## VBugs

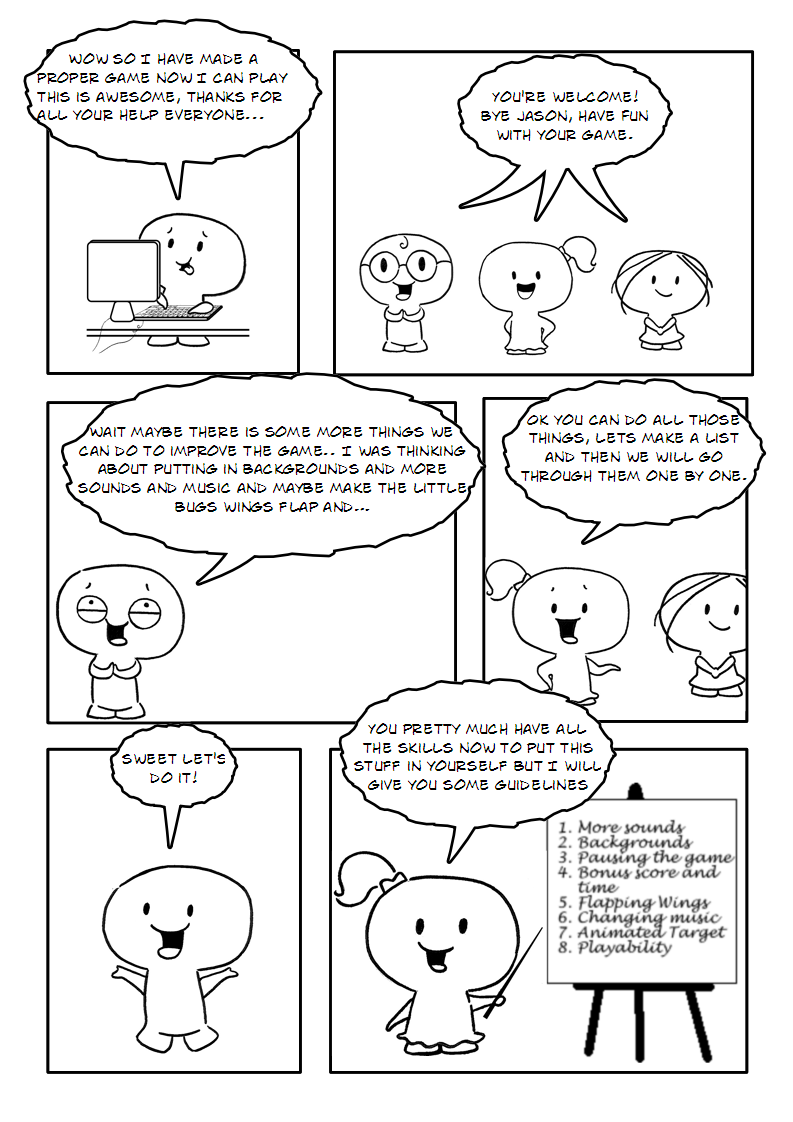
## Chapter 9

###### comic 1 - 1a.pngExtensions and Additions



# Summary:

This chapter will give you some ideas for extensions and additions to your already functioning Bugs game. Some resources and guidelines have been included.



## Extensions and Additions

### 1. More Sounds

A game over sound(gameover.wav) and a penalty sound (penalty.wav) have been provided for you in your resources folder. Your task is to make these sounds play at the appropriate time (when the penalty screen comes up and the loser screen comes up).

### 2. Backgrounds

Load a background into your game. Three images have been provided to you in the Resources folder (You can make your own or add as many different ones as you like with a program like Fireworks or MSPaint). Your task is to figure out how to get them to display in the background of the game and when the level completes to go to the next background image. You will need to tell your program when it has run out of images to go back to the first image so it loops.

