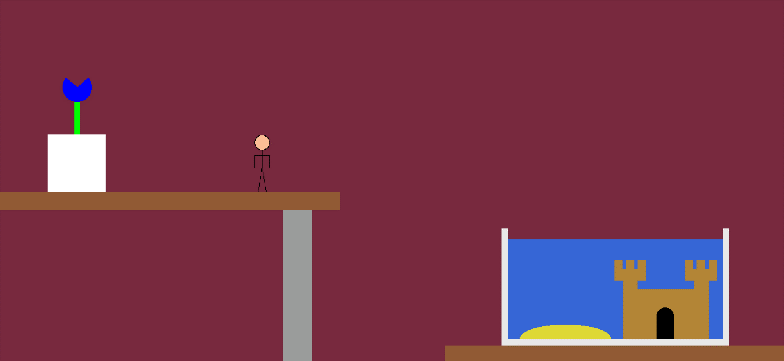
Softwareontwikkeling

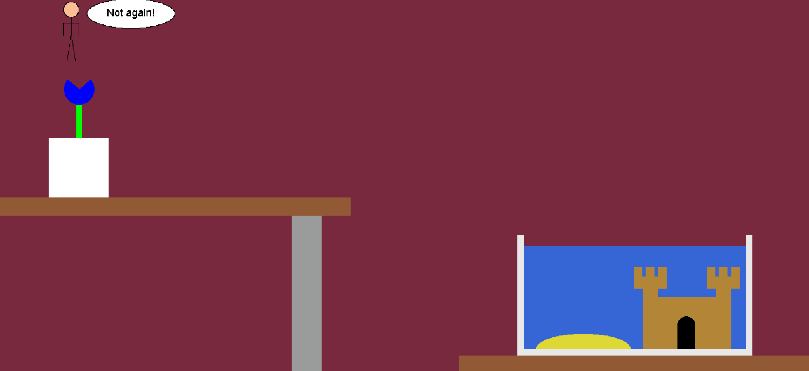
Animatie

******First Second**

**Afbeelding met tekst, lucht, buiten, berg

Automatisch gegenereerde beschrijvingAfbeelding met tekst, verbandtrommel, teken, vectorafbeeldingen

Automatisch gegenereerde beschrijving ThreeOne ThreeTwo**

** ThreeThree**

**Animatie.java GBlockCharacter.java GCharacter.java**

package Animatie;

/\* File: Animation.java

\* --------------------

\* This GCompound creates the character for the animation.

\* --------------------

\* Author: Kenjy Depoorter

\*/

import acm.graphics.\*;

import java.awt.Color;

public class GCharacter extends GCompound {

public GCharacter() {

GOval head = new GOval (0, 0, 26, 26);

head.setFillColor(Color.decode("#fcbe95"));

head.setFilled(true);

add(head);

GLine torso = new GLine (13, 26, 13, 55);

add(torso);

GLine leftLeg = new GLine (13, 55, 6, 100);

add(leftLeg);

GLine rightLeg = new GLine (13, 55, 20, 100);

add(rightLeg);

GLine leftBicep = new GLine (13, 36, 0, 36);

add(leftBicep);

GLine leftForearm = new GLine (0, 36, 0, 57);

add(leftForearm);

GLine rightBicep = new GLine (13, 36, 26, 36);

add(rightBicep);

GLine rightForearm = new GLine (26, 36, 26, 57);

add(rightForearm);

}

}

package Animatie;

/\* File: Animation.java

\* --------------------

\* This GCompound creates a square version of the character for the animation.

\* --------------------

\* Author: Kenjy Depoorter

\*/

import acm.graphics.\*;

import java.awt.Color;

public class GBlockCharacter extends GCompound {

public GBlockCharacter() {

GRect head = new GRect(20, 0, 50, 50);

head.setFillColor(Color.decode("#fcbe95"));

head.setFilled(true);

add(head);

GRect torso = new GRect(20, 50, 50, 90);

torso.setFillColor(Color.decode("#9a9c9b"));

torso.setFilled(true);

add(torso);

GRect leftArm = new GRect(0, 50, 20, 90);

leftArm.setFillColor(Color.decode("#9a9c9b"));

leftArm.setFilled(true);

add(leftArm);

GRect rightArm = new GRect(70, 50, 20, 90);

rightArm.setFillColor(Color.decode("#9a9c9b"));

rightArm.setFilled(true);

add(rightArm);

GRect leftLeg = new GRect(20, 140, 25, 90);

leftLeg.setFillColor(Color.decode("#3666d6"));

leftLeg.setFilled(true);

add(leftLeg);

GRect rightLeg = new GRect(45, 140, 25, 90);

rightLeg.setFillColor(Color.decode("#3666d6"));

rightLeg.setFilled(true);

add(rightLeg);

}

}

package Animatie;

/\* File: Animation.java

\* --------------------

\* In this program the user can watch a small animation.

