“Whack the Kat”

A program of the clone popular arcade game whack-a-mole

Cheemay (Zi Wei) Song

20334601

CS171 Computer Systems I

B.Sc.

Computer Science and Software Engineering

Department of Computer Science
Maynooth University
Maynooth, Co. Kildare
Ireland

Declaration

I hereby certify that the project to be accessed as part of my CS171 Computer Systems module, is entirely of my own work and the work was not taken from the work of others. The work has been cited and acknowledged to be my own work.

Signed: Cheemay Song *Date:* 20/11/2020

Acknowledgements

I would like to thank my younger brother Ricky Song for suggesting this project idea to me. I managed to complete this project within the time given by my lecturers, and I would like to express my gratitude to my friends and family for all the support they have given me. I would like to thank my friend Rosemary Bello for reviewing this word document. I would also like to thank a third-party member, "paulgoux" on the discourse processing.org website, who helped me on some issues on the programming.

Abstract

This report details the development of a simple clone version of the popular arcade game Whac-a-Mole with meerkats being the replacement of the moles and the concept of them getting slapped as a change. The game over screen displays the score, replay, and exit button. The game allows the player to restart the game or exit the game which means that the player can play the game infinite number of times or choose to close the game when they have played enough. This report includes contributions of third-party members who successfully solved some of the project code's issues. The report includes testing of the simple game and references to third party members. This report ends with some suggestions for future works.

1 Introduction

It might be easy to visualise a simple whack-a-mole game from the arcade but there are online alternatives which also provides a similar function. The ability to feel the surroundings is very important in playing this game but not everyone is able to go to arcades or buy a whack-a-mole game set from the local toy store. The aim of this project is to develop a clone version of the popular arcade game whack-a-mole that has the exact same functions as the ones in the arcades.

2 Research

Whack-a-mole is a popular arcade game. The game was invented in Japan in 1975 by Kazuo Yamada. The game concepts were designed by Yamada's sketches, he envisioned a play area and a display screen for the score. The play area has five holes with plastic moles popping out of them randomly. There are many websites that provide online versions of such game.

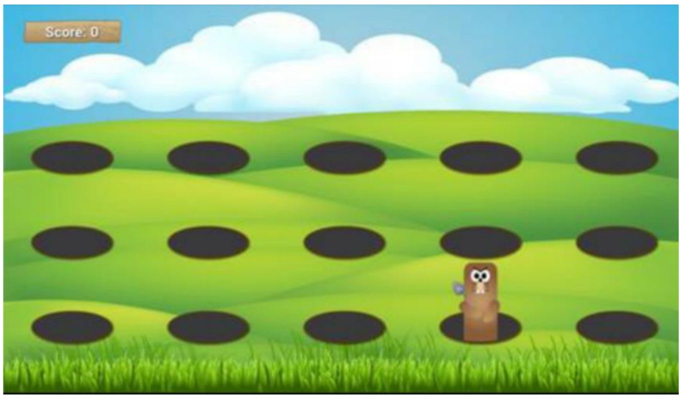

Figure 2.1 A simple online Whack-a-Mole game

Most of these Whack-a-Mole online games only display one's score without a timer, no indication of the game stopping. This would mean the game would continue until the player chooses to close the website. This project includes both scoreboard and timer on the play area to allow the player to challenge themselves next time. The project includes an exit button for the player to exit the game as they please. The change to the original is the hammer to a hand and the moles to meerkats. The game's scenery is that of a desert.

Specification

Overview of the code

Testing

Conclusion

3 Technical background and problem statement

The problem that arose first was getting the meerkats to appear in specific locations but randomly. For the meerkats to appear in a specific location, the position of the meerkats needed to be set. Placing the meerkats in all the hole positions was much harder to solve than it seems when the appearance of the meerkats must be random.

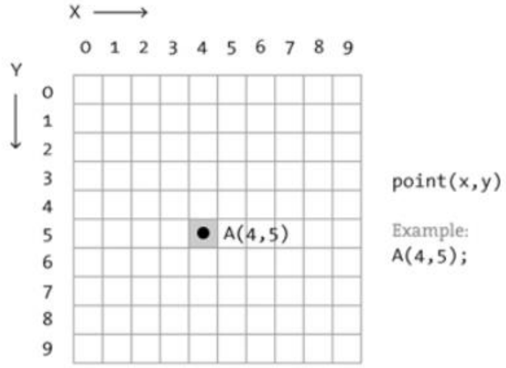

Figure 3.1 Processing co-ordinate system

The simplest way to manipulate the positions of the meerkats at random requires the co-ordinate system. Unlike the usual mathematical graph, the origin for the processing co-ordinate system is at the top left-hand corner of the screen.