*Hints: You will need a new variable for background number and an If Else statement to say when the background number is over a certain amount set it back to 1 Else add 1 to the background number.*

### 3. Pausing the Game

Create a pause screen that comes up when a user hits “P”. The pause screen should pauses the game timer display a pause screen that loops until the user either closes the window or hits “P” again. Make sure you un-pause the timer after the loop. Pausing can be called by using:

Core.PauseTimer(gameTimer)

And at the end:

Core.UnpauseTimer(gameTimer)

There is also a sound for pause (pause.wav) in your resources folder. Set it to play when the game is paused.

### 4. Bonus Score and Time

Set game so that when the user has killed a certain amount of bugs they get a score bonus and the timer restarts. Set it so when they kill a certain amount of bugs they get a different bonus 100 for 10, 500 for 20 etc. Let the player know when they have got a bonus by using a similar screen to the penalty screen. You can do the different score events by using a select case statement as demonstrated by the following pseudocode:

Select Case bugsKilled

Case 10

add 100 to the score and

call the bonus screen

Case 20

add 500 to the score and

call the bonus screen

End Select

Add the bonus sound (resources folder) to the bonus screen. Also it’s better if you stop and restart the timer when the bonus screen comes up. That way they get a time bonus as well.

### 5. Flapping Wings

The bugs when they fly around at the moment are a static image. We would like to replace this image with that of a flying bug. The flying bug image has been provided in your resources folder (bugFly.png). Add this image (Figure 1) to your project and use it to replace the current static bug image. You will need to go into Bug.vb to do this. The bugs are 40 x 40pixels and there are five bugs that make up the animation. You should set the bugs so that each bug shows thirty times before moving on to the next (see Chapter 5)

