

SEEM 3460/ESTR 3504 (2020)

Assignment 2

Due on November 6th, 2020 14:00

1. General Information

- There will be mark penalty for uninformed late submission.
- You must follow the guideline in this file, or there will be a mark penalty.
- You should develop your program under UNIX-like environment in server linux03-05.se.cuhk.edu.hk.
- Your assignment will be compiled by **gcc** and executed in UNIX-like systems.

2. Faculty Guideline for Plagiarism

- If a student is found plagiarizing, a very heavy punishment will be imposed. The definition of plagiarism includes copying of the whole or parts of programming assignments from other's work or the Web. The penalty will apply to both the one who copies the work and the one whose work is being copied.

3. Problem Overview

The objective of this assignment is to write a program to perform a magic show.

The demo of expected magic show can be found in this folder:

~seem3460/distribute/asgn2_demo

You can get access to see the magic show by running the following commands

```
> cd ~seem3460/distribute/asgn2_demo
> ./magic
```

You **MUST ONLY USE** the provided library (**display.c**) for handling the graphical user interface (GUI). You must follow the specification below.

- The GTK+2.0 library is required for this project. Currently only linux03-linux05 can compile your programs.
- Material for this assignment can be found in this folder:
~seem3460/distribute/asgn2

- To download the files to your own home directory, run the following commands

```
> cp -r ~seem3460/distribute/asgn2 ~/
> cd ~/asgn2
```

- You will get the following files in your folder ~/asgn2:
display.h and **display.c** contain helper functions to write GUI programs with GTK.
david.jpg and files in the **card** folder are images used in the program.
message.txt contains the messages displayed in the program.
functest.c is a demonstration on how to display text and images using **display.h/ display.c**
magic.c is the source code file you need to edit to **finish this assignment**.
Makefile is the make configuration to help you compile and run the program.
- NONE of the provided files should be changed except **magic.c**. You are expected to write a program in the file **magic.c**, which can be compiled to an executable magic by running **make magic**. This command will automatically do the following 3 **gcc** commands to compile your

program. Then you run the executable file **./magic** to perform the magic show specified in section 4.

```
sepc92:/gds/wwxu/asgn2[44] > make magic
gcc -c magic.c
gcc -c display.c `pkg-config --cflags --libs gtk+-2.0`
gcc -o magic magic.o display.o `pkg-config --cflags --libs gtk+-2.0`
sepc92:/gds/wwxu/asgn2[45] > ./magic
```

- The **functest.c** is provided for your reference on how to display text and image. You can compile and run the **functest** by running **make functest** and followed by **./functest**.

```
sepc92:/gds/wwxu/asgn2[47] > make functest
gcc -c functest.c
gcc -o functest functest.o display.o `pkg-config --cflags --libs gtk+-2.0`
sepc92:/gds/wwxu/asgn2[48] > ./functest
```

- For submission, you only need to submit the file **magic.c**.

4. Problem Specification

You may divide the whole assignment into three steps:

- Reading and displaying user's input. (Section 4.1)
- Displaying the text and image (get familiar with the given functions). (Section 4.2)
- Designing the logic of picking the hidden card and separating the remaining cards to two sets (you should fulfill the requirements). (Section 4.3)

