**Navigation icon**

private final AnimatorSet animatorSet = new AnimatorSet();

private Context context;

private View sheet;

private Interpolator interpolator;

private int height;

private boolean backdropShown = false;

private Drawable openIcon;

private Drawable closeIcon;

NavigationIconClickListener(Context context, View sheet) {

this(context, sheet, null);

}

NavigationIconClickListener(Context context, View sheet, @Nullable Interpolator interpolator) {

this(context, sheet, interpolator, null, null);

}

public NavigationIconClickListener(

Context context, View sheet, @Nullable Interpolator interpolator,

@Nullable Drawable openIcon, @Nullable Drawable closeIcon) {

this.context = context;

this.sheet = sheet;

this.interpolator = interpolator;

this.openIcon = openIcon;

this.closeIcon = closeIcon;

DisplayMetrics displayMetrics = new DisplayMetrics();

((Activity) context).getWindowManager().getDefaultDisplay().getMetrics(displayMetrics);

height = displayMetrics.heightPixels;

}

@Override

public void onClick(View view) {

backdropShown = !backdropShown;

// Cancel the existing animations

animatorSet.removeAllListeners();

animatorSet.end();

animatorSet.cancel();

updateIcon(view);

final int translateY = height -

context.getResources().getDimensionPixelSize(R.dimen.product\_grid\_reveal\_height);

ObjectAnimator animator = ObjectAnimator.ofFloat(sheet, "translationY", backdropShown ? translateY : 0);

animator.setDuration(500);

if (interpolator != null) {

animator.setInterpolator(interpolator);

}

animatorSet.play(animator);

animator.start();

//miscController.MessageBox((MainActivity)MainActivity.GLOBALS.get("app"),"Ebony",MainActivity.GLOBALS.get("actualFragment").toString(), AlertStyle.DIALOG);

}

private void updateIcon(View view) {

if (openIcon != null && closeIcon != null) {

if (!(view instanceof ImageView)) {

throw new IllegalArgumentException("updateIcon() must be called on an ImageView");

}

if (backdropShown) {

((ImageView) view).setImageDrawable(closeIcon);

} else {

((ImageView) view).setImageDrawable(openIcon);

}

}

}

**Mis juegos**

private List<MiJuego> miJuegos;

private Context context;

public MisJuegosAdapter(List<MiJuego> miJuegos) {

this.miJuegos = miJuegos;

}

@NonNull

@Override

public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {

View view = LayoutInflater.from(parent.getContext())

.inflate(R.layout.row\_mis\_juegos,parent,false);

context = parent.getContext();

return new ViewHolder(view);

}

@Override

public void onBindViewHolder(@NonNull ViewHolder holder, int position) {

MiJuego miJuego = miJuegos.get(position);

String imgUri = "@drawable/"+ miJuego.getImagen();

int imgResource = context.getResources().getIdentifier

(imgUri,null,context.getPackageName());

holder.imgMisjuegos.setImageResource(imgResource);

holder.txtTitulo.setText(miJuego.getTitulo());

holder.rbClasificacion.setRating(miJuego.getClasificacion());

holder.txtDescripcion.setText(miJuego.getDescripcion());

}

@Override

public int getItemCount() {

return miJuegos.size();

}

public class ViewHolder extends RecyclerView.ViewHolder {

private View view;

private AppCompatImageView imgMisjuegos;

private TextView txtTitulo;

private AppCompatRatingBar rbClasificacion;

private TextView txtDescripcion;

public ViewHolder(@NonNull View itemView) {

super(itemView);

imgMisjuegos= itemView.findViewById(R.id.imgJuego);

txtTitulo = itemView.findViewById(R.id.txtDescripcion);

rbClasificacion = itemView.findViewById(R.id.rbClasificacion);

txtDescripcion = itemView.findViewById(R.id.txtDescripcion);

this.view= itemView;

}

}

**Juegos adapter**

private List<Juego> juegos;

private Context context;

public JuegosAdapter(List<Juego> juegos) {

this.juegos = juegos;

}

@NonNull

@Override

public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {

View view = LayoutInflater.from(parent.getContext())

.inflate(R.layout.row\_juego,parent,false);

context = parent.getContext();

return new ViewHolder(view);

}

@Override

public void onBindViewHolder(@NonNull ViewHolder holder, int position) {

Juego juego = juegos.get(position);

String imgUri = "@drawable/"+ juego.getImagen();

int imgResource = context.getResources().getIdentifier

(imgUri,null,context.getPackageName());

holder.imgJuego.setImageResource(imgResource);

holder.txtTitulo.setText(juego.getTitulo());

holder.rbClasificacion.setRating(juego.getClasificacion());

holder.txtDescripcion.setText(juego.getDescripcion());

}

@Override

public int getItemCount() {

return juegos.size();

}

public class ViewHolder extends RecyclerView.ViewHolder {

private View view;

private AppCompatImageView imgJuego;

private TextView txtTitulo;

private AppCompatRatingBar rbClasificacion;

private TextView txtDescripcion;

public ViewHolder(@NonNull View itemView) {

super(itemView);

imgJuego = itemView.findViewById(R.id.imgJuego);

txtTitulo = itemView.findViewById(R.id.txtTitulo);

rbClasificacion = itemView.findViewById(R.id.rbClasificacion);

txtDescripcion= itemView.findViewById(R.id.txtDescripcion);

this.view = itemView;

}

}

**Categorias adapter**

**private List<Categoria> categorias;**

**private Context context;**

**public CategoriasAdapter(List<Categoria> categorias) {**

**this.categorias = categorias;**

**}**

**@NonNull**

**@Override**

**public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {**

**View view = LayoutInflater.from(parent.getContext())**

**.inflate(R.layout.row\_categoria,parent,false);**

**context = parent.getContext();**

**return new ViewHolder(view);**

**}**

**@Override**

**public void onBindViewHolder(@NonNull ViewHolder holder, int position) {**

**Categoria categoria = categorias.get(position);**

**String imgUri = "@drawable/"+categoria.getImagen();**

**int imgResource = context.getResources().getIdentifier**

**(imgUri,null,context.getPackageName());**

**holder.imgCategoria.setImageResource(imgResource);**

**holder.txtNombreCategoria.setText(categoria.getNombreCategoria());**

**}**

**@Override**

**public int getItemCount() {**

**return categorias.size();**

**}**

**public class ViewHolder extends RecyclerView.ViewHolder {**

**private View view;**

**private AppCompatImageView imgCategoria;**

**private TextView txtNombreCategoria;**

**public ViewHolder(@NonNull View itemView) {**

**super(itemView);**

**imgCategoria= itemView.findViewById(R.id.imgCategoria);**

**txtNombreCategoria= itemView.findViewById(R.id.txtNombrecategoria);**

**this.view = itemView;**

**}**

**}**