You

I'm trying to build an android app in android studio bjut it keeps crashing while testing, here is my code:

```
package com.has_akh.flying_football;

import com.badlogic.gdx.ApplicationAdapter;
import com.badlogic.gdx.Gdx;
import com.badlogic.gdx.graphics.Color;
import com.badlogic.gdx.graphics.Texture;
import com.badlogic.gdx.graphics.g2d.SpriteBatch;
import com.badlogic.gdx.graphics.glutils.ShapeRenderer;
import com.badlogic.gdx.math.Rectangle;
import com.badlogic.gdx.utils.ScreenUtils;

import java.util.Random;

public class MyGdxGame extends ApplicationAdapter {
  SpriteBatch batch;
  ShapeRenderer shapes = new ShapeRenderer();
  Texture background;
  Texture[] footballs;
  Rectangle[] barriers;
  Texture goalSupport;
  Texture goal;
  int thisFootball;
  float ballY;
  float velocity;
  int gameState;
  Random randomGenerator;
  int goalVelocity;
  int numOfGoals = 6;
  float[] supportX = new float[numOfGoals];
  float[] supportHeight = new float[numOfGoals];
  int distanceBetweenGoals;

  @Override
  public void create () {
    batch = new SpriteBatch();
    background = new Texture("Background.jpg");
    footballs = new Texture[2];
    goalSupport = new Texture("GoalSupport.png");
    goal = new Texture("Goal.png");
    footballs[0] = new Texture("Football.png");
    footballs[1] = new Texture("Football2.png");

    ballY = (Gdx.graphics.getHeight()/2) - 100;
    randomGenerator = new Random();
    goalVelocity = 4;
    distanceBetweenGoals = Gdx.graphics.getWidth() / 3;
    barriers = new Rectangle[numOfGoals];

    for (int i = 0; i < numOfGoals; i++) {
      supportX[i] = (Gdx.graphics.getWidth()/2) - 100 + i * distanceBetweenGoals;
      supportHeight[i] = randomGenerator.nextInt(600);
      barriers[i] = new Rectangle();
    }
  }
```

```
@Override
public void render () {
  int width = Gdx.graphics.getWidth();
  int height = Gdx.graphics.getHeight();
  int centreX = (width/2) - 150;
  int centreY = (height/2) - 100;

  batch.begin();
  batch.draw(background, 0, 0, width, height);

  if (gameState != 0) {
    if (Gdx.input.justTouched()) {
      velocity = -30;
    }

    for (int i = 0; i < numOfGoals; i++) {
      if (supportX[i] < -goal.getWidth()) {
        supportX[i] += numOfGoals * distanceBetweenGoals;
        supportHeight[i] = randomGenerator.nextInt(600);
      } else {
        supportX[i] -= goalVelocity;
      }

      batch.draw(goalSupport, supportX[i], 0, 200, supportHeight[i]);
      batch.draw(goal, supportX[i], supportHeight[i], 300, 300);

      //barriers[i] = new Rectangle(supportX[i], supportHeight[i], supportX[i], supportHeight[i]);
    }

    if (ballY >= (height-100)) {
      ballY = height - 100;
    }
    if (Gdx.input.isTouched()) {
      thisFootball = 1;
    } else {
      thisFootball = 0;
    }
    if (ballY > 0 || velocity < 0) {
      velocity++;ballY -= velocity;
    }
  } else {
    if (Gdx.input.justTouched()) {
      gameState = 1;
    }
  }

  batch.draw(footballs[thisFootball], centreX, ballY, 300, 200);
  batch.end();
```

Copilot

Alright, let's dive in. I see a potential issue in your code that might be causing the crash.

It looks like you're not updating the Rectangle instances in your barriers array. You have a commented-out line that could be key:

```
// barriers[i] = new Rectangle(supportX[i], supportHeight[i], supportX[i], supportHeight[i]);
```

Instead, it should be something like this:

```
barriers[i].set(supportX[i], supportHeight[i], goal.getWidth(), goal.getHeight());
```

This way, you update the position and size of each barrier based on the current position of the goal supports.

Uncomment this line and modify it as shown. Try running your code again after making this change.

Hope that helps! Let me know how it goes.

You

That made no difference, it still keeps crashing, usually when it crashes it's because of an infinite loop or something similar

Copilot

Infinite loops are like the digital equivalent of getting lost in a maze with no way out. Fun times, eh?

Alright, let's hunt this down. An infinite loop might be occurring due to some logic that keeps running endlessly. I see a potential problem in your render method. The loop could be tied to the goalVelocity and updating the supportX positions. If the logic never meets the condition to exit the loop, it crashes.

Check your for loop:

```
for (int i = 0; i < numOfGoals; i++) {
    if (supportX[i] < -goal.getWidth()) {
        supportX[i] += numOfGoals * distanceBetweenGoals;
        supportHeight[i] = randomGenerator.nextInt(600);
    } else {
        supportX[i] -= goalVelocity;
    }
}
```

If this logic continually fails to reset the supportX[i] positions correctly, it might trap your game in an infinite loop. Add some logging to see if the condition is being met as you expect:

```
Gdx.app.log("Loop Debug", "supportX[" + i + "]: " + supportX[i]);
```

This might give you insight into where the infinite loop is happening. Worth a shot!

You

How do I find the logs then?

Copilot

Good call! To see the logs, you need to set up your Android logging properly. Here's a quick way to do it:

1. **Add logging statement:** Place Gdx.app.log("Loop Debug", "supportX[" + i + "]: " + supportX[i]); within your render loop where you want to check your values.
2. **Run your app:** Launch your app as you normally would. The logs will start streaming.
3. **Check the logcat:** In Android Studio, go to the "Logcat" tab at the bottom of your screen. Make sure your connected device or emulator is selected from the dropdown menu.
4. **Filter your logs:** Use the search bar in Logcat to filter by the tag you used in your log statement ("Loop Debug"). This way, you can quickly see your debug information without all the noise from other logs.

By doing this, you'll get a better sense of what's going on inside your render loop and hopefully spot the infinite loop or other issues causing your app to crash.

Good luck, and let me know how it goes!

You

It's still not working, I have no idea what's wrong, there's nothing in the logs either

Copilot

Sounds like we're missing something small but critical. Try adding try-catch blocks around the code that might be causing the crashes to catch any exceptions that are not currently being logged. Something like:

```
try {
  // Your code here
} catch (Exception e) {
  Gdx.app.log("Error", e.getMessage());
}
```

Wrap sections like this around your render() method, for loops, and anywhere else where the crash could be occurring. This way, even if there's a hidden bug, it should show up in your logs.