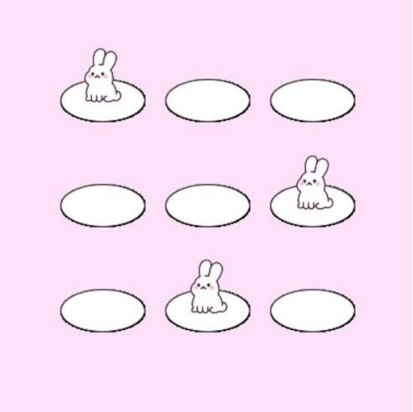

Figure 3.2 Bunny in random locations

The problem was hard to solve when applying the random function especially when one mistake can cause the meerkats to only appear in one specific location. The solution to the problem is to set each meerkat to a random function variable.

The other problem faced was the countdown timer working backwards. The game over screen would appear when the time is at 0 however it appears at the start of the game. The timer would countdown from -1 and further negatively. To provide a solution the problem, the use of Boolean operators was effective.

4 Solution

The game structure consists of three main components: the start screen, the play area, and the game over screen. The play area holds the main idea of the clone Whack-a-Mole game. Like the moles in the original Whack-a-Mole game, the meerkats in the project need to appear at a specific location randomly.

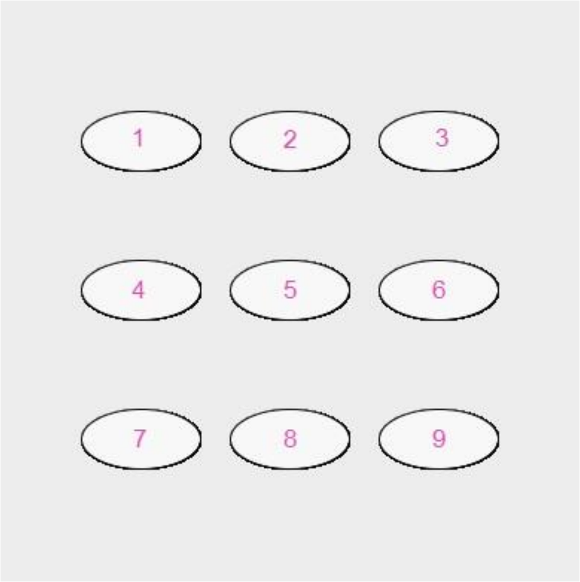

Figure 4.1 Random function applied to position

The random function is used but each meerkat is set at a given a variable number from 1 to 10. Each meerkat is given a location in the play area and would only appear when they are called upon. The meerkats are in position and the random function allows them to appear randomly from 1 to 10. This way the game can perform its duties as a clone version of the Whack-a-Mole game. However, one problem was not solved and that was the tenth position of the meerkat, it appears at the top left corner of the screen. This problem was a thorn in the whole game and covering it up with a shape was done.

The problem involving the countdown timer was solved with the use of Boolean operators. As the timer starts at 30, it continues to decrease until it reaches 0. The Boolean operator for relational ordering "<=" was used so when the timer is equal to or less than 0, the game over screen appears.

Originally the Boolean operator used was "== 0", when it reaches 0 the game over screen would appear but a third-party member on the discourse.processing.org forum suggested it to be "<=" instead. The reasoning for this is that the countdown timer for the game would need to be less than or equal to 0 for the game over screen to appear, if the timer is equal to exactly 0, only then would the countdown continue to decrease.

5 Evaluation

The game was tested by slapping the meerkats receiving a score of ten while waiting for the timer to end. The test cases are shown in figure 5.1, the game over screen displays the scoreboard with a score of ten as well as the replay and exit button.

