COMP8049
Embedded Systems Engineering
Project – Part B

Completion Date: 20th December 2024

Value: 25 marks

Question 1

Explain how the code in defender_map_lab enables interrupt driven control over the timer, UART and lcd devices. Explain how the code implements and uses a “back buffer” for displaying game entities on the screen.

(3 marks)

Given the code in defender_map_lab.zip you are to (see the TODO comments in the code):

1. Add code to “save” the pacman game entities. That is if the defender sprite touches them. They are removed from the game.
2. Take out the timer value and put in the score e.g. how many pacman entities have been saved.
3. The defender entity should not be allowed to collide with an obstacle.
4. Sort the pacman array based on the x coordinate so that the pacman positions are in order. They are currently in “random” order i.e. the order in which they were added to the layer in tiled .
5. Combine the Tiles and pacman_tiles layers. Specifically there are two “for loops” in the Draw_all function. The 2nd loop iterates over the pacman_tiles array and draws a pacman tile (tile id 74) at the appropriate location. The pacman_tiles array is mostly zero and thus it is inefficient to loop over this array without applying some heuristic.
If you combine the arrays into one then on return from the function catch_pacman you will need to watch for changes to the isalive attribute in the appropriate pacmans array entry. The catch_pacman function will return the index into the pacmans array for the pacman entity that the ghost sprite is currently trying to “catch”. The catch_pacman function will return -1 when there is no pacman entity to be “caught”.

To modify/view the map in tiled – (note there is no need to do this – no marks), then see appendix 1.

(7 marks)

Question 2

The attached java code (ast.zip) is an interpreter for c. The code takes as input a clang generated Abstract Syntax Tree (AST) in a text file (out.txt in the ast2 directory) and traverses the tree while interpreting the nodes. The code only “works” for simple expressions and with integer variables. You are to:-

a) modify the code to handle more complex expressions, “i*f statements*”, “*while loops*” and “*for loops*”.

b) java classes have an outline gen() method which shows how the code could be used as a compiler to generate ARM assembly. Outline how you would generate code for *if* statements.

We will use clang to generate an AST for a given c simple c function. We use the -ast-dump command line argument to generate a textual representation of the AST, which is stored in a file. For example:

```
clang-check -ast-dump for2.c --extra-arg="-fno-color-diagnostics" > out.txt
```

The attached code reads in this text file and constructs a tree with a java object for each node in the clang AST. For example, the VarDecl class models a c declaration of a variable. When the interpreter sees a VarDecl AST node it creates a corresponding java variable to model it.

If the c variable is used in an expression in t1.c, the AST will have a corresponding DeclRefExpr node to model that use. When the interpreter sees the DeclRefExpr node in the AST it uses the current value of corresponding java variable when evaluating the expression.

For example, the c statement:

```
j = 9;
```

will be modelled using the following AST nodes (dumped using the -ast-dump flag to the compiler):

```
-BinaryOperator 0x30d7968 <line:3:1, col:3> 'int' '='
   | |-DeclRefExpr 0x30d7920 <col:1> 'int' lvalue Var 0x30d7890 'j' 'int'
   | `-IntegerLiteral 0x30d7948 <col:3> 'int' 9
```

The interpreter will “evaluate” each of these nodes and place the value 9 into the java variable j, which was created in the corresponding VarDecl statement.

(7 marks)

Question 3 - Answer Part A or Part B

Part A

Given the code in sdh-file.zip. Compile and run the code with mk.

The firmware includes a sd card driver for the versatilepb board. The c program in the file createfs.c creates a simple file system and places it in the file sdimage.

The file sdimage is used as the data associated with the sd card within the firmware. That is QEMU emulates the sd card using the file sdimage.

a) Write code for the ***ls*** command in the firmware which lists all the files in the file system and their sizes.

b) Write code for the ***cat*** command in the firmware which lists the contents of a given file.

c) Write code for the **mv** command in the firmware which renames a given file.

d) Write code for a **copy** command.

The createfs program takes a number of files as argument and creates the filesystem from these files. Compile createfs and run with:

```
gcc createfs.c
./a.out t.c vid.c
```

If you get the following error

> “qemu-system-arm: Invalid SD card size: 10.5 KiB SD card size has to be a power of 2, e.g. 16 KiB. You can resize disk images with 'qemu-img resize <imagefile> <new-size>'”

Then try:

```
qemu-img resize sdimage 16000
```

(8 marks)

Part B

a)

Give an overview of a STM32 based PCB Design using a tool such as KiCad. You do not need to install KiCad.

https://www.youtube.com/watch?v=avH-iordHFU

https://youtu.be/zRdJrE80Vjk

https://www.youtube.com/watch?v=aVUqaB0IMh4&t=2157s

https://youtu.be/vaCVh2SAZY4

https://youtu.be/gFmm91c_mr8

(4 marks)

b)

Give an overview of an aspect of drone firmware design. You may chose any drone firmware e.g. the PX4 architecture (

https://docs.px4.io/master/en/concept/architecture.html). The review is to focus on some aspect of the drone firmware design e.g. on sensors, actuators (e.g. motors) or communication mechanisms.

(4 marks)

Appendix 1

To modify/view the map in tiled – Note there is no need to do this – no marks

You can view/edit the map in tiled by opening the file ObjectLayer.tmx.
If you modify the map in tiled you will need to use the File→export option to generate corresponding c code. To do this you will need to copy the file tiled-to-gbdk-c-file-export.js (in the attached zip file) to the extensions directory.
You can find the extensions directory by going to
Open Tiled and go to Edit > Preferences > Plugins

The file tiled-to-gbdk-c-file-export.js has been updated to output an objects layer.

See here for original code and install instructions

https://github.com/djedditt/tiled-to-gba-export