Министерство образования Республики Беларусь

Учреждение образования

“БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ”

Факультет компьютерного проектирования

Дисциплина: Компьютерные сети

**ЛАБОРАТОРНАЯ РАБОТА № 2**

Android - рисовалка

Выполнил:

студент гр. 714301

Гельдымурадов С.

Проверил:

Минск 2020

Используется язык Java 1.8.

Реализован функционал слоев, отмены слоя, возврат отменненного слоя. Так же сохранение и открытие файлов из памяти телефона. Из фигур доступны линии, карандаш, круг и квадрат. Каждый слой отображается на в панеле слоев, чуть выше панели интсрументов. Доступен выбор цвета для инструментов.



Рисунок 1. Главное окно программы.

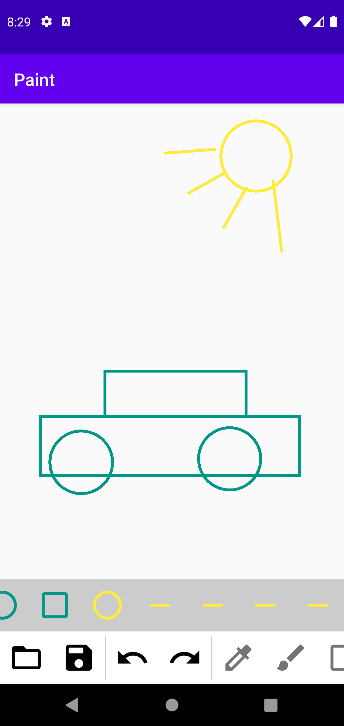


Рисунок 2. Пример рисунка.

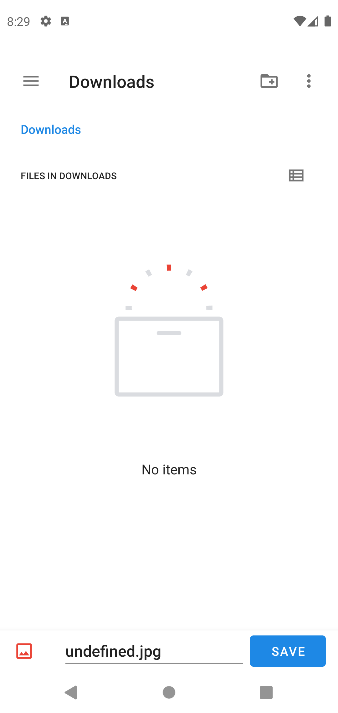


Рисунок 3. Окно сохранения файла.

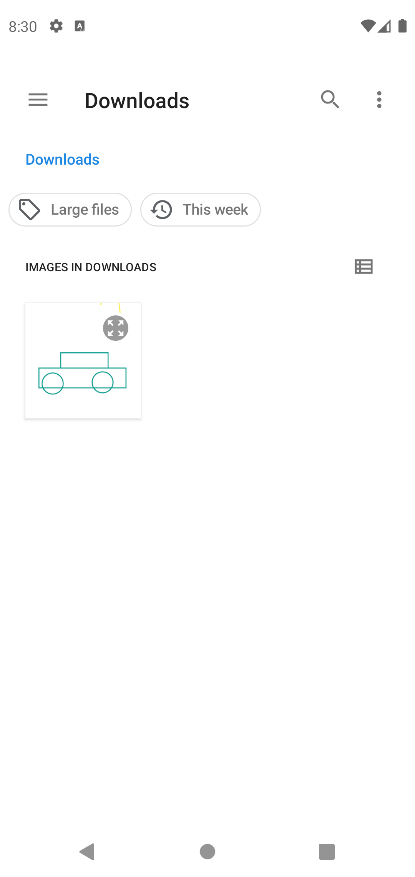


Рисунок 4. Окно открытия файла.

