

How I won Telegram's \$50,000 Sticker App Contest using Flutter

Rafal Bednarczuk • April 06, 2023

Telegram Contests

Telegram runs its own platform contest.com with developer challenges aimed at adding or improving features of Telegram ecosystem.

The contests are paid and prize pools are split between participants based on their results.

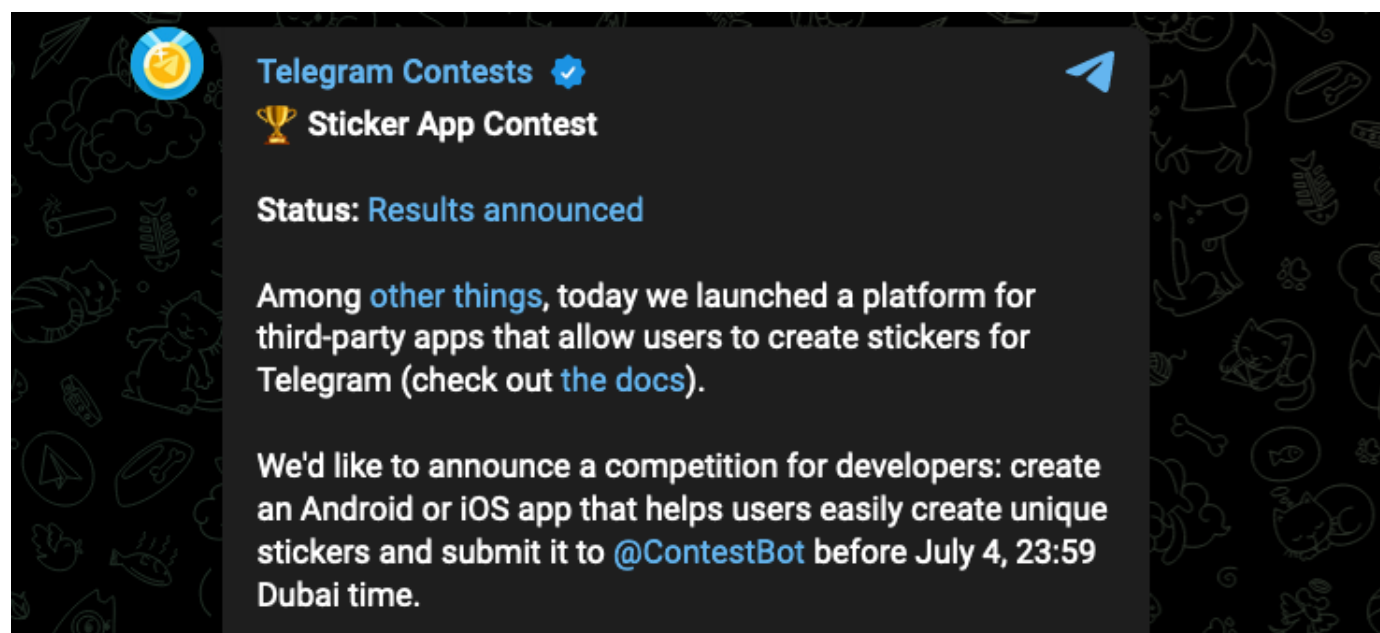
Sticker App Contest

On June 25 2021 Sticker App Contest was [announced](#).

The task was to create a mobile app that let users create and export Telegram stickers.

The contest had a prize pool of \$50,000, and participants had less than 10 days to develop their app.

I decided to take on this challenge using Flutter.



TStick app

The app I created, [TStick](#), is an image manipulation app designed for telegram static stickers.

My goal was to provide good UI/UX and make the features I implemented polished.

Using Telegram native iOS and Android libraries in Flutter

The core feature of a sticker app is exporting stickers.

Telegram team provided iOS and Android libraries for loading images in Telegram app and creating a sticker set from them. However, accessing native iOS and Android libraries in Flutter requires creating a Flutter plugin. Fortunately, Iurii Dorofeev created such a plugin and with a few small changes in my fork, it was suitable to be used in TStick app.

Translating, resizing and rotating

In TStick translation, resizing and rotation is used for the whole sticker, text, and added images.

Flutter provides high-level GestureDetector callbacks to handle these transformations.

```
GestureDetector(  
  onStart: (details) {  
    // Get initial touch point and save in _focalPointStart  
  },  
  onUpdate: (details) {  
    final scale = details.scale;  
    final translate = details.localFocalPoint - _focalPointStart;  
    final angle = details.rotation;  
    // Use scaling, translation and angle to transform scene, text or image  
  },  
  onEnd: (details) {  
    // End and save scaling, translation and rotation  
  },  
)
```

Transformer is used to apply manipulation data.

```
Transform.translate(  
  offset: translation,  
  child: Transform.rotate(  
    angle: angle,  
    child: Transform.scale(  
      scale: scale,  
      child: child,  
    ),  
  ),  
)
```

0:00



Drawing

Users can draw with a specific brush size and color.

Drawing data is taken from Listener widget.

```
Listener(  
  onPointerDown: (details){  
    // Get initial point  
  },  
  onPointerMove: (details){  
    // Get draw points and draw a line through all the points  
  },  
  onPointerUp: (details){  
    // End drawing  
  },  
)
```

The drawing itself is done with CustomPainter.

```
final path = Path();
path.moveTo(points.first.dx, points.first.dy);
points.skip(1).forEach((point) {
    path.lineTo(point.dx, point.dy);
});
canvas.drawPath(path, paint);
```

0:00



Text

Users can add text with a specific color and font.
The text can be transformed before being saved.