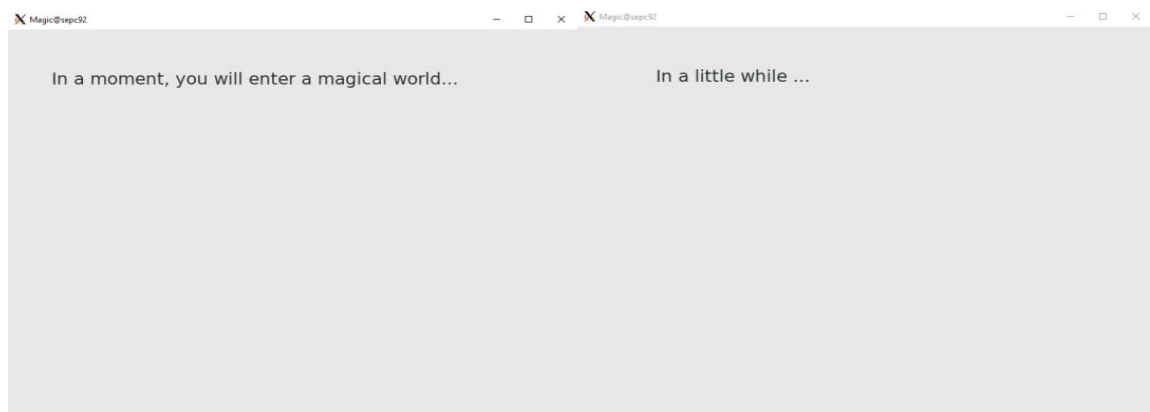
4.1 Flow of the magic show

In the lecture, you should have seen a full demo of the expected program.

- You are supposed to input your favourite fruit and the program will display “Your input is: X” in the console where X is your favourite fruit.

```
First of all, please tell me your favourite fruit: orange
Your input is: orange
```

- Your program is supposed to display text messages listed in **message.txt**. One line each time. You can hard code the message content in the source code. Therefore, you do not need to handle C file I/O.



- Your program is supposed to use function **wait_keypress** between two continuous messages i.e. when you press the space key, the program will continue to show the next message or image.

- When displaying the last message in **messages.txt**, your program is supposed to display the image of David Copperfield.



- In the specified points, your program is supposed to display some cards on the window according to the requirements in section 4.2 and section 4.3.

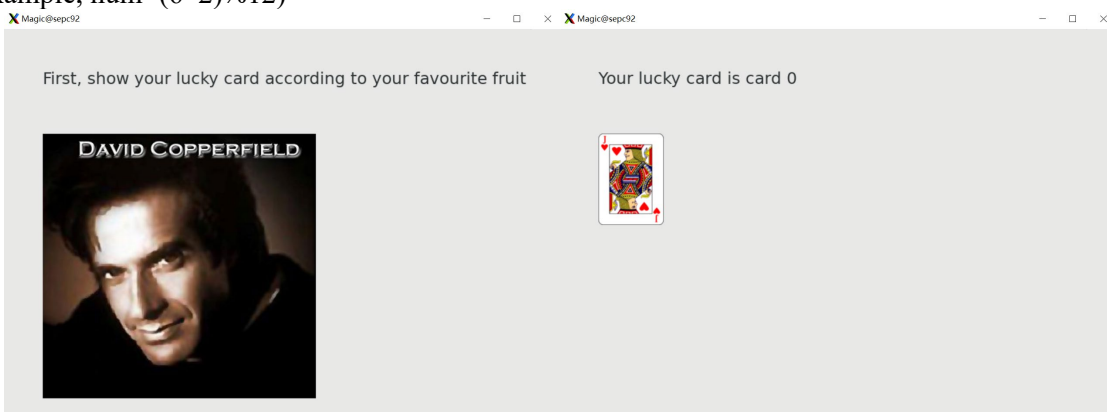
***Hint 1:** Read **functest.c** and you will know how to display text and images. To avoid extra confusing warning messages, function of click button at top-right corner of the window is disabled. To terminate the program, please press Ctrl+C in the console or finish running the whole program by pressing space key.

***Hint 2:** You can use function **sprintf** to write a message into an array of char.

4.2 Display Your Lucky Card

After displaying some messages in **message.txt**. Your program is supposed to display a message "First, show your lucky card according to your favourite fruit" (it is already written in **magic.c**), then it is supposed to show "your lucky card is card X" and an image **X.png** in the folder **card** where X is your lucky number. X is defined as the number of characters of your favourite fruit input multiplied by 2 and then modulo 12.

