```
from gamemodel import *
from graphics import *


class GameGraphics:
    def __init__(self, game):
        self.game = game

        # open the window
        self.win = GraphWin("Cannon game" , 640, 480, autoflush=False)
        self.win.setCoords(-110, -10, 110, 155)
        self.win.setBackground(color_rgb(224, 255, 255))

        # screentexts
        self.firingtext = Text(Point(0,75), 'Let the battle begin')
        self.firingtext.draw(self.win)
        self.playerscore = None
        self.waitingtext = None

        # draw the terrain
        line = Line(Point(-110,0), Point(110,0))
        line.draw(self.win)
        grass = Rectangle(Point(-110,-10), Point(110,0))
        grass.setFill(color_rgb(143, 237, 143))
        grass.draw(self.win)
        sun = Image(Point(100, 150), 'sun.png')
        sun.draw(self.win)
        cannon0 = Image(Point(-97, -3), 'cannonplayer0.png')
        cannon0.draw(self.win)
        cannon1 = Image(Point(97, -3), 'cannonplayer1.png')
        cannon1.draw(self.win)
        flagblue = Image(Point(-100, 5), 'blueflag.png')
        flagblue.draw(self.win)
        flagred = Image(Point(100, 5), 'redflag.png')
        flagred.draw(self.win)

        self.draw_cannons = [self.drawCanon(0), self.drawCanon(1)]
        self.draw_scores  = [self.drawScore(0), self.drawScore(1)]
        self.draw_projs   = [None, None]

    def drawCanon(self,playerNr):

# Importerade bilder in i spelet istället (har dock en kod för metoden nedan om du
vill se)
        cannon0 = Image(Point(-97, -3), 'cannonplayer0.png')
        cannon0.draw(self.win)
        cannon1 = Image(Point(97, -3), 'cannonplayer1.png')
        cannon1.draw(self.win)
        flagblue = Image(Point(-100, 5), 'blueflag.png')
        flagblue.draw(self.win)
        flagred = Image(Point(100, 5), 'redflag.png')
        flagred.draw(self.win)

        return None

# Här är min kod för drawCanon enligt den egentliga uppgiften:
        # player = self.game.getPlayers()[playerNr]
        # color = Player.getColor(player)
        # xPosCannon = Player.getX(player)
        # xPosComp = self.game.getCannonSize()/2

        # Position of cannon corners
        # xPosP1 = xPosCannon - (xPosComp)/2
        # yPosP1 = 0
        # xPosP2 = xPosCannon + (xPosComp)/2
        # yPosP2 = xPosComp

        # draw the cannon
        #self.cannon = Rectangle(Point(xPosP1,yPosP1), Point(xPosP2,yPosP2))
        #self.cannon.setFill(color)
        #self.cannon.draw(self.win)

        #return self.cannon

    def drawScore(self,playerNr):
        player = self.game.getPlayers()[playerNr]
        if playerNr == 0:
            compensate = 13
        else:
            compensate = -13
        text = Text(Point(Player.getX(player)+compensate,-6),'Score: ' +
str(Player.getScore(player)))
        text.draw(self.win)
        return text

    def fire(self, angle, vel):
        player = self.game.getCurrentPlayer()
        playernumber = self.game.getCurrentPlayerNumber()
        proj = player.fire(angle, vel)

        if self.waitingtext != None:
            self.waitingtext.undraw()
        if self.playerscore != None:
           self.playerscore.undraw()
        self.firingtext.undraw()
        if playernumber == 0:
            firingcolor = 'Blue is Firing'
        else:
            firingcolor = 'Red is Firing'
        self.firingtext = Text(Point(0,75), firingcolor)
        self.firingtext.draw(self.win)

        circle_X = proj.getX()
        circle_Y = proj.getY()

        if self.draw_projs[playernumber] != None:
            self.draw_projs[playernumber].undraw()

        self.circle = Circle(Point(circle_X, circle_Y), self.game.getBallSize())
        self.circle.setFill(Player.getColor(player)) #color_rgb(120, 135, 153))
        self.circle.draw(self.win)
        self.draw_projs[playernumber] = self.circle

        while proj.isMoving():
            proj.update(1/50)

            # move is a function in graphics. It moves an object dx units in x
direction and dy units in y direction
```

```
            self.circle.move(proj.getX() - circle_X, proj.getY() - circle_Y)

            circle_X = proj.getX()
            circle_Y = proj.getY()

            update(50)

        return proj

    def updateScore(self,playerNr):
        player = self.game.getPlayers()[playerNr]
        self.draw_scores[playerNr].undraw()
        self.firingtext.undraw()
        if playerNr == 0:
            compensate = 13
            if self.playerscore != None:
                self.playerscore.undraw()
            self.playerscore = Text(Point(0,80),'Blue Scored!')