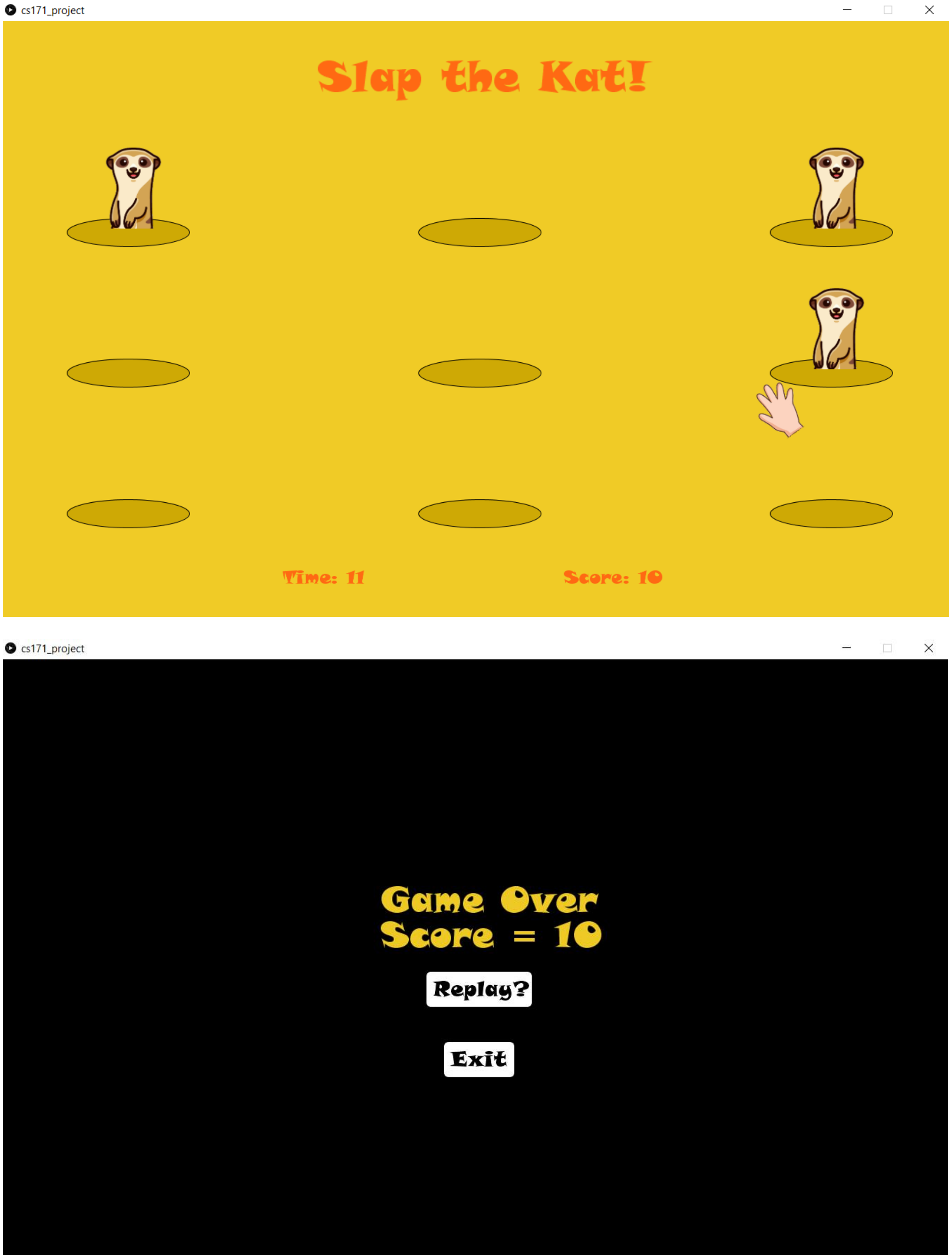

Figure 5.1 The meerkats slapped resulted in the total score of ten. Countdown timer at 0 transports to another screen. The game over screen displays replay button, exit button and the scoreboard.

6 Conclusion

A simple clone version of the popular arcade game has been created for the enjoyment of individuals. The game includes the start screen, play area and the game over screen which was needed to clone the popular arcade game. There could be further development of the game such as creating meerkats that the player shall not slap. The code could be manipulated so that the meerkat touched would decrease the player's points.

In conclusion, if the code were extended to include different levels of difficulty, the game would be much more pleasant to play.

References

[1] Website – https://forum.processing.org./two/discussion/20024/

[2] Website – https://forum.processing.org/one/topic/how-to-create-countdown-timer/

[3] Website – https://www.openprocessing.org/sketch/106739

[4] Book – Learning Processing: A Beginner's Guide to Programming Images, Animation, and Interaction by Daniel Shiffman