If you still don't see anything, another option could be narrowing it down by commenting out larger sections to see which part is causing the crash, then re-adding smaller parts to hone in on the issue. This iterative approach can help pinpoint the culprit.

Keep me updated! Let's crack this.

You

This is my error:

```
FATAL EXCEPTION: main Process: com.has_akh.flying_football, PID: 8809 java.lang.RuntimeException:
Unable to start activity ComponentInfo{com.has_akh.flying_football/com.has_akh.flying_football.AndroidLauncher}:
java.lang.NullPointerException: Attempt to invoke interface method 'int
com.badlogic.gdx.graphics.GL20.glCreateShader(int)' on a null object reference
at android.app.ActivityThread.performLaunchActivity(ActivityThread.java:4048)
at android.app.ActivityThread.handleLaunchActivity(ActivityThread.java:4235)
at android.app.servertransaction.LaunchActivityItem.execute(LaunchActivityItem.java:112)
at android.app.servertransaction.TransactionExecutor.executeNonLifecycleItem(TransactionExecutor.java:174)
at android.app.servertransaction.TransactionExecutor.executeTransactionItems(TransactionExecutor.java:109)
at android.app.servertransaction.TransactionExecutor.execute(TransactionExecutor.java:81)
at android.app.ActivityThread$H.handleMessage(ActivityThread.java:2636)
at android.os.Handler.dispatchMessage(Handler.java:107)
at android.os.Looper.loopOnce(Looper.java:232)
at android.os.Looper.loop(Looper.java:317)
at android.app.ActivityThread.main(ActivityThread.java:8705)
at java.lang.reflect.Method.invoke(Native Method)
at com.android.internal.os.RuntimeInit$MethodAndArgsCaller.run(RuntimeInit.java:580)
at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:886)
Caused by: java.lang.NullPointerException: Attempt to invoke interface method
'int com.badlogic.gdx.graphics.GL20.glCreateShader(int)' on a null object reference
at com.badlogic.gdx.graphics.glutils.ShaderProgram.loadShader(ShaderProgram.java:204)
at com.badlogic.gdx.graphics.glutils.ShaderProgram.compileShaders(ShaderProgram.java:183)
at com.badlogic.gdx.graphics.glutils.ShaderProgram.(ShaderProgram.java:166)
at
com.badlogic.gdx.graphics.glutils.ImmediateModeRenderer20.createDefaultShader(ImmediateModeRenderer20.java:24
1)
at com.badlogic.gdx.graphics.glutils.ImmediateModeRenderer20.(ImmediateModeRenderer20.java:55)
at com.badlogic.gdx.graphics.glutils.ShapeRenderer.(ShapeRenderer.java:116)
at com.badlogic.gdx.graphics.glutils.ShapeRenderer.(ShapeRenderer.java:111)
at com.badlogic.gdx.graphics.glutils.ShapeRenderer.(ShapeRenderer.java:107)
at com.has_akh.flying_football.MyGdxGame.(MyGdxGame.java:16)
at com.has_akh.flying_football.AndroidLauncher.onCreate(AndroidLauncher.java:14)
at android.app.Activity.performCreate(Activity.java:9002)
at android.app.Activity.performCreate(Activity.java:8980)
at android.app.Instrumentation.callActivityOnCreate(Instrumentation.java:1526)
at android.app.ActivityThread.performLaunchActivity(ActivityThread.java:4030)
... 13 more
```

Copilot

The error indicates a NullPointerException when trying to invoke the glCreateShader method. This suggests the GL20 interface object is not properly initialized when your app starts.

The problem lies in the ShapeRenderer initialization in your create() method. It seems like the ShapeRenderer is being created before the OpenGL context is fully set up. To fix this, delay the creation of the ShapeRenderer object until after the OpenGL context is initialized:

First, remove this line from your variable declarations:

```
ShapeRenderer shapes = new ShapeRenderer();
```

Then, move it to your create() method after Gdx.gl is available:

```
@Override
public void create () {
    batch = new SpriteBatch();
    background = new Texture("Background.jpg");
    footballs = new Texture[2];
    goalSupport = new Texture("GoalSupport.png");
    goal = new Texture("Goal.png");
    footballs[0] = new Texture("Football.png");
    footballs[1] = new Texture("Football2.png");
    ballY = (Gdx.graphics.getHeight()/2) - 100;
    randomGenerator = new Random();
    goalVelocity = 4;
    distanceBetweenGoals = Gdx.graphics.getWidth() / 3;
    barriers = new Rectangle[numOfGoals];
    for (int i = 0; i < numOfGoals; i++) {
        supportX[i] = (Gdx.graphics.getWidth()/2) - 100 + i * distanceBetweenGoals;
        supportHeight[i] = randomGenerator.nextInt(600);
        barriers[i] = new Rectangle();
    }
    shapes = new ShapeRenderer(); // Initialize ShapeRenderer here
}
```

By ensuring that ShapeRenderer is initialized after the OpenGL context, you should avoid the NullPointerException.

Give this a shot and let's see if your app stays on its feet this time!

You

(Now Saturday, Oct 26)
I have the android manifest file here but I want to change the logo, I currently hae an image called logo.png stored in my assests folder:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
  <uses-feature android:glEsVersion="0x00020000" android:required="true" />
  <application
    android:allowBackup="true"
    android:fullBackupContent="true"
    android:icon="@drawable/ic_launcher"
    android:isGame="true"
    android:appCategory="game"
    android:label="@string/app_name"
    android:theme="@style/AppTheme"
    tools:ignore="UnusedAttribute">
    <activity
      android:name="com.has_akh.flying_football.AndroidLauncher"
      android:label="@string/app_name"
      android:screenOrientation="landscape"
      android:configChanges="keyboard|keyboardHidden|navigation|orientation|screenSize|screenLayout"
      android:exported="true">
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>
  </application>
</manifest>
```

Copilot

Got it. You'll need to move the logo.png image to the res/drawable folder. Then, update your Android Manifest file to reference the new logo:

1. Move logo.png to res/drawable folder.

2. Update the android:icon attribute in your AndroidManifest.xml to:

```
android:icon="@drawable/logo"
```

Your manifest should look like this:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
     xmlns:tools="http://schemas.android.com/tools">
  <uses-feature android:glEsVersion="0x00020000" android:required="true" />
  <application
    android:allowBackup="true"
    android:fullBackupContent="true"
    android:icon="@drawable/logo"
    android:isGame="true"
    android:appCategory="game"
    android:label="@string/app_name"
    android:theme="@style/AppTheme"
    tools:ignore="UnusedAttribute">
    <activity
      android:name="com.has_akh.flying_football.AndroidLauncher"
      android:label="@string/app_name"
      android:screenOrientation="landscape"
      android:configChanges="keyboard|keyboardHidden|navigation|orientation|screenSize|screenLayout"
      android:exported="true">
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>
  </application>
</manifest>
```

This should set your app’s icon to your logo.png image. Check it out and see if it works!