For example, if your favourite fruit is orange, the number of characters is 6. Then the lucky number is $(6*2) \bmod 12 = 0$. In C programming language, you can use operator **%** to obtain the results. (for example, `num=(6*2)%12`)



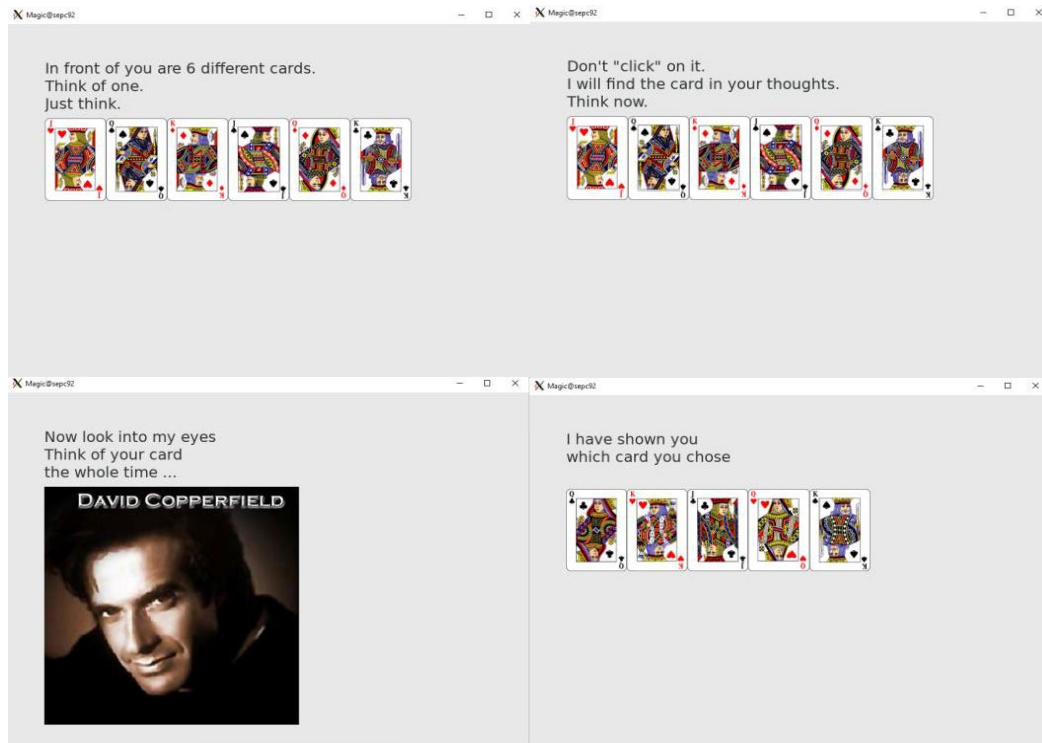
At this stage, your program is also supposed to continue by pressing space key.

***Hint 1:** Be careful of the **type** of your input. If the program reads a string, then you need some functions to get the length before computing the lucky number.

***Hint 2:** You may use **sprintf** to concatenate the string "Hello, " and the user input. You can refer to the **functest.c** file to see how we concatenate the string "Slide " and the integer. Remember to change **%d** to **%s** as we treat the user input as string instead of integer.

***Hint 3:** To display a card on GUI, you need to read **functest.c** and understand the usage of function **display_card**.

4.3 Card Display Randomness



In this magic, the trick is to hide all cards displayed in the first time. Here are the requirements on the card displays:

- You must show 6 cards in the first display and 5 cards in the second display
- You must not repeatedly display any card in the two card displays
- The card that you never display must be randomly selected from all 12 picture cards (this means by repeatedly running your program, every card will be the hidden card in some runs)
- The colour of the cards shown must be balanced (i.e. you must display 3 red cards and 3 black cards in the first display)
- The letter of the cards shown must be balanced (i.e. you must display 2 Jack's, 2 Queen's and 2 King's in the first display)
- Besides card display, your program is supposed to show messages in **content.txt**. Messages are separated by blank lines. You can customize the time to show these messages during card randomness display. You can hard code the message content in the source code. Therefore, you do not need to handle C file I/O.

You also have some freedoms on the card display:

- The card displays will only be considered as sets. You are free to design the card order in each display.
- The combination of colour and letter is free. E.g. you may set both Jack's red in the first display.
- The randomness requirement is only on the hidden card (the card never displayed). Therefore, it is up to you whether to make remaining things (display set, card order) random given a fixed hidden card.

***Hint 1:** Use function **srand()** to initialize random integer generator and use **rand()** to generate random number. Remember to take modulo 12 since the generated random number can be very large.

