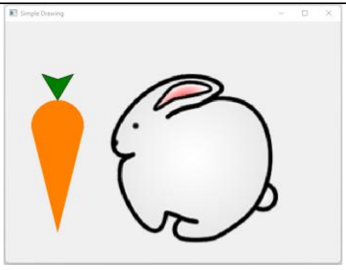
Software Engineering Lab #4
3rd February 2022
Graphics with Qt for Python

Program 4.1: Simple Drawing in Qt for Python

Source Code

```python
import sys
from PySide6.QtCore import *
from PySide6.QtWidgets import *
from PySide6.QtGui import *

class Simple_drawing_window(QWidget):
    def __init__(self):
        QWidget.__init__(self, None)
        self.setWindowTitle("Simple Drawing")
        self.rabbit = QPixmap("images/rabbit.png")

    def paintEvent(self, e):
        p = QPainter()
        p.begin(self)

        p.setPen(QColor(0, 0, 0))
        p.setBrush(QColor(0, 127, 0))
        p.drawPolygon(
            [QPoint( 70, 100), QPoint(100, 110),
            QPoint(130, 100), QPoint(100, 150),]
        )

        p.setPen(QColor(255, 127, 0))
        p.setBrush(QColor(255, 127, 0))
        p.drawPie(50, 150, 100, 100, 0, 180 * 16)

        p.drawPolygon(
            [QPoint(50, 200), QPoint(150, 200), QPoint(100, 400),]
        )

        p.drawPixmap(QRect(200, 100, 320, 320), self.rabbit)
        p.end()

def main():
    app = QApplication(sys.argv)

    w = Simple_drawing_window()
    w.show()

    return app.exec()

if __name__ == "__main__":
    sys.exit(main())
```