**Листинг кода.**

**https://github.com/bsuir-temp/ol2**

**public class MainActivity extends AppCompatActivity implements ColorPickerDialogListener {**

**private static final int READ\_REQUEST\_CODE = 52;**

**private static final int WRITE\_REQUEST\_CODE = 53;**

**private Canvas canvas = null;**

**public void onCreate(Bundle savedInstanceState) {**

**super.onCreate(savedInstanceState);**

**setContentView(R.layout.activity\_main);**

**this.canvas = findViewById(R.id.canvas);**

**this.canvas.setAdapter((RecyclerView)findViewById(R.id.layers));**

**}**

**@Override**

**public void onActivityResult(int requestCode, int resultCode, Intent resultData) {**

**super.onActivityResult(requestCode, resultCode, resultData);**

**try {**

**if (requestCode == READ\_REQUEST\_CODE && resultCode == Activity.RESULT\_OK) {**

**if (resultData != null) {**

**PaintLayer layer = new PaintLayer();**

**layer.bitmap = getBitmapFromUri(resultData.getData());**

**layer.type = PaintLayer.drawingtype.OPEN;**

**layer.paint = new Paint();**

**if (this.canvas.paints.size() != this.canvas.layerPosition) {**

**this.canvas.paints.add(this.canvas.layerPosition, layer);**

**this.canvas.paints.subList(this.canvas.layerPosition + 1, this.canvas.paints.size()).clear();**

**} else**

**this.canvas.paints.add(layer);**

**this.canvas.layerPosition++;**

**this.canvas.adapter.notifyDataSetChanged();**

**}**

**}else**

**if (requestCode == WRITE\_REQUEST\_CODE && resultCode == Activity.RESULT\_OK) {**

**if (resultData != null) {**

**setBitmapUri(resultData.getData());**

**}**

**}**

**}catch (Throwable th){**

**th.printStackTrace();**

**//error(th.getLocalizedMessage());**

**}**

**}**

**private Bitmap getBitmapFromUri(Uri uri) throws IOException {**

**ParcelFileDescriptor parcelFileDescriptor = getContentResolver().openFileDescriptor(uri, "r");**

**FileDescriptor fileDescriptor = parcelFileDescriptor.getFileDescriptor();**

**Bitmap image = BitmapFactory.decodeFileDescriptor(fileDescriptor);**

**parcelFileDescriptor.close();**

**return image;**

**}**

**public static Bitmap getBitmapFromView(View view) {**

**Bitmap returnedBitmap = Bitmap.createBitmap(view.getWidth(), view.getHeight(),Bitmap.Config.ARGB\_8888);**

**android.graphics.Canvas canvas = new android.graphics.Canvas(returnedBitmap);**

**Drawable bgDrawable =view.getBackground();**

**if (bgDrawable!=null)**

**bgDrawable.draw(canvas);**

**else**

**canvas.drawColor(Color.WHITE);**

**view.draw(canvas);**

**return returnedBitmap;**

**}**

**private void setBitmapUri(Uri uri) {**

**try {**

**ParcelFileDescriptor pfd = this.getContentResolver().openFileDescriptor(uri, "w");**

**FileOutputStream fileOutputStream = new FileOutputStream(pfd.getFileDescriptor());**

**Bitmap temp = getBitmapFromView(this.canvas);**

**temp.compress(Bitmap.CompressFormat.JPEG, 100, fileOutputStream);**

**fileOutputStream.flush();**

**fileOutputStream.close();**

**pfd.close();**

**} catch (Exception e) {**

**e.printStackTrace();**

**//error(e.getLocalizedMessage());**

**}**

**}**

**public void undo(View view) { this.canvas.undo(); }**

**public void redo(View view) { this.canvas.redo(); }**

**public void torect(View view) { this.canvas.type = PaintLayer.drawingtype.RECT; }**

**public void tocirc(View view) { this.canvas.type = PaintLayer.drawingtype.CIRC; }**

**public void toline(View view) { this.canvas.type = PaintLayer.drawingtype.LINE; }**

**public void tobrus(View view) { this.canvas.type = PaintLayer.drawingtype.BRUSH; }**

**public void save(View view) {**

**Intent intent = new Intent(Intent.ACTION\_CREATE\_DOCUMENT);**

**intent.addCategory(Intent.CATEGORY\_OPENABLE);**

**intent.setType("image/jpeg");**

**intent.putExtra(Intent.EXTRA\_TITLE, "undefined.jpg");**

**startActivityForResult(intent, WRITE\_REQUEST\_CODE);**

**}**

**public void open(View view) {**

**Intent intent = new Intent(Intent.ACTION\_OPEN\_DOCUMENT);**

**intent.addCategory(Intent.CATEGORY\_OPENABLE);**

**intent.setType("image/\*");**

**startActivityForResult(intent, READ\_REQUEST\_CODE);**

**}**

**public void picker(View view) {**

**ColorPickerDialog.newBuilder()**

**.setColor(this.canvas.color)**

**.setDialogType(ColorPickerDialog.TYPE\_PRESETS)**

**.setAllowCustom(true)**

**.setAllowPresets(true)**

**.setColorShape(ColorShape.CIRCLE)**

**.setDialogId(1)**

**.show(this);**

**}**

**public void error(String text)**

**{**

**AlertDialog alert = new AlertDialog.Builder(getBaseContext()).create();**

**alert.setTitle("Error "+text);**

**alert.show();**

**}**

**@Override**

**public void onColorSelected(int dialogId, int color) { this.canvas.color = color; }**

**@Override**

**public void onDialogDismissed(int dialogId) { }**

**}**

**public class Canvas extends View {**

**public int layerPosition = 0;**

**public ArrayList<PaintLayer> paints = new ArrayList<>();**

**public PaintLayer.drawingtype type = PaintLayer.drawingtype.BRUSH;**

**public LayersAdapter adapter = new LayersAdapter(this);**

**public int color = Color.BLACK;**

**public Canvas(Context context) {**

**super(context);**

**init();**

**}**

**public Canvas(Context context, AttributeSet attrs) {**

**super(context, attrs);**

**init();**

**}**

**public Canvas(Context context, AttributeSet attrs, int defStyleAttr) {**

**super(context, attrs, defStyleAttr);**

**init();**

**}**

**public Canvas(Context context, AttributeSet attrs, int defStyleAttr, int defStyleRes) {**

**super(context, attrs, defStyleAttr, defStyleRes);**

**init();**

**}**

**private void init() {**

**setOnTouchListener(getTouchListener());**

**}**

**public void undo() {**

**int i = this.layerPosition;**

**if (i > 0) {**

**this.layerPosition = i - 1;**

**adapter.notifyDataSetChanged();**

**}**

**invalidate();**

**}**

**public void redo() {**

**int size = this.paints.size();**

**int i = this.layerPosition;**

**if (size > i) {**

**this.layerPosition = i + 1;**

**adapter.notifyDataSetChanged();**

**}**

**invalidate();**

**}**

**public void onDraw(android.graphics.Canvas canvas) {**

**int i = 0;**

**while (i < this.paints.size() && i < this.layerPosition) {**

**this.paints.get(i).draw(canvas);**

**i++;**

**}**

**}**

**public void setAdapter(RecyclerView recyclerView)**

**{**

**recyclerView.setAdapter(adapter);**

**final LinearLayoutManager layoutManager = new LinearLayoutManager(getContext(), LinearLayoutManager.HORIZONTAL, false);**