References

1. github.com 2. github.com 3. github.com 4. github.com

You

(Now Sunday, Oct 27)
THis is my layout structure where I've got different logo sizes for different purposes such as small icon, large icon, etc. [Uploaded Image Expired]

Copilot

Great, your structure is spot-on for supporting multiple screen densities. To update your app icon with the various logo sizes, ensure you’ve placed the logo.png images in their corresponding drawable folders like so:

- res/drawable-mdpi/logo.png
- res/drawable-hdpi/logo.png
- res/drawable-xhdpi/logo.png
- res/drawable-xxhdpi/logo.png
- res/drawable-xxxhdpi/logo.png

Then, reference the icon in your AndroidManifest.xml using:

```
    android:icon="@drawable/logo"
```

This way, your app uses the appropriate logo size for different screen resolutions. Your app should now display the correct icon on different devices. Let’s see how it looks!

You

It's showing teh old logo still, do I need to update my logo_foreground.xml:

```
<vector xmlns:android="http://schemas.android.com/apk/res/android"
  android:width="108dp"
  android:height="108dp"
  android:viewportWidth="108"
  android:viewportHeight="108">
  <path android:pathData="M22,48.667l2.987,0l0,10.667l-2.987,0z"
    android:fillColor="#000000"
    android:strokeColor="#00000000"
    android:fillAlpha="1"/>
  <path android:pathData="M26.907,52.72l2.987,0l0,6.613l-2.987,0z"
    android:fillColor="#000000"
    android:strokeColor="#00000000"
    android:fillAlpha="1"/>
  <path android:pathData="M26.907,48.667l2.987,0l0,2.56l-2.987,0z"
    android:fillColor="#000000"
    android:strokeColor="#00000000"
    android:fillAlpha="1"/>
  <path android:pathData="M31.813,48.667L31.813,52.72 31.813,55.067 31.813,56.767 31.813,59.333l2.992,0 2.117,0c1.654,0 2.998,-1.481 2.998,-3.307 0,-1.826 -1.344,-3.307 -2.998,-3.307l-2.117,0L34.805,48.667ZM34.805,55.067l1.269,0c0.469,0 0.848,0.384 0.848,0.853 0,0.469 -0.379,0.847 -0.848,0.847l-1.269,0z"
    android:fillColor="#000000"
    android:strokeColor="#00000000"
    android:fillAlpha="1"/>
  <path android:pathData="m44.192,48.667c-1.65,0 -2.992,1.481 -2.992,3.307 0,0.023 -0,0.044 0,0.067 0,0.004 -0,0.009 0,0.013l0,3.893c-0.001,0.027 0,0.053 0,0.08 0,1.826 1.341,3.307 2.992,3.307l2.112,0 0.247,0 2.739,0 0.247,0 2.112,0c1.651,0 2.992,-1.481 2.992,-3.307 0,-1.826 -1.341,-3.307 -2.992,-3.307l-1.199,0 -0.48,0 -2.372,0l0,2.347l2.372,0 0.48,0 0.353,0c0.468,0 0.846,0.384 0.846,0.853 0,0.469 -0.378,0.847 -0.846,0.847l-0.833,0 -0.433,0 -0.247,0 -2.739,0 -0.247,0 -0.433,0 -0.833,0c-0.459,0 -0.832,-0.363 -0.846,-0.821l0,-3.893 0,-0.013c0.021,-0.45 0.391,-0.807 0.846,-0.807l0.833,0 0.433,0 1.293,0l0,0.007L54.207,51.24L54.207,48.667l-4.917,0 -1.692,0 -1.293,0 -2.112,0z"
    android:fillColor="#e74a45"
    android:strokeColor="#00000000"
    android:fillAlpha="1"/>
  <path android:pathData="M56.133,48.667L56.133,51.238l5.406,0 1.859,0 1.105,0 0.43,0 0.827,0c0.452,0 0.82,0.356 0.84,0.806l0,0.013 0,3.891c-0.014,0.456 -0.384,0.819 -0.84,0.819l-0.827,0 -0.43,0 -1.899,0 -1.065,0 -2.442,0L59.098,52.724L56.133,52.724l0,4.044 0,1.752 0,0.813l5.406,0 1.065,0 1.899,0 2.098,0c1.639,0 2.971,-1.48 2.971,-3.305 0,-0.027 0.001,-0.053 0,-0.08L69.573,52.058c0,-0.004 -0,-0.009 0,-0.013 0,-0.022 0,-0.044 0,-0.067 0,-1.825 -1.332,-3.305 -2.971,-3.305l-2.098,0 -1.105,0l0,-0.007L56.133,48.667Z"
    android:fillColor="#e74a45"
    android:strokeColor="#00000000"
    android:fillAlpha="1"/>
  <path android:pathData="M69.572,48.667L73.72,48.667L77.787,52.733 81.853,48.667l4.147,0l-5.333,5.333 5.333,5.333L81.853,59.333L77.787,55.267 73.72,59.333l-4.147,0l5.333,-5.333z"
    android:fillColor="#e74a45"
    android:strokeColor="#00000000"/>
</vector>
```

Copilot

Updating logo_foreground.xml might be necessary if your app is using adaptive icons introduced in Android 8.0 (API level 26). But, if you just want to update the app's icon, here’s what you need to do:

1. **Make sure logo.png is in the res/mipmap folder** instead of the assets folder. You should have:

    - res/mipmap-mdpi/logo.png
    - res/mipmap-hdpi/logo.png
    - res/mipmap-xhdpi/logo.png
    - res/mipmap-xxhdpi/logo.png
    - res/mipmap-xxxhdpi/logo.png

2. **Update your AndroidManifest.xml** to use the new logo:

```
android:icon="@mipmap/logo"
```

3. **Create adaptive icons** if necessary:

    - *res/mipmap-anydpi-v26/ic_launcher.xml* for background and foreground.
    - *res/drawable/ic_launcher_foreground.xml* and *ic_launcher_background.xml* to define vector paths.

