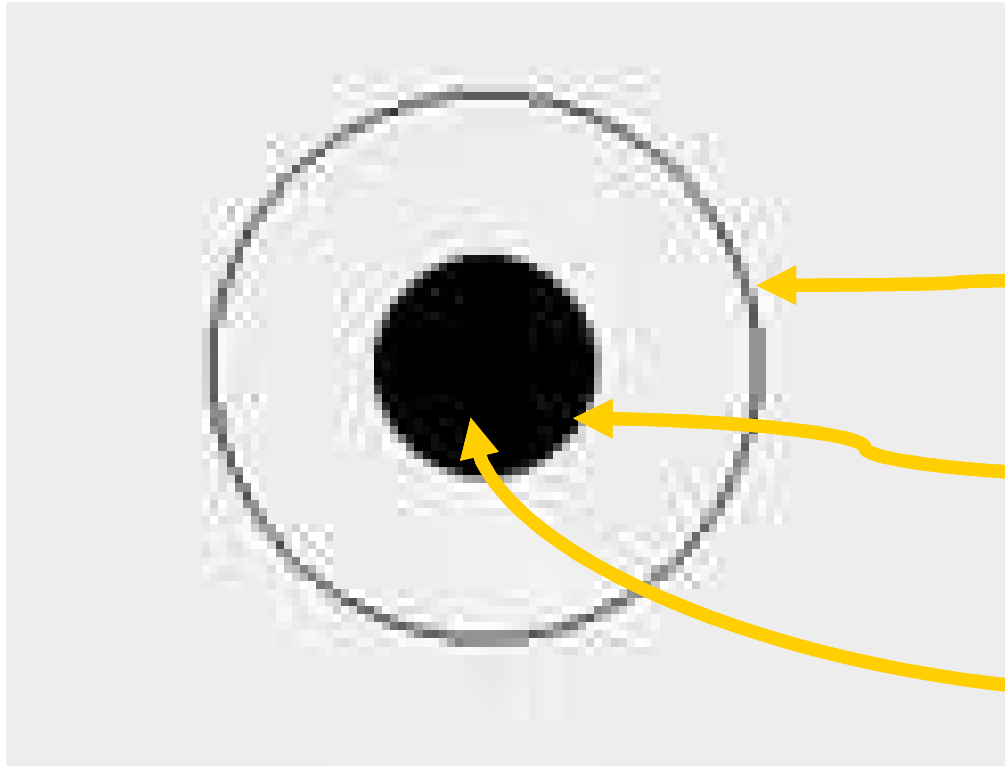


myjoystick.py

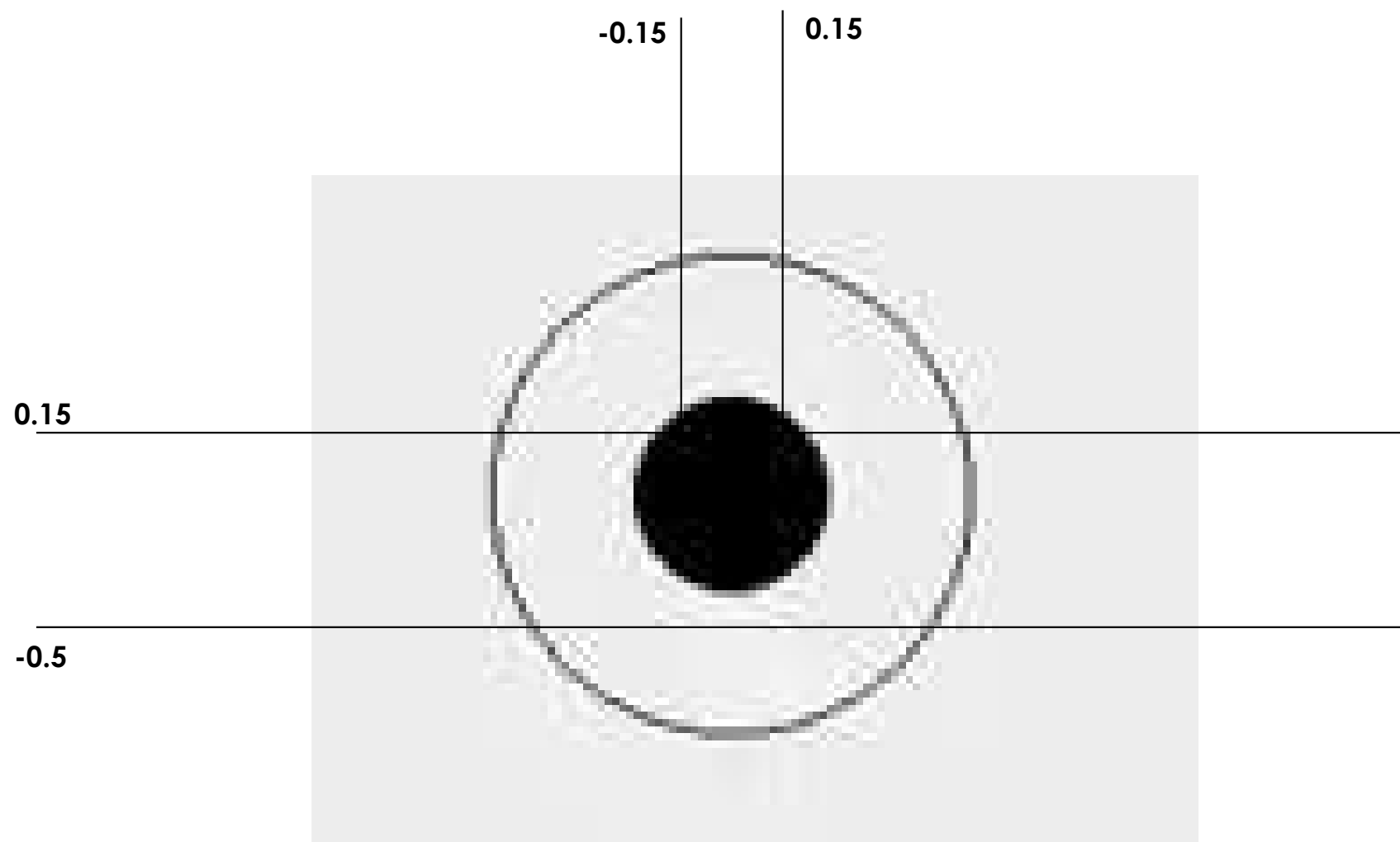


조이스틱 위젯
MyJoystick

조이스틱 손잡이 경계
_boundJoystick

조이스틱 손잡이
_centerEllips

조이스틱 중심
_center





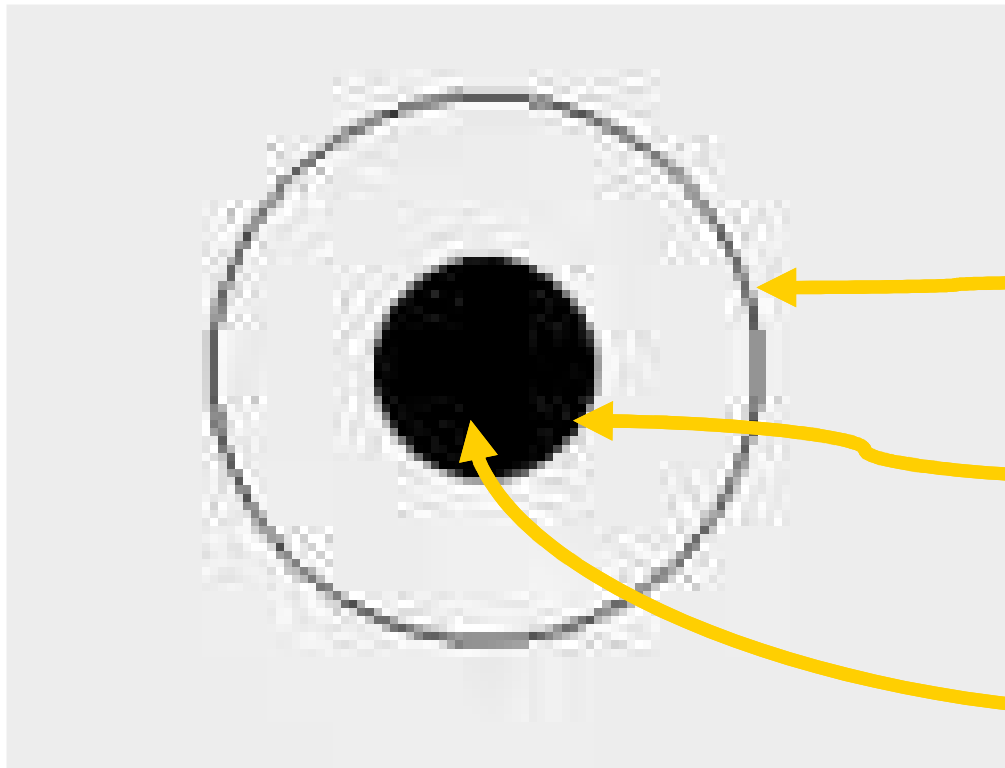
상세 분석

파이썬 조이스틱 코딩하기

✓ 파이썬 조이스틱 코딩하기

- 조이스틱 모듈 살펴보기 – `myjoystick.py`

`myjoystick.py`



조이스틱 위젯
`MyJoystick`

조이스틱 손잡이 경계
`_boundJoystick`

조이스틱 손잡이
`_centerEllips`

조이스틱 중심
`_center`

파이썬 조이스틱 코딩하기 - myjoystick.py

✓ 초기화

```
# 조이스틱 초기화 함수
def __init__(self, cbJoyPos=None, parent=None):
    super(MyJoystick, self).__init__(parent)
    self.setMinimumSize(200, 200)
    self.movingOffset = QPointF(0, 0)
    self.grabCenter = False
    self.__maxDistance = 50

