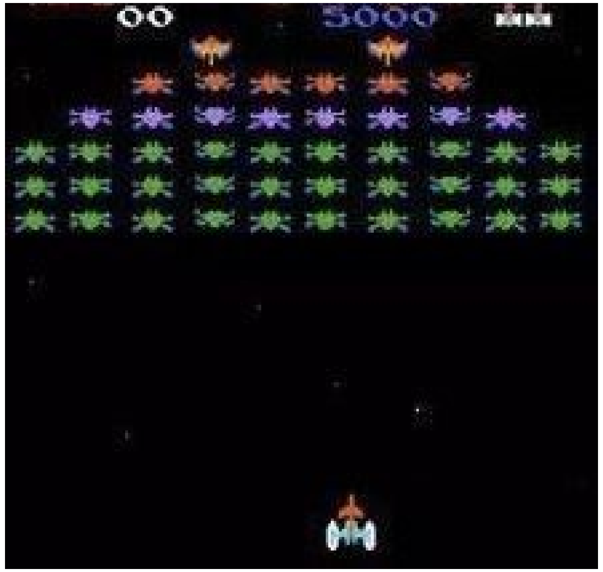
Final Project Tutorial

What is Allegro?

- Allegro = **A**llegro **L**ow-**Le**vel **G**ame **Ro**utines
- Created in 1990s.
- Written in C and supporting several platform, like windows, macos, linux-like platform and so on.



Before we start

- 1. bool type
    - A kind of data type in lots of programming languages
        - Like c++, java and so on.
    - It can only be assigned by two values:
        - 0 – means false.
        - 1 – means true.
        - You can just assign true or false to a bool variable. (e.g. bool a = true;)
    - Allegro implements a bool data type in itself.

Hello world

- In this section, we will learn how to program our first allegro program.

- For each allegro with display, we need to **initial allegro**, **initial display**, **set and show the display,** and finally, **destroy** all things have been initialized or created.

- Let’s do it!!

Hello world

- 1. Initial allegro
  - Step 1: *#include <allegro5/allegro.h>*

  - Step 2: Initialize.
  - It’s necessary to call the initialization function before doing anything else with the library.
  - After you calling *al_init():*
  - Great!!! The allegro is initialized. Now you can use any function you want in <allegro5/allegro.h>.

Hello world – cont.

- 2. Initial display
  - Step 1: Declare a pointer for display:
    - *ALLEGRO_DISPLAY* display = NULL;*

  - Step 2: Construct the display:
    - *ALLEGRO_DISPLAY* al_create_display(int width, int height);*

  - E.g. :
    - ALLEGRO_DISPLAY* display = NULL; //declare a display pointer
    - display = al_create_display(30, 50);   // construct the display

Hello world – cont.

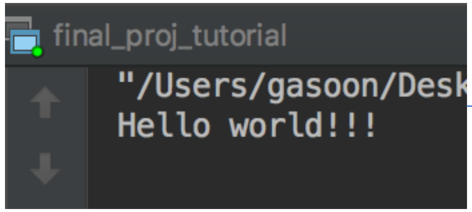
- Step 3: Set and show the display
  - To get color:
    - *ALLEGRO_COLOR al_map_rgb(unsigned char r, unsigned char g, unsigned char b);*
    - r, g, b are the integers belonging to [0, 255]
  - To set the whole window into the color:
    - *void al_clear_to_color(ALLEGRO_COLOR color);*
  - To show it:
    - *void al_flip_display(void);*

Hello world – cont.

- Step 4: Destroy
  - Using function below to destroy your display:
    - *void al_destroy_display(ALLEGRO_DISPLAY *display)*
  - Please destroy all things we have initialized or created before the prog ram closed.

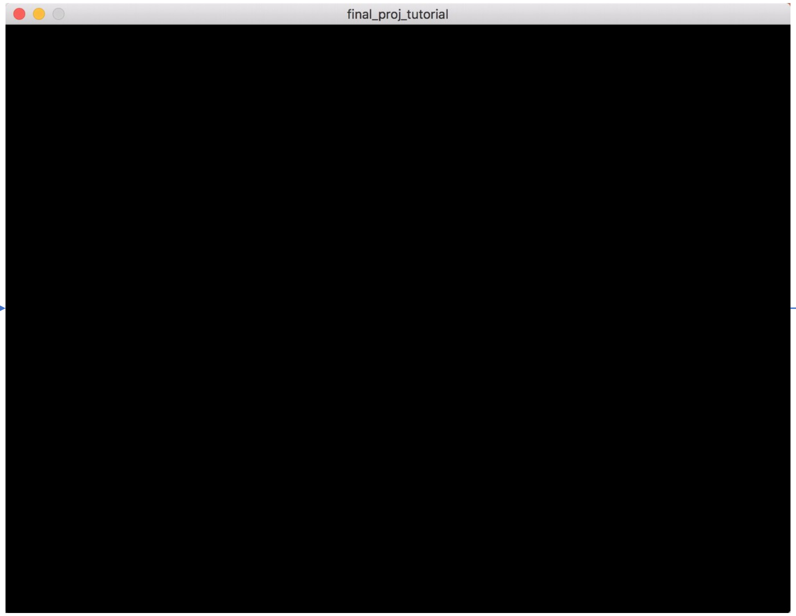
Task 1 – hello world

- In this task, we will program our first allegro project -- hello world!!

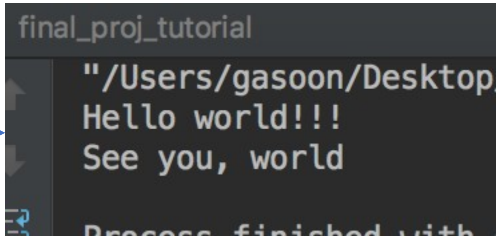


```
"/Users/gasoon/Desk
Hello world!!!
```

Wait 5 sec…



Wait 5 sec…



```
"/Users/gasoon/Desktop
Hello world!!!
See you, world

Process finished with
```

Task 1– hello world

- Please accomplish the task_1.c file.

- There’s lots of hint, hoping it might help you.

- Feel free to ask us if there’s any problem.

Draw something!!!

- In this section, we will learn how to show some words, graphic s or even images in our window.

Draw something!!! – cont.

- 1. draw text on the window:
  - Step 1: include the following head files:
    - *#include <allegro5/allegro_font.h>*
      #include <allegro5/allegro_ttf.h>
  - Step 2: call the following function to initial:
    - *al_init_font_addon();* // initialize the font addon
      al_init_ttf_addon(); // initialize the ttf (True Type Font) addon

Draw something!!! – cont.

- 1. draw text on the window:
  - Step 3:
  - *void al_draw_text(const ALLEGRO_FONT *font,   ALLEGRO_COLOR color, float x, float y, int flags, char const *text)*

  - *ALLEGRO_FONT *font*: the font of text will be displayed.
  - Using this to construct:
    - *ALLEGRO_FONT *al_load_ttf_font(char const *filename, int size, int flags)*
    - filename = the address of your own .ttf file
    - size = the size of letter.
      - Notice: You have to construct 2 kinds of font if you want use the same .ttf file but in different siz e.
    - flags = 0 is enough.

Draw something!!! – cont.