For *ic_launcher.xml*, update to:

```
<adaptive-icon xmlns:android="http://schemas.android.com/apk/res/android">
   <background android:drawable="@color/ic_launcher_background"/>
   <foreground android:drawable="@drawable/logo"/>
</adaptive-icon>
```

If your old icon still shows, **clean and rebuild your project** in Android Studio to ensure all changes are applied.

Try these steps out and your new logo should be good to go!

You

I don't want to use ic_launcher I want to use logo

Copilot

Got it, you want to sidestep ic_launcher altogether.

1. **First**, ensure your logo.png files are in the res/mipmap directories for various screen densities:

    - res/mipmap-mdpi/logo.png
    - res/mipmap-hdpi/logo.png
    - res/mipmap-xhdpi/logo.png
    - res/mipmap-xxhdpi/logo.png
    - res/mipmap-xxxhdpi/logo.png

2. **Next**, update your AndroidManifest.xml to use logo instead of ic_launcher:

```
<application
   android:allowBackup="true"
   android:fullBackupContent="true"
   android:icon="@mipmap/logo"
   android:isGame="true"
   android:appCategory="game"
   android:label="@string/app_name"
   android:theme="@style/AppTheme"
   tools:ignore="UnusedAttribute">
```

3. **Finally**, clean and rebuild your project in Android Studio to ensure all changes are picked up.

Try these steps and your new logo should be front and center!

You

I want to keep my logo in drawable but use logo.png instead of ic_launcher.png

Copilot

I hear you. Let's make sure your logo.png is shining from the drawable folder:

1. **Ensure logo.png is placed** in:
   - *res/drawable/logo.png*
2. **Update your AndroidManifest.xml** to point to the logo.png in the drawable directory:

```
<application
   android:allowBackup="true"
   android:fullBackupContent="true"
   android:icon="@drawable/logo"
   android:isGame="true"
   android:appCategory="game"
   android:label="@string/app_name"
   android:theme="@style/AppTheme"
   tools:ignore="UnusedAttribute">
```

Now, clean and rebuild your project in Android Studio, and you should see your shiny new logo in action. Let’s give it a whirl and see how it looks!

You

I still have the same structure as before, I want to use logo.png and logo.xml to get my new logo as the icon in various circumstance [Uploaded Image Expired]

Copilot

Got it! Let’s make sure your logo fits perfectly in different scenarios. Here's how you can incorporate logo.png and logo.xml:

1. **Create the necessary drawables**: Ensure your logo images are in res/drawable for different densities:
   - res/drawable-mdpi/logo.png
   - res/drawable-hdpi/logo.png
   - res/drawable-xhdpi/logo.png
   - res/drawable-xxhdpi/logo.png
   - res/drawable-xxxhdpi/logo.png
2. **Setup logo.xml**: Make sure your logo.xml correctly references your logo:

```
<layer-list xmlns:android="http://schemas.android.com/apk/res/android">
   <item android:drawable="@android:color/white"/>
   <item>
      <bitmap
         android:gravity="center"
         android:src="@drawable/logo"/>
   </item>
</layer-list>
```

3. **Update AndroidManifest.xml**: Point to your logo.xml as the icon:

```
<application
   android:allowBackup="true"
   android:fullBackupContent="true"
   android:icon="@drawable/logo"
   android:isGame="true"
   android:appCategory="game"
   android:label="@string/app_name"
   android:theme="@style/AppTheme"
   tools:ignore="UnusedAttribute">
```

4. **Add adaptive icons** (optional): If you need adaptive icons, set up:

- res/mipmap-anydpi-v26/logo.xml:

```
<adaptive-icon xmlns:android="http://schemas.android.com/apk/res/android">
    <background android:drawable="@drawable/logo"/>
    <foreground android:drawable="@drawable/logo_foreground"/>
</adaptive-icon>
```

5. **Rebuild your project**: Clean and rebuild your project in Android Studio to apply the changes.

This should update your app's icon using logo.png in different contexts and ensure consistency across all devices. Give it a shot!

References

github.com

You

How would I update logo_foreground.xml, I previously showed what values are in there

Copilot

You've got it. To make sure your logo_foreground.xml correctly references your new logo.png, here's what you can do:

1. **Update logo_foreground.xml** to use your new logo:

```
<vector xmlns:android="http://schemas.android.com/apk/res/android"
    android:width="108dp"
    android:height="108dp"
    android:viewportWidth="108"
    android:viewportHeight="108">
    <path
        android:pathData="M0,0h108v108H0z"
        android:fillColor="@android:color/transparent"/>
    <path
        android:pathData="M22,48.667l2.987,0l0,10.667l-2.987,0z"
        android:fillColor="#000000"
        android:strokeColor="#00000000"
        android:fillAlpha="1"/>
    <!-- Adjust other pathData elements as necessary -->
</vector>
```

Make sure to adjust the pathData elements to fit your specific design requirements, but keep the vector dimensions consistent with your desired logo size. This ensures that your app will properly display the new logo as the foreground icon.

After making these changes, clean and rebuild your project in Android Studio to see the updated logo.

Give it a try and see how it looks!

References

1. github.com 2. github.com

You

How do I adjust the pathData elements for all of the logo.png files I just showed in my strcuture?

Copilot

To adjust the pathData elements for your logo_foreground.xml to match your logo.png files across various densities, you'll need to convert your logo image into a vector format, which will generate the necessary path data.

Here’s a general approach:

1. **Use an Image to Vector Converter**: Tools like Adobe Illustrator, Inkscape, or online converters (like Vector Magic) can help.
   - Import your logo.png into the tool.
   - Use the tool to trace the bitmap and create a vector graphic.
   - Export the result as an SVG file.
2. **Convert SVG to Android Vector Drawable**:
   - You can use online tools like SVG2VectorDrawable to convert your SVG into an Android-compatible vector drawable XML file.
   - Replace your current path data with the one generated.
3. **Replace Path Data in logo_foreground.xml**:
   - Open the converted vector drawable XML file.
   - Copy the path data and other necessary attributes.
   - Replace the corresponding sections in your logo_foreground.xml.

