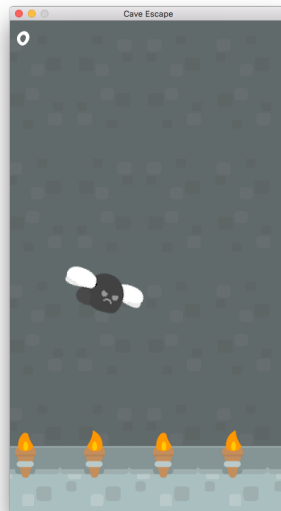


SWINBURNE UNIVERSITY

Cave Escape Presentation Guide

Authors:

Jake RENZELLA and Reuben WILSON



March 9, 2016

1 Introduction

Cave Escape is a clone of the popular mobile game Flappy Bird which has been made in Pascal using SwinGame. This presentation uses snapshots of the games progress as it was developed.

The aim of the presentation is to show the audience the basics of how game development works, just how simple making a game can be, and how much fun they can have.

Throughout the course of the presentation, the Presenter will present different iterations of the code of the game, in different levels of completion and talk about what each new addition to the code does to the program/game.

All versions of code shown to the audience will be complete, meaning they are executable. The purpose is to allow the presenter to visualise with the audience what different blocks of code actually do to the game.

The presenter should have enough procedural programming knowledge and experience to talk through all lines of the code, and answer any questions about the code. It would be best to determine the general programming skill of the audience at large.

1.1 Installation

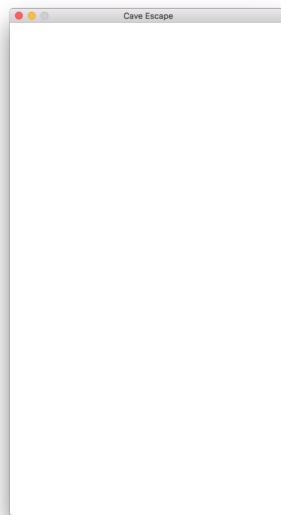
The development environment of Swingame using Pascal requires a few tools to be installed that the presenter or audience may not have. Thankfully, there are detailed instructions in the form of videos which run through installing the necessary tools to compile SwinGame.

The presenter will not need these tools to present, this package is pre-compiled and ready to go, however if the audience would like to alter the source code for their own versions of the game, they should be directed to the videos on [YouTube](#) and chose the installation videos.

2 Presentation

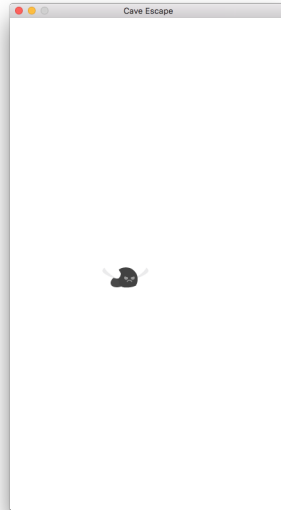
File 1

The first code file provided is a basic game loop in SwinGame. Running the application will open a graphics window, however it will not display anything (as nothing has been drawn to the window!). At this stage the presenter should talk about how the lines of code are run in sequence. (procedural programming). It is important to mention how procedures work and shift the code operation.



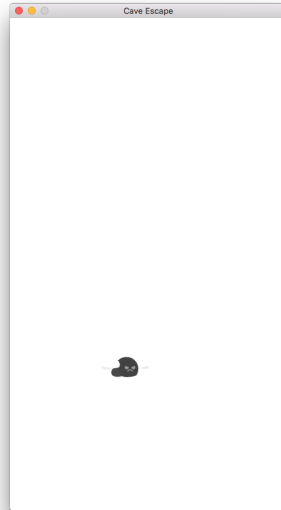
File 2

The second code file provided adds and draws a sprite of our player. Nothing is moving. Talk about how we added the sprite's image to the resource bundle, and how we load the image in our code.



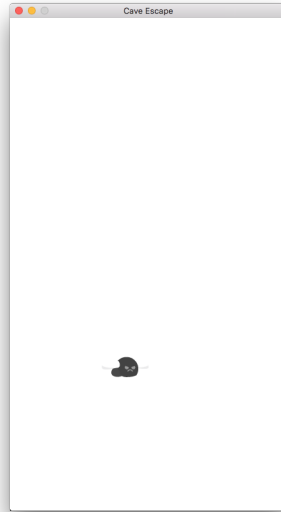
File 3

The third code file provided adds the functionality which makes the player fall (gravity), but no user input is handled so the player just falls off the screen! Talk about how this is a start, however is pretty boring.



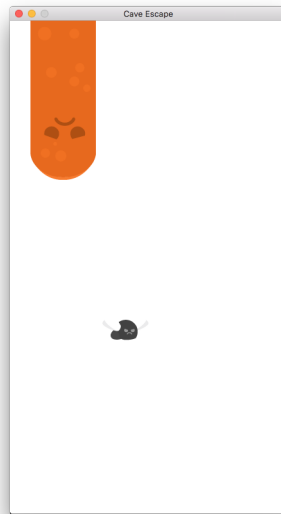
File 4

The fourth code file provided allows the player to jump when the user presses the space bar. Talk about how this is achieved in the code, how we alter the vertical speed of the sprite, where this data is stored etc.



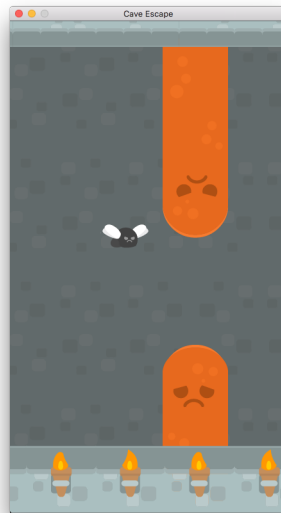
File 5

Version 5 of the code file adds in the poles bitmaps to the resources folder, and adds the code for their placement and drawing on the screen. It also adds the code for moving them off the screen and resetting them. This is a big jump in code so spend some time talking about how we place them, how we detect that they have moved off the screen, and how we reset them.



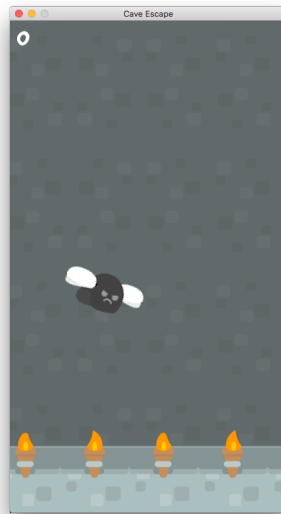
File 6

This version adds the code which adds the background sprites, the floor and the roof. Quickly mention how assets are important to a game.



File 7, Final Version

The final version adds collision with the poles and roof and floor. Talk about collision quickly. It also adds the music and score. Version 7 is the final source code for Cave Escape. We have animations, music, and a score to create a much more enjoyable and functional game. Talk about how simple a game like this can be (only 250 or so lines of code).



Conclusion

At this point, take any questions, ask the audience how we could possibly modify the game, how we could create other games etc.