Output

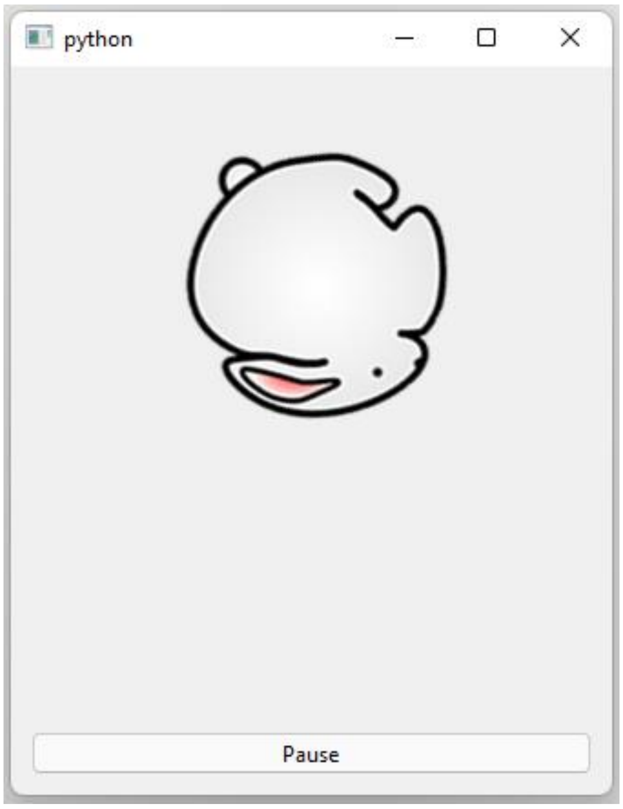


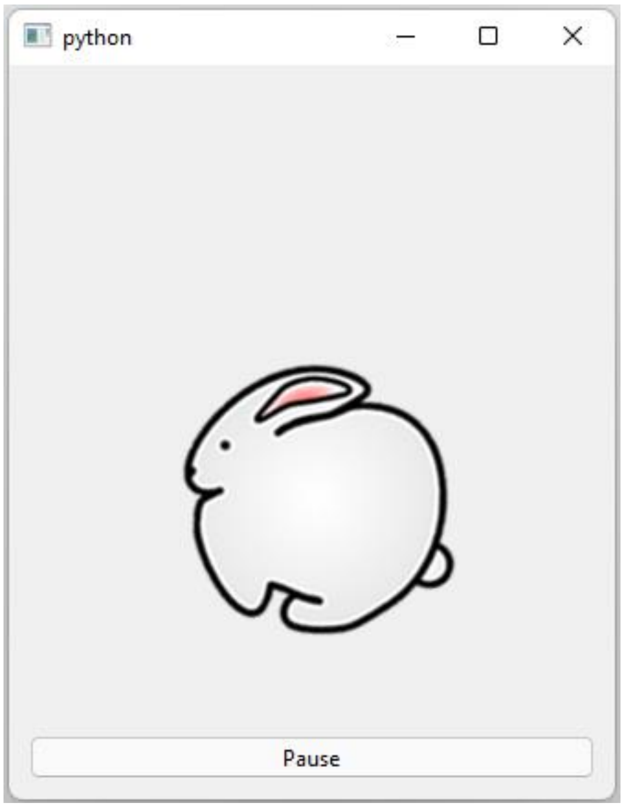
Program 4.2: Image, Animation and Sound

Source Code

```
import sys
from PySide6.QtCore import *
from PySide6.QtWidgets import *
from PySide6.QtGui import *
from PySide6.QtMultimedia import QSoundEffect

class Animation_area(QWidget):
    def __init__(self):
        QWidget.__init__(self, None)

        self.frame_no = 0
        self.images = [
            QPixmap("images/frame-" + str(i + 1) + ".png")
            for i in range(20)
        ]

        timer = QTimer(self)
        timer.timeout.connect(self.update_value)
        timer.start(75)
        #self.ps = PlaySoundThread("sounds/rabbit_jump.wav")
        self.QSE = QSoundEffect()
        self.QSE.setSource(QUrl.fromLocalFile("sounds/rabbit_jump.wav"))

    def paintEvent(self, e):
        p = QPainter()
        p.begin(self)
        p.drawPixmap(QRect(0, 0, 320, 320), self.images[self.frame_no])
        p.end()

    def update_value(self):
        self.frame_no += 1
        if self.frame_no >= 20:
            self.frame_no = 0
            self.QSE.play() #("sounds/rabbit_jump.wav")

        self.update()

class Simple_animation_window(QWidget):
    def __init__(self):
        QWidget.__init__(self, None)

        self.anim_area = Animation_area()

        layout = QVBoxLayout()
        layout.addWidget(self.anim_area)

        layout.addWidget(QPushButton("Pause"))

        self.setLayout(layout)
        self.setMinimumSize(330, 400)

def main():
    app = QApplication(sys.argv)

    w = Simple_animation_window()
    w.show()

    return app.exec()
```