            self.playerscore.draw(self.win)
        else:
            compensate = -13
            if self.playerscore != None:
                self.playerscore.undraw()
            self.playerscore = Text(Point(0,80),'Red Scored!')
            self.playerscore.draw(self.win)
        self.text = Text(Point(Player.getX(player)+compensate,-6),'Score: ' +
str(Player.getScore(player)))
        self.text.draw(self.win)
        self.draw_scores[playerNr] = self.text
        return self.text
        # update the score on the screen


    def explode(self):
        currentplayer = self.game.getCurrentPlayer()
        color = Player.getColor(currentplayer)
        otherplayer = self.game.getOtherPlayer()
        xPos = Player.getX(otherplayer)

        r = self.game.getBallSize()
        for r in range(self.game.getCannonSize()*2):
            circle = Circle(Point(xPos, 0), r)
            circle.setFill(color)
            circle.draw(self.win)
            update(50)
            circle.undraw()
            r += 0.5

    def play(self):
        while True:
            player = self.game.getCurrentPlayer()
            oldAngle,oldVel = player.getAim()
            wind = self.game.getCurrentWind()

            # InputDialog(self, angle, vel, wind) is a class in gamegraphics
            inp = InputDialog(oldAngle,oldVel,wind)
            # interact(self) is a function inside InputDialog. It runs a loop until
the user presses either the quit or fire button
            if inp.interact() == "Fire!":
                angle, vel = inp.getValues()
                inp.close()
            elif inp.interact() == "Quit":
                exit()

            player = self.game.getCurrentPlayer()
            other = self.game.getOtherPlayer()
            proj = self.fire(angle, vel)
            distance = other.projectileDistance(proj)

            if distance == 0.0:
                player.increaseScore()
                self.explode()
                self.updateScore(self.game.getCurrentPlayerNumber())
                self.game.newRound()

            self.game.nextPlayer()
            self.firingtext.undraw()
            playernumber = self.game.getCurrentPlayerNumber()
            if playernumber == 0:
                text = 'Waiting for Blue...'
            else:
                text = 'Waiting for Red...'
            if self.waitingtext != None:
                self.waitingtext.undraw()
            self.waitingtext = Text(Point(0,75), text)
            self.waitingtext.draw(self.win)


class InputDialog:
    def __init__ (self, angle, vel, wind):
        self.win = win = GraphWin("Fire", 200, 300)
        win.setCoords(0,4.5,4,.5)
        win.setBackground(color_rgb(255,255,224))

        Text(Point(1,1), "Angle").draw(win)
        self.angle = Entry(Point(3,1), 5).draw(win)
        self.angle.setFill(color_rgb(240,247,255))
        self.angle.setText(str(angle))

        Text(Point(1,2), "Velocity").draw(win)
        self.vel = Entry(Point(3,2), 5).draw(win)
        self.vel.setFill(color_rgb(240,247,255))
        self.vel.setText(str(vel))

        Text(Point(1,3), "Wind").draw(win)
        self.height = Text(Point(3,3), 5).draw(win)
        self.height.setText("{0:.2f}".format(wind))

        self.fire = Button(win, Point(1,4), 1.25, .5, "Fire!")
        self.fire.activate()
        self.quit = Button(win, Point(3,4), 1.25, .5, "Quit")
        self.quit.activate()

    def interact(self):
        while True:
            pt = self.win.getMouse()
            if self.quit.clicked(pt):
                return "Quit"
            if self.fire.clicked(pt):
                return "Fire!"

    def getValues(self):
        a = float(self.angle.getText())
        v = float(self.vel.getText())
        return a,v

    def close(self):
        self.win.close()


class Button:

    def __init__(self, win, center, width, height, label):
        w,h = width/2.0, height/2.0
        x,y = center.getX(), center.getY()
        self.xmax, self.xmin = x+w, x-w
        self.ymax, self.ymin = y+h, y-h
        p1 = Point(self.xmin, self.ymin)
        p2 = Point(self.xmax, self.ymax)
        self.rect = Rectangle(p1,p2)
        self.rect.setFill('lightgray')
        self.rect.draw(win)
        self.label = Text(center, label)
        self.label.draw(win)
        self.deactivate()

    def clicked(self, p):
        return self.active and \
               self.xmin <= p.getX() <= self.xmax and \
               self.ymin <= p.getY() <= self.ymax

    def getLabel(self):
        return self.label.getText()

    def activate(self):
        self.label.setFill('black')
        self.rect.setWidth(2)
        self.active = 1

    def deactivate(self):
        self.label.setFill('darkgrey')
        self.rect.setWidth(1)
        self.active = 0


GameGraphics(Game(11,3)).play()
```