**recyclerView.setLayoutManager(layoutManager);**

**}**

**private View.OnTouchListener getTouchListener() {**

**return new View.OnTouchListener() {**

**public boolean onTouch(View v, MotionEvent event) {**

**if (event.getAction() == MotionEvent.ACTION\_DOWN) {**

**PaintLayer layer = new PaintLayer();**

**layer.type = type;**

**layer.paint = new Paint();**

**layer.paint.setColor(color);**

**layer.paint.setStyle(Paint.Style.STROKE);**

**layer.paint.setStrokeWidth(9.0f);**

**if(paints.size()!=layerPosition) {**

**paints.add( layerPosition,layer);**

**paints.subList(layerPosition+1,paints.size()).clear();**

**}**

**else**

**paints.add(layer);**

**layerPosition++;**

**adapter.notifyDataSetChanged();**

**}**

**paints.get(paints.size() - 1).onTouch(v, event);**

**invalidate();**

**return true;**

**}**

**};**

**}**

**}**

**public class PaintLayer {**

**private static final int UNDEFINED = 0;**

**public coord coord = new coord();**

**public Paint paint;**

**public Bitmap bitmap;**

**public Path path = new Path();**

**public drawingtype type = drawingtype.BRUSH;**

**public enum drawingtype {**

**BRUSH,**

**LINE,**

**RECT,**

**CIRC,**

**FILL,**

**OPEN**

**}**

**public void onTouch(View v, MotionEvent event) {**

**int action = event.getAction();**

**if (action == MotionEvent.ACTION\_DOWN) {**

**this.coord.startX = event.getX();**

**this.coord.startY = event.getY();**

**this.path.reset();**

**this.path.moveTo(this.coord.startX,this.coord.startY);**

**} else if (action == MotionEvent.ACTION\_UP)**

**{**

**this.coord.endX = event.getX();**

**this.coord.endY = event.getY();**

**this.path.lineTo(this.coord.endX,this.coord.endY);**

**} else if(action == MotionEvent.ACTION\_MOVE) {**

**this.coord.endX = event.getX();**

**this.coord.endY = event.getY();**

**this.path.lineTo(this.coord.endX,this.coord.endY);**

**}**

**}**

**public double calculateDistanceBetweenPoints(**

**float x1,**

**float y1,**

**float x2,**

**float y2) {**

**float xDelta = x2 - x1;**

**float yDelta = y2 - y1;**

**return Math.sqrt(xDelta \* xDelta + yDelta \* yDelta);**

**}**

**public void draw(Canvas canvas) {**

**if(type!=drawingtype.OPEN)**

**if(this.coord.startX == UNDEFINED || this.coord.endX == UNDEFINED)return;**

**switch (type)**

**{**

**case RECT:**

**canvas.drawRect(this.coord.startX, this.coord.startY, this.coord.endX, this.coord.endY, this.paint);break;**

**case CIRC:**

**final float radius = Float.parseFloat(String.valueOf(calculateDistanceBetweenPoints(this.coord.startX,this.coord.startY,this.coord.endX,this.coord.endY)));**

**canvas.drawCircle(this.coord.startX, this.coord.startY, radius, this.paint);break;**

**case BRUSH:**

**canvas.drawPath(this.path, this.paint);break;**

**case LINE:**

**canvas.drawLine(this.coord.startX, this.coord.startY, this.coord.endX, this.coord.endY, this.paint);break;**

**case OPEN:**

**canvas.drawBitmap(this.bitmap,0,0, paint);**

**}**

**}**

**static class coord {**

**public float endX = UNDEFINED;**

**public float endY = UNDEFINED;**

**public float startX = UNDEFINED;**

**public float startY = UNDEFINED;**

**}**

**}**

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

**public ImageView icon;**

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

**super(itemView);**

**icon = itemView.findViewById(R.id.layericon);**

**}**

**}**

**public class LayersAdapter extends RecyclerView.Adapter<LayerVH> {**

**private Canvas canvas;**

**public LayersAdapter(Canvas canvas) {**

**this.canvas = canvas;**

**}**

**@NonNull**

**@Override**

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

**return new LayerVH(LayoutInflater.from(parent.getContext())**

**.inflate(R.layout.layer, parent, false));**

**}**

**@Override**

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

**holder.icon.setColorFilter(canvas.paints.get(position).paint.getColor());**

**switch (canvas.paints.get(position).type)**

**{**

**case BRUSH:holder.icon.setImageResource(R.drawable.ic\_brush);break;**

**case CIRC:holder.icon.setImageResource(R.drawable.ic\_circle);break;**

**case RECT:holder.icon.setImageResource(R.drawable.ic\_rect);break;**

**case LINE:holder.icon.setImageResource(R.drawable.ic\_line);break;**

**case OPEN:holder.icon.setImageResource(R.drawable.ic\_open);break;**

**}**

**}**

**@Override**

**public int getItemCount() {**

**return canvas.layerPosition;**

**}**

**}**