Here’s a simplified example of what your logo_foreground.xml might look like after updating:

```
<vector xmlns:android="http://schemas.android.com/apk/res/android"
    android:width="108dp"
    android:height="108dp"
    android:viewportWidth="108"
    android:viewportHeight="108">
    <path
        android:pathData="M10,10L50,50L90,10Z"
        android:fillColor="#000000"/>
    <!-- Add more path data here as needed -->
</vector>
```

Clean and rebuild your project in Android Studio to see the updated icons in action.

Give this a go, and you should see your new logo perfectly integrated as a vector drawable.

References

1. github.com 2. github.com

You

Can I have the full example for my use case?

Copilot

Absolutely. Let’s put this all together for you.

1. **Ensure logo.png files are in the res/drawable folders for various screen densities**:
   - res/drawable-mdpi/logo.png
   - res/drawable-hdpi/logo.png
   - res/drawable-xhdpi/logo.png
   - res/drawable-xxhdpi/logo.png
   - res/drawable-xxxhdpi/logo.png
2. **Create logo.xml to reference the new logo**:

```
<layer-list xmlns:android="http://schemas.android.com/apk/res/android">
   <item android:drawable="@android:color/white"/>
   <item>
      <bitmap
         android:gravity="center"
         android:src="@drawable/logo"/>
   </item>
</layer-list>
```

3. **Update your AndroidManifest.xml to use logo.xml**:

```
<application
   android:allowBackup="true"
   android:fullBackupContent="true"
   android:icon="@drawable/logo"
   android:isGame="true"
   android:appCategory="game"
   android:label="@string/app_name"
   android:theme="@style/AppTheme"
   tools:ignore="UnusedAttribute">
```

4. **Create logo_foreground.xml for adaptive icons**:

```
<vector xmlns:android="http://schemas.android.com/apk/res/android"
   android:width="108dp"
   android:height="108dp"
   android:viewportWidth="108"
   android:viewportHeight="108">
   <!-- Add pathData elements here after converting your logo.png to SVG and then to vector drawable -->
   <path
      android:pathData="M22,48.667l2.987,0l0,10.667l-2.987,0z"
      android:fillColor="#000000"
      android:strokeColor="#00000000"
      android:fillAlpha="1"/>
   <path
      android:pathData="M26.907,52.72l2.987,0l0,6.613l-2.987,0z"
      android:fillColor="#000000"
      android:strokeColor="#00000000"
      android:fillAlpha="1"/>
   <path
      android:pathData="M26.907,48.667l2.987,0l0,2.56l-2.987,0z"
      android:fillColor="#000000"
      android:strokeColor="#00000000"
      android:fillAlpha="1"/>
   <path
      android:pathData="M31.813,48.667L31.813,52.72 31.813,55.067 31.813,56.767 31.813,59.333l2.992,0
2.117,0c1.654,0 2.998,-1.481 2.998,-3.307 0,-1.826 -1.344,-3.307 -2.998,-3.307l-
2.117,0L34.805,48.667ZM34.805,55.067l1.269,0c0.469,0 0.848,0.384 0.848,0.853 0,0.469 -0.379,0.847 -0.848,0.847l-
1.269,0z"
      android:fillColor="#000000"
      android:strokeColor="#00000000"
      android:fillAlpha="1"/>
   <!-- Continue adding other paths as necessary -->
</vector>
```

5. **(Optional) Create an adaptive icon logo.xml for anydpi**:

```
<adaptive-icon xmlns:android="http://schemas.android.com/apk/res/android">
   <background android:drawable="@drawable/logo"/>
   <foreground android:drawable="@drawable/logo_foreground"/>
</adaptive-icon>
```

6. **Rebuild your project**: Clean and rebuild your project in Android Studio to apply all the changes.

This should cover all bases for using logo.png and logo.xml effectively. Hope this does the trick!

References

github.com

You

(Now Monday, Oct 28)
When I put the following code in my logo.xml, I get an error saying that logo isn't allowed to reference itself:

```
<layer-list xmlns:android="http://schemas.android.com/apk/res/android">
  <item android:drawable="@android:color/white"/>
  <item> <bitmap android:gravity="center" android:src="@drawable/logo"/>
  </item>
</layer-list>
```

Copilot

This error makes sense because the logo.xml is referencing itself, which causes a loop. Let's correct this by not directly referencing logo.xml within itself. Instead, you can reference the logo.png directly.

Here's a solution:

1. Ensure logo.png files are in the res/drawable folder.
2. Update logo.xml to:

```
<layer-list xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:drawable="@android:color/white"/>
    <item>
        <bitmap
            android:gravity="center"
            android:src="@drawable/logo"/>
    </item>
</layer-list>
```

Note: Ensure there is no loop by not referencing logo.xml within itself.

Update your AndroidManifest.xml to point to logo.xml:

```
<application
    android:allowBackup="true"
    android:fullBackupContent="true"
    android:icon="@drawable/logo"
    android:isGame="true"
    android:appCategory="game"
    android:label="@string/app_name"
    android:theme="@style/AppTheme"
    tools:ignore="UnusedAttribute">
```

This setup ensures no self-referencing loop and appropriately uses logo.png. Now, clean and rebuild your project. This should solve the referencing issue.

References

github.com

You

(Now Tuesday, Oct 29)
I'm still getting teh following error:
Drawable logo should not reference itself
More... (Ctrl+F1)
Inspection info: There should be no cycles in resource definitions as this can lead to runtime exceptions.
Issue id: Resource

Cycle Vendor: Android Open Source
Project Contact: https://groups.google.com/g/lint-dev
Feedback: https://issuetracker.google.com/issues/new?component=192708

Copilot

It sounds like you're encountering a **resource cycle** issue in your Android project. This error occurs when a drawable resource references itself, either directly or indirectly, which can lead to runtime exceptions.

Here are a few steps you can take to resolve this issue:

1. **Check Drawable References**: Ensure that none of your drawable XML files reference themselves or create a loop in references. For example, if you have a drawable file my_image.xml, it should not reference my_image.xml within itself.
2. **Clean and Rebuild Project**: Sometimes, simply cleaning and rebuilding your project can resolve resource-related issues. You can do this by going to Build > Clean Project and then Build > Rebuild Project in Android Studio.
3. **Check for Typos**: Make sure there are no typos in your drawable file names and references. Even a small typo can cause this error.
4. **Update Dependencies**: Ensure that all your project dependencies are up to date. Sometimes, outdated dependencies can cause unexpected issues.
5. **Review Resource Definitions**: Carefully review your resource definitions to ensure there are no cycles. This includes checking for any indirect references that might create a loop.

