

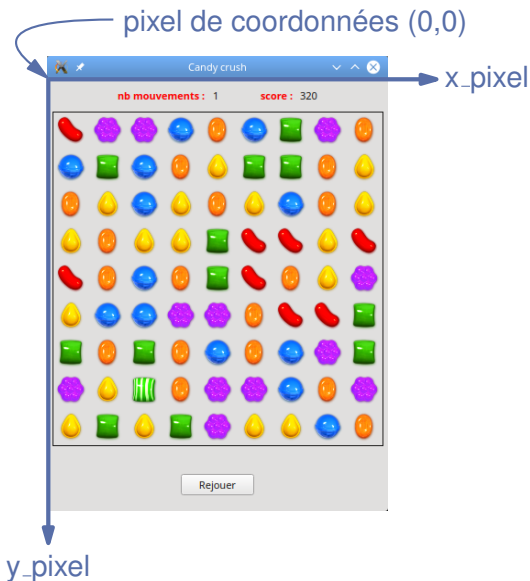
# Aide à l'utilisation de PyQt5



Ing-info — Ingénierie informatique

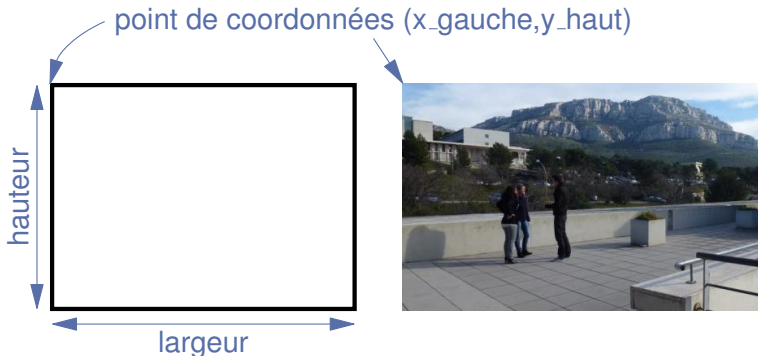
Christophe Gonzales

# Fenêtre et pixels



# Joli dessin (Qt drawing)

- ▶ Affichages  $\implies$  utilisation de Qt (pyQt5)
- ▶ Fenêtre  $\implies$  Widget Qt  $\implies$  Painter
  - ▶ `painter.drawRect(x_gauche, y_haut, largeur, hauteur)`
  - ▶ `painter.drawPixmap(x_gauche, y_haut, image)`



# Manipulation d'images (1/2)

- ▶ Image = QPixmap = objet du module QtGui
- ▶ Créer une image à partir d'un fichier `puget.png` :  
`image = QtGui.QPixmap("puget.png")`
- ▶ Modifier la taille d'une image (largeur de 60 pixels) :  
`petite_image = image.scaledToWidth(60)`
- ▶ Récupérer la largeur en pixels d'un QPixmap :  
`largeur = image.width()`
- ▶ Récupérer la hauteur en pixels d'un QPixmap :  
`hauteur = image.height()`

## Rotation d'un QPixmap :

- 1 Créer une transformation Qt :

```
transform = QtGui.QTransform()
```

- 2 Indiquer que c'est une rotation de  $\alpha$  **degrés** :

```
transform.rotate(45)
```

- 3 Le QPixmap applique la transformation et crée un nouveau QPixmap :

```
new_image = image.transformed(transform)
```



# Les goûts et les couleurs...

- ▶ Afficher un rectangle en rouge :

- 1 Créer un nouveau stylo :

```
QtGui.QPen(couleur, taille, style)
```

```
stylo = QtGui.QPen(QtCore.Qt.red, 4, QtCore.Qt.SolidLine)
```

- 2 Affecter le stylo au painter :

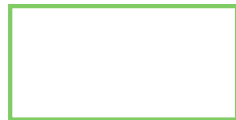
```
painter.setPen(stylo)
```

- ▶ Stylo vide : taille = 0, style = QtCore.Qt.NoPen

- ▶ Créer une couleur personnalisée :

```
couleur = QtGui.QColor(rr, gg, bb)
```

```
rr, gg, bb ∈ {0, ..., 255}
```



```
couleur = QtGui.QColor(127, 203, 100)
stylo = QtGui.QPen(couleur, 4, QtCore.Qt.SolidLine)
painter.setPen(stylo)
painter.drawRect(0, 0, 100, 50)
```

# Brosser dans le sens du poil

- Remplir un rectangle avec une couleur donnée :

- 1 Créer une brosse :

```
QtGui.QBrush(couleur, style)
```

```
brosse = QtGui.QBrush(QtCore.Qt.red, QtCore.Qt.SolidPattern)
```

- 2 Affecter la brosse au painter :

```
painter.setBrush(brosse)
```

```
stylo = QtGui.QPen(QtCore.Qt.red, 4, QtCore.Qt.SolidLine)
painter.setPen(stylo)
brosse = QtGui.QBrush(QtCore.Qt.blue, QtCore.Qt.SolidPattern)
painter.setBrush(brosse)
painter.drawRect(0, 0, 100, 50)
```



- ▶ Afficher du texte – 1ère version :

```
painter.drawText(x, y, texte)
```



Le texte est dessiné avec le stylo du painter

- ▶ Contrôler facilement la position du texte :

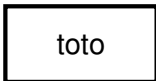
- 1 Créer un rectangle :

```
rect = QtCore.QRect(x_gauche, y_haut, largeur, hauteur)
```

- 2 Dessiner le texte en l'alignant dans le rectangle :

```
painter.drawText(rect, QtCore.Qt.AlignCenter, texte)
```

```
rect = QtCore.QRect(0, 0, 100, 50)
painter.drawRect(rect)
texte = "toto"
painter.drawText(rect, QtCore.Qt.AlignCenter, texte)
```





- ▶ Récupérer la police de caractères utilisée par le painter :

```
font = painter.font()
```

- ▶ Modifier la taille de la font :

```
font.setPointSize(12)
```

- ▶ Mettre la font en gras :

```
font.setBold(True)
```

- ▶ Affecter une font au painter :

```
painter.setFont(font)
```

- ▶ Déterminer la taille d'un texte avec une font donnée :

```
metrics = QtGui.QFontMetrics(font)
```

```
width = metrics.width(texte)
```

```
height = metrics.height()
```