\* The animation eventually lets the user choose which path the character will take.

\* I used lots of private voids to make this.

\* --------------------

\* Author: Kenjy Depoorter

\*/

import java.awt.Color;

import acm.graphics.\*;

import acm.program.GraphicsProgram;

import acm.util.RandomGenerator;

public class Animation extends GraphicsProgram {

// This is a collection of all the GCompounds

GCharacter character = new GCharacter();

GCloudOne cloudOne = new GCloudOne();

GCloudTwo cloudTwo = new GCloudTwo();

GCloudThree cloudThree = new GCloudThree();

GBlockCharacter blockCharacter = new GBlockCharacter();

GSpeechBubble speechbubble = new GSpeechBubble();

// This is a collection of the recurring objects

GOval speechConsole = new GOval(210, 380, 150, 50);

GLabel consoleSpeechText = new GLabel("Look in the console!");

// This is a collection of all the used GImages

GImage oakDoor = new GImage("C:\\Users\\GAMER\\eclipse-workspace\\Images\_eclipse\\Oak\_Door.jpg");

GImage secondDoor = new GImage("C:\\Users\\GAMER\\eclipse-workspace\\Images\_eclipse\\Second\_Door.jpg");

GImage finalDoor = new GImage("C:\\Users\\GAMER\\eclipse-workspace\\Images\_eclipse\\Final\_Door.jpg");

GImage minecraft = new GImage("C:\\Users\\GAMER\\eclipse-workspace\\Images\_eclipse\\minecraft.png");

GImage obamium = new GImage("C:\\Users\\GAMER\\eclipse-workspace\\Images\_eclipse\\SmallObamium.gif");

GImage timer = new GImage("C:\\Users\\GAMER\\eclipse-workspace\\Images\_eclipse\\timer.png");

GImage birchTree = new GImage("C:\\Users\\GAMER\\eclipse-workspace\\Images\_eclipse\\Birch\_Tree.png");

GImage oakTree = new GImage("C:\\Users\\GAMER\\eclipse-workspace\\Images\_eclipse\\Oak\_Tree.png");

GImage background = new GImage("C:\\Users\\GAMER\\eclipse-workspace\\Images\_eclipse\\background.jpg");

GImage picture = new GImage("C:\\Users\\GAMER\\eclipse-workspace\\Images\_eclipse\\picture.png");

// This is a collection of all the RandomGenerators

RandomGenerator cloudHeight = new RandomGenerator();

RandomGenerator treeType = new RandomGenerator();

public void run() {

setSize(1400, 700);

sceneOne();

pause(1000);

movementOne();

pause(500);

remove(character);

pause(750);

removeAll();

sceneTwo();

pause(1000);

add(character, -50, getHeight() - 226);

pause(500);

for (int i = 0; i <= 100; i++) {

character.move(2, 0);

pause(5);

}

pause(500);

speechConsole.setFillColor(Color.decode("#ffffff"));

speechConsole.setFilled(true);

add(speechConsole);

consoleSpeechText.setFont("Arial-14");

add(consoleSpeechText, 222, 410);

doorChoice();

}

// This private void creates a table for the first scene

private void addTable() {

GRect tableLeg = new GRect(500, 370, 50, getHeight());

tableLeg.setColor(Color.decode("#9a9c9b"));

tableLeg.setFillColor(Color.decode("#9a9c9b"));

tableLeg.setFilled(true);

add(tableLeg);

GRect tabletop = new GRect(0, 340, 600, 30);

tabletop.setColor(Color.decode("#915a34"));

tabletop.setFillColor(Color.decode("#915a34"));

tabletop.setFilled(true);

add(tabletop);

}

// This private void creates an aquarium for the first scene

private void addAquarium() {

addGlass();

addWaterAndDecoration();

}

// This private void creates the glass for the aquarium

private void addGlass() {

GRect glassMiddle = new GRect(getWidth() - 500, getHeight() - 41, 400, 10);

glassMiddle.setColor(Color.decode("#e8e8e8"));

glassMiddle.setFillColor(Color.decode("#e8e8e8"));

glassMiddle.setFilled(true);

add(glassMiddle);

GRect glassLeft = new GRect(getWidth() - 500, getHeight() - 236, 10, 195);

glassLeft.setColor(Color.decode("#e8e8e8"));

glassLeft.setFillColor(Color.decode("#e8e8e8"));

glassLeft.setFilled(true);

add(glassLeft);

GRect glassRight = new GRect(getWidth() - 110, getHeight() - 236, 10, 195);

glassRight.setColor(Color.decode("#e8e8e8"));

glassRight.setFillColor(Color.decode("#e8e8e8"));

glassRight.setFilled(true);

add(glassRight);

}

// This private void adds the water and decoration in the aquarium

private void addWaterAndDecoration() {

GRect water = new GRect(getWidth() - 489, getHeight() - 217, 378, 175);

water.setColor(Color.decode("#3666d6"));

water.setFillColor(Color.decode("#3666d6"));

water.setFilled(true);

add(water);

addCastle();

GArc sand = new GArc(160, 50, 0, 180);

sand.setColor(Color.decode("#ded835"));

sand.setFillColor(Color.decode("#ded835"));

sand.setFilled(true);

add(sand, 918, 573);

}

// This private void creates the sandcastle

private void addCastle() {

GRect castleBase = new GRect(1100, 510, 150, 88);

castleBase.setColor(Color.decode("#b38536"));

castleBase.setFillColor(Color.decode("#b38536"));

castleBase.setFilled(true);

add(castleBase);

castleTowers();

doorWay();

}

// This private void creates the castle towers

private void castleTowers() {

leftCastleTower();

rightCastleTower();

}

// This private void creates the left castle tower

private void leftCastleTower() {

GRect towerShaftLeft = new GRect(1100, 495, 25, 20);

towerShaftLeft.setColor(Color.decode("#b38536"));

towerShaftLeft.setFillColor(Color.decode("#b38536"));

towerShaftLeft.setFilled(true);

add(towerShaftLeft);

GRect towerBaseLeftL = new GRect(1085, 475, 55, 20);

towerBaseLeftL.setColor(Color.decode("#b38536"));

towerBaseLeftL.setFillColor(Color.decode("#b38536"));

towerBaseLeftL.setFilled(true);

add(towerBaseLeftL);

GRect towerTopLeftL = new GRect(1085, 459, 13, 15);

towerTopLeftL.setColor(Color.decode("#b38536"));

towerTopLeftL.setFillColor(Color.decode("#b38536"));

towerTopLeftL.setFilled(true);

add(towerTopLeftL);

GRect towerTopMiddleL = new GRect(1105, 459, 14, 15);

towerTopMiddleL.setColor(Color.decode("#b38536"));

towerTopMiddleL.setFillColor(Color.decode("#b38536"));

towerTopMiddleL.setFilled(true);

add(towerTopMiddleL);

GRect towerTopRightL = new GRect(1126, 459, 14, 15);

towerTopRightL.setColor(Color.decode("#b38536"));

towerTopRightL.setFillColor(Color.decode("#b38536"));

towerTopRightL.setFilled(true);

add(towerTopRightL);

}

// This private void creates the right castle tower

private void rightCastleTower() {

GRect towerShaftRight = new GRect(1225, 495, 25, 20);

towerShaftRight.setColor(Color.decode("#b38536"));

towerShaftRight.setFillColor(Color.decode("#b38536"));

towerShaftRight.setFilled(true);

add(towerShaftRight);

GRect towerBaseLeftR = new GRect(1210, 475, 55, 20);

towerBaseLeftR.setColor(Color.decode("#b38536"));

towerBaseLeftR.setFillColor(Color.decode("#b38536"));

towerBaseLeftR.setFilled(true);

add(towerBaseLeftR);

GRect towerTopLeftR = new GRect(1210, 459, 13, 15);

towerTopLeftR.setColor(Color.decode("#b38536"));

towerTopLeftR.setFillColor(Color.decode("#b38536"));

towerTopLeftR.setFilled(true);

add(towerTopLeftR);

GRect towerTopMiddleR = new GRect(1230, 459, 14, 15);

towerTopMiddleR.setColor(Color.decode("#b38536"));

towerTopMiddleR.setFillColor(Color.decode("#b38536"));

towerTopMiddleR.setFilled(true);

add(towerTopMiddleR);

GRect towerTopRightR = new GRect(1251, 459, 14, 15);

towerTopRightR.setColor(Color.decode("#b38536"));

towerTopRightR.setFillColor(Color.decode("#b38536"));

towerTopRightR.setFilled(true);

add(towerTopRightR);

}

// This private void creates the castle's doorway

private void doorWay() {

GOval doorWayArc = new GOval(1159, 543, 30, 30);

doorWayArc.setColor(Color.BLACK);

doorWayArc.setFillColor(Color.BLACK);

doorWayArc.setFilled(true);

add(doorWayArc);

GRect doorWayBase = new GRect(1159, 558, 30, 40);

doorWayBase.setColor(Color.BLACK);

doorWayBase.setFillColor(Color.BLACK);

doorWayBase.setFilled(true);

add(doorWayBase);

}

// This private void creates the pot of the flower

private void pot() {

GRect pot = new GRect(85, 238, 101, 101);

pot.setColor(Color.WHITE);

pot.setFillColor(Color.WHITE);

pot.setFilled(true);

add(pot);

}

// This private void creates the flower

private void flower() {

GRect flowerBase = new GRect(132, 167, 9, 70);

flowerBase.setColor(Color.GREEN);

flowerBase.setFillColor(Color.GREEN);

flowerBase.setFilled(true);

add(flowerBase);

GArc flowerPetals = new GArc(51, 51, 140, 260);

flowerPetals.setColor(Color.BLUE);

flowerPetals.setFillColor(Color.BLUE);

flowerPetals.setFilled(true);

add(flowerPetals, 111, 129);

}

// This private void creates the flowerpot

private void addFlowerPot() {

pot();

flower();

}

// This private void is a collection of the first series of movement

private void movementOne() {

for (int i = 0; i <= 220; i++) {

character.move(2, 0);

pause(5);

}

pause(750);

jump();

pause(500);

for (int a = 0; a <= 90; a++) {

character.move(1.5, 0.11);

pause(7);

}

}

// This private simulates a jump

private void jump() {

for (int j = 0; j <= 40; j++) {

character.move(2, -3);

pause(5);

}

for (int k = 0; k <= 25; k++) {

character.move(2, -2);

pause(5);

}

for (int l = 0; l <= 20; l++) {

character.move(2, -1);

pause(5);

}

for (int n = 0; n <= 10; n++) {

character.move(2, 0);

pause(5);

}

for (int o = 0; o <= 20; o++) {

character.move(2, 1);

pause(5);

}

for (int p = 0; p <= 25; p++) {

character.move(2, 2);

pause(5);

}

for (int q = 0; q <= 40; q++) {

character.move(2, 3);

pause(5);

}

for (int q = 0; q <= 20; q++) {

character.move(2, 4);

pause(5);

}

for (int q = 0; q <= 40; q++) {

character.move(1, 4);

pause(5);

}

}

// This private void creates the first scene

private void sceneOne() {

GRect achtergrondOne = new GRect(0, 0, getWidth(), getHeight());

achtergrondOne.setColor(Color.decode("#78293e"));

achtergrondOne.setFillColor(Color.decode("#78293e"));

achtergrondOne.setFilled(true);

add(achtergrondOne);

GRect plank = new GRect(getWidth() - 600, getHeight() - 30, 600, 30);

plank.setColor(Color.decode("#915a34"));

plank.setFillColor(Color.decode("#915a34"));

plank.setFilled(true);

add(plank);

addTable();

addAquarium();

add(character, 100, 239);

addFlowerPot();

}

// This private void creates the second scene

private void sceneTwo() {

GRect achtergrondTwo = new GRect(0, 0, getWidth(), getHeight());

achtergrondTwo.setColor(Color.decode("#916b29"));

achtergrondTwo.setFillColor(Color.decode("#916b29"));

achtergrondTwo.setFilled(true);

add(achtergrondTwo);

GRect castleFloor = new GRect(0, getHeight() - 125, getWidth(), 125);

castleFloor.setColor(Color.decode("#785a24"));

castleFloor.setFillColor(Color.decode("#785a24"));

castleFloor.setFilled(true);

add(castleFloor);

GRect castleRoof = new GRect(0, 0, getWidth(), 125);

castleRoof.setColor(Color.decode("#785a24"));

castleRoof.setFillColor(Color.decode("#785a24"));

castleRoof.setFilled(true);

add(castleRoof);

addDoorsAndIcons();

addNumbers();

}

// This private void creates both doors and their respective icons

private void addDoorsAndIcons() {

add(oakDoor, 520, getHeight() - 250);

add(secondDoor, 800, getHeight() - 250);

add(finalDoor, 1120, getHeight() - 250);

add(minecraft, 528, getHeight() - 320);

add(obamium, 817, getHeight() - 320);

add(timer, 1134, getHeight() - 320);

}

// This private void creates the numbers next to the doors

private void addNumbers() {

GLabel numberOne = new GLabel("1");

numberOne.setFont("Arial-32");

add(numberOne, 595, getHeight() - 220);

GLabel numberTwo = new GLabel("2");

numberTwo.setFont("Arial-32");

add(numberTwo, 895, getHeight() - 220);

GLabel numberThree = new GLabel("3");

numberThree.setFont("Arial-32");

add(numberThree, 1195, getHeight() - 220);

}

// This private void lets the user's choice change the animation

private void doorChoice() {

int keuze = 0;

keuze = readInt("Which door will you choose? ");

remove(speechConsole);

remove(consoleSpeechText);

switch (keuze) {

case 1: {

sceneThreeOne();

break;

}

case 2: {

sceneThreeTwo();

break;

}

case 3: {

sceneThreeThree();

break;

}

}

}

// This private void creates the first of three possible final scenes

private void sceneThreeOne() {

pause(500);

for (int i = 0; i <= 168; i++) {

character.move(2, 0);

pause(5);

}

pause(500);

remove(character);

pause(750);

removeAll();

GRect achtergrondThree = new GRect(0, 0, getWidth(), getHeight());

achtergrondThree.setColor(Color.decode("#2da3e3"));

achtergrondThree.setFillColor(Color.decode("#2da3e3"));

achtergrondThree.setFilled(true);

add(achtergrondThree);

GRect sun = new GRect(getWidth() - 185, 25, 90, 90);

sun.setColor(Color.decode("#fff675"));

sun.setFillColor(Color.decode("#fff675"));

sun.setFilled(true);

add(sun);

int cloudHeightOne = cloudHeight.nextInt(20, 80);

add(cloudOne, 80, cloudHeightOne);

int cloudHeightTwo = cloudHeight.nextInt(20, 80);

add(cloudTwo, 440, cloudHeightTwo);

int cloudHeightThree = cloudHeight.nextInt(20, 80);

add(cloudThree, 800, cloudHeightThree);

addGrond();

int treeTypeChoice = treeType.nextInt(1, 4);

switch (treeTypeChoice) {

case 1, 2: {

add(birchTree, 800, 126);

break;

}

case 3, 4: {

add(oakTree, 790, 112);

break;

}

}

add(blockCharacter, 440, -270);

for (int c = 0; c < 193; c++) {

blockCharacter.move(0, 3);

pause(3);

}

pause(500);

GRect squareSpeechBubble = new GRect(540, 300, 150, 50);

squareSpeechBubble.setFillColor(Color.decode("#ffffff"));

squareSpeechBubble.setFilled(true);

add(squareSpeechBubble);

GLabel pixelText = new GLabel("Where am I?");

pixelText.setFont("Arial-18");

add(pixelText, 566, 330);

}

// This private void creates the second of three possible final scenes

private void sceneThreeTwo() {

pause(500);

for (int i = 0; i <= 308; i++) {

character.move(2, 0);

pause(5);

}

pause(500);

remove(character);

pause(750);

removeAll();

add(background);

add(picture, 640, 534);

add(obamium, 1326, 150);

pause(1200);

GRect speechHello = new GRect(540, 480, 150, 50);

speechHello.setFillColor(Color.decode("#ffffff"));

speechHello.setFilled(true);

add(speechHello);

GLabel helloText = new GLabel("Uhh... Hello?");

helloText.setFont("Arial-18");

add(helloText, 566, 510);

}

// This private void creates the third of three possible final scenes

private void sceneThreeThree() {

pause(500);

for (int i = 0; i <= 468; i++) {

character.move(2, 0);

pause(5);

}

pause(500);

remove(character);

pause(750);

removeAll();

GRect achtergrondOne = new GRect(0, 0, getWidth(), getHeight());

achtergrondOne.setColor(Color.decode("#78293e"));

achtergrondOne.setFillColor(Color.decode("#78293e"));

achtergrondOne.setFilled(true);

add(achtergrondOne);

GRect plank = new GRect(getWidth() - 600, getHeight() - 30, 600, 30);

plank.setColor(Color.decode("#915a34"));

plank.setFillColor(Color.decode("#915a34"));

plank.setFilled(true);

add(plank);

addTable();

addAquarium();

add(character, 110, -101);

addFlowerPot();

pause(750);

add(speechbubble, 150, 0);

pause(1000);

for (int p = 0; p <= 56; p++) {

character.move(0, 2);

pause(2);

}

for (int k = 0; k <= 85; k++) {

character.move(0, 2);

speechbubble.move(0, 2);

pause(2);

}

for (int o = 0; o <= 26; o++) {

character.move(0, 2);

pause(2);

}

remove(speechbubble);

pause(1500);

run();

}

// This private void creates the ground for the first end scene

private void addGrond() {

GRect dirtFirstLayer = new GRect(0, getHeight() - 80, getWidth(), 80);

dirtFirstLayer.setColor(Color.decode("#4f2e20"));

dirtFirstLayer.setFillColor(Color.decode("#4f2e20"));

dirtFirstLayer.setFilled(true);

add(dirtFirstLayer);

GRect grassFirstLayer = new GRect(300, getHeight() - 100, getWidth() - 300, 20);

grassFirstLayer.setColor(Color.decode("#57a324"));

grassFirstLayer.setFillColor(Color.decode("#57a324"));

grassFirstLayer.setFilled(true);

add(grassFirstLayer);

GRect dirtSecondLayerLeft = new GRect(0, getHeight() - 180, 300, 180);

dirtSecondLayerLeft.setColor(Color.decode("#4f2e20"));

dirtSecondLayerLeft.setFillColor(Color.decode("#4f2e20"));

dirtSecondLayerLeft.setFilled(true);

add(dirtSecondLayerLeft);

GRect dirtSecondLayerRight = new GRect(getWidth() - 200, getHeight() - 180, 200, 100);

dirtSecondLayerRight.setColor(Color.decode("#4f2e20"));

dirtSecondLayerRight.setFillColor(Color.decode("#4f2e20"));

dirtSecondLayerRight.setFilled(true);

add(dirtSecondLayerRight);

GRect grassSecondLayerLeft = new GRect(100, getHeight() - 200, 200, 20);

grassSecondLayerLeft.setColor(Color.decode("#57a324"));

grassSecondLayerLeft.setFillColor(Color.decode("#57a324"));

grassSecondLayerLeft.setFilled(true);

add(grassSecondLayerLeft);

GRect grassSecondLayerRight = new GRect(getWidth() - 200, getHeight() - 200, 200, 20);

grassSecondLayerRight.setColor(Color.decode("#57a324"));

grassSecondLayerRight.setFillColor(Color.decode("#57a324"));

grassSecondLayerRight.setFilled(true);

add(grassSecondLayerRight);

GRect dirtThirdLayerLeft = new GRect(0, getHeight() - 280, 100, 120);

dirtThirdLayerLeft.setColor(Color.decode("#4f2e20"));

dirtThirdLayerLeft.setFillColor(Color.decode("#4f2e20"));

dirtThirdLayerLeft.setFilled(true);

add(dirtThirdLayerLeft);

GRect grassThirdLayerLeft = new GRect(0, getHeight() - 300, 100, 20);

grassThirdLayerLeft.setColor(Color.decode("#57a324"));

grassThirdLayerLeft.setFillColor(Color.decode("#57a324"));

grassThirdLayerLeft.setFilled(true);

add(grassThirdLayerLeft);

}

}

**GCloudOne.java GCloudTwo.java GCloudThree.java**

package Animatie;

/\* File: Animation.java

\* --------------------

\* This GCompound creates the third cloud for the animation.

\* --------------------

\* Author: Kenjy Depoorter

\*/

import acm.graphics.\*;

import java.awt.Color;

public class GCloudThree extends GCompound {

public GCloudThree() {

GRect layerOne = new GRect(80, 0, 40, 40);

layerOne.setColor(Color.decode("#e0e0e0"));

layerOne.setFillColor(Color.decode("#e0e0e0"));

layerOne.setFilled(true);

add(layerOne);

GRect layerTwo = new GRect(40, 40, 160, 40);

layerTwo.setColor(Color.decode("#e0e0e0"));

layerTwo.setFillColor(Color.decode("#e0e0e0"));

layerTwo.setFilled(true);

add(layerTwo);

GRect layerThree = new GRect(0, 80, 240, 40);

layerThree.setColor(Color.decode("#e0e0e0"));

layerThree.setFillColor(Color.decode("#e0e0e0"));

layerThree.setFilled(true);

add(layerThree);

}

}

package Animatie;

/\* File: Animation.java

\* --------------------

\* This GCompound creates the second cloud for the animation.

\* --------------------

\* Author: Kenjy Depoorter

\*/

import acm.graphics.\*;

import java.awt.Color;

public class GCloudTwo extends GCompound {

public GCloudTwo() {

GRect layerOne = new GRect(40, 0, 80, 40);

layerOne.setColor(Color.decode("#e0e0e0"));

layerOne.setFillColor(Color.decode("#e0e0e0"));

layerOne.setFilled(true);

add(layerOne);

GRect layerTwo = new GRect(0, 40, 160, 40);

layerTwo.setColor(Color.decode("#e0e0e0"));

layerTwo.setFillColor(Color.decode("#e0e0e0"));

layerTwo.setFilled(true);

add(layerTwo);

}

}

package Animatie;

/\* File: Animation.java

\* --------------------

\* This GCompound creates the first cloud for the animation.

\* --------------------

\* Author: Kenjy Depoorter

\*/

import acm.graphics.\*;

import java.awt.Color;

public class GCloudOne extends GCompound {

public GCloudOne() {

GRect layerOne = new GRect(0, 0, 200, 40);

layerOne.setColor(Color.decode("#e0e0e0"));

layerOne.setFillColor(Color.decode("#e0e0e0"));

layerOne.setFilled(true);

add(layerOne);

GRect layerTwo = new GRect(80, 40, 80, 40);

layerTwo.setColor(Color.decode("#e0e0e0"));

layerTwo.setFillColor(Color.decode("#e0e0e0"));

layerTwo.setFilled(true);

add(layerTwo);

}

}

**GSpeechBubble.java**

package Animatie;

/\* File: Animation.java

\* --------------------

\* This GCompound creates the round speech bubble for the animation.

\* --------------------

\* Author: Kenjy Depoorter

\*/

import acm.graphics.\*;

import java.awt.Color;

public class GSpeechBubble extends GCompound{

public GSpeechBubble() {

GOval bubble = new GOval(0, 0, 150, 50);

bubble.setFillColor(Color.decode("#ffffff"));

bubble.setFilled(true);

add(bubble);

GLabel notAgain = new GLabel("Not again!");

notAgain.setFont("Arial-18");

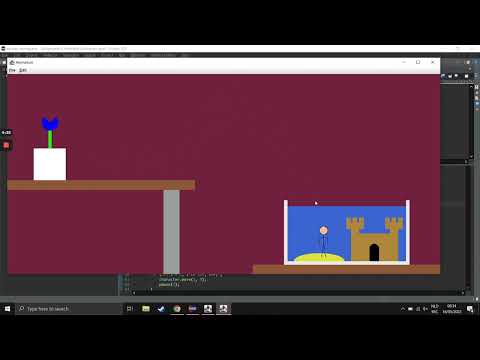
add(notAgain, 34, 30);

}

}

[](https://www.youtube.com/embed/FblpYTzhZQA?feature=oembed) **Scenario 1 Scenario 2**

**[](https://www.youtube.com/embed/_ifYsWEihlU?feature=oembed)**

**[](https://www.youtube.com/embed/g5XtgjAQcIk?feature=oembed)Scenario 3**