If you've tried these steps and are still encountering the error, could you share a snippet of the problematic drawable XML file? That way, I can help you troubleshoot further.

References

1. github.com 2. stackoverflow.com

You

Where should logo.xml be and how do I then reference logo.png diretly instead?

Copilot

The logo.xml file should be placed in the res/drawable directory of your Android project. To reference logo.png directly instead, you can follow these steps:

1. Place the logo.png file in the res/drawable directory.

2. Reference logo.png in your XML layouts or other resource files using the following syntax: @drawable/logo.

For example, if you want to use logo.png in an ImageView in your layout XML file, you can do it like this:

```
<ImageView
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:src="@drawable/logo" />
```

This way, your logo.png will be directly referenced without needing to create an intermediate logo.xml file. Any other questions about your project? Let's dive deeper.

You

I want to use logo.png as the front facing icon for my app

Copilot

Got it. To set logo.png as the app icon, follow these steps:

1. **Place logo.png in the correct directory**:

    - Add your logo.png to the res/mipmap directory. You might have different subfolders here (like mipmap-hdpi, mipmap-mdpi, etc.). You should add appropriately sized versions of your logo to each of these folders to support different screen resolutions.

2. **Update your AndroidManifest.xml**:
    - Open your AndroidManifest.xml file, and within the <application> tag, set the android:icon attribute to reference your logo. It should look something like this:

```
<application
  android:icon="@mipmap/logo"
  ... >
```

3. **Clean and Rebuild Your Project**:
    - Go to Build > Clean Project and then Build > Rebuild Project in Android Studio to ensure the changes are applied.

That should set logo.png as your app icon. Anything else you want to tweak or explore in your app?

You

How do I add the version number to the app and in which file?

Copilot

To add the version number to your Android app, you need to edit the build.gradle file. Here's how:

1. **Open build.gradle (Module: app)**: This file is located under the "Gradle Scripts" section in your Android Studio project view.
2. **Find the defaultConfig block**: In this block, you can set the version information. You'll see something like this:

```
android {
  compileSdkVersion 30
  defaultConfig {
    applicationId "com.your.package.name"
    minSdkVersion 16
    targetSdkVersion 30
    versionCode 1
    versionName "1.0"
  }
  ...
}
```

3. **Set versionCode and versionName**:
    - versionCode: Increment this integer value for every release. This is used internally and not shown to users.
    - versionName: Set this to a string that represents the version name shown to users.

For example:

```
android {
  compileSdkVersion 30
  defaultConfig {
    applicationId "com.your.package.name"
    minSdkVersion 16
    targetSdkVersion 30
    versionCode 2
    versionName "1.1"
  }
  ...
}
```

4. **Sync Your Project**: After making these changes, click "Sync Now" in the bar that appears at the top of the file.

By updating the versionCode and versionName in your build.gradle (Module: app) file, you control your app's versioning for both internal tracking and user-visible updates.

Good luck, and let me know if you need any more assistance!

You

(Now Wednesday, Oct 30)
How can I reword and shorten this request for an IT service ticket:

I previously logged service request 1023584 about this issue, I left the Trust on the 20th of September as my internship ended but have since returned on a part-time basis, this is what the issue was about - "Taskbar notification icons always showing on loginI want specific icons on my taskbar to be in the hidden section, the section that can be shown by a popup arrow on the right hand side of the taskbar. However, every time I shutdown and then log back on or restart and log back on, the icons are showing again. Shortly after logging on, I go to Taskbar settings then Select which icons appear on the taskbar (in the Notification area section) then I toggle the setting Always show all icons in the notification area off. I do this almost every time I log on, but still the setting is automatically turned on when I log on again on another day or another time on the same day. Can you check the default settings for my device and make sure this is always toggled off as it's really annoying, the icons take up extra space on my taskbar and it looks really cramped then."It was recommended that I would have to get a new laptop as my profile was corrupt. Could it ideally be a Windows 11 laptop as part of BCFT pilot as I do a lot of technical work and Windows 11 features would be useful to say the least

Copilot

Sure thing, here's a more concise version of your IT service ticket request:

Service Request: 1023584

I previously left the Trust on September 20th after my internship ended but have returned part-time. My issue is with taskbar notification icons always showing upon login. I need specific icons to be hidden, but they reappear every time I log in, despite toggling the setting to hide them.

It was recommended that I get a new laptop as my profile was corrupt. Ideally, this would be a Windows 11 laptop as part of the BCFT pilot, as I perform a lot of technical work and would benefit from Windows 11 features.

Thank you for your assistance.

This should cover all the necessary details succinctly.

You

How about a one line summary of the request

Copilot

Service Request Summary: Issue with taskbar notification icons resetting on login; requesting a new Windows 11 laptop due to a corrupt profile.

How's that?

You

Could you reduce that further?

Copilot

Request: Taskbar icons reset on login; need a Windows 11 laptop due to profile corruption.

That’s as tight as it gets!

You

I'm planning a Microsoft SharePoint tutorial for work colleagues from all different disciplines of the organisation, my topic for this one is Pages and Lists, how can I plan the tutorial based on this planning format:

- Team Sites cannot be created by workers, refer to IT department for this
- However, they can be added to, edited and permissions managed
- News can be created and shared (discussed later on)
- SharePoint sites include Pages, Lists and Document libraries
- Showcase of Children's System Support site
- Further sections on Pages, Lists and Documents
- For checklist items, mark them as done after shown it in tutorial
- Adding to Sites
- If listed as an owner of site, can add to it
- If press new button in top left, can add lists, document libraries, pages, spaces, news posts, news links and also apps
- Won't go into apps or spaces in this but rest covered later (different videos in series)
- More About Sites
- Sidebar is navigation bar
- CSS one contains the following tabs:
- Home (always there)
- Pages (always there)
- Admin Guides (Optional page, subjugated under Pages)
- AI Searches (Optional Label - created by me, subjugated under Pages)
- EHM Guides [AI Search] (Optional Page - created by me, under AI Searches)
- LCS Guides [AI Search] (Optional Page - created by me, under AI Searches)
- Team Documents (always there)
- Team Calendar (usually there)
- Tasks (sometimes there)
- Partner Access (Optional Page, List)
- Recycle Bin (always there, can't remove)
- Edit button – allows to edit the navigation bar
- If any items are subjugated as sub links, will appear in an initially hidden drop-down menu from the item they are a sub link of, the arrow on the far left can be clicked to open this up
- Can Edit the navigation bar: If click edit button can do the following:
- Can drag and drop navigation items – reorder them and nest them under other navigation items
- Choose to cancel any changes, click cancel button in bottom right
- Choose to save any changes, click save button in bottom left
- Enable/Disable site navigation audience targeting, use toggle near bottom
- Click on 3 dots on any items, this brings up a menu with the following options (individual to each item):
    - Edit (always there)
    - Remove (always there)
    - Move up (always there unless item already at top)
    - Move down (always there unless item already at bottom)
    - Make sub link (usually there – not there if item is Home or if cannot be further subjugated due to there already being 3 levels of items)
    - Promote sub link (only there if an item is already a sub link to push it out a level)
- Add a navigation item, shows up as a horizontal line with plus sign in middle, brings up new pop-up side menu with different options:
- A drop-down to choose whether the item should be a link or simply a label (non-linked navigation item)
- Address (only available if link is chosen above), text box to paste a webpage address (URL) in to where you want the link to go
- Display name, text box to type in the what should be displayed as the name of the item in the navigation bar
- Open in new tab checkbox (only available if link is chosen above), tick this to allow link to open in new tab automatically, otherwise leave unchecked
- Ok and cancel buttons (self-explanatory)

Copilot

Here’s a structured plan for your SharePoint tutorial on Pages and Lists:

1. **Introduction**
    - Briefly explain the purpose and benefits of SharePoint.
    - Mention that team sites can’t be created by workers but can be edited, added to, and have permissions managed.
2. **SharePoint Overview**
    - Highlight that SharePoint sites include Pages, Lists, and Document Libraries.
    - Showcase the Children's System Support site as an example.
3. **Pages Section**
    - Explain what Pages are and their role in SharePoint.
    - Demonstrate how to add new pages, edit existing ones, and manage permissions.
    - Checklist:
        - Adding Pages.
        - Editing Pages.
        - Managing Permissions.
4. **Lists Section**
    - Define Lists and their importance.
    - Show how to create new lists, add items, and customize columns.
    - Checklist:
        - Creating Lists.
        - Adding Items to Lists.
        - Customizing Columns.
5. **Adding to Sites**
    - Explain that owners can add new elements to the site.
    - Walk through the process of adding lists, document libraries, pages, news posts, and news links.
    - Mention that apps and spaces are covered in a different video series.
    - Checklist:
        - Adding Lists.
        - Adding Document Libraries.
        - Adding Pages.
        - Adding News Posts.
        - Adding News Links.
6. **Site Navigation**
    - Explain the sidebar navigation bar and its components.
    - Walk through how to edit the navigation bar:
        - Adding navigation items.
        - Rearranging navigation items.
        - Creating sub-links.
    - Checklist:
        - Editing Navigation Bar.
        - Adding Navigation Items.
        - Reordering Navigation Items.
        - Creating Sub-links.
7. **Custom Sections**
    - Discuss optional pages like Admin Guides, AI Searches, and Partner Access.
    - Showcase how to enable/disable audience targeting for site navigation.
8. **Practical Demonstration**
    - Live demo of adding pages and lists.
    - Walk through the process of managing taskbar settings.
9. **Q&A Session**
    - Open the floor for any questions and clarifications.
10. **Conclusion**
    - Recap the key points covered.
    - Provide resources for further learning and exploration.

With this plan, your colleagues will get a comprehensive understanding of how to effectively use Pages and Lists in SharePoint. Ready to lead them through this digital journey?

You

I've talked about some of the other sections in previous tutorials, this one I only want to focus on Pages and Lists, teh plan is for what I'm talking about while shoing the demonstration

Copilot

Got it. Let’s streamline your plan to focus specifically on Pages and Lists for your SharePoint tutorial:

1. **Introduction**
   - Briefly explain the purpose of this tutorial: "Today, we'll be diving deep into Pages and Lists in SharePoint."
2. **Pages Section**
   - **What Are Pages?**
     - Explain the concept and purpose of Pages in SharePoint.
   - **Creating and Managing Pages**
     - Demonstrate how to create a new page:
       - Go to the site.
       - Click on "New" and select "Page."
       - Choose a page template.
     - **Editing Pages**
       - Show how to edit an existing page:
         - Open the page.
         - Click "Edit" in the top-right corner.
         - Make changes and save.
     - **Permissions**
       - Explain how to manage permissions for Pages:
         - Go to the page.
         - Click on "Share" or "Permissions."
         - Adjust settings accordingly.
     - **Checklist:**
       - Create a new page.
       - Edit an existing page.
       - Manage page permissions.
3. **Lists Section**
   - **What Are Lists?**
     - Define Lists and explain their importance.
   - **Creating and Customizing Lists**
     - Demonstrate how to create a new list:
       - Go to the site.
       - Click on "New" and select "List."
       - Name the list and choose a template.
     - **Adding Items to Lists**
       - Show how to add items to a list.
     - **Customizing Columns**
       - Explain how to customize columns in a list.
     - **Permissions**
       - Discuss how to manage permissions for Lists:
         - Go to the list settings.
         - Adjust permissions.
     - **Checklist:**
       - Create a new list.
       - Add items to the list.
       - Customize list columns.
       - Manage list permissions.

4. **Live Demonstration**
   - Perform a live demo showing the creation and management of Pages and Lists:
     - Create a new page.
     - Edit and save the page.
     - Create a new list.
     - Add items to the list.
     - Customize columns in the list.
5. **Q&A Session**
   - Open the floor for questions specific to Pages and Lists.
6. **Conclusion**
   - Recap the key points covered in the tutorial.
   - Provide additional resources for further learning.

By focusing on these key areas, your colleagues will gain a thorough understanding of how to effectively use Pages and Lists in SharePoint. Good luck with your tutorial!

You

How about for News and Discussion Boards?

Copilot

Alright, let’s focus on planning a tutorial around SharePoint’s News and Discussion Boards.