***Hint 2:** You may use the following equation to identify the card id:

$4n + k$, where $0 \leq n \leq 2$ and $0 \leq k \leq 3$

n represents the rank (0 is J, 1 is Q and 2 is K)

k represents the category (0 is heart, 1 is spade, 2 is diamond and 3 is club)

***Hint 3:** To display several cards at the same time, you need to read **functest.c** and understand the usage of function **display_card**.

4.4 Provided Module – display.h/display.c

In this assignment, you are required to show something on the GUI. To simplify your work, a module – **display.h/display.c** are provided.

```
#define MAX_CARD_TO_HOLD 6
#define true 1
#define false 0
#define SIGNAL_IS_WINDOW_DESTROYED false
#define SIGNAL_IS_NOT_WINDOW_DESTROYED true

void init_display(int argc, char *argv[]);
void display_text(char *message, int x, int y);
void display_card(int position, int card_id);
void display_david(int x, int y);
void display_card_clear();
void move_card_set(int x, int y);
int wait_keypress();
```

Function	Description	Parameters
init_display	Initialize the window	You need not manage argc and agrv. Just pass the same value as you get from main
display_text	Show a message	message: the message to show x, y: the coordinates of the text. (0,0) is at the top-left corner of window. (50,50) is recommended to display messages.
display_card	Show a card	position: 0 to 5 index of cards in a hand card_id: a number from 0 to 11. Each represents a poker card you will use.
display_david	Show photo of David Copperfield When you use this, all card display will be cleared	x, y: the coordinates of the display
display_card_clear	Clear all card display	
move_card_set	Set the position of the card set	x, y: the coordinates of the display. (0,0) is at the top-left corner of window
wait_keypress	Wait for a SPACE key press before continuing. Normally you do not need to care about the returned value of this function. If the user clicks close button at top-right corner of the window, the function will return SIGNAL_IS_WINDOW_DESTROYED, otherwise it will return SIGNAL_IS_NOT_WINDOW_DESTROYED	.

4.5 Additional Requirements for ESTR 3504

For ESTR 3504 students, you are supposed to read messages in **message.txt** and **content.txt** via C file I/O functions (e.g. **fscanf**, **fgets**) instead of hard coding them in the source code files.

The messages in content.txt are separated by blank lines. Each message may span multiple lines. Function display text supports showing a message that contains multiple lines separated by '\n' on GUI.

4.6 Submission

Please follow the submission procedures, so that we could ensure your assignment is received properly. Please use your own account to submit.

- Only **magic.c** is required.
- Before submission, **magic.c** MUST be renamed as <student- ID>-asgn2-magic.c (e.g. 1155XXXXXX-asgn2-magic.c)
- To submit the file, type the following command:
~seem3460/submit asgn2 <student- ID>-asgn2-magic.c
- When the program asks for your full student ID, type in your full student ID as follows:
What is your full student ID? 11XXXXXXX
- Then the program will summarize your personal information and asks for your confirmation. Type in “Y” to confirm your information. If you typed in your information wrongly, type “N” or press Ctrl+C to exit the program.
Your student ID: 11XXXXXXX; Your account name: XXXXX
Is the above information correct? (Y/N) Y
- NOTE: Submit using your own UNIX account, otherwise information will be wrong.
- Then you should see a message like this:

Connecting to SEEM3460 Submission Server...

sftp channel opened and connected.

Uploading...

Done! Thanks.

- If you see any error message apart from the above, then your assignment may not be submitted properly, try to submit again.
- Re-submission is not encouraged but allowed. Any submission will OVERWRITE previous submissions. Therefore, only the last submission will be graded.

If you have difficulties submitting your assignment, please send your student ID and <student-ID>-asgn2-magic.c to wwxu@se.cuhk.edu.hk.

5. Question & Answer

If you have any question, first check the following Q&A webpage

https://docs.google.com/document/d/1aRDeKBMxBxVub12rwdg7gK3Fxl5XEE6uqpL_ba4siFk/edit?usp=sharing

If you cannot find solution in the above webpage, send email to wwwxu@se.cuhk.edu.hk, after your problem being solved, the solution will be posted on the above webpage.