Laboratorijska vježba 5

UI dizajn pattern-i i principi dizajna Android aplikacija; Napredno instrumented testiranje

Zadatak 1

Omogućite tranzicije s animacijom između liste elemenata i detalja elemenata korištenjem proizvoljne animacije, animacije sa zajedničkom slikom i animacije sa zajedničkim nazivom i slikom

U prvom koraku se trebaju omogućiti animacije i tranzicije. Animacije i tranzicije se mogu definisati u sklopu `theme.xml` ili kroz kôd. U slučaju da želimo omogućiti tranzicije u glavnoj aktivnosti prije nego izvršimo `setContentView` naredbu ćemo dodati sljedeći kôd kroz koji se definiše vrsta tranzicije.

```
with(window) {
    requestFeature(Window.FEATURE_ACTIVITY_TRANSITIONS)
    // postavit ćemo exitTranziciju
    exitTransition = Fade()
}
```

Sljedeće tranzicije su dostupne:

- `explode` - Pomjera view u/iz centra ekrana
- `slide` - Pomjera view u/iz krajeva ekrana
- `fade` - Dodaje ili uklanja view promjenom vidljivosti

Moguće je postaviti sljedeće tipove tranzicija:

- `enterTransition`
- `exitTransition`
- `sharedElementEnterTransition`
- `sharedElementExitTransition`

U slučaju da koristimo starije verzije Androida treba provjeriti da li je moguće vršiti animacije. Animacije su dostupne od 5.0 verzije odnosno Lollipop. Prije nego se izvrše iste vrši se provjera sa:

```
// Check if we're running on Android 5.0 or higher
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
    // Apply activity transition
} else {
    // Swap without transition
}
```

Obzirom da je aplikacija na vježbi s min_sdk 7.0 ova provjera se neće vršiti.

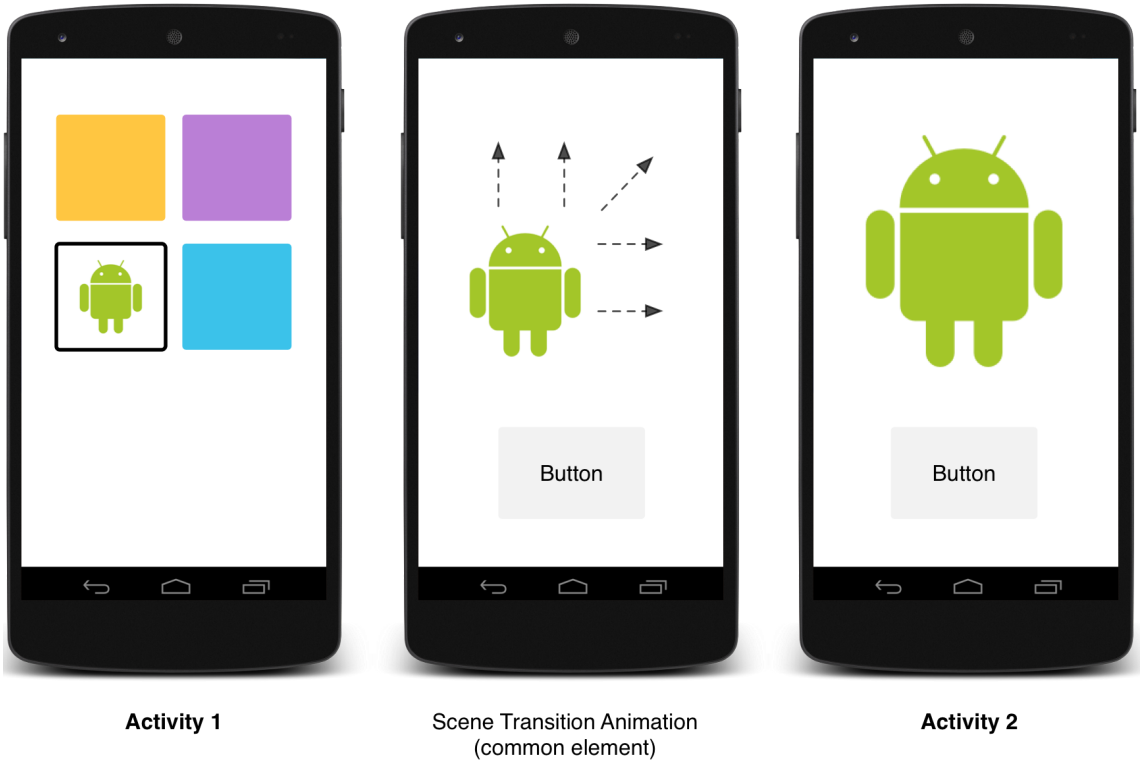
Napomena: Možete kreirati i vlastite tranzicije.

U sklopu `showMovieDetails` funkcije u RecentFragment i FavoriteFragment ćemo dodati sljedeću liniju umjesto klasičnog startanja aktivnosti:

```
startActivity(intent,
          ActivityOptions.makeSceneTransitionAnimation(activity).toBundle())
```

Kada pokrenemo aplikaciju vidjet ćemo da se sada izvršava odgovarajuća animacija.

Znamo da je slika postera jednaka u listi i u detaljima. Iskoristimo nju kao zajedničku stavku u tranziciji.



Ako želimo imati tranziciju sa zajedničkim elementima trebamo uraditi sljedeće:

1. Omogućiti tranzicije
2. Specificirati zajednički element
3. Po potrebi definisati vlastitu tranziciju kao XML resurs
4. Dodijeljiti zajednički naziv elementima u oba layout-a `android:transitionName`
5. Koristiti `ActivityOptions.makeSceneTransitionAnimation()` funkciju

U odgovarajućim .xml file-ovima layout-a ćemo dodati za slike sljedeću liniju:

```
android:transitionName="poster"
```

Ovim označavmo da je ovo zajednički element.

Obzirom da u metodi `showMovieDetails` nemamo direktni pristup `ImageView`-u postera u `RecyclerView`, promjenit ćemo našu funkciju višeg reda da pored filma prima i view.

U konstruktoru adaptera ćemo imati sljedeće:

```
private val onItemClicked: (movie:Movie,view:View) -> Unit
```

Postavljanje listener-a ćemo vršiti na sljedeći način:

```
holder.itemView.setOnClickListener{ onItemClicked(movies[position],holder.movieImage)
}
```

Unutar fragmenta ćemo proslijediti adapteru metodu na sljedeći način:

```
favoriteMoviesAdapter = MovieListAdapter(arrayListOf()) { movie,view ->
showMovieDetails(movie,view) }
```

Metoda `showMovieDetails` će sada izgledati ovako:

```
private fun showMovieDetails(movie: Movie, view: View) {
    val intent = Intent(activity, MovieDetailActivity::class.java).apply {
        putExtra("movie_title", movie.title)
    }
    val options = ActivityOptions
            .makeSceneTransitionAnimation(activity, view, "poster")
    startActivity(intent, options.toBundle())
}
```

Sada `makeSceneTransitionAnimation` funkciji prosljeđujemo odgovarajući view elementa koji je jednak i njihov zajednički naziv.

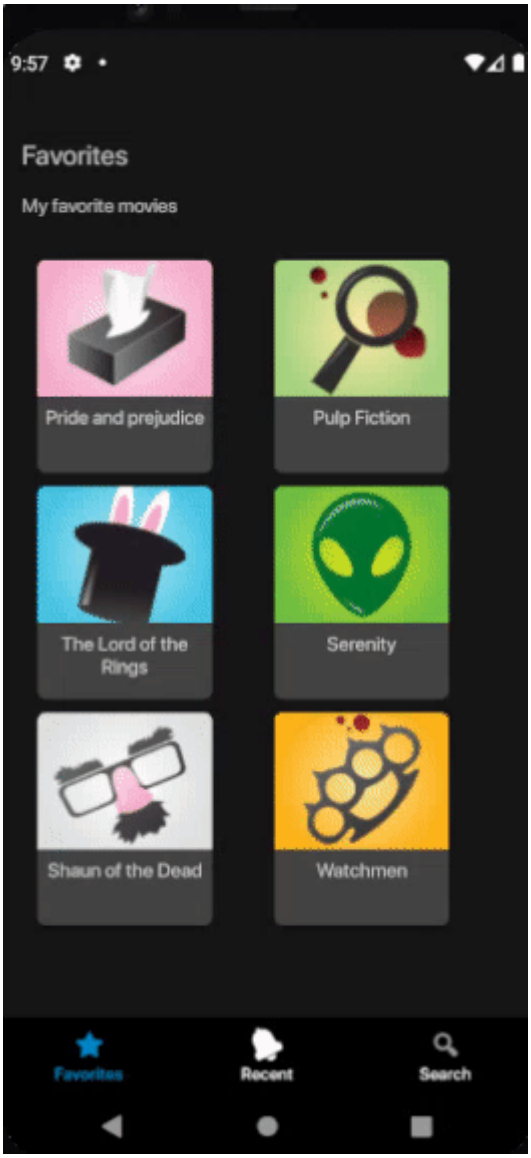
Uradimo isto sada s nazivom filma. U odgovarajućim elementima u layout-ima ćemo dodati da je naziv tranzicije `title`. Odgovarajuće `onClick` stavke ćemo proširiti s još jednim view-om analogno prethodno urađenom. Metoda `showMovieDetails` će sada izgledati ovako:

```
private fun showMovieDetails(movie: Movie, view1: View,view2:View) {
    val intent = Intent(activity, MovieDetailActivity::class.java).apply {
        putExtra("movie_title", movie.title)
    }
    val options = ActivityOptions
            .makeSceneTransitionAnimation(activity,  UtilPair.create(view1, "poster"),
                UtilPair.create(view2, "title"))
    startActivity(intent, options.toBundle())
}
```

`UtilPair` nije ništa drugo nego `Android.Pair` importovan kao `UtilPair`, da ne bi došlo do problema kod imenovanja.

```
import android.util.Pair as UtilPair
```

Izgled aplikacije na kraju zadatka:



Zadaci

1. Napravite instrumented testove kojim ćete testirati da se ispravno prikazuje lista glumaca u filmovima
2. Napravite instrumented test kojim ćete testirati da aplikacija reaguje na `Intent.ACTION_SHARE`