako bi započeli proces povezivanja, dodajemo gumb pomoću kojeg ćemo to izvršiti.



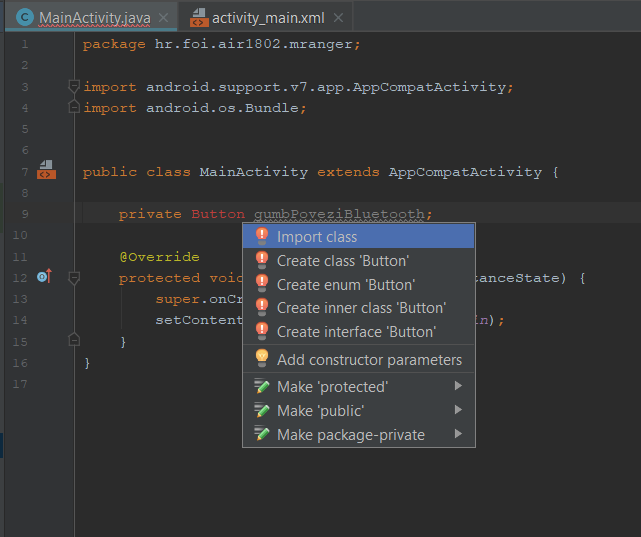
Slika 28. Dodavanje gumba

Istim principu kako smo dodali kontrolu za slikovni okvir, dodajemo kontrolu gumba na zaslon. Sidrimo je isto kao i kod kontrole slikovnog okvira. Također dodajemo joj *id* (taj naziv *id*-a će poslije biti jako važan u kodu), kao i tekst koji će se prikazivati na njemu, te ostale atribute po želji.

## 4.4 Prijelaz s jednog zaslona na drugi

Slika 29. Kontrola za gumb

Pozicioniramo se unutar *MainActivity.java* datoteke, te u njoj pišemo pripadajući kod za prijelaz na zaslon *KontroleActivity.java.*