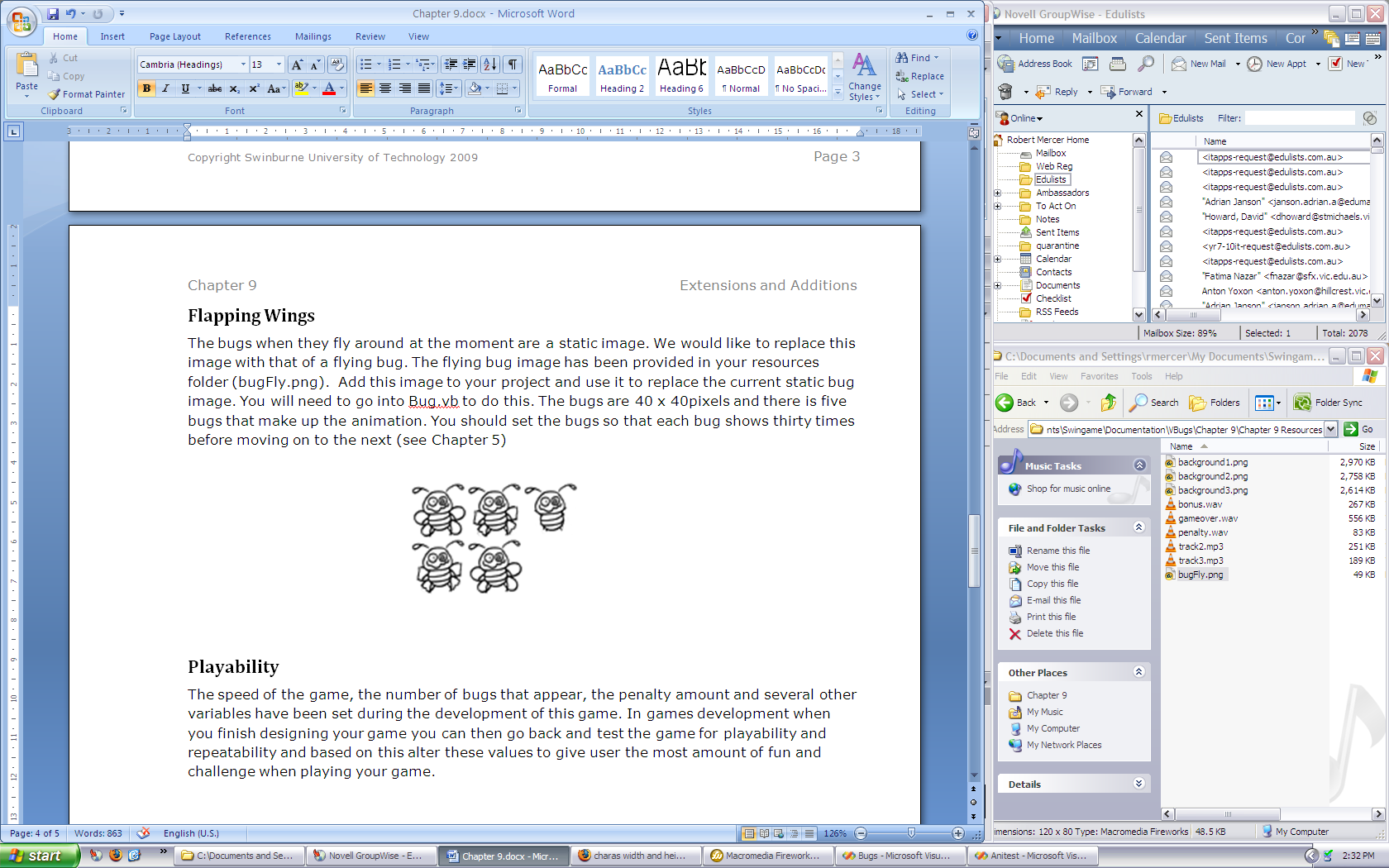


Figure 1

### 6. Changing Music

At the moment the game only has one piece of music it plays (lion.mp3). In the resources folder there is 2 more pieces of music (track2.mp3 and track3.mp3). We want to set our program so that when it changes levels it will play the next track. After adding the two new pieces of music to our project we need to rename them so they are consistent (track1, track2 and track3). Then we can create a new variable (trackNumber) like the one we did for the backgrounds that represents the track number we are currently playing. Then make a new sub that is called when the level is set up that adds a number to this variable and then starts the new track(Audio.PlayMusic(GameMusic("track" & trackNumber), -1)). Remember we need to set this so that if trackNumber is greater than 3 then it goes back to track number 1(like we did for the backgrounds so they loop).

In the game loop we can also set it so that the user can flick to the next track by hitting a key, perhaps the right arrow key. This is easy now we have a sub that tells the program to go to the next track. Now you have 3 pieces of music in your game you can easily add more mp3s from your own library, just remember to keep the naming consistent.