[5] Website – https://en.wikipedia.org/wiki/Whack-A-Mole

[6] Website – https://processing.org/reference/PImage/

Appendix A Source code listing

```
//Simple clone game of Whack-A-Mole
//C. Song

//SoundFile file;
import ddf.minim.analysis.*;
import ddf.minim.*;
Minim       minim;
AudioPlayer victory;

PFont font;
PImage menu;
PImage slap;
PImage meerkat;
PImage smashed;
int stage; /*This is to indicate the start screen or the play area
--
>the site that gave the info //https://forum.processing.org/two/discussion/20024/ur
gent-please-i-want-to-create-a-startscreen-for-my-pong-game-what-am-i-doing-wrong
*/
int x;
int y;
int x1;
int y1;
int x2;
int y2;
int lastTime;
int counter = 30; /*Counter equals the amount of time a player has to slap the
meerkats
-->this could also be in void setup();*/
int lastCounter;
int delayAmount;
int diameter = 120;
int score = 0;
//the use of boolean for datatype to be either true or false, useful for the progra
m to
boolean whack = false;
boolean whacked1 = false;
boolean whacked2 = false;
boolean clickScore = false;

//Create window
void setup()
{
  //Soundfile from the data folder of the sketch
  minim = new Minim(this);
  victory = minim.loadFile("music.wav"); //The music file is loaded into memory
  victory.loop(); //Sounfile is played in a loop
  victory.setGain(-7);//This decreases the volume of the original music track

  stage = 1;
  size(1350,850);
  lastCounter = millis(); //this returns the time in milliseconds
  speed();
  smooth();
  noStroke();
  noCursor();

  //Loading images and image resizing
  menu = loadImage ("menu.jpg");
  menu.resize(1350,850); //this resizes the image to fit the whole screen
  slap = loadImage ("slap.png");
  slap.resize(0,120);
  meerkat = loadImage ("meerkat.png");
  meerkat.resize(0,120);
  smashed = loadImage ("smashed.png");
  smashed.resize(0,1);
}//end

void draw()
{
  if (stage == 1){ //if stage is 1 it will display the start screen
    background(menu);
    fill(0,0,0);
    textSize(60);
    text("Welcome to Slap the Kat!", 270, 200);
    textSize(30);
    text("Press any key to start the game", 440, 300);
    if (keyPressed == true){ //when any key is pressed, the stage is 2
      stage = 2;
    }
  }
  else if (stage == 2){ //when stage is 2 the start screen changes to the play area
    background(240, 203, 38);
    stroke(0);
    fill(206, 169, 5);
    //Drawing of the 9 holes
    //First row of holes
    ellipse(180,300,174,40);
    ellipse(680,300,174,40);
    ellipse(1180,300,174,40);
    //Second row of holes
    ellipse(180,500,174,40);
    ellipse(680,500,174,40);
    ellipse(1180,500,174,40);
    //Third row of holes
    ellipse(180,700,174,40);
    ellipse(680,700,174,40);
    ellipse(1180,700,174,40);

    /*Kayla Arias inspired code
    --
>this part was inspired by Kayla Arias on the openprocessing.org because it was
    a bit difficult to get the images of meerkats to appear in specific location bu
t randomly
    --
>this piece of code displays the meerkats only when the player whacks the area wher
e meerkats show up*/
    if (whack == true) {
      image(smashed, x, y);
    }
    else {
      image(meerkat, x, y);
    }
    if (whacked1 == true) {
      image(smashed, x1, y1);
    }
    else {
      image(meerkat, x1, y1);
    }
    if (whacked2 == true) {
      image(smashed, x2, y2);
    }
    else {
      image(meerkat, x2, y2);
    }

    /*The slapping of meerkats
    -->processing.org has info on this piece of code
    -->the image of the hand is rotated -
25 and the popMatrix() allows the hand to have the push/pop effect
    */
    pushMatrix();
    if (mousePressed) {
      rotate(-25);
    }
    image(slap, mouseX-50, mouseY-120);
    popMatrix();

    //Play area screen
    font = loadFont("SnapITC-Regular-48.vlw");
    fill(255, 104, 19);
    textFont(font,32);
    textSize(26);
    text("Score: "+score, 800, 800);
    text("Time: "+counter, 400, 800);
    textSize(60);
    text("Slap the Kat!", 450, 100);

    /*Kayla Arias inspired code
    --
>as mentioned it was difficult to get the meerkats to appear in specific locations
but randomly
    --
>if mousePressed then the click will have to be within the diameter of the meerkat
    */
    if ((mousePressed == true) && (dist(mouseX, mouseY, x, y) <= diameter)) {
      whack = true;
    }
    else {
      whack = false;
    }

    if ((mousePressed == true) && (dist(mouseX, mouseY, x1, y1) <= diameter)) {