    self.timer = QTimer(self)
    self.timer.setInterval(10)
    self.timer.timeout.connect(self.timeout)
    self.timer.start()
```

self.cbJoyPos = cbJoyPos

함수포인터 역할

함수명 등록

타이머 등록

주기적으로 조이스틱 손잡이 위치를 알려주는 함수

```
def timeout(self):
    sender = self.sender()
    if id(sender) == id(self.timer):
        if self.cbJoyPos != None :
            self.cbJoyPos(self.joystickPosition())
```

```
def cbJoyPos(joystickPosition):
    print(joystickPosition)
```

Callback 등록

MouseMoveEvent

self.cbJoyPos(self.joystickPosition())

조이스틱 손잡이 현재 위치

```
def joystickPosition(self):
    if not self.grabCenter:
        return (0, 0)
    normVector = QLineF(self._center(), self.movingOffset)
    currentDistance = normVector.length()
    angle = normVector.angle()

    distance = min(currentDistance / self.__maxDistance, 1.0)

    posX = math.cos(angle*math.pi/180)*distance
    posY = math.sin(angle*math.pi/180)*distance

    return (posX, posY)
```

파이썬 조이스틱 코딩하기 - myjoystick.py

✓ paintEvent

```
# 조이스틱을 그리는 함수
def paintEvent(self, event):
    painter = QPainter(self)
    bounds = QRectF(
        -self.__maxDistance,      # 50
        -self.__maxDistance,
        self.__maxDistance * 2,
        self.__maxDistance * 2
    ).translated(self._center())
    painter.drawEllipse(bounds)
    painter.setBrush(Qt.black)
    painter.drawEllipse(self._centerEllipse())
```