0:00



Manual background cut

To manually cut the background, users must draw the cutting area. Cutting area data points are captured with Listener widget.

To make a background cut, CustomClipper is used.

The cutting path is just a line through the points.

```
Path getClip(Size size) {
    final path = Path();
    path.moveTo(points.first.dx, points.first.dy);
    points.skip(1).forEach((point) {
        path.lineTo(point.dx, point.dy);
    });
    path.close();
    return path;
}
```

0:00



Auto background removal

Auto background removal is another method for cutting the background.

This feature is server-side and uses rembg running on the server. Rembg provides API access that takes an image as input and returns the same image with the background removed.

The only thing that I had to keep in mind is to compress the image before sending it to the server as full-size images take a lot of time to upload and process.

0:00



Outline

This feature creates a white border around non-transparent parts of the sticker. This is useful when a background manual or automatic cut was used and the shape of the sticker needs to be highlighted.

It is also helpful when our sticker is dark and we want to make it visible when posted in a chat with a dark background.

The outline is created by drawing a line through outline pixels.

To calculate outline pixels, an image of the current scene is captured and compressed for ease of pixel-by-pixel analysis. Outline pixels are transparent pixels with non-transparent pixels around them.

0:00



Exporting sticker packs

Stickers packs are exported from the sticker pack page.

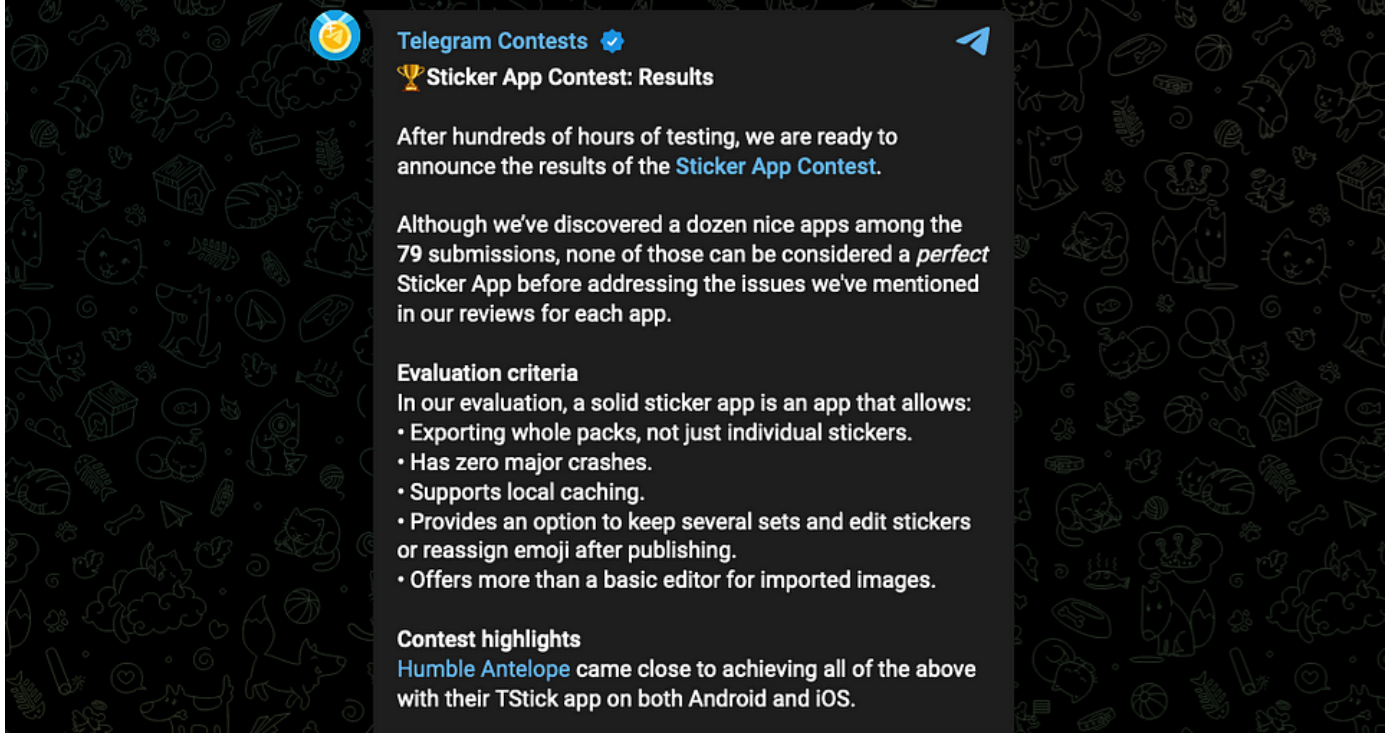
0:00



Sticker App Contest results

On November 11 2021, the results were posted and TStick placed 1st out of 79 submissions.

The app was highlighted as the one that came close to achieving all of the evaluation criteria and won the largest share of the \$50,000 prize pool.



How to install the app

If you want to create a Telegram stickerpack using TStick by yourself, the app is available on Apple App Store and Google Play:

<https://apps.apple.com/app/tstick/id1575132889>

<https://play.google.com/store/apps/details?id=xyz.freelanceinfluence.tstick>

Reddit post

https://www.reddit.com/r/programming/comments/13lr6cn/the_technical_details_of_the_app_that_won/