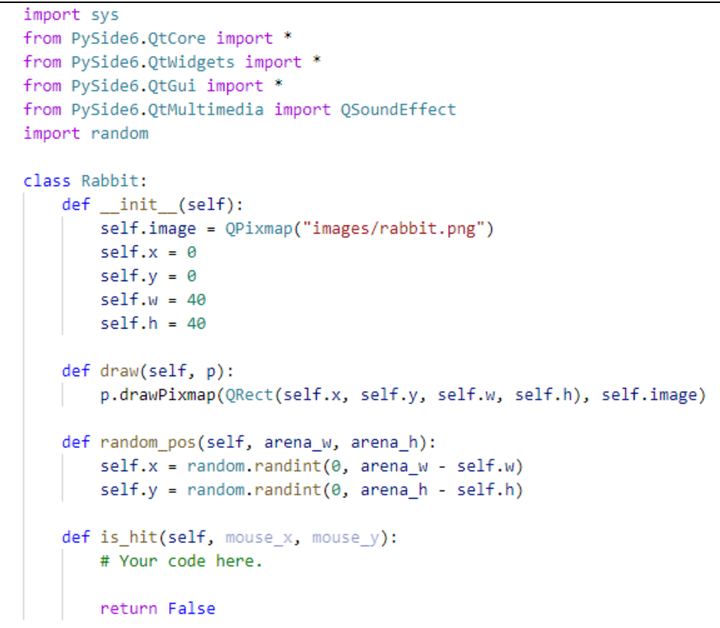
Output



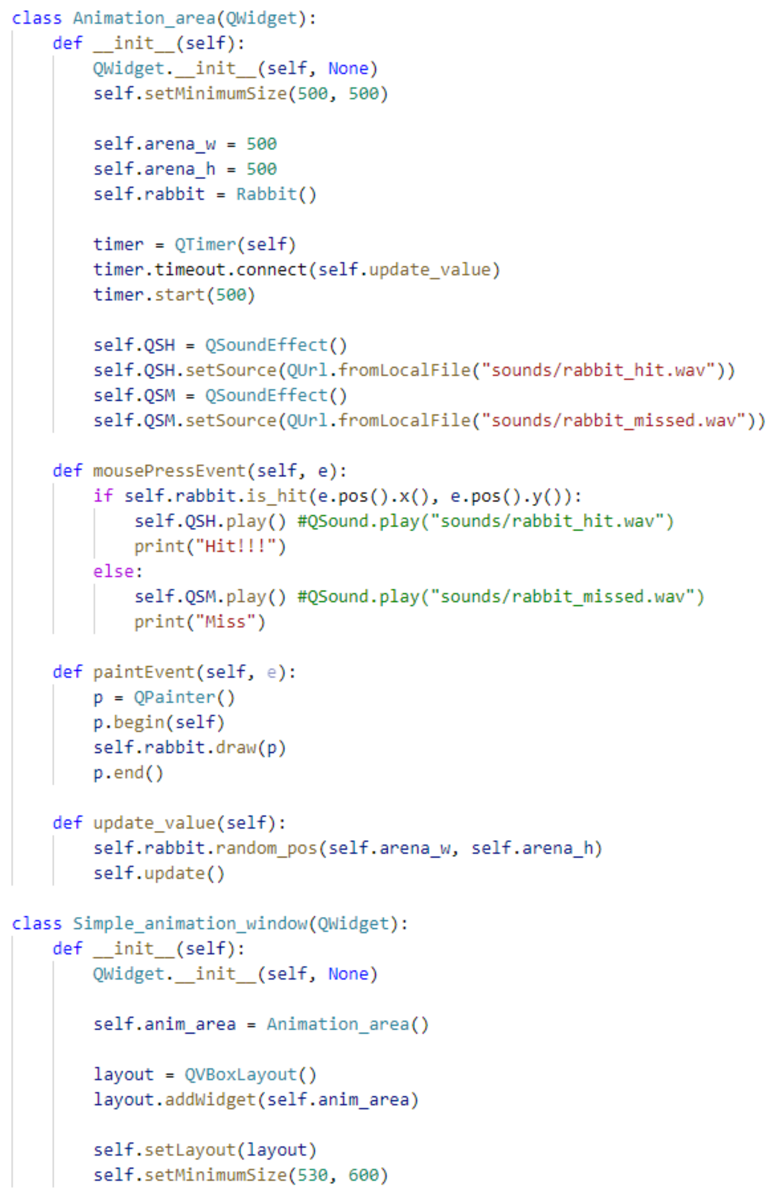


Program 4.3: More Fun with Qt

Source Code



```python
import sys
from PySide6.QtCore import *
from PySide6.QtWidgets import *
from PySide6.QtGui import *
from PySide6.QtMultimedia import QSoundEffect
import random

class Rabbit:
    def __init__(self):
        self.image = QPixmap("images/rabbit.png")
        self.x = 0
        self.y = 0
        self.w = 40
        self.h = 40

    def draw(self, p):
        p.drawPixmap(QRect(self.x, self.y, self.w, self.h), self.image)

    def random_pos(self, arena_w, arena_h):
        self.x = random.randint(0, arena_w - self.w)
        self.y = random.randint(0, arena_h - self.h)

    def is_hit(self, mouse_x, mouse_y):
        # Your code here.

        return False
```



```python
class Animation_area(QWidget):
    def __init__(self):
        QWidget.__init__(self, None)
        self.setMinimumSize(500, 500)

        self.arena_w = 500
        self.arena_h = 500
        self.rabbit = Rabbit()

        timer = QTimer(self)
        timer.timeout.connect(self.update_value)
        timer.start(500)

        self.QSH = QSoundEffect()
        self.QSH.setSource(QUrl.fromLocalFile("sounds/rabbit_hit.wav"))
        self.QSM = QSoundEffect()
        self.QSM.setSource(QUrl.fromLocalFile("sounds/rabbit_missed.wav"))

    def mousePressEvent(self, e):
        if self.rabbit.is_hit(e.pos().x(), e.pos().y()):
            self.QSH.play() #QSound.play("sounds/rabbit_hit.wav")
            print("Hit!!!")
        else:
            self.QSM.play() #QSound.play("sounds/rabbit_missed.wav")
            print("Miss")

    def paintEvent(self, e):
        p = QPainter()
        p.begin(self)
        self.rabbit.draw(p)
        p.end()

    def update_value(self):
        self.rabbit.random_pos(self.arena_w, self.arena_h)
        self.update()

class Simple_animation_window(QWidget):
    def __init__(self):
        QWidget.__init__(self, None)

        self.anim_area = Animation_area()

        layout = QVBoxLayout()
        layout.addWidget(self.anim_area)

        self.setLayout(layout)
        self.setMinimumSize(530, 600)
```

```
def main():
    app = QApplication(sys.argv)

    w = Simple_animation_window()
    w.show()

    return app.exec()

if __name__ == "__main__":
    sys.exit(main())
```

Output