paintEvent는 어플리케이션에서 그림을 그리기 위해 사용되는 이벤트 핸들러

1. 위젯이 처음 생성될 때
2. **update** 메서드가 호출될 때
3. 크기가 변경될 때
4. 다른 위젯이 위에 덮일 때

paintEvent는 보통 개발자가 오버라이딩(overriding)하여 사용

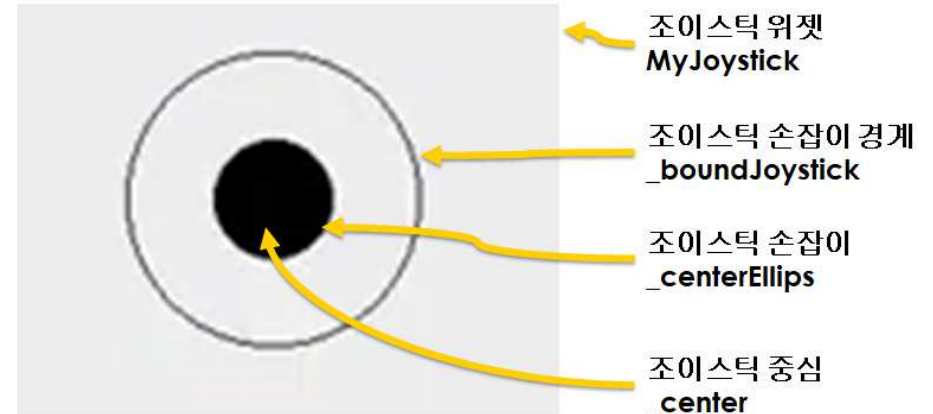
파이썬 조이스틱 코딩하기 - myjoystick.py

✓ _centerEllipse, _center, _boundJoystick

```
# 조이스틱 손잡이 영역
def _centerEllipse(self):
    if self.grabCenter:
        return QRectF(-20, -20, 40, 40).\
            translated(self.movingOffset)
    return QRectF(-20, -20, 40, 40).translated(self._center())

# 조이스틱 중심
def _center(self):
    return QPointF(self.width()/2, self.height()/2)

# 조이스틱 손잡이 움직임 경계 제한
def _boundJoystick(self, point):
    limitLine = QLineF(self._center(), point)
    if (limitLine.length() > self.__maxDistance):
        limitLine.setLength(self.__maxDistance)
    return limitLine.p2()
```



파이썬 조이스틱 코딩하기 - myjoystick.py

✓ mousePressEvent, mouseReleaseEvent, mouseMoveEvent

```
# 조이스틱 손잡이 잡는 함수
def mousePressEvent(self, ev):
    self.grabCenter = self._centerEllipse().contains(ev.pos())
    return super().mousePressEvent(ev)

# 조이스틱 손잡이 놓는 함수
def mouseReleaseEvent(self, event):
    self.grabCenter = False
    self.movingOffset = QPointF(0, 0)
    self.update()

# 조이스틱 손잡이 움직이는 함수
def mouseMoveEvent(self, event):
    if self.grabCenter:
        self.movingOffset = self._boundJoystick(event.pos())
        self.update()
    if self.cbJoyPos != None :
        self.cbJoyPos(self.joystickPosition())
```


파이썬 조이스틱 코딩하기 – joystick_pos.py

✓ joystick_pos.py - 모듈사용

```
from PyQt5.QtWidgets import *  
import sys  
from myjoystick import MyJoystick
```

```
def cbJoyPos(joystickPosition) :  
    print(joystickPosition)
```

```
# Create main application window  
app = QApplication([])  
app.setStyle(QStyleFactory.create("Cleanlooks"))  
mw = QMainWindow()  
mw.setWindowTitle('RC Car Joystick')  
mw.setGeometry(100, 100, 300, 200)
```

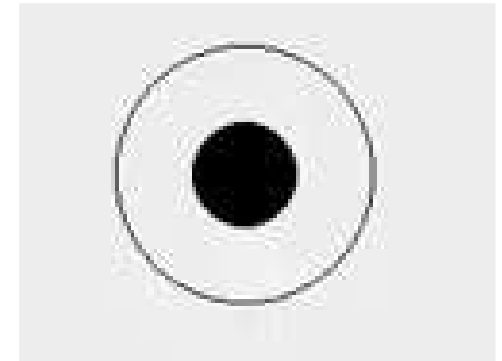
```
# Create and set widget layout  
# Main widget container  
cw = QWidget()  
ml = QGridLayout()  
cw.setLayout(ml)  
mw.setCentralWidget(cw)
```

```
# Create joystick  
joystick = MyJoystick(cbJoyPos)  
ml.addWidget(joystick,0,0)  
mw.show()
```

```
# Start Qt event loop  
sys.exit(app.exec_())
```

```
# 콜백 함수, 조이스틱 위치값을 인자로 받아 출력  
# 좌표값 X, Y축 -1에서 1 사이
```

```
# 응용프로그램 객체와 주 윈도우 실행  
# "Cleanlooks" 스타일을 설정
```



```
# 주 윈도우에 놓일 위젯 생성  
# mw 메인 윈도우의 중앙에 cw 위젯이 위치하며,  
# cw 위젯에는 ml 그리드 레이아웃이 적용되어 있습니다.
```

```
# 조이스틱 인스턴스 생성
```

```
# 윈도우를 화면에 표시
```

```
# 마우스, 키보드, 타이머 등 처리
```

