Game() {  
 setVisible(true);  
 firstTimer.start();  
 setBackground(Color.*white*);  
 addKeyListener(this);  
 setFocusable(true);  
  
  
 cactus.add(new Cactus(1000, commonCactusY));  
 for (int i = 0; i < 20; i++) {  
 cactus.add(new Cactus(cactus.getLast().getxPos() + rand.nextInt(3000) + 200, commonCactusY));  
 }  
  
  
}

Konstruktor, tworzymy pierwsze przeszkody

public void gameReset() {  
 mMeter.x = 0;  
 firstTimer.start();  
 cactus.clear();  
 cactus.add(new Cactus(1000, commonCactusY));  
 for (int i = 0; i < 20; i++) {  
 cactus.add(new Cactus(cactus.getLast().getxPos() + rand.nextInt(3000) + 200, commonCactusY));  
 }  
 setInGame(true);  
}

Metoda do zrestartowania gry   
  
public void checkCollisions() {  
  
 Rectangle r3 = mainDino.getBounds();  
  
 for (Cactus cactus : cactus) {  
  
 Rectangle r2 = cactus.getBounds();  
  
 if (r3.intersects(r2)) {  
  
 setInGame(false);  
 }  
 }  
}

Sprawdzanie kolizji z obiektami   
public void deleteCactus() {  
 for (Iterator<Cactus> cactusa = cactus.iterator(); cactusa.hasNext(); ) {  
 Cactus next = cactusa.next();  
  
 if (next.getxPos() < 0) {  
 cactusa.remove();  
 }  
 }  
}

Usuwanie kaktusów poza obszarem

@Override  
public void keyPressed(KeyEvent e) {  
  
  
 if (e.getKeyCode() == KeyEvent.*VK\_SPACE*) {  
  
 if (!mainDino.isStartJump() && mainDino.getUpdown())  
 mainDino.dinoMovePlease(this.dinoSpeed);  
  
 }  
 if (e.getKeyCode() == KeyEvent.*VK\_R*) {  
 if (!inGame) {  
 gameReset();  
 }  
 }  
  
  
}

Zachowanie programu po wciśnięciu danego klawisza

public void doDrawings(Graphics g) {  
 Graphics2D g2d = (Graphics2D) g;  
 g2d.setColor(Color.*gray*);  
 g2d.fillRect(0, 800, 1600, 50);  
  
 //g2d.fillRect(mainDino.getxPos(), mainDino.getyPos(), 60, 60);  
  
 g2d.drawImage(Dino.*getDino*(), mainDino.getxPos(), mainDino.getyPos() + 10, null);  
  
  
 for (Cactus x : cactus  
 ) {  
 // g2d.fillRect(x.getxPos(),x.getyPos(),50,50);  
 g2d.drawImage(Cactus.*getCactus*(), x.getxPos(), x.getyPos(), null);  
  
 }  
  
 if (!inGame) {  
 g2d.setColor(Color.*RED*);  
 g2d.setFont(new Font("TimesRoman", Font.*PLAIN*, 150));  
 g2d.drawString("GAME OVER", 350, 500);  
 g2d.setFont(new Font("TimesRoman", Font.*PLAIN*, 30));  
 g2d.drawString("Press R to continue", 400, 550);  
 }  
 g2d.setFont(new Font("TimesRoman", Font.*PLAIN*, 30));  
 g2d.setColor(Color.*yellow*);  
 g2d.drawString("Distance: " + mMeter.x, 1000, 30);  
}

Rysowanie wszystkich obiektów

@Override  
public void actionPerformed(ActionEvent e) {  
 if (inGame) {  
 mMeter.x += dinoSpeed;  
  
 if (mainDino.isStartJump()) {  
 if (mainDino.getUpdown() && mainDino.getyPos() > 500) {  
 mainDino.setyPos(mainDino.getyPos() - mainDino.getDy());  
 } else if (mainDino.getyPos() == 500 && mainDino.getUpdown()) {  
 mainDino.setUpDown(false);  
 mainDino.setDy(-2 \* this.dinoSpeed);  
  
 } else if (!mainDino.getUpdown() && mainDino.getyPos() != 740) {  
 mainDino.setyPos(mainDino.getyPos() - mainDino.getDy());  
 } else if (!mainDino.getUpdown() && mainDino.getyPos() == 740) {  
 mainDino.setUpDown(true);  
 mainDino.setStartJump(false);  
 mainDino.setDy(this.dinoSpeed);  
 }  
 }  
  
 for (Cactus x : cactus  
 ) {  
 x.setxPos(-10);  
  
 }  
 checkCollisions();  
 deleteCactus();  
  
 if (cactus.size() < 20)  
 cactus.add(new Cactus(cactus.getLast().getxPos() + rand.nextInt(3000) + 200, commonCactusY));  
  
 repaint();  
 } else {  
 firstTimer.stop();  
 repaint();  
 }  
  
  
}

Główna pętla gry