1. **Introduction**
   - Briefly state the purpose of the tutorial: "Today, we're exploring News and Discussion Boards in SharePoint."
2. **News Section**
   - **What Is SharePoint News?**
     - Explain the concept and benefits of using News in SharePoint to keep everyone informed.
   - **Creating News Posts**
     - Demonstrate how to create a new News post:
       - Navigate to the home page or News web part.
       - Click on "Add" or "New" and select "News post."
       - Choose a template and start adding content.
   - **Publishing and Sharing News**
     - Show how to publish the news post:
       - Add a title, text, images, and other web parts.
       - Click "Post" to publish.
     - Explain how to share News posts with the team:
       - Use the "Share" button.
       - Discuss other sharing options, like email and Teams.
   - **Checklist:**
     - Create a new News post.
     - Add content to the News post.
     - Publish and share the News post.
   - **Managing News**
     - Show how to edit or delete News posts:
       - Navigate to the post.
       - Click on "Edit" or the ellipsis (three dots) and select "Delete."
3. **Discussion Boards Section**
   - **What Are Discussion Boards?**
     - Define Discussion Boards and explain their importance for collaboration.
   - **Creating Discussion Boards**
     - Demonstrate how to create a new Discussion Board:
       - Navigate to the site contents.
       - Click on "New" and select "App."
       - Choose "Discussion Board" and name it.
   - **Posting and Replying to Discussions**
     - Show how to post a new discussion:
       - Go to the Discussion Board.
       - Click on "New discussion."
       - Add a subject and body, then click "Save."

        - Explain how to reply to discussions:
            - Open a discussion.
            - Click on "Reply," add your response, and click "Save."
    - **Managing Discussions**
        - Discuss how to manage and moderate discussions:
            - Edit or delete discussions.
            - Use settings to manage permissions and notifications.
    - **Checklist:**
        - Create a new Discussion Board.
        - Post a new discussion.
        - Reply to a discussion.
        - Manage and moderate discussions.
4. **Live Demonstration**
    - Perform a live demo showing the creation and management of News posts and Discussion Boards:
        - Create and publish a News post.
        - Create a new Discussion Board.
        - Post and reply to discussions.
5. **Q&A Session**
    - Open the floor for questions specific to News and Discussion Boards.
6. **Conclusion**
    - Recap the key points covered in the tutorial.
    - Provide additional resources for further learning.

With this plan, your colleagues will get a clear and practical understanding of how to use News and Discussion Boards in SharePoint. Ready to bring them up to speed?

You

I have an ellipse in java, how can I convert it to a Circle while keeping teh similar dimensions?

Copilot

Alright, converting an Ellipse to a Circle in Java while maintaining similar dimensions is straightforward. Here’s the approach:

1. **Find the Diameter**: Use the shortest dimension of the ellipse (either the width or height) as the diameter of the circle.

2. **Create the Circle**: Use the calculated diameter to create a circle. If your ellipse is wider than it is tall, use the height; if it’s taller, use the width.

Here’s a quick example:

```
import java.awt.geom.Ellipse2D;
import java.awt.geom.Rectangle2D;
import java.awt.geom.Ellipse2D.Double;

public class EllipseToCircle {
    public static void main(String[] args) {
        // Original Ellipse
        Ellipse2D ellipse = new Ellipse2D.Double(10, 20, 100, 50); // x, y, width, height

        // Find the diameter (shortest dimension)
        double diameter = Math.min(ellipse.getWidth(), ellipse.getHeight());

        // Create a Circle (use Ellipse2D for simplicity)
        Ellipse2D circle = new Ellipse2D.Double(ellipse.getX(), ellipse.getY(), diameter, diameter);

        System.out.println("Original Ellipse: " + ellipse);
        System.out.println("Converted Circle: " + circle);
    }
}
```

This code snippet converts the ellipse to a circle, keeping the position the same and using the smaller dimension as the diameter. Simple and effective! Give it a try and see how it works out.

You

I want to use a Circle class to reprsent teh converted circle

Copilot

Got it! Let's create a Circle class and convert the ellipse to a circle using that class.

Here's an example:

1. **Create the Circle class**:

```
public class Circle {
    private double x;
    private double y;
    private double radius;

    public Circle(double x, double y, double radius) {
        this.x = x;
        this.y = y;
        this.radius = radius;
    }

    // Getters and Setters
    public double getX() {
        return x;
    }

    public void setX(double x) {
        this.x = x;
    }

    public double getY() {
        return y;
    }

    public void setY(double y) {
        this.y = y;
    }

    public double getRadius() {
        return radius;
    }

    public void setRadius(double radius) {
        this.radius = radius;
    }

    @Override
    public String toString() {
        return "Circle{" + "x=" + x + ", y=" + y + ", radius=" + radius + '}';
    }
}
```

2. **Convert Ellipse to Circle**: Here's how you can convert an Ellipse2D object to a Circle using the Circle class:

```
import java.awt.geom.Ellipse2D;

public class EllipseToCircle {
    public static void main(String[] args) {
        // Original Ellipse
        Ellipse2D ellipse = new Ellipse2D.Double(10, 20, 100, 50); // x, y, width, height

        // Find the diameter (shortest dimension)
        double diameter = Math.min(ellipse.getWidth(), ellipse.getHeight());

        // Create a Circle
        Circle circle = new Circle(ellipse.getX(), ellipse.getY(), diameter / 2);

        System.out.println("Original Ellipse: " + ellipse);
        System.out.println("Converted Circle: " + circle);
    }
}
```

This code does the following:

1. Defines a Circle class with properties x, y, and radius.
2. Converts the ellipse to a circle by using the smallest dimension as the diameter.
3. Instantiates a Circle object with the appropriate position and radius.

This way, you maintain similar dimensions while representing the converted shape as a Circle object. Try it out and see how it works!

References

1. github.com 2. github.com 3. github.com

You

(Now Thursday, Oct 31)
I'm trying to perform an exercise for my coursework involving hash tables in computer science. I've been given a primary hash function (h1(x) = x mod 11) and a secondary hash function (h2(x) = (x mod 3) + 1) but both result in a collision. I've found this on Reddit but I need to reference a paper that contains more information about it: H = (H1(k) + i*H2(k))%n

Copilot

I found a paper titled **"Collision Resolution Techniques in Hash Table: A Review"** by Ahmed Dalhatu Yusuf, Saleh Abdullahi, Moussa Mahamat Boukar, and Salisu Ibrahim Yusuf. This paper provides an in-depth analysis of various collision resolution techniques, including the use of secondary hash functions.

You can reference this paper for more information on the formula **H = (H1(k) + i*H2(k))%n** and other collision resolution strategies.

Would you like help with anything else related to your coursework?

References

academia.edu

Conversation by Fifa Utuber with Copilot on Friday, Oct 25 2024