|  |
| --- |
| Conestoga College |
| Set-Jab Final Project |
| PROG2215 – Graphics, Animation and Sound |

|  |
| --- |
| William Pring  4-19-2016 |

Contents

[Overview 3](#_Toc448840167)

[Functional Requirement 3](#_Toc448840168)

[3.1 Import/Convert images and audio 3](#_Toc448840169)

[3.3 Developed procedural animation sequence 5](#_Toc448840170)

[3.4 Export final Product 6](#_Toc448840171)

# Overview

The purpose of this assignment is to recreate JIB-JAB but with a certain requirement such as Resizing, moving, spinning, playing music, exporting to AVI file and playing with a custom player.

# Functional Requirement

## 3.1 Import/Convert images and audio

I load all of the images to a bitmap pointer, I load the music at the end after I create the AVI file. With these two images pointer I set it to another bitmap but using it as a thumbnail. I use the OnEraseBkgnd which is part of the afx\_msg to be double buffering. For the resizing of the images I used the onSize function which is part of the afx\_msg

This statement bellow will add the music to the AVI after I finish with the creation of the video with the AVI. These statement belong to the AddingMusicFinish

AddAviWav(avi, "music.wav", SND\_FILENAME);

CloseAvi(avi);

This function is called every time the screen gets resized. What it is doing to changing the screen size in particular the background. It also used when the project is started off by executing the timer and setting the size of the heads and bitmaps.

void CChildView::OnSize(UINT nType, int x, int y)

{

RECT screenSize;

GetWindowRect(&screenSize);

int xWidth = screenSize.right - screenSize.left;

int yHeight = screenSize.bottom - screenSize.top;

if (xWidth > 0 && yHeight > 0)

{

delete head;

delete head1;

sizeX = xWidth \* 0.2;

sizeY = yHeight \* 0.2;

head = (Bitmap\*)orgHead->GetThumbnailImage(sizeX, sizeY);

head1 = (Bitmap\*)orgHead1->GetThumbnailImage(sizeX, sizeY);

}

if (init)

{

SetTimer(1, 1000 / 30, NULL);

myHead.initProject(xWidth, yHeight);

myHead1.initProject(xWidth, yHeight);

init = false;

}

delete displayNewBackground;

displayNewBackground = (Bitmap\*)bmpBackground->GetThumbnailImage(xWidth, yHeight);

}

This check Collison is checking if the head will collide with one another if so it will then call a function which is part of the myHead and myHead1 to change the directions. But this is a little flawed because it possible to continue on to the same path because changeDirAndSpeed() randomly change the directions of the head velX and velY. If there is a Collison that dose occur I return a bool which the onTimer will know about and execute accordingly.

bool CChildView::CheckCollision()

{

bool retStatus = false;

if (myHead.getX() < (myHead1.getX() + sizeX) && (myHead.getX() + sizeX) > myHead1.getX())

{

if (myHead.getY() < (myHead1.getY() + sizeY) && (sizeY + myHead.getY()) > myHead1.getY())

{

myHead.changeDirAndSpeed();

myHead1.changeDirAndSpeed();

retStatus = true; //Collided with box

}

}

return retStatus;

}

This is the original screen



//This screen is the shrink down image for the previous screen



## 3.3 Developed procedural animation sequence

Everytime I create a new head I randomly assigned new coordinates. Every 5 seconds in the onTimer I change the direction and the speed of the head and every 3 seconds it spins left or right. The head bounce and reflect off the wall. I set the onTimer to 1000/30.

The timer starting

SetTimer(1, 1000 / 30, NULL);

The onTimer that will execute and trigger different event and function base on the time and frame that have been executed. This keep track of a counter that I keep updating which is the frames that is current been executed.

if (counter % (30 \* 5) == 0)

{

myHead.changeDirAndSpeed();

}

if (counter % (30 \* 3) == 0)

{

randLeftOrRight = (rand() % (3 - 1)) + 1;

}

bool status = CheckCollision();

if (status == true)

{

myHead.changeDirAndSpeed();

}

if (counter > (500)) {

AddingMusicFinish();

KillTimer(1);

}

else {

myHead.move();

myHead1.move();

this->Invalidate();

}



## 3.4 Export final Product

I use the Lucian Wischik library for the AVI export. What I do is have another bitmap and redraw what I draw on the screen the AVI file. After a certain amount of frame, I stop creating the video and add a music file to it and export as an AVI file. I then create my own function called AddingMusicFinish which will add a music file and then close the AVi file. There was limitation to my AVI file. Even though I have 30 frames

## 3.5 Export final Product

I did not meet this requirement as I did not create a view for my music player. I used the school already install VLC to play my video.

# Bibliography

fscholz, t. c. (2015, Aug 4). *mozilla*. Retrieved from 2D collision detection: https://developer.mozilla.org/en-US/docs/Games/Techniques/2D\_collision\_detection

msdn. (n.d.). *msdn*. Retrieved from https://msdn.microsoft.com/en-us/library/d06h2x6e(v=vs.100).aspx

Wheeler, J. (1998, July 30). *Using Timers in MFC Applications*. Retrieved from http://www.developer.com/net/cplus/article.php/603531/Using-Timers-in-MFC-Applications.htm

Wischik, L. (n.d.). *AVI example code for creating AVI files*. Retrieved from wischik: http://www.wischik.com/lu/programmer/avi\_utils.html#code