

In [1]:

```
import cv2
import numpy as np
import imutils
```

In [2]:

```
image = cv2.imread("./data/more_shapes.png", cv2.IMREAD_COLOR)
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
ret, mask = cv2.threshold(gray, 50, 255, cv2.THRESH_BINARY)
cv2.imshow("mask", mask)
cv2.waitKey()
cv2.destroyAllWindows()
```

Bounding box

In [3]:

```
# find external contours in the image
cnts, _ = cv2.findContours(mask, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

# clone the original image
clone = image.copy()

# loop over the contours
for c in cnts:
    # fit a bounding box to the contour
    (x, y, w, h) = cv2.boundingRect(c)
    cv2.rectangle(clone, (x, y), (x + w, y + h), (0, 255, 0), 2)

# show the output image
cv2.imshow("Bounding Boxes", clone)
cv2.waitKey()
cv2.destroyAllWindows()
```

Oriented Bounding Box

In [5]:



```
# clone the original image
clone = image.copy()

# loop over the contours
for c in cnts:
    # fit a rotated bounding box to the contour and draw a rotated bounding box
    rect = cv2.minAreaRect(c)
    box = cv2.boxPoints(rect)
    #box = np.int0(box)
    box = np.int0(box)
    cv2.drawContours(clone, [box], -1, (0, 255, 0), 2)

# show the output image
cv2.imshow("Oriented Bounding Boxes", clone)
cv2.waitKey()
cv2.destroyAllWindows()
```

Bounding Sphere

In []:



```
# clone the original image
clone = image.copy()

# loop over the contours
for c in cnts:
    (x,y), r = cv2.minEnclosingCircle(c)
    center = (int(x), int(y))
    r = int(r)
    cv2.circle(clone, center, r, (255, 0, 0), 3)

# show the output image
cv2.imshow("Bounding Sphere", clone)
cv2.waitKey()
cv2.destroyAllWindows()
```

Convex Hull

In []:



```
# clone the original image
clone = image.copy()

# loop over the contours
for c in cnts:
    hull = cv2.convexHull(c)
    cv2.drawContours(clone, [hull], -1, (0, 0, 255), 2)

# show the output image
cv2.imshow("Convexhull", clone)
cv2.waitKey()
cv2.destroyAllWindows()
```

In []:

