**SHORT DESCRIPTION:**

Our project is entitled as “Shot the Balloon”. This game is for everyone. The instructions could easily be followed by any person may it be a child or an adult. From the title itself, the main goal of this game is to shot the balloon where each balloon correspond a point depending on its colour and how fast it’ll fly up.

**SCOPE AND LIMITATION:**

This is a one player game that could be played by anyone with no age limit as long as he/she is computer literate. To be able to play the game, one should follow its instructions. The player should hit the balloon using the mouse through clicking it with the guide of the arrow at the right side. He/She could move it by moving the mouse upward or downward.

**RECOMMENDATION:**

We are planning to revise our game project and make it more challenging to the player. We will improve the background color and the graphics. We will add some music and sound effects on it. We are also planning to put three lives for the player and have 3 levels; easy, average and difficult.

**Shoot the Balloons**

A Final Project

Submitted by:

Arrienda, Milry T.

Moso, Karen L.

Rosales, Pamela Christelle A.

Submitted to:

Prof. Malikey M. Maulana

October 15, 2011

**CODING:**

/\* Project entitled " SHOOT THE BALLOONS"

Submitted by:

Arrienda, Milry T.

Moso, Karen L.

Rosales, Pamela Christelle A.

- this is a simple shooting game

\*/

//global variables

ball[] Balloon = new ball[8];

shooter bow;

Shot [] arrowShot = new Shot [15];

int totalShots = 0;

int pageNumber, score, lives;

float txtX, txtY;

boolean enterPressed;

boolean rPressed;

//images of the background

PImage intro;

PImage bg;

PImage bg2;

PImage bg3;

PImage end;

//images of the balloon

PImage balloon;

PImage balloonpop;

//images of the arrow and bow

PImage arrow\_n\_bow;

PImage arrowMoving;

//music and sound effects

import ddf.minim.analysis.\*;

import ddf.minim.\*;

Minim minim;

AudioPlayer bgmusic;

AudioSample ballPop;

//fonts used

PFont header;

PFont txt;

void setup()

{

size(600, 700, P2D);

smooth();

//fonts

header = loadFont("Aharoni-Bold-48.vlw");

txt = loadFont("AgencyFB-Bold-48.vlw");

//images

intro = loadImage("SKY.png");

bg = loadImage("bg.jpg");

bg2 = loadImage("bg2.jpg");

bg3 = loadImage("bg3.jpg");

end = loadImage("end.jpg");

//graphics

arrow\_n\_bow = loadImage("bownarrow.png");

arrowMoving = loadImage("arrow.png");

balloonpop = loadImage("pop.png");

balloon = loadImage("balloon.png");

//balloon

for (int i=0;i<Balloon.length;i++)

{

int x = (int)random(10, 400);

float y = random(800, 1000);

int a = (int)random(0, 255);

int b = (int)random(0, 255);

int c = (int)random(0, 255);

Balloon[i] = new ball(x, y, a, b, c);

Balloon[i].speed\_ball = (int)random(0.3,2);

}

// bow and arrow

bow = new shooter();

//arrow shots

for(int i = 0; i < arrowShot.length; i++)

{

arrowShot[i] = new Shot(random(0,700));

println("shot instance name is: shot" + i);

}

//music and sound effects

minim = new Minim(this);

ballPop = minim.loadSample("BalloonPop.mp3", 512);

bgmusic = minim.loadFile("bgmusic.mp3", 510000 );

bgmusic.play();

bgmusic.loop();

//variables

pageNumber = 1;

score = 0;

lives = 5;

}

void draw()

{

background(0);

if (pageNumber == 1) // intro

{

page1();

}

if (pageNumber == 2) // easy level

{

page2();

}

if (pageNumber == 3) // average level

{

page3();

}

if (pageNumber == 4) // difficult level

{

page4();

}

if (pageNumber == 5) // gameover page

{

page5();

}

if (pageNumber == 6) // WINNER! page

{

page6();

}

if (pageNumber == 7) // quick game end page

{

page7();

}

if (pageNumber == 8) // quick game page

{

page8();

}

}

//intro

void page1()

{

image(intro, 0, 0);

txtX = 100;

txtY = 100;

textFont(header);

fill(0);

text("Shoot the Balloons!", txtX, txtY);

textFont(txt);

textSize(30);

text("Click to Play Arcade.", txtX + 50, txtY+50);

text("Press ENTER to play quick game.", txtX + 50, txtY+100);

textFont(header);

text("Instructions:", txtX, txtY+250);

textFont(txt);

textSize(30);

text("Use your mouse to play this game.", txtX+50, txtY+300);

//if mouse is clicked, proceed to the next page which is the easy level

if(mousePressed) pageNumber = 2;

//if enter is pressed, proceed to the quick game

if(enterPressed)

{

pageNumber = 8;

}

}

//easy level page

void page2()

{

// tint(200, 250);

image(bg, 0, 0);

noTint();

//display the bow

bow.yposBow = mouseY-65;

bow.display();

txtX = 100;

txtY = 30;

//text

textFont(header);

textSize(30);

text("Shoot the Balloons: Easy Level", txtX, txtY); //displays teh current level

textFont(txt);

textSize(15);

text("Score (Balloons Hit):" + score + " (Balloons needed " + (35-score) + ")", 50, txtY+20); //displays the score

text("Lives:" + lives, 50, txtY+40); //displays the current lives

//balloons

for (int i=0;i<Balloon.length;i++)

{

Balloon[i].move();

Balloon[i].display();

Balloon[i].show();

Balloon[i].checkBallHit();

}

//arrow shots

for (int l=0; l<arrowShot.length; l++){

{

if (arrowShot[l].fired){

arrowShot[l].display();

arrowShot[l].move();

}

// println("shot instance display name is: shot" + totalShots);

}

}

//if the player's score reached 35 then it will go to the next level- average

if(score == 35){

pageNumber = 3;

score = 0;

}

}

//average level page

void page3()

{

background(255);

// tint(200, 250);

image(bg2, 0, 0);

noTint();

//displays the bow

bow.yposBow = mouseY-65;

bow.display();

txtX = 100;

txtY = 30;

textFont(header);

textSize(30);

text("Shoot the Balloons: Average Level", txtX-50, txtY); //displays the current level

textFont(txt);

fill(255);

textSize(15);

text("Score (Balloons Hit):" + score + " (Balloons needed " + (25-score) + ")", 50, txtY+20); //displays the score

text("Lives:" + lives, 50, txtY+40); //displays the current lives

noFill();

//balloons

for (int i=0;i<Balloon.length;i++)

{

Balloon[i].move();

Balloon[i].display();

Balloon[i].show();

Balloon[i].checkBallHit2();

}

//arrow shots

for (int l=0; l<arrowShot.length; l++){

{

if (arrowShot[l].fired){

arrowShot[l].display();

arrowShot[l].move();

}

// println("shot instance display name is: shot" + totalShots);

}

}

//if the score reached 25 then proceed to the next level - difficult

if(score == 25){

pageNumber = 4;

score = 0;

}

}

//difficult level page

void page4()

{

background(255);

//tint(200, 250);

image(bg3, 0, 0);

noTint();

//displays the bow

bow.yposBow = mouseY-65;

bow.display();

txtX = 100;

txtY = 30;

textFont(header);

noFill();

textSize(30);

//fill(255);

text("Shoot the Balloons: Difficult Level", txtX-40, txtY); //displays the current level

textFont(txt);

textSize(15);

text("Score (Balloons Hit):" + score + " (Balloons needed " + (20-score) + ")", 50, txtY+20); //displays the score

text("Lives:" + lives, 50, txtY+40); //displays the current lives

//balloons

for (int i=0;i<Balloon.length;i++)

{

Balloon[i].move();

Balloon[i].display();

Balloon[i].show();

Balloon[i].checkBallHit3();

}

//arrow shots

for (int l=0; l<arrowShot.length; l++){

{

if (arrowShot[l].fired){

arrowShot[l].display();

arrowShot[l].move();

}

// println("shot instance display name is: shot" + totalShots);

}

}

//if the score reached 20 then the player finishes all the level

if(score == 20){

pageNumber = 6;

score = 0;

}

}

//gameover page

void page5()

{

background(0);

//tint(200, 250);

image(end, 0, 0);

noTint();

txtX = 300;

txtY = 200;

textFont(header);

fill(255);

textSize(60);

text("Sorry! :(", txtX-100, txtY);

textFont(txt);

textSize(40);

text("GAMEOVER!", txtX-80, txtY+100);

text("Press r to play again", txtX-150, txtY+180);

//if r is pressed then go to the first page

if(rPressed)

{

pageNumber = 1;

score = 0;

lives = 5;

}

}

//WINNERS!

void page6()

{

background(0);

//tint(200, 250);

image(end, 0, 0);

noTint();

txtX = 200;

txtY = 200;

textFont(header);

fill(255);

text("Congratulations! (o.0)", txtX-150, txtY);

textFont(header);

textSize(25);

text("You have finished all the Levels!", txtX-80, txtY+100);

text("Press r to play again.", txtX-60, txtY+150);

if(rPressed)

{

pageNumber = 1;

score = 0;

lives = 5;

}

}

//quick game

void page8()

{

background(255);

//tint(200, 250);

image(bg3, 0, 0);

noTint();

//displays the bow

bow.yposBow = mouseY-65;

bow.display();

txtX = 100;

txtY = 30;

textFont(header);

fill(0);

textSize(30);

text("Shoot the Balloons: Quick Game", txtX-20, txtY);

textFont(txt);

textSize(15);

text("Score (Balloons Hit):" + score, txtX-50, txtY+20); //displays the score

//text("Lives:" + lives, txtX, txtY);

//balloons

for (int i=0;i<Balloon.length;i++)

{

Balloon[i].move();

Balloon[i].display();

Balloon[i].quick();

Balloon[i].checkBallHit2();

}

//arrow shots

for (int l=0; l<arrowShot.length; l++){

{

if (arrowShot[l].fired){

arrowShot[l].display();

arrowShot[l].move();

}

// println("shot instance display name is: shot" + totalShots);

}

}

}

//quick game end page

void page7()

{

background(0);

//tint(200, 250);

image(end, 0, 0);

noTint();

txtX = 200;

txtY = 200;

textFont(header);

fill(255);

textSize(80);

text("Sorry! :(", txtX-50, txtY);

textSize(30);

textFont(txt);

text("GAMEOVER!", txtX, txtY+150);

text("You have hit: " + score + " balloons", txtX-90, txtY+220);

text("Press r to play again", txtX-60, txtY+280);

if(rPressed)

{

pageNumber = 1;

score = 0;

lives = 5;

}

}

//----------------------------------------------------------------

//declaring a class of balloon

class ball {

//properties of balloon

int xpos\_ball; //x-coordinate

float ypos\_ball; //y-coordinate

float speed\_ball; //speed of the balloon

float direction\_ball = -1; //going upwards

int a, b, c; //color of the balloon

int ballWidth = 70; //width

int ballHeight = 190; //height

//constructor

ball(int x, float y, int a1, int b1, int c1)

{

xpos\_ball = x;

ypos\_ball = y;

a = a1;

b = b1;

c = c1;

}

//movement of the balloon

void move()

{

ypos\_ball += (speed\_ball \* direction\_ball);

}

//displays the balloons

void display()

{

tint(a, b, c);

image(balloon, xpos\_ball, ypos\_ball, ballWidth, ballHeight);

noTint();

}

//this method will show tha if the balloon reached the sky then one life will be taken

void show()

{

if(ypos\_ball < -100)

{

ypos\_ball = random(800, 1000);

lives = lives - 1;

if(lives == 0)

{

lives = lives;

pageNumber = 5;

}

xpos\_ball = (int)random(10, 450);

}

if (pageNumber == 1)

{

ypos\_ball = height;

}

if (pageNumber == 5)

{

ypos\_ball = height;

}

}

//balloons are not seen in the window for pages 1, 5, 6, 7

void show2()

{

ypos\_ball = height;

}

//for quick game page

void quick()

{

if(ypos\_ball < -100)

{

pageNumber = 7;

}

}

//to check the collision of the arrow and the balloons of page 2

void checkBallHit()

{

if(dist(xpos\_ball,ypos\_ball,arrowShot[totalShots].arrowshotX,arrowShot[totalShots].arrowshotY) < ballWidth-15){

image(balloonpop, xpos\_ball, ypos\_ball, ballWidth, ballHeight);

arrowShot[totalShots].arrowshotX = 500;

ballPop.trigger();

//speed\_ball = (int)random(0.3, 2);

ypos\_ball = random(700, 900);

xpos\_ball = (int)random(10, 400);

arrowShot[totalShots].fired = false;

score += 1; //for every balloon hit one point is added to the score

}

}

//to check the collision of the arrow and the balloons of page 3

void checkBallHit2()

{

if(dist(xpos\_ball,ypos\_ball,arrowShot[totalShots].arrowshotX,arrowShot[totalShots].arrowshotY) < ballWidth-15){

image(balloonpop, xpos\_ball, ypos\_ball, ballWidth, ballHeight);

ballPop.trigger();

arrowShot[totalShots].arrowshotX = 500;

speed\_ball = (int)random(1, 3); //increases the speed

ypos\_ball = random(700, 900);

xpos\_ball = (int)random(10, 400);

arrowShot[totalShots].fired = false;

score += 1; //for every balloon hit one point is added to your score

}

}

//to check the collision of the arrow and the balloons of page 4

void checkBallHit3()

{

if(dist(xpos\_ball,ypos\_ball,arrowShot[totalShots].arrowshotX,arrowShot[totalShots].arrowshotY) < ballWidth-15){

image(balloonpop, xpos\_ball, ypos\_ball, ballWidth, ballHeight);

ballPop.trigger();

arrowShot[totalShots].arrowshotX = 500;

speed\_ball = (int)random(2, 5); //increases the speed

ypos\_ball = random(700, 900);

xpos\_ball = (int)random(10, 400);

arrowShot[totalShots].fired = false;

score += 1; // for every balloon hit one point is added to your score

}

}

}

//declaring bow and arrow as a class

class shooter

{

//properties of the class bow and arrow

float xposBow;

float yposBow;

int bowWidth;

int bowHeight;

shooter()

{

xposBow = 500;

bowWidth = 100;

bowHeight = 150;

}

//method that displays the bow

void display()

{

float YR = constrain(yposBow, 50, height-150);

image(arrow\_n\_bow, xposBow, YR, bowWidth, bowHeight);

}

}

//Shot class for arrows

class Shot

{

int arrowshotX;

float arrowshotY;

int arrowWidth = 100;

int arrowHeight = 150;

int arrowSpeed;

int arrowDirection = -1;

boolean fired; // also indicates if the bullet should be displayed or not ...

Shot(float tempShotY)

{

arrowshotX = 500;

arrowshotY = tempShotY;

arrowSpeed = 10;

fired = false;

}

//movement of the arrow

void move()

{

arrowshotX += (arrowSpeed \* arrowDirection);

if (arrowshotX < -100){

fired = false;

arrowshotX = 500;

}

}

//displays the arrow

void display()

{

//Display the arrow

float YR = constrain(arrowshotY, 50, height-150);

image(arrowMoving, arrowshotX, YR, arrowWidth, arrowHeight);

}

}

//if mouse is released an arrow is shot

void mouseReleased()

{

totalShots+=2;

totalShots %= arrowShot.length;

arrowShot[totalShots].arrowshotY = mouseY-65;

arrowShot[totalShots].arrowshotX = 500;

arrowShot[totalShots].fired = true;

//mouseCheck = true;

}

void keyPressed()

{

if (keyCode == ENTER ) enterPressed = true;

if (key == 'r' ) rPressed = true;

}

void keyReleased()

{

if (keyCode == ENTER ) enterPressed = false;

if (key == 'r' ) rPressed = false;

}

void stop()

{

// always close Minim audio classes when you are done with them

ballPop.close();

bgmusic.close();

minim.stop();

super.stop();

}