**PictureInPicture**

hsize = width \*(int)swPercent;

vsize = height \* (int)shPercent;

xOff = (int)xOffPercent;

yOff = (int)yOffPercent;

hsize = width;

vsize = height;

xOff = xOffset;

yOff = yOffset;

unit == UnitsIn.screen\_percentage



T

Y

T

T

Y

Y

T

Y

float justifiedCenter = (sw \* 0.5f) - (hsize \* 0.5f);

hloc =(int)(justifiedCenter - xOff);

int justifiedRight = (sw - hsize);

hloc = (justifiedRight - xOff);

hloc = xOff;

hAlignment.center

hAlignment.right

hAlignment.left

horAlign

verAlign

T

T

T

Y

Y

Y

float justifiedMiddle = (sh \* 0.5f) - (vsize \* 0.5f);

vloc = (int)(justifiedMiddle - yOff);

int justifiedTop = sh - vsize;

vloc = (justifiedTop - (yOff));

GetComponent<Camera>().pixelRect = new Rect(hloc,vloc,hsize,vsize);

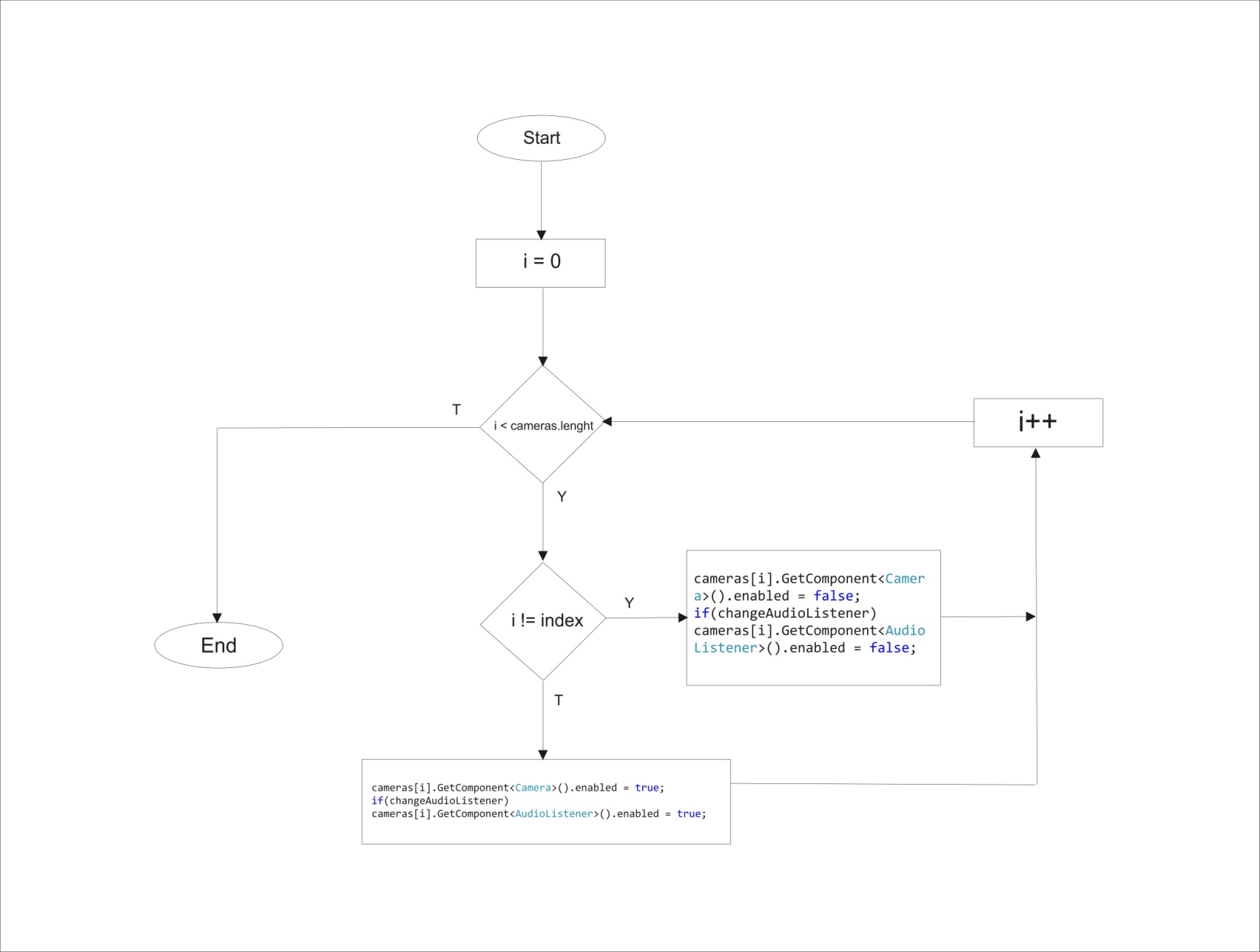
vloc = yOff;

vAlignment.middle

vAlignment.bottom

vAlignment.top

**CameraSwitch**

****

**ScreenTexture**

RectTransform frameTransform = frameGUI.GetComponent<RectTransform>();

photoGUI.SetActive(false);

int sw = Screen.width;

int sh = Screen.height;

Rect framing = frameTransform.rect;

Vector2 pivot = frameTransform.pivot;

Vector2 origin = frameTransform.anchorMin;

origin.x \*= sw;

origin.y \*= sh;

float xOffset = pivot.x \* framing.width;

origin.x += xOffset;

float yOffset = pivot.y \* framing.height;

origin.y += yOffset;

framing.x += origin.x;

framing.y += origin.y;

int textWidth = (int)framing.width;

int textHeight = (int)framing.height;

Texture2D texture =

new Texture2D(textWidth,textHeight);

yield return new WaitForEndOfFrame();

Vector3 photoScale = new Vector3 (framing.width \* ratio, framing.height \* ratio, 1);

texture.ReadPixels(framing, 0, 0);

texture.Apply();

photoGUI.SetActive (true);

photoGUI.GetComponent<RectTransform>().localScale = photoScale; photoGUI.GetComponent<RawImage>().texture = texture;