### 7. Animated Target

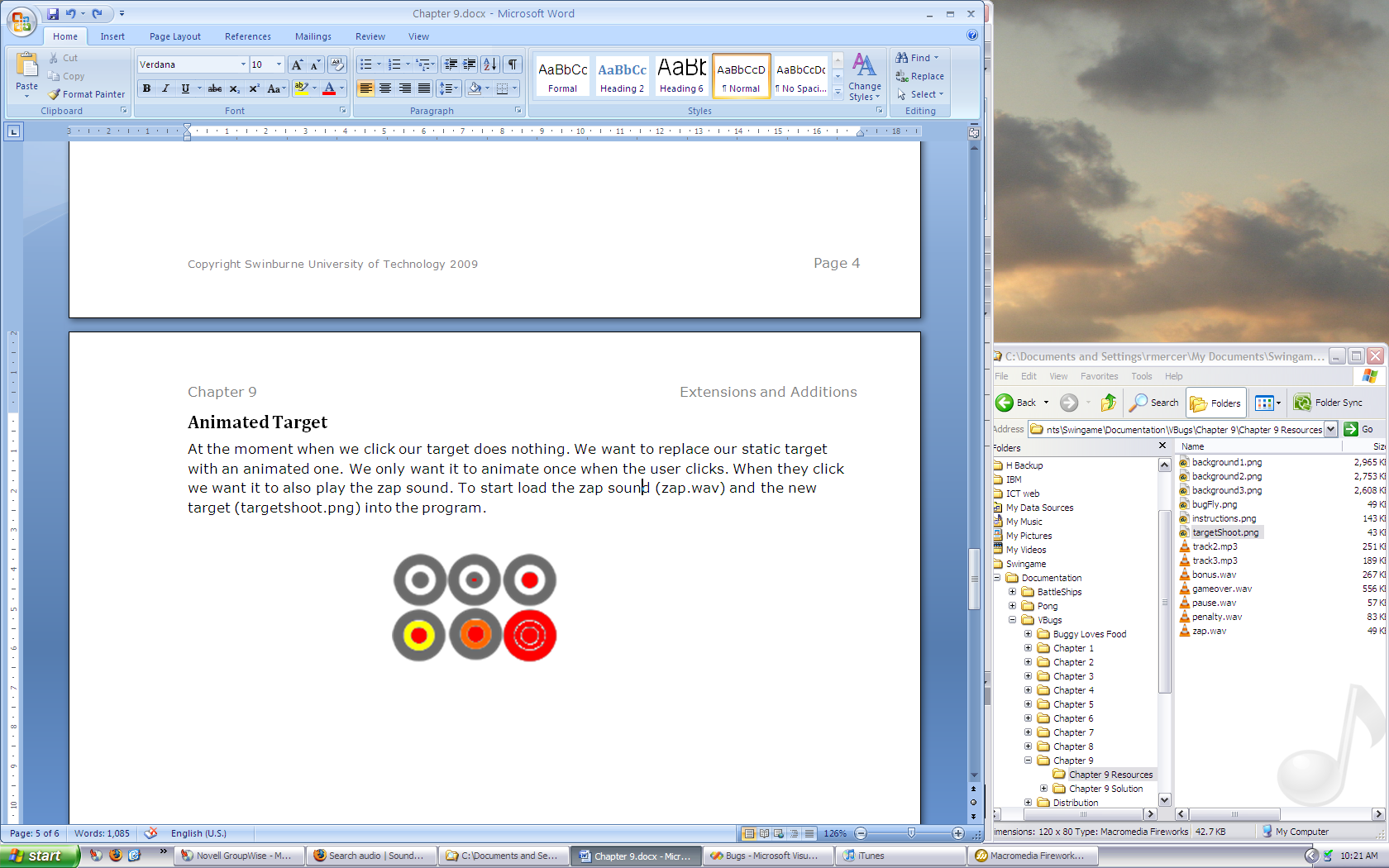
At the moment when we click our target does nothing. We want to replace our static target with an animated one (Figure 2). We only want it to animate once when the user clicks. When they click we want it to also play the zap sound. To start load the zap sound (zap.wav) and the new target (targetshoot.png) into your program.

Figure 2

Next we need to create two new Private variables in the GameLogic.vb; one for new the sprite targetShoot and one for ZapNow. ZapNow will be a Boolean variable and it will store either true or false depending of if the user has clicked. If they have clicked then ZapNow will be set to “True” until the targetShoot animation has been played.

In the main sub before the game loop we need to add the animation details to program:

targetShoot = Graphics.CreateSprite(GameImage("targetShoot"), 8, 7, 40, 40)

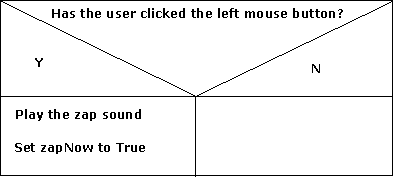
Then we can go into the DrawMouse() sub and alter it so it plays this animation when we want it to. We want to get rid of the line that draws the original target and replace it with the following logic:

Figure 3

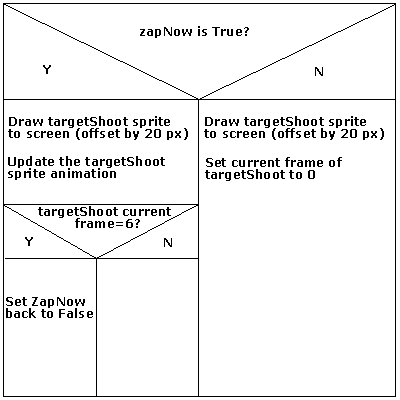
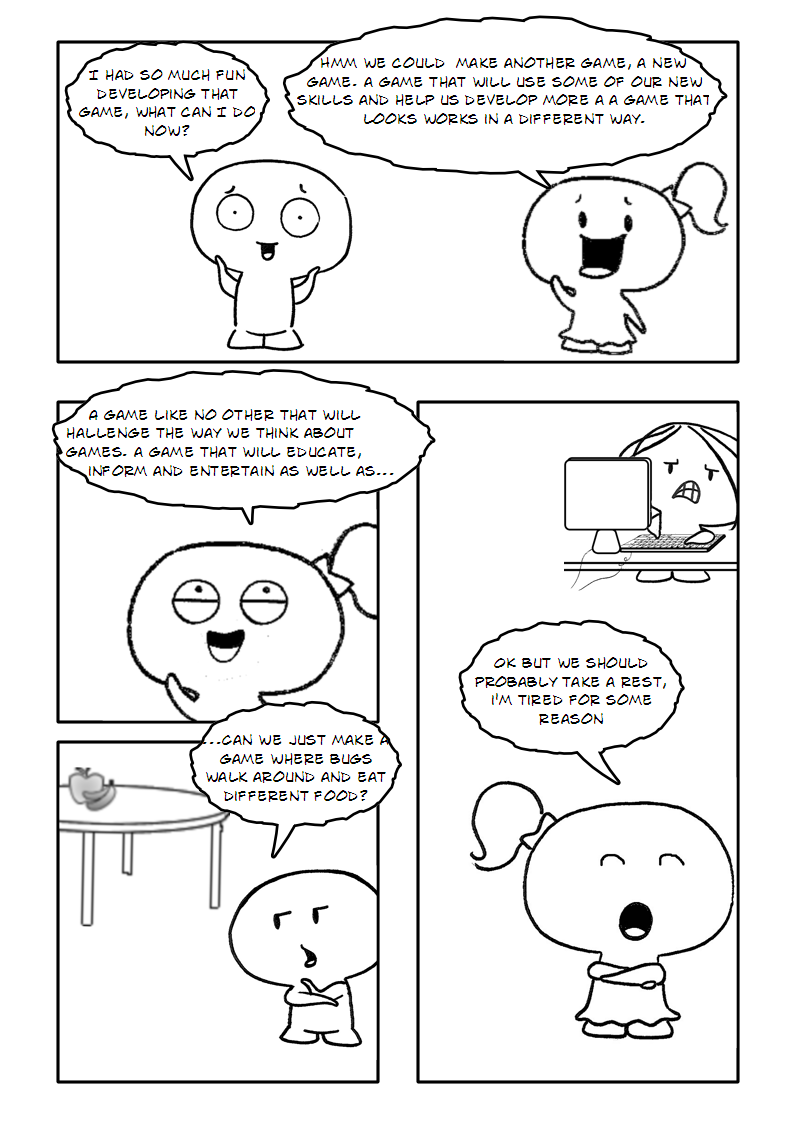


Figure 4

### 8. Playability

The speed of the game, the number of bugs that appear, the penalty amount and several other variables have been set during the development of this game. In games development when you finish designing your game you can then go back and test the game for playability and repeatability and based on this alter these values to give user the most amount of fun and challenge when playing your game.



## Use your Imagination

Remember that this is **your** program now you can get it to do whatever you like. See if you can think of any other cool additions you can make to the code. Have fun and when you have finished with this game you can move onto “**Part 2 - Buggy Likes to Eat**” which will build on your already existing knowledge on game development. Remember to check out [www.swingame.com](http://www.swingame.com) for any cool new development with the SwinGame Software Development Kit which you are fast becoming an expert in.

## SwinGame Competition

Check out the competition details on the SwinGame site(www.swingame.com). You might be interested in making your own game by yourself or with your friends and submitting it for a chance to win. The competition is a great way to challenge your creative game development skills and maybe even win some great prizes. The site offers forums and message boards which are supported by experts in the SDK if you get stuck in the development of your game you can ask questions there and you will get an answer and some solid advice. You can check out previous year’s winners and their games also on the site. Good Luck!