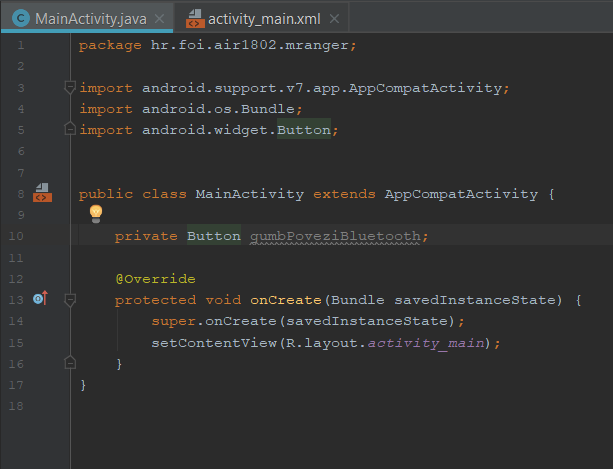


Slika 30. ALT + Enter kombinacija tipaka

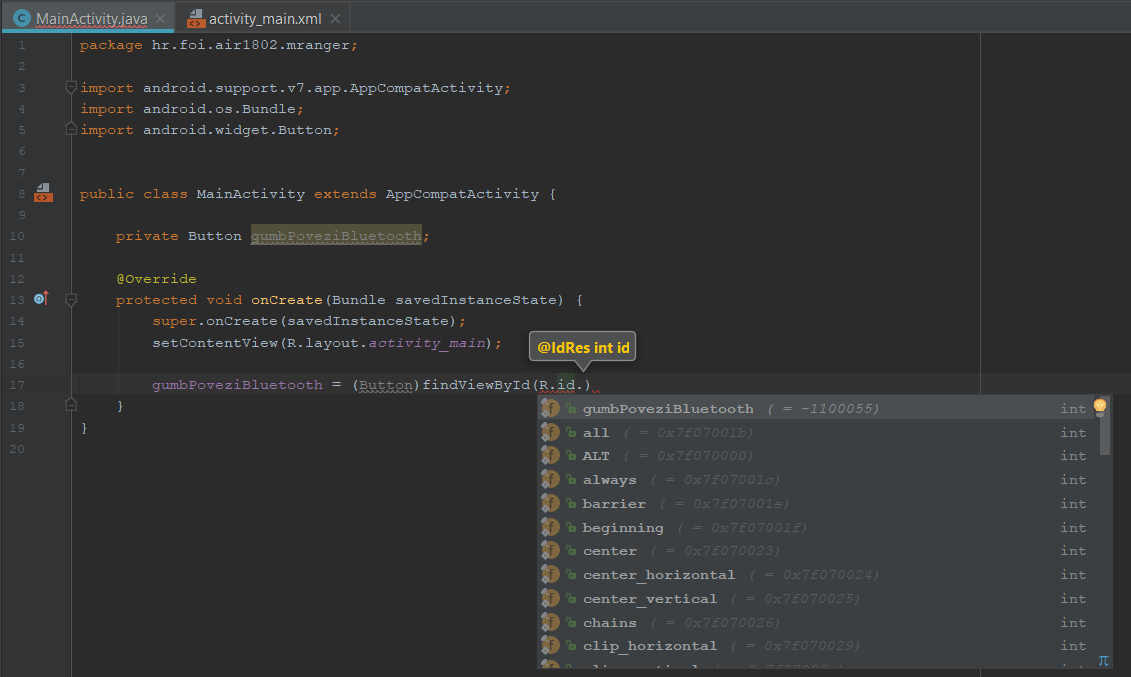
Napomena: ova kombinacija tipki je vrlo važna, ona omogućuje brzo izvršavanje unaprijed predviđenih opcija, te služi kao prva opcija koju ćemo poduzeti ukoliko dođe do nekih greški.

Na primjer, na slici poviše je napisana sljedeća linija koda gdje stvaramo novu privatnu varijablu tipa *Button* koju nazivamo po želji, ali zbog jednostavnosti isto onako kao što je i *id* našeg gumba.

Vidimo kako se pojavila greška kod tipa varijable jer nismo posebno uključili biblioteku koja se koristi za rad s tim tipom kontrole, te jednostavnim *Alt + Enter* dobivamo nekoliko mogućih opcija za rješavanje tog problema. U ovom slučaju je potrebno odabrati *Import Class* kako bi dodali biblioteku i otklonili pogrešku.

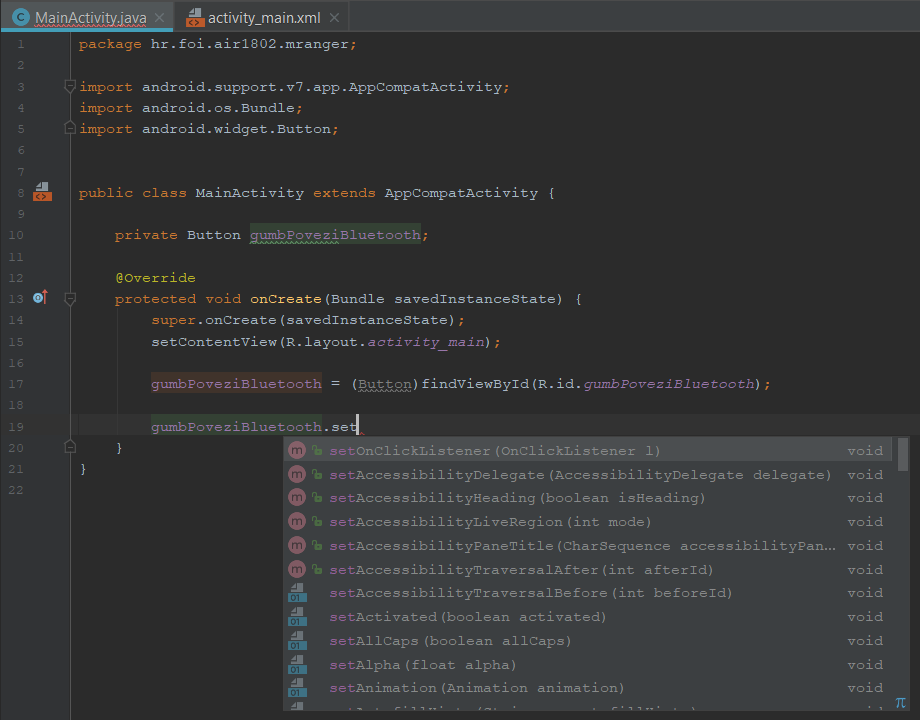


Slika 31. Dodana biblioteka

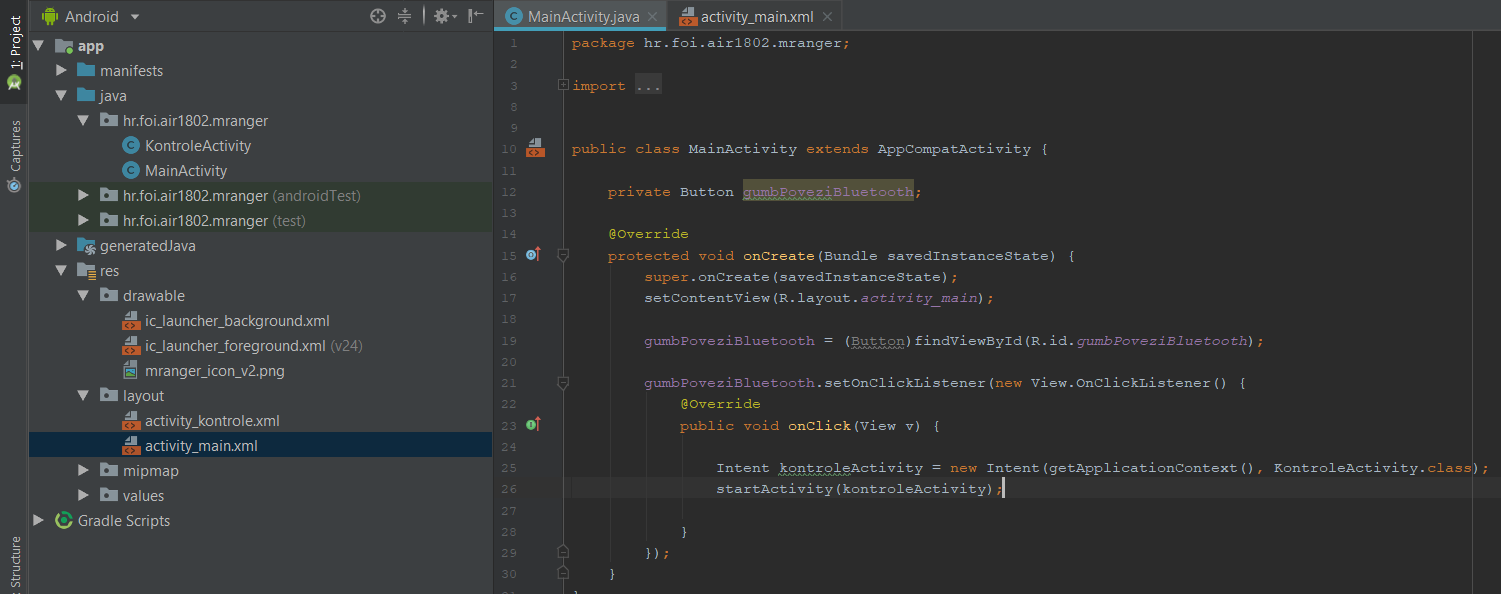


Slika 32. Dohvaćanje gumba preko id-a

Varijabli koju smo prethodno stvorili u kodu dajemo vrijednost, odnosno povezujem je sa kontrolom pripadajućeg gumba koji se nalazi na našem zaslonu. Primjetimo kako korištenje točke kod određenih elemenata nam također daje više mogučnosti što olakšava tipkanje koda kao i pronalaženje one prave.



Slika 33. Dodavanje setOnClickListener metode

Dodavanje *setOnClickListener* metode na gumb omogućujemo izvršavanje aktivnosti koje ćemo mi odrediti prilikom klika na određeni gumb na zaslonu.

Slika 34. Dodavanje koda koji se izvršava na klik gumba

Unutar onClick metode nalazi kod pomoću kojeg se prilikom pritiska na gumb prelazi sa početnog zaslona na željeni, što je u ovom slučaju *KontroleActivity.java.*