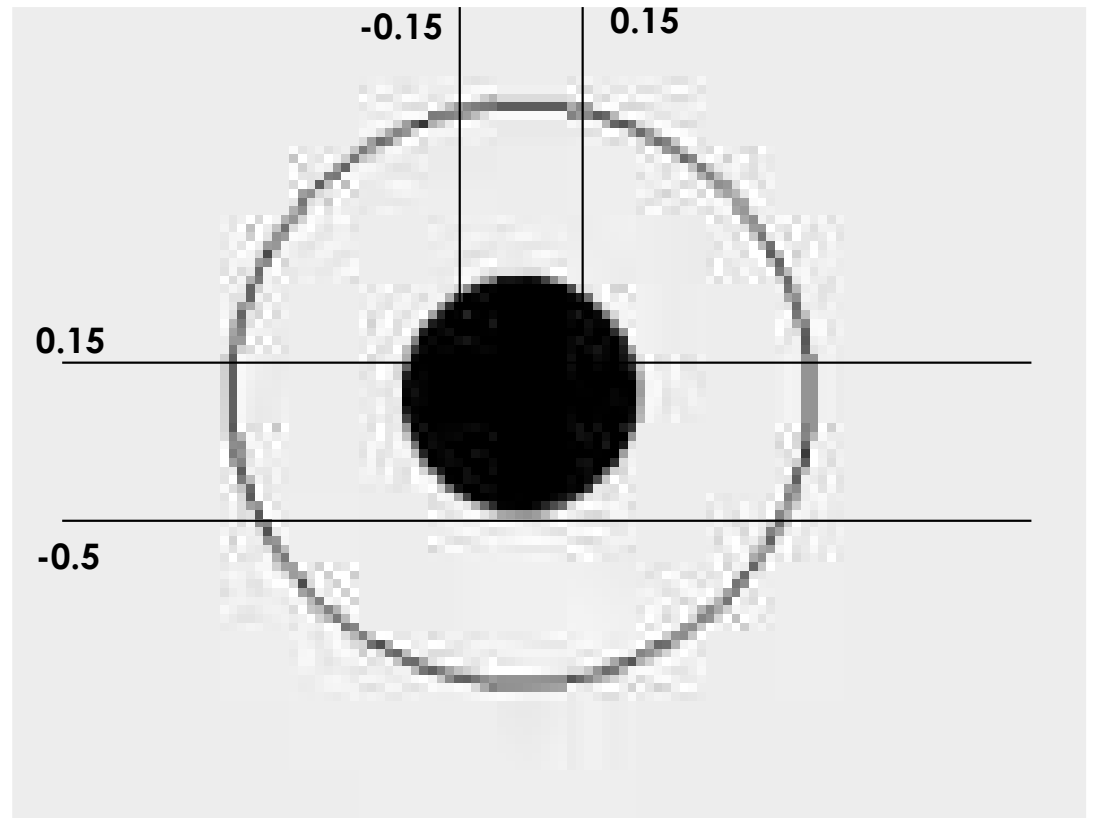
파이썬 조이스틱 코딩하기

✓ 파이썬 조이스틱 코딩하기

- 조이스틱 모듈 살펴보기 – `myjoystick.py`
- 조이스틱 테스트하기
 - RC카 방향값 얻어오기 – `joystick_dir.py`

```
# 콜백 함수, 조이스틱 위치값을 인자로 받아 출력
def cbJoyPos(joystickPosition):
    # posX, posY는 -1 에서 1 사이
    posX, posY = joystickPosition

    # 자동차 방향
    right, left = -1, -1
    if posY < -0.5:
        right, left = 1, 1
        print('brake')
    elif posY > 0.15:
        if -0.15 <= posX <= 0.15:
            right, left = 0, 0
            print('forward')
        elif posX < -0.15:
            right, left = 1, 0
            print('left')
        elif posX > 0.15:
            right, left = 0, 1
            print('right')
    else: # -0.5 <= posY <= 0.15
        print('stop driving')
```





감사합니다.