      whacked1 = true;
    }
    else {
      whacked1 = false;
    }

    if ((mousePressed == true)  && (dist(mouseX, mouseY, x2, y2) <= diameter)) {
      whacked2 = true;
    }
    else {
      whacked2 = false;
    }

    if (millis() - lastTime > delayAmount) {
      //Meerkats appear randomly in specific loacations
      int randomMole= int (random(1, 10)); //From locations 1-10 the meerkats are
appearing randomly
      if (randomMole == 1)
      {
        x = 140;
        y = 175;
        x1 = 1140;
        y1 = 375;
        x2 = 1140;
        y2 = 175;
      }

      if (randomMole == 2)
      {
        x = 640;
        y = 175;
        x1 = 1140;
        y1 = 575;
      }


      else if (randomMole == 3)
```

```
  {
    x = 1140;
    y = 175;
    x1 = 1140;
    y1 = 375;
  }


  else if (randomMole == 4)
  {
    x = 140;
    y = 375;
  }


  else if (randomMole == 5)
  {
    x = 640;
    y = 375;
  }

  else if (randomMole == 6)
  {
    x = 1140;
    y = 375;
  }
  else if (randomMole == 7)
  {
    x = 140;
    y = 575;
    x1 = 140;
    y1 = 175;
    x2 = 1140;
    y2 = 575;
  }
  else if (randomMole == 8)
  {
    x = 640;
    y = 575;
  }
  else if (randomMole == 9)
  {
    x = 1140;
    y = 575;
  }
  else if (randomMole == 10)
  {
    x = 140;
    y = 175;
    x1 = 1140;
    y1 = 375;
    x2 = 1140;
    y2 = 175;
  }
  speed();//this controls the speed of the meerkats popping out
}

/*Timer countdown
-->Processing.org has this info
-->int lastCounter minus the int lastCounter less than 1000*/
if (millis() - lastCounter > 1000) {
  counter -= 1; //counter = counter -1; --> same as -=
  lastCounter = millis(); //lastCounter in milliseconds
}
/*A mistake in the program i wasn&apos;t able to fix on my own
-->a meerkat appears in the top left corner for some reason
-->i tried to use this piece of code which covers it up*/
  noStroke();
```

```
   fill(240, 203, 38);
   rect(0,0,100,150);

  /*I needed help on this part of the code
  -->the timer was going backwards negatively
  -->paulgoux on discourse.openprocessing.com forum helped resolve this
  --
>the mistake was if(counter == 0) which i changed to if(counter <= 0) to fix it
  --
>counter had to be less than or equal for the timer to not get backwards negatively

  even after displaying game over screen*/
  if (counter <= 0){//when time is less than or equal it will display game over
screen
   background(0);
   cursor();
   textSize(50);
   text("Game Over", 540, 360);
   text("Score = " + score, 540, 410);

   //This piece of is for the replay button on the game over screen
   //Draw replay button
   rectMode(CENTER);
   fill(255);
   rect(680, height-380, 150, 50, 7);
   fill(0, 0, 0);
   textSize(30);
   text("Replay?", 615, height-370);

   //Click to replay the game
   if (mousePressed) {
    if (mouseX > (width/2 - 75) &&  mouseX < (width/2 + 75) && mouseY < height-
380 + 25 && mouseY > height-380 - 25) {
    fill(0);
    rect(670, height-380, 150, 50, 7);
    reset();//if the replay button is clicked the game resets
    }
   }

   //This piece of is for the exit button on the game over screen
   //Draw exit button
   rectMode(CENTER);
   fill(255);
   rect(680, height-280, 100, 50, 7);
   fill(0, 0, 0);
   textSize(30);
   text("Exit", 640, height-270);

   //Click to exit the game
   if (mousePressed) {
    if (mouseX > (width/2 - 75) &&  mouseX < (width/2 + 75) && mouseY < height-
280 + 25 && mouseY > height-280 - 25) {
    fill(0);
    rect(680, height-280, 100, 50, 7);
    exit();//like the ball and bat lab, exit() is used here to click off the
game
    }
   }
  }
 }
}//end

/*Discourse.openprocessing.com has this info
-->the speed of the meerkat appearing is controlled by this void speed()
*/
void speed() {
 lastTime = millis();
 delayAmount = int(random(1, -2) * 1000);
```

```
}

/*This is what happens when replay button is clicked on
--
>everything will be like the start of the game, score is 0 and timer is starts from
 30 secs
*/
void reset() {
  score = 0;
  counter = 30;
  noStroke();
  noCursor();
}

/*Discourse.openprocessing.org has this info
-->for each click on the meerkat, the score is added by 1
*/
void mouseClicked() {
  clickScore = true;
  if ((clickScore == true) && (whack == true)) {
    score += 1; //could also be score = score +1
  }
  if ((clickScore == true) && (whacked1 == true)) {
    score += 1;
  }
  if ((clickScore == true) && (whacked2 == true)) {
    score += 1;
  }
}
```