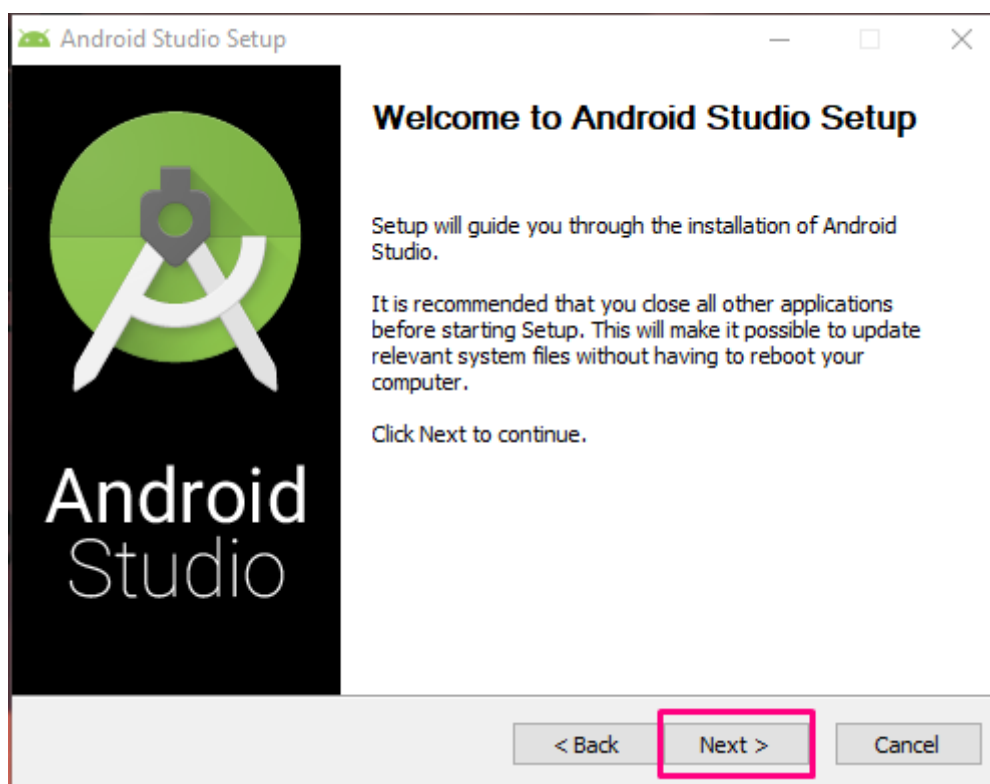


# 1 Uvod

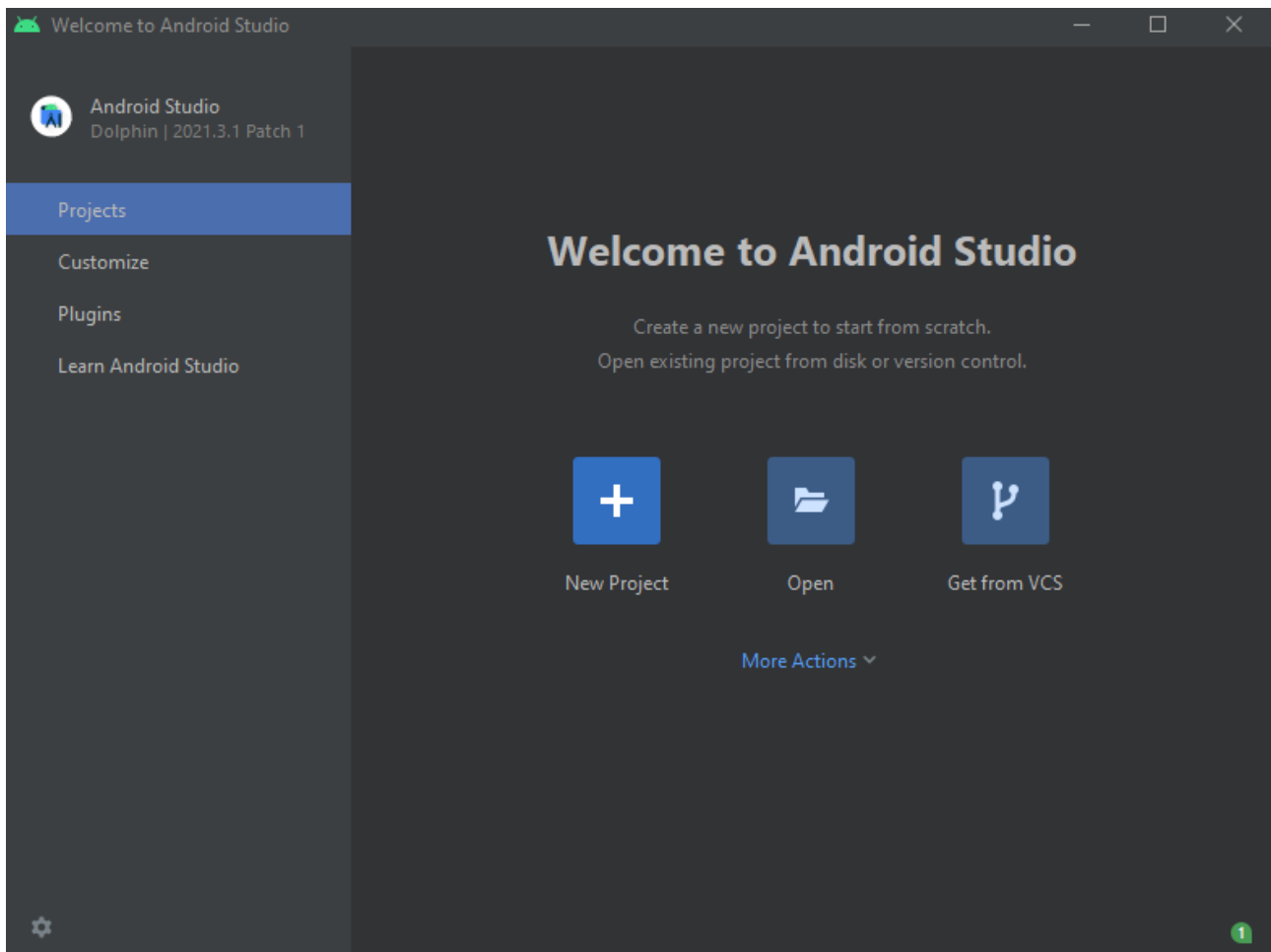
Razvoj mobilnih aplikacija jedna je od radionica u sklopu projekta Studentsko poduzetništvo - 404 koji se izvodi 2023. u suradnji Fakulteta elektrotehnike, računarstva i informacijskih tehnologija Osijek (FERIT), Fakulteta primijenjene matematike i informatike u Osijeku (MATHOS) te Ekonomskog fakulteta u Osijeku (EFOS). Cilj radionice jest pokazati učenicima osnovne koncepte razvoja mobilnih aplikacija s fokusom na platformu Android te ih potaknuti na samostalno učenje i istraživanje.

Ovaj minijturni priručnik prati aktivnosti radionice gdje će se kroz kreiranje nove Android aplikacije učenici upoznati s osnovnim pojmovima kao što su izgradnja korisničkog sučelja, povezivanje programskog koda i elemenata korisničkog sučelja, ali i sa složenijim elementima aplikacija poput uporabe senzora ili prikaza podataka dostupnih na Internetu. Aplikacija koja će biti izrađena sastoji se od tri zasebna ekrana, ekrana za inspiraciju koji će poslužiti za prolazak kroz osnove kreiranja korisničkog sučelja, ekrana za igru koji će poslužiti za upoznavanje sa senzorima te ekrana za šalu koji će poslužiti za dohvaćanje podataka s interneta i njihov prikaz. Svijet razvoja mobilnih aplikacija puno je širi od sadržaja koji pokriva radionica pa će u posljednjem poglavlju biti prikazane poveznice na vanjske resurse koje je moguće koristiti za samostalan rad i učenje. Za sve naknadne informacije i pitanja, slobodno se javite putem e-maila na [bruno.zoric@ferit.hr](mailto:bruno.zoric@ferit.hr).

## 2 Instalacija alata



Slika 1: Instalacija Android Studija - 1. korak



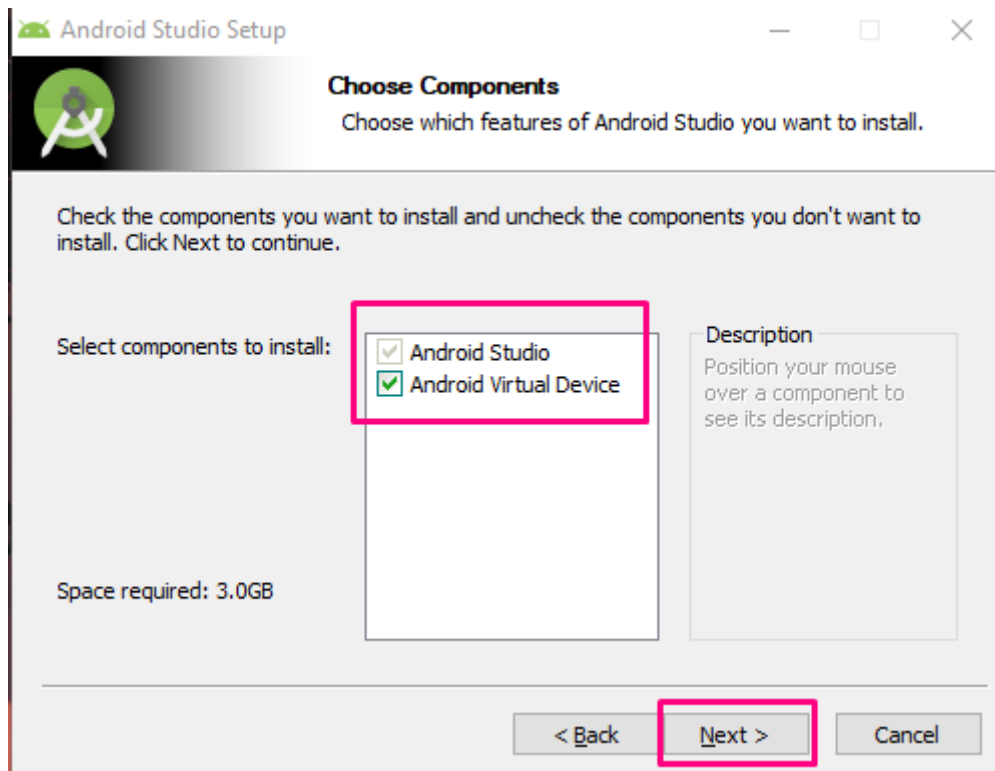
Slika 2: Instalacija Android Studija - 2. korak

### 3 Kreiranje projekta

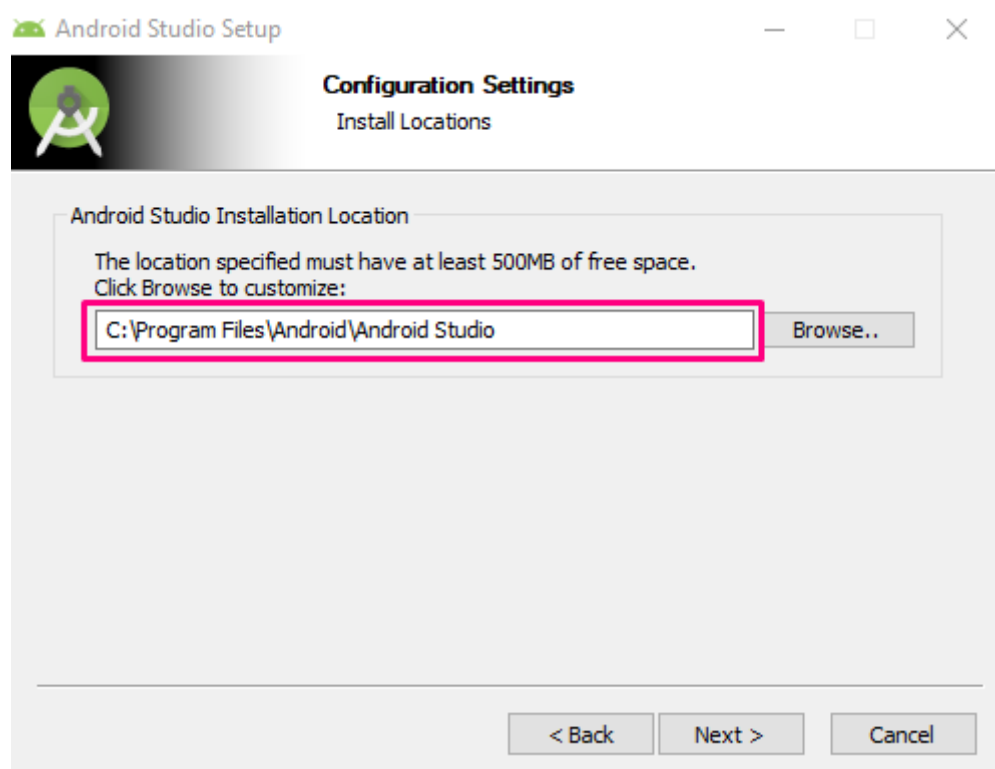
### 4 Uvođenje ekrana i navigacija

Android aplikacije u pravilu imaju ekrane koji služe za prikaz sadržaja korisniku. Za potrebe prikaza pojedinog ekrana moguće je koristiti različite gotove komponente koje se zovu **Activity** ili **Fragment**. Iako je ranije primaran način za oblikovanje i izgradnju korisničkih sučelja bio napisati XML kod koji ih opisuje i koji se vezuje uz pojedini **Activity** ili **Fragment**, danas to više nije tako. U novije vrijeme koristi se pristup gdje je svaka komponenta sučelja, uključujući i cijele ekrane, predstavljena funkcijom. Ovo je omogućeno korištenjem biblioteke imena **Jetpack Compose**, a taj pristup i navedena biblioteka bit će korišteni i u sklopu ove radionice. U ovom slučaju, mora postojati samo jedan **Activity** koji će služiti kao polazna točka i kontejner za sve ekrane aplikacije. Sama aplikacija imat će pet ekrana. Glavni ekran (**MainScreen**) sadržavat će komponente za navigaciju i određivati koji je početni ekran aplikacije. Početni ekran (**HomeScreen**) sadržavat će gumbe za navigaciju prema pojedinim aktivnostima aplikacije. Tri ekrana koja predstavljaju aktivnosti u aplikaciji bit će redom inspirirajući ekran (**InspireScreen**), zaigrani ekran (**PlayScreen**) te nasmijani ekran (**LaughScreen**).

Prvi korak koji ćemo napraviti je priprema projekta. Prilikom izrade aplikacije za bilo koju platformu programer se oslanja na kod i alate koje su razvili drugi programeri prije njega, a koji dolaze u obliku pripremljenih biblioteka. S obzirom da ih je razvio netko "izvana", često se nazivaju vanjskim ovisnostima (engl. *external dependency*). U suprotnom bi izrada bila osjetno dugotrajnija i zamornija.

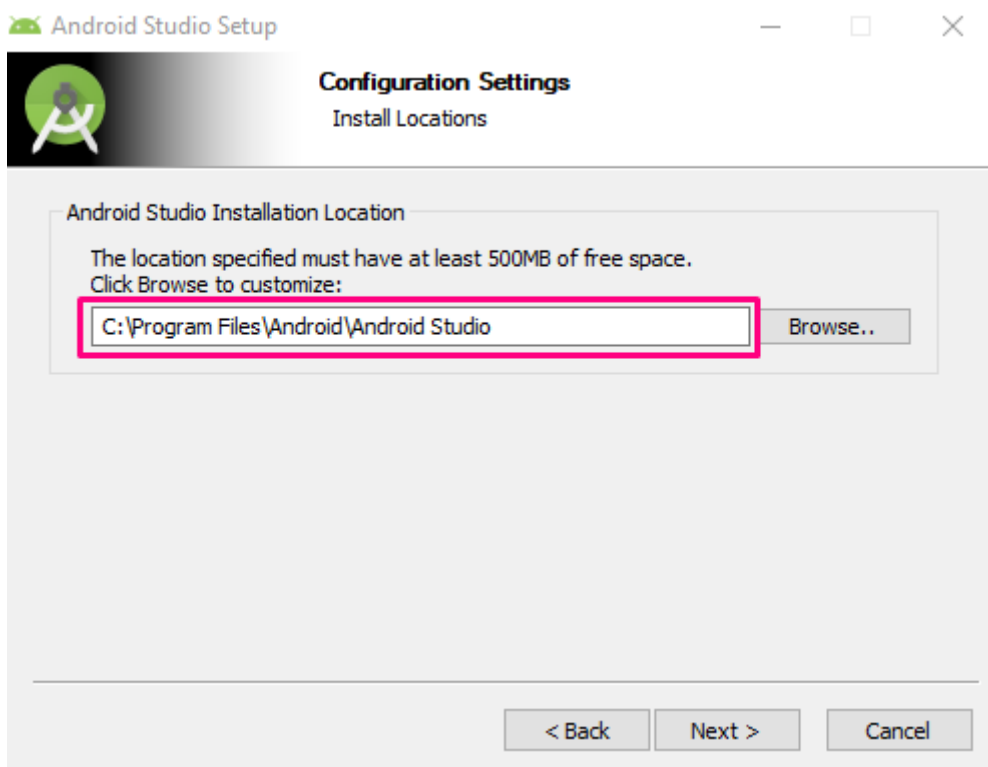


Slika 3: Instalacija Android Studija - 3. korak

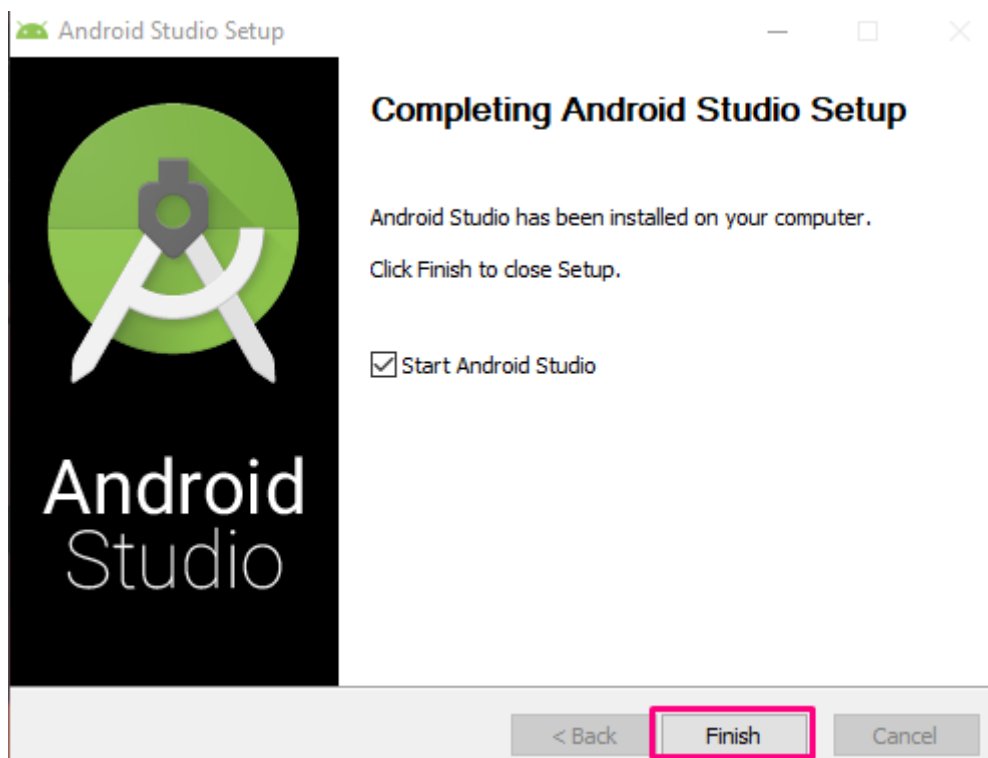


Slika 4: Instalacija Android Studija - 3. korak

Kod Android aplikacija, ove se vanjske ovisnosti uključuju u projekt dodavanjem nekoliko linija koda u datoteku `build.gradle` (Module: `app`) gdje se opisuje što točno želimo uključiti i koristiti u našem projektu. Sve potrebne vanjske ovisnosti za cjelokupnu radionicu dane su izlistanjem 1.



Slika 5: Instalacija Android Studija - 4. korak



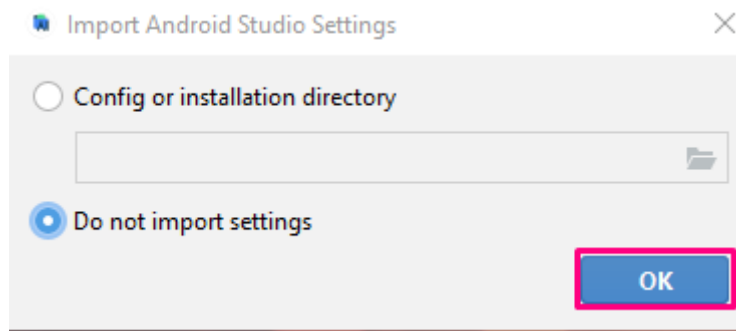
Slika 6: Instalacija Android Studija - 5. korak

Listing 1: Ovisnosnosti o vanjskim bibliotekama u datoteci build.gradle (Module:app)

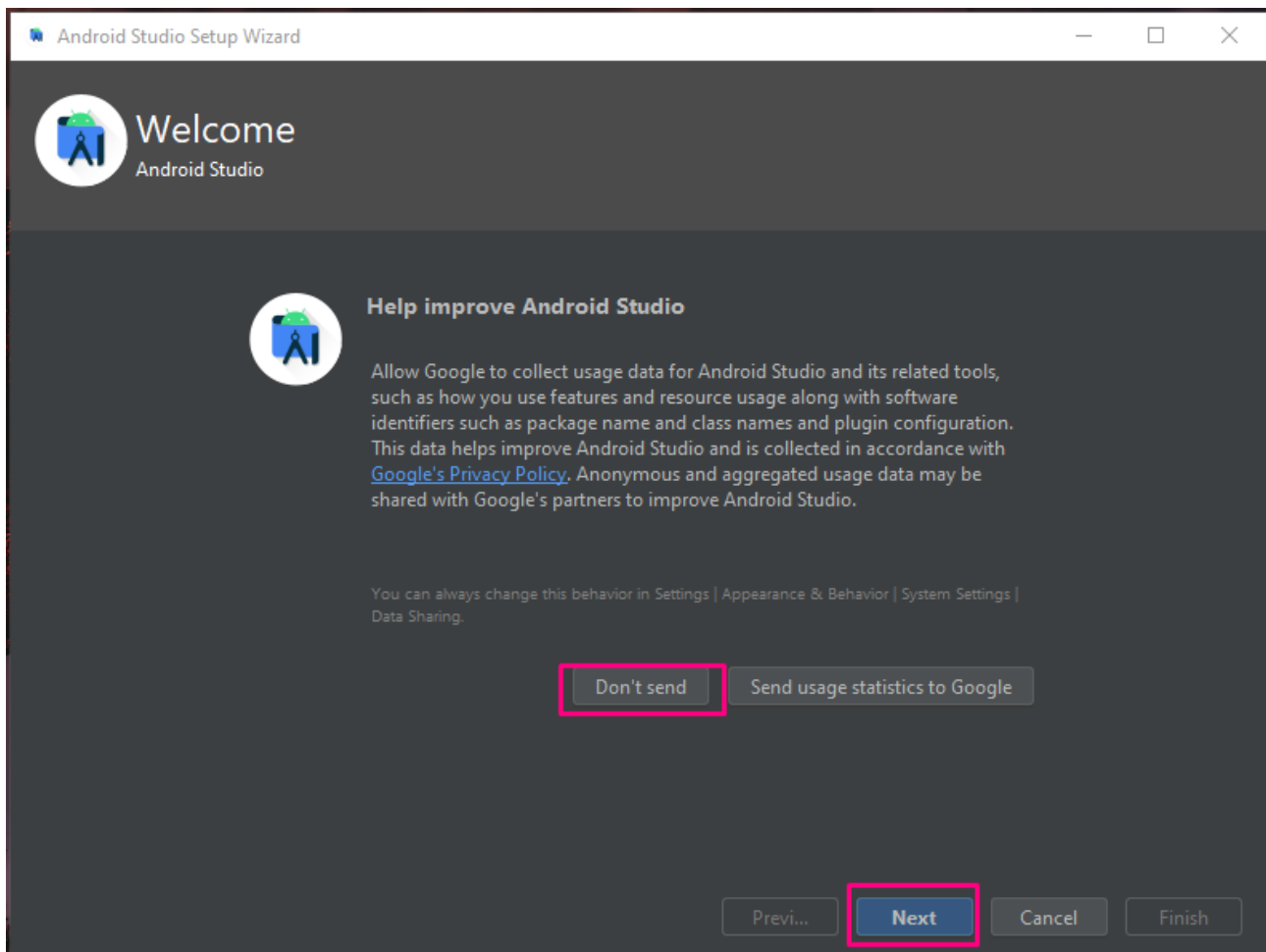
```

1 /*****
2 /*  Specific dependencies for the workshop:*/
3 /*****

```



Slika 7: Instalacija Android Studija - 6. korak

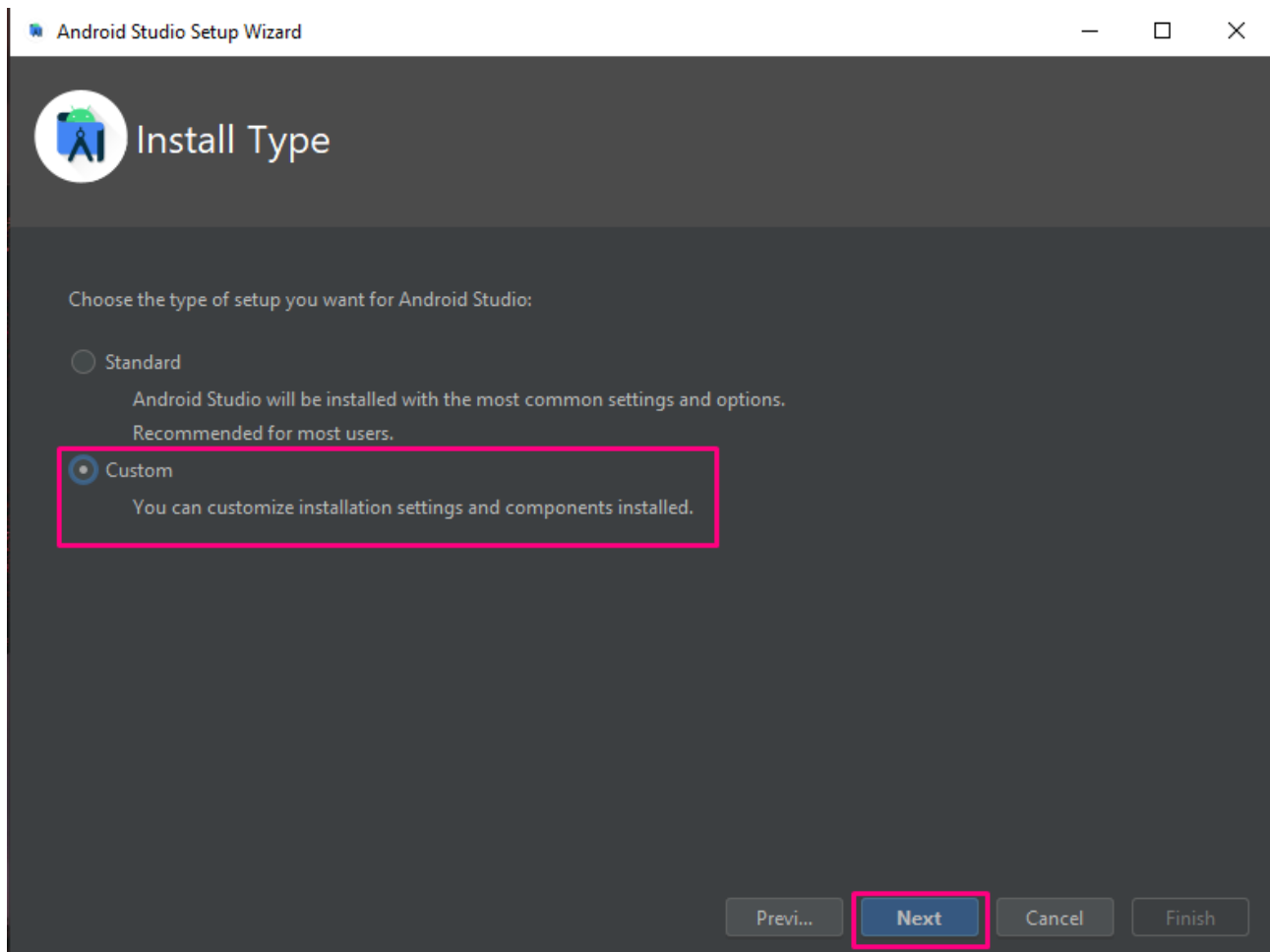


Slika 8: Instalacija Android Studija - 7. korak

```

4 // Navigation with jetpack compose:
5 def nav_version = "2.5.3"
6 implementation "androidx.navigation:navigation-compose:$nav_version"
7
8 // Ktor for networking:
9 def ktor_version = '2.2.2'
10 implementation "io.ktor:ktor-client-core:$ktor_version"
11 implementation("io.ktor:ktor-client-android:$ktor_version")
12 implementation "io.ktor:ktor-client-serialization:$ktor_version"

```



Slika 9: Instalacija Android Studija - 8. korak

```

13 implementation "io.ktor:ktor-serialization-gson:$ktor_version"
14 implementation("io.ktor:ktor-client-content-negotiation:$ktor_version")
15 implementation "io.ktor:ktor-client-logging-jvm:$ktor_version"

```

Kako bismo mogli pristupiti organizaciji ekrana naše aplikacije, najprije ćemo definirati sve ekrane i njihove jedinstvene oznake. Ove jedinstvene oznake nazivaju se još i rutama. Ruta predstavlja jedinstvenu oznaku svakog ekrana i omogućuje komponenti za navigaciju da točno zna kamo treba odvesti korisnika. Ove rute definirat ćemo u posebnoj datoteci koju ćemo nazvati `Navigation.kt`, a koja je prikazana izlistanjem 2.

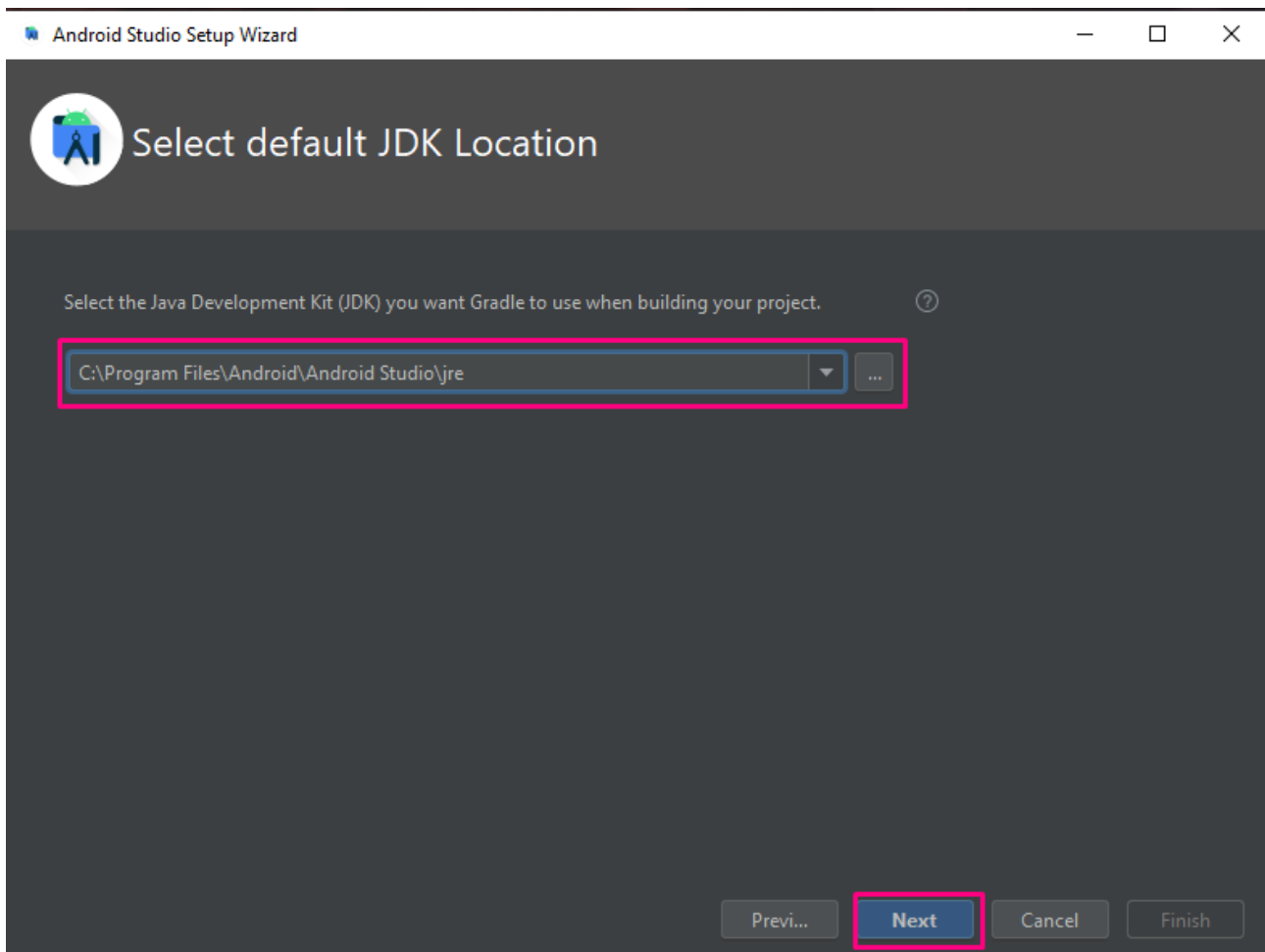
Listing 2: Navigacija - `Navigation.kt`

```

1 sealed class Screen(val route: String) {
2     object Home : Screen("home")
3     object Inspire : Screen("inspire")
4     object Laugh : Screen("laugh")
5     object Play : Screen("throw")
6 }

```

Svaka funkcija koja predstavlja dio korisničkog sučelja označena je oznakom `@Composable`. S obzirom da ćemo za početak samo kreirati prazne ekrane za pojedine aktivnosti koje ćemo nadograđivati i popunjavati sadržajem kako radionica bude odmicala, kreirajmo redom sva tri ekrana za aktivnosti,



Slika 10: Instalacija Android Studija - 9. korak

prema izlistanjima 3, 4, 5. Svaki od ovih ekrana kreira se u zasebnoj datoteci čije ime odgovara imenu ekrana, dakle `InspireScreen`, `PlayScreen`, `LaughScreen`.

Listing 3: Inspirirajući ekran - `InspireScreen.kt`

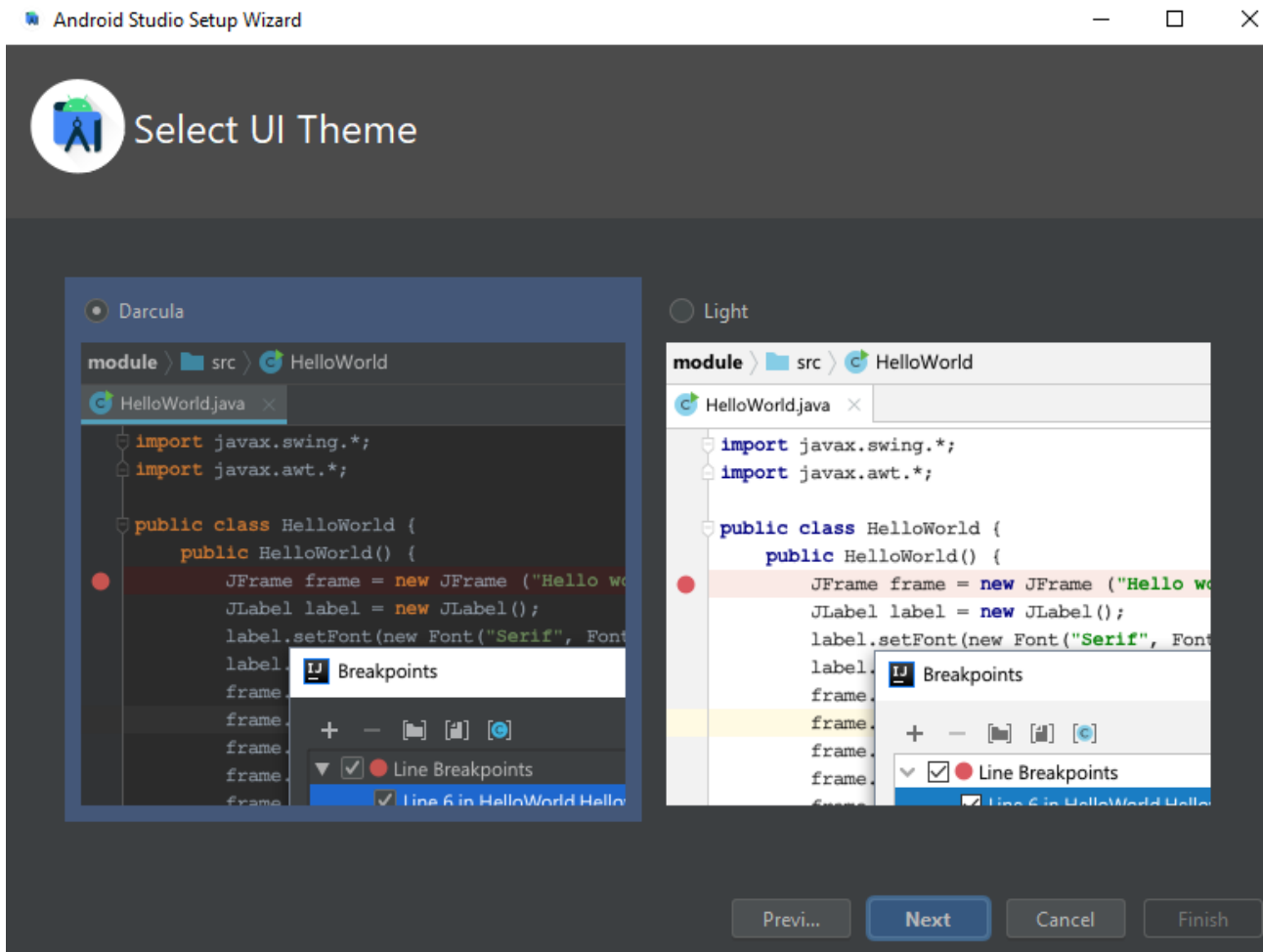
```
1 @Composable
2 fun InspireScreen() {
3     Text(text = "Inspire□screen")
4 }
```

Listing 4: Zaigrani ekran - `PlayScreen.kt`

```
1 @Composable
2 fun PlayScreen() {
3     Text(text = "Play□screen")
4 }
```

Listing 5: Nasmijani ekran - `LaughScreen.kt`

```
1 @Composable
2 fun LaughScreen() {
3     Text(text = "Laugh□screen")
4 }
```



Slika 11: Instalacija Android Studija - 10. korak

Idući korak je kreiranje početnog ekrana. Riječ je o ekranu koji će sadržavati gumbe koji će korisniku omogućiti odlazak na neki od ranije pripremljenih ekrana u aplikaciji. Izgled početnog ekrana dan je izlistanjem ???. Ovaj je ekran nešto kompleksniji od dosadašnjih. Vidimo da, kako bi taj ekran radio, traži od nas da mu damo `NavController`. Riječ je o komponenti za navigaciju koju ćemo pripremiti kasnije i predati ju početnom ekranu svaki put kada ga zatrebamo. Osim toga, sučelje ovog ekrana sadrži više sadržaja no prethodni ekrani. Za kreiranje sučelja koriste se druge `Composable` funkcije koje pozivamo unutar naše funkcije `HomeScreen()`. Najprije, želimo kreirati kontejner za sav sadržaj i postaviti mu pozadinsku sliku. Ovo postizemo funkcijom `Box()` i postavljanjem slike kao prvog elementa u nju (poziv funkciji `Image()`). Uz sliku, funkcija `Box()` sadržavat će i stupac s gumbima, za što se koristi funkcija `Column()`. Unutar potonje iskoristit će se tri poziva funkciji `NavigateButton()` koju smo sami definirali u datoteci `NavigateButton.kt`. Naime, čest je slučaj da se pojedine komponente koriste na više mjesta. Kako bi se izbjeglo ponavljanje koda na svim tim mjestima (npr. postavljanje boje, veličine teksta i slično), moguće je definirati komponentu zasebno u obliku funkcije sa željenim vrijednostima ovih postavku i onda ju pozvati gdje god je ona potrebna. Navigacijski gumb definiran je u datoteci `NavigateButton.kt`, a izgled je dan izlistanjem koda 7.

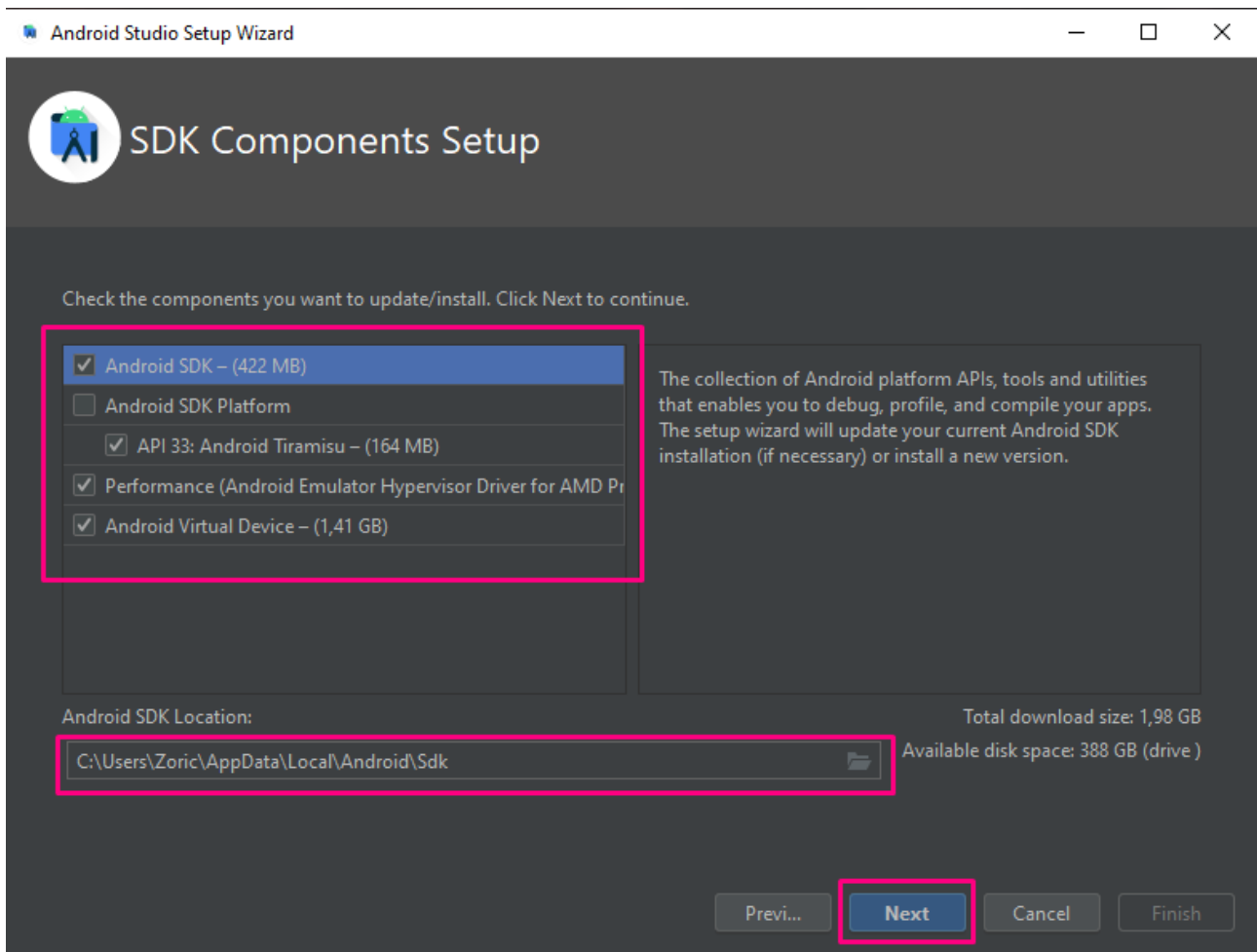
Listing 6: Početni ekran - `HomeScreen.kt`

```

1 @Composable
2 fun HomeScreen(navController: NavController) {
3     Box(modifier = Modifier.fillMaxSize()) {

```



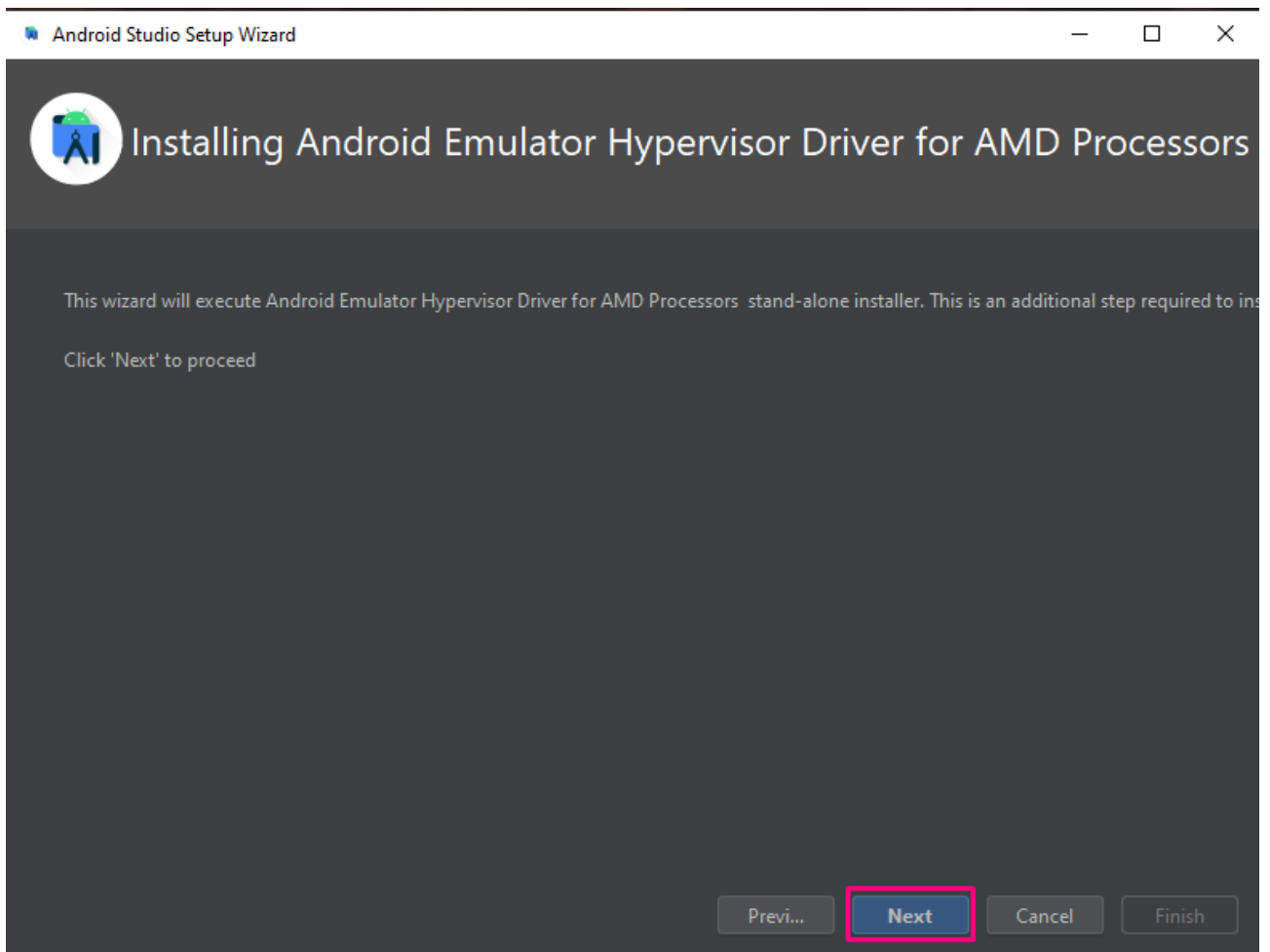


Slika 12: Instalacija Android Studija - 11. korak

```

4 Image(
5     painter = painterResource(id = R.drawable.background),
6     contentDescription = "Background┐gradient┐image",
7     modifier = Modifier.matchParentSize(),
8     contentScale = ContentScale.Crop
9 )
10 Column(
11     modifier = Modifier.fillMaxSize(),
12     verticalArrangement = Arrangement.Center,
13     horizontalAlignment = Alignment.CenterHorizontally
14 ) {
15     NavigateButton(
16         text = stringResource(id = R.string.label_inspire),
17         onClick = { navController.navigate(Screen.Inspire.route) }
18     )
19     NavigateButton(
20         text = stringResource(id = R.string.label_laugh),
21         onClick = { navController.navigate(Screen.Laugh.route) }
22     )

```

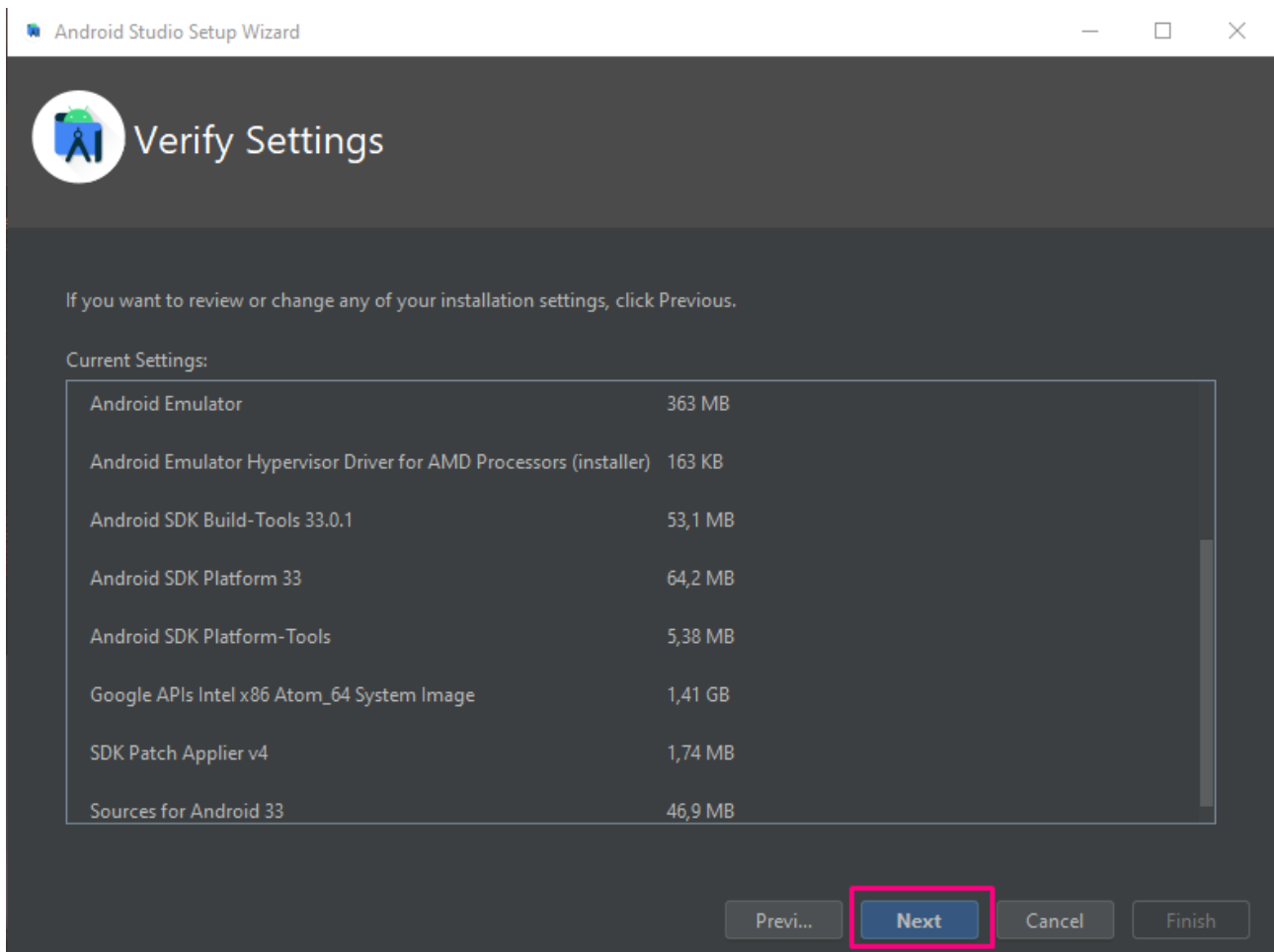


Slika 13: Instalacija Android Studija - 12. korak

```
23     )
24     NavigateButton(
25         text = stringResource(id = R.string.label_play),
26         onClick = { navController.navigate(Screen.Play.route) }
27     )
28 }
29 }
30 }
```

Listing 7: Gumb za navigaciju - NavigateButton.kt

```
1 @Composable
2 fun NavigateButton(text: String, onClick: () -> Unit) {
3     Button(
4         onClick = onClick,
5         Modifier.padding(dimensionResource(id = R.dimen.padding_medium)),
6         colors = ButtonDefaults.buttonColors(
7             backgroundColor = colorResource(id = R.color.rosewater),
8             contentColor = colorResource(id = R.color.cream)
9         )
10 }
```



Slika 14: Instalacija Android Studija - 13. korak

```

10 ) {
11     Text(text = text , fontSize = 20.sp)
12 }
13 }

```

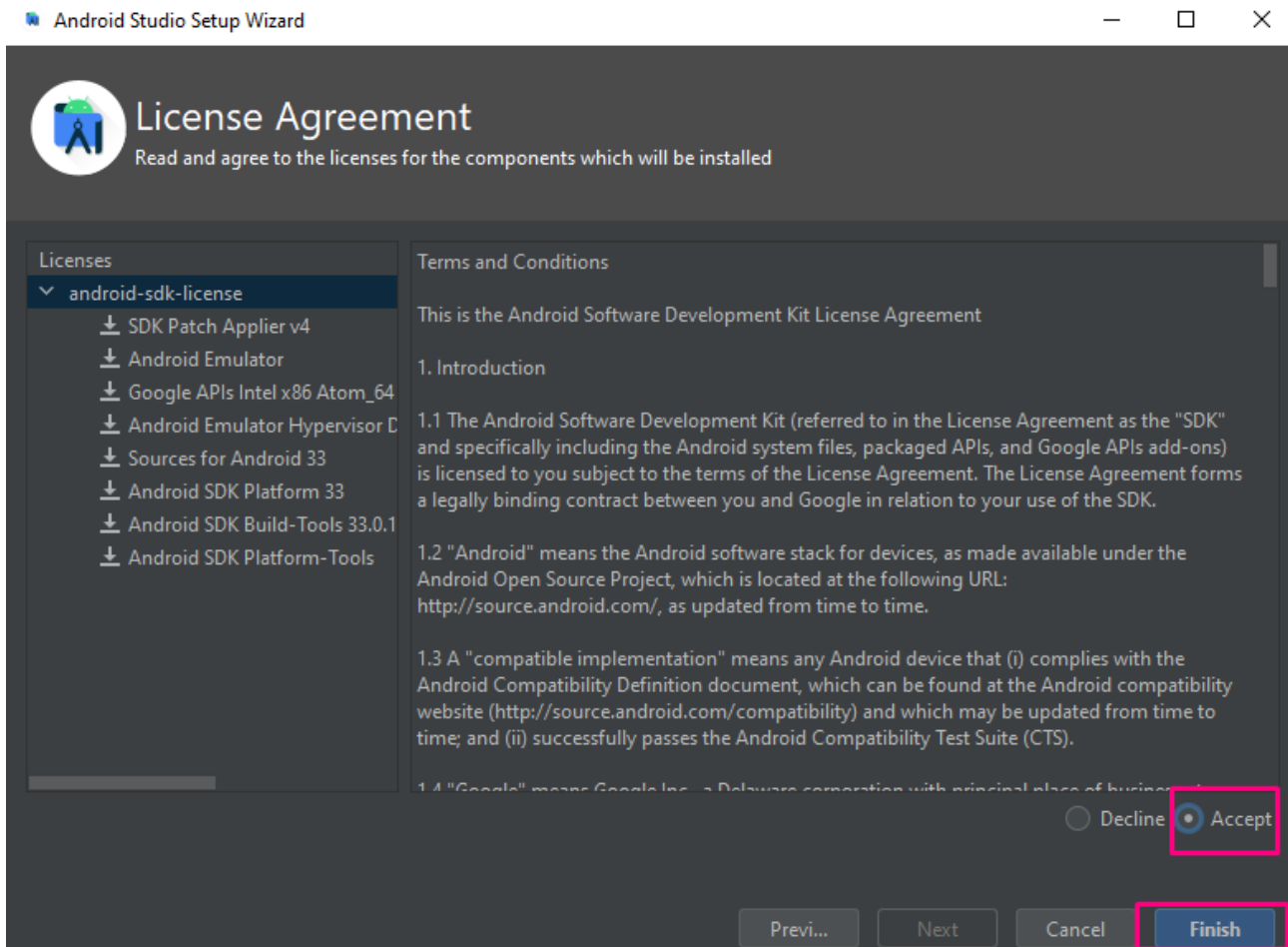
Kada je sve ostalo pripremljeno, potrebno je sve i povezati u cjelinu i to kreiranjem glavnog ekrana. Za kreiranje glavnog ekrana rabi se stoga funkcija dana izlistanjem 8. U toj funkciji kreirana je komponenta za navigaciju kroz aplikaciju te je navedeno koji ekran je početni, a koji su ekrani na koje je moguće navigirati. Možete primijetiti da se ovdje kreira i postavlja navigacijska komponenta koja je potrebna početnom ekranu aplikacije.

Listing 8: Glavni ekran - mainScreen.kt

```

1 @Composable
2 fun MainScreen() {
3     val navController = rememberNavController()
4     NavHost(navController = navController , startDestination = Screen.Home.route) {
5         composable(Screen.Home.route) { HomeScreen(navController) }
6         composable(Screen.Laugh.route) { LaughScreen() }
7         composable(Screen.Inspire.route) { InspireScreen() }
8         composable(Screen.Play.route) { PlayScreen() }
9     }

```



Slika 15: Instalacija Android Studija - 14. korak

10 }

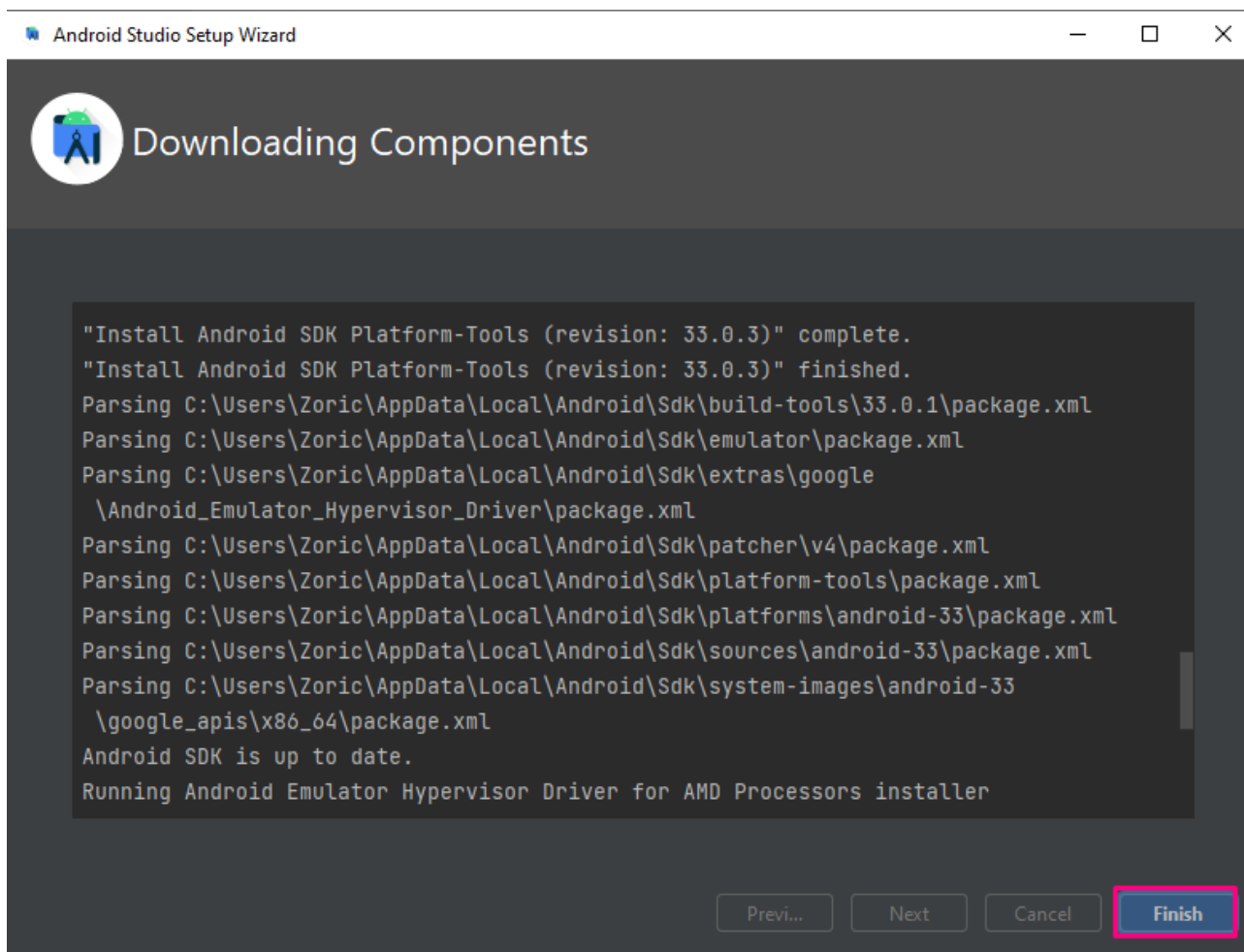
Preostalo je još samo reći Android sustavu da, kada se pokrene aplikacija, prikaže najprije glavni ekran (ovaj nema sadržaja) koji će onda odmah navigirati ka početnom ekranu. Ovo se postiže izmjenom linije koda u datoteci `MainActivity.kt` prema izlistanju ?? tako da se poziva ranije definirana funkcija `MainScreen()`.

Listing 9: Glavna aktivnost - `MainActivity.kt`

```

1 class MainActivity : ComponentActivity() {
2     override fun onCreate(savedInstanceState: Bundle?) {
3         super.onCreate(savedInstanceState)
4
5         setContent {
6             LaughlyTheme {
7                 // A surface container using the 'background' color from the theme
8                 Surface(
9                     modifier = Modifier.fillMaxSize(),
10                    color = MaterialTheme.colors.background
11                ) {
12                    MainScreen()
13                }
14            }
15        }
16    }
17 }

```



Slika 16: Instalacija Android Studija - 15. korak

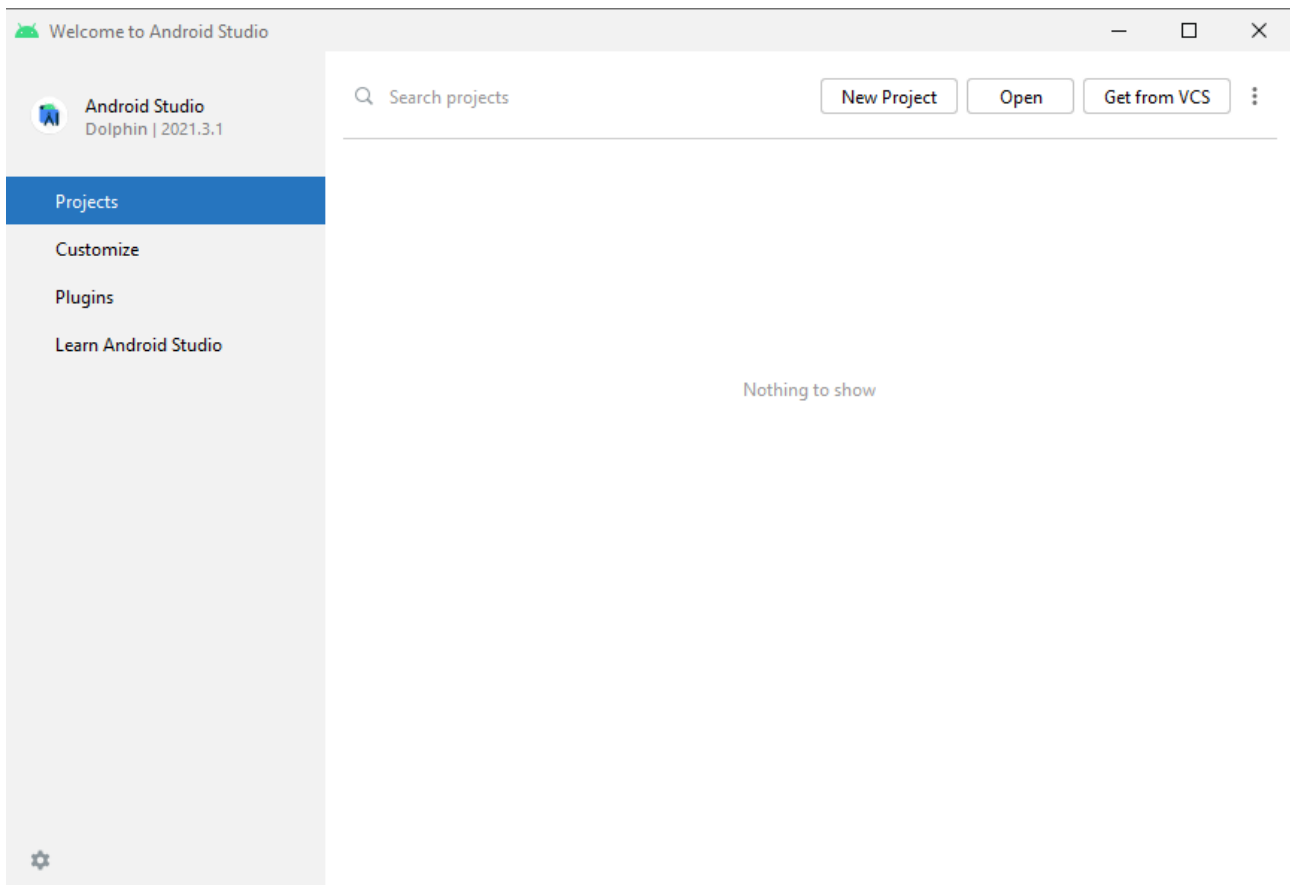
```

14     }
15   }
16 }
17 }
  
```

## 5 Značajka - inspiracija

Kao prvu aktivnost koju nudimo unutar naše aplikacije, omogućit ćemo korisnicima da se inspiriraju poznatom osobom koja je za nas inspirativna. Na radionici možete samostalno odabrati osobu koju želite, prikupiti potrebne materijale i zatim iskoristiti upute i kod dan na ovim stranicama kako biste aplikaciju učinili osobnijom.

Krenut ćemo s informacijom da je fiksni sadržaj u Android aplikacijama izdvojen u datoteke za specifične oblike sadržaja. Primjerice, tekst koji prikazujete na gumbima, dimenzije margina ili slike koje predstavljaju fiksne grafičke elemente u aplikacijama izdvajaju se izvan koda aplikacije. Razlog je taj što iz brojnih razloga u nekim situacijama trebate drugačiji izgled ili prikaz. Primjerice, korisnici u Njemačkoj očekuju sadržaj na njemačkom jeziku, dok bi stanovnicima Hrvatske to predstavljalo problem i oni očekuju hrvatski jezik. Svaka vrijednost koju kanite rabiti predstavljena je resursom koji ima jedinstveni identifikator. Ovo omogućuje programeru da koristi resurs preko identifikatora, a prepusti sustavu da, ovisno o postavkama sustava, brine o tome koji će se konkretan resurs učitati.



Slika 17: Kreiranje novog projekta - 1. korak

Tekst se tako izdvaja u datoteku **strings.xml** koja je smještena u direktorij **res/values**. Cjelokupni izgled ove datoteke dan je izlistanjem ?? i uključuje tekst za cijelu aplikaciju.

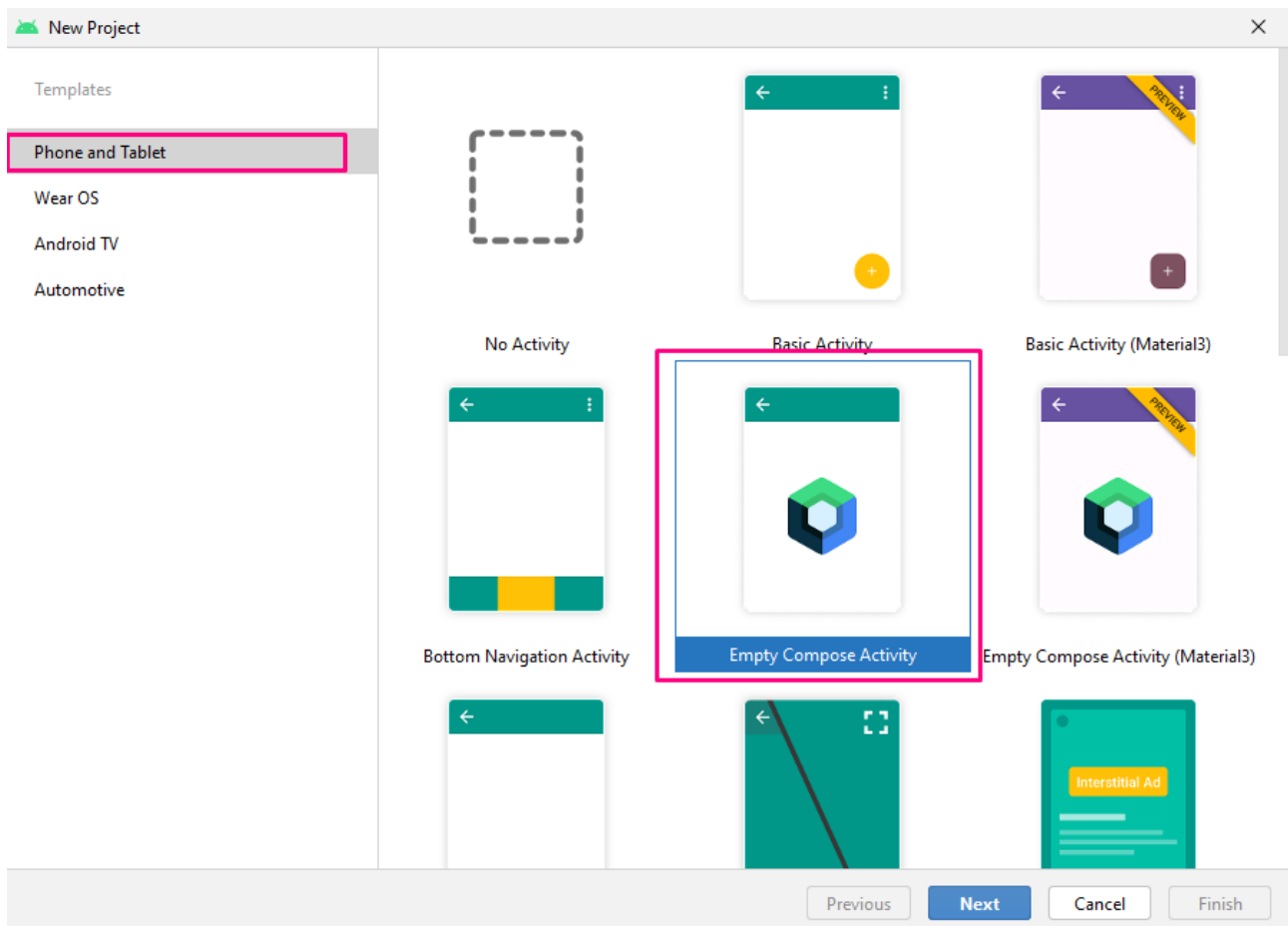
Listing 10: Glavna aktivnost - MainActivity.kt

```

1 <resources>
2 <string name="app_name">Laughly</string>
3 <!-- Custom strings: -->
4 <string name="label_inspire">Get inspired!</string>
5 <string name="label_laugh">Laugh!</string>
6 <string name="label_play">Play!</string>
7 <string name="label_quote">Quote</string>
8 <string name="name_turing">Alan Turing</string>
9 <string name="info_turing">"He was the most important mathematician of the second world war"
10 <string name="quote_turing_1">Sometimes it is the people no one imagines anything of
11 <string name="quote_turing_2">A computer would deserve to be called intelligent if it
12 <string name="quote_turing_3">Unless in communicating with it one says exactly what
13 <string name="label_reroll">Re-roll</string>
14 <string name="description_joke_icon">New joke icon</string>
15 <string name="label_refresh_joke">Get new joke!</string>
16 </resources>

```

Sadržaj koji će biti prikazan u Android aplikacijama uobičajeno se ne definira unutar ekrana koji definira kako će sadržaj izgledati (razdvajanje prezentacije od implementacije). Za potrebe razdvaja-



Slika 18: Kreiranje novog projekta - 2. korak

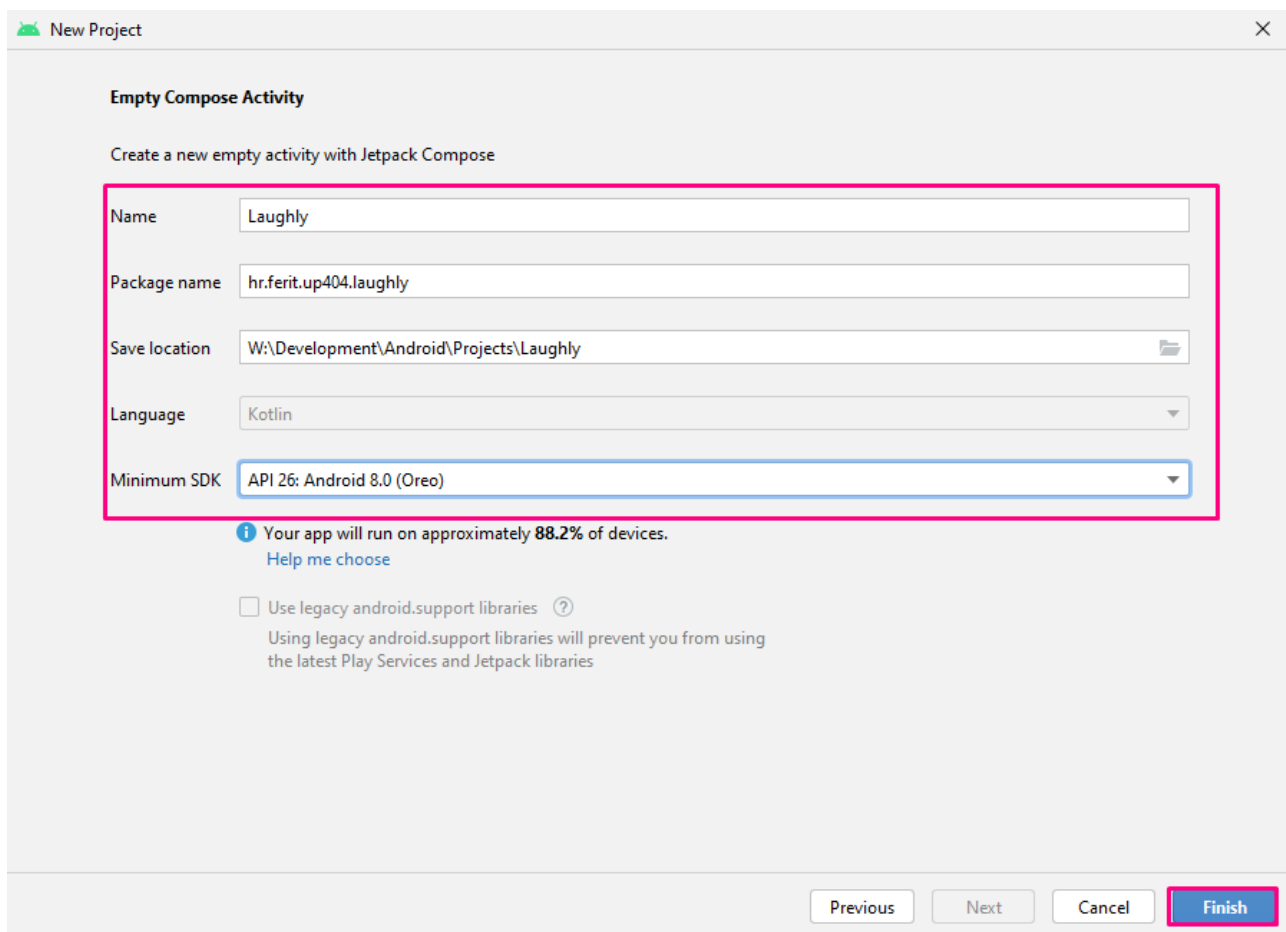
nja ovog sadržaja, ali i ostvarivanje brojnih drugih prednosti, koriste se komponente koje se nazivaju **ViewModel**. Tijekom radionice kreirat ćemo odgovarajuće **ViewModel** komponente za svaki ekran aplikacije. Za **InspireScreen** taj je **ViewModel** dan izlistanjem `??`. Može se uočiti kako taj **ViewModel** referencira resurse koje ćemo unutar ekrana iskoristiti za prikaz sadržaja te omogućuje da se dohvati resurs nasumičnog citata u bilo kojem trenutku kroz funkciju `getQuote()`.

Listing 11: Glavna aktivnost - MainActivity.kt

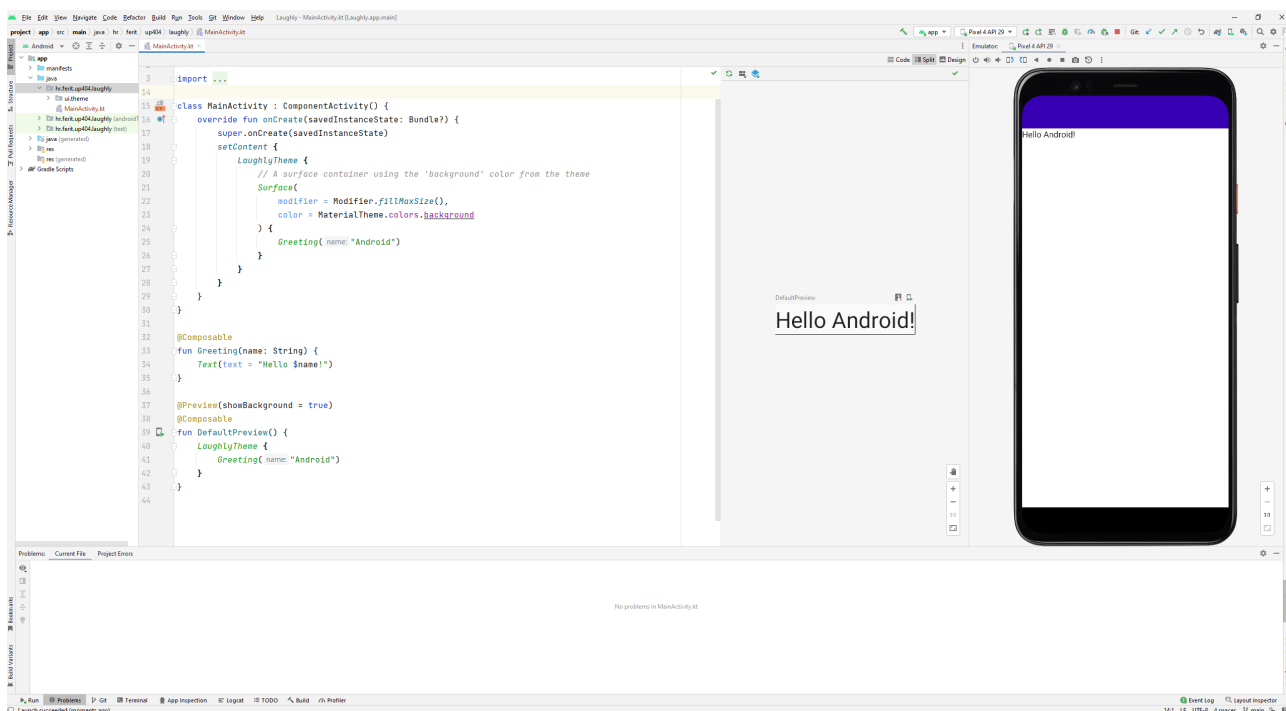
```

1 class InspireViewModel : ViewModel() {
2
3     val name = R.string.name_turing
4     val info = R.string.info_turing
5     val image = R.drawable.turing
6     private val quotes = listOf<Int>(
7         R.string.quote_turing_1,
8         R.string.quote_turing_2,
9         R.string.quote_turing_3
10    )
11
12    fun getQuote() = quotes.random()
13 }

```



Slika 19: Kreiranje novog projekta - 3. korak



Slika 20: Kreiranje novog projekta - 4. korak

Jednom kada je `ViewModel` definiran, pristupamo definiranju ekrana koji ga rabi. U ovom slučaju je riječ o inspirirajućem ekranu `InspireScreen`. Inspire screen sastoji se od nekolicine komponenti koje



omogućuju prikaz svog sadržaja smještenog unutar ranije danog `ViewModel`-a. Sam `ViewModel` ubacuje se u ekran svaki put kada ga je potrebno prikazati, a za te potrebe rabi se funkcija `viewModel()`.

Listing 12: Inspirirajući ekran - `InspireScreen.kt`

```
1 @Composable
2 fun InspireScreen(viewModel: InspireViewModel = viewModel()) {
3     Column(
4         modifier = Modifier.fillMaxSize()
5         .background(colorResource(id = R.color.cream))
6         .padding(dimensionResource(id = R.dimen.padding_medium))
7     ) {
8         Row(
9             verticalAlignment = Alignment.CenterVertically,
10            horizontalArrangement = Arrangement.Center,
11            modifier = Modifier.weight(1f)
12        ) {
13            Image(
14                painter = painterResource(id = viewModel.image),
15                contentDescription = stringResource(id = viewModel.name),
16                contentScale = ContentScale.Crop,
17                modifier = Modifier.size(100.dp, 100.dp).clip(CircleShape)
18            )
19            Text(
20                text = stringResource(viewModel.name),
21                fontSize = 48.sp,
22                textAlign = TextAlign.Center,
23                color = colorResource(id = R.color.rosewater),
24                modifier = Modifier.fillMaxWidth()
25            )
26        }
27        Text(
28            text = stringResource(id = viewModel.info),
29            fontSize = 14.sp,
30            modifier = Modifier
31                .weight(3f)
32                .fillMaxWidth()
33                .padding(dimensionResource(id = R.dimen.padding_medium))
34        )
35        Column(
36            horizontalAlignment = Alignment.CenterHorizontally,
37            verticalArrangement = Arrangement.SpaceEvenly,
38            modifier = Modifier.weight(1f)
39        ) {
40            var quoteResourceId by remember { mutableStateOf(viewModel.getQuote()) }
41            Text(
```

```

42     text = stringResource(id = quoteResourceId),
43     fontSize = 16.sp,
44     textAlign = TextAlign.Center,
45     color = colorResource(id = R.color.hotpink),
46     modifier = Modifier.fillMaxWidth()
47 )
48 Button(onClick = { quoteResourceId = viewModel.getQuote() }) {
49     Text(text = stringResource(R.string.label_quote))
50 }
51 }
52 }
53 }

```

## **6 Značajka - igra**

## **7 Značajka - šala**

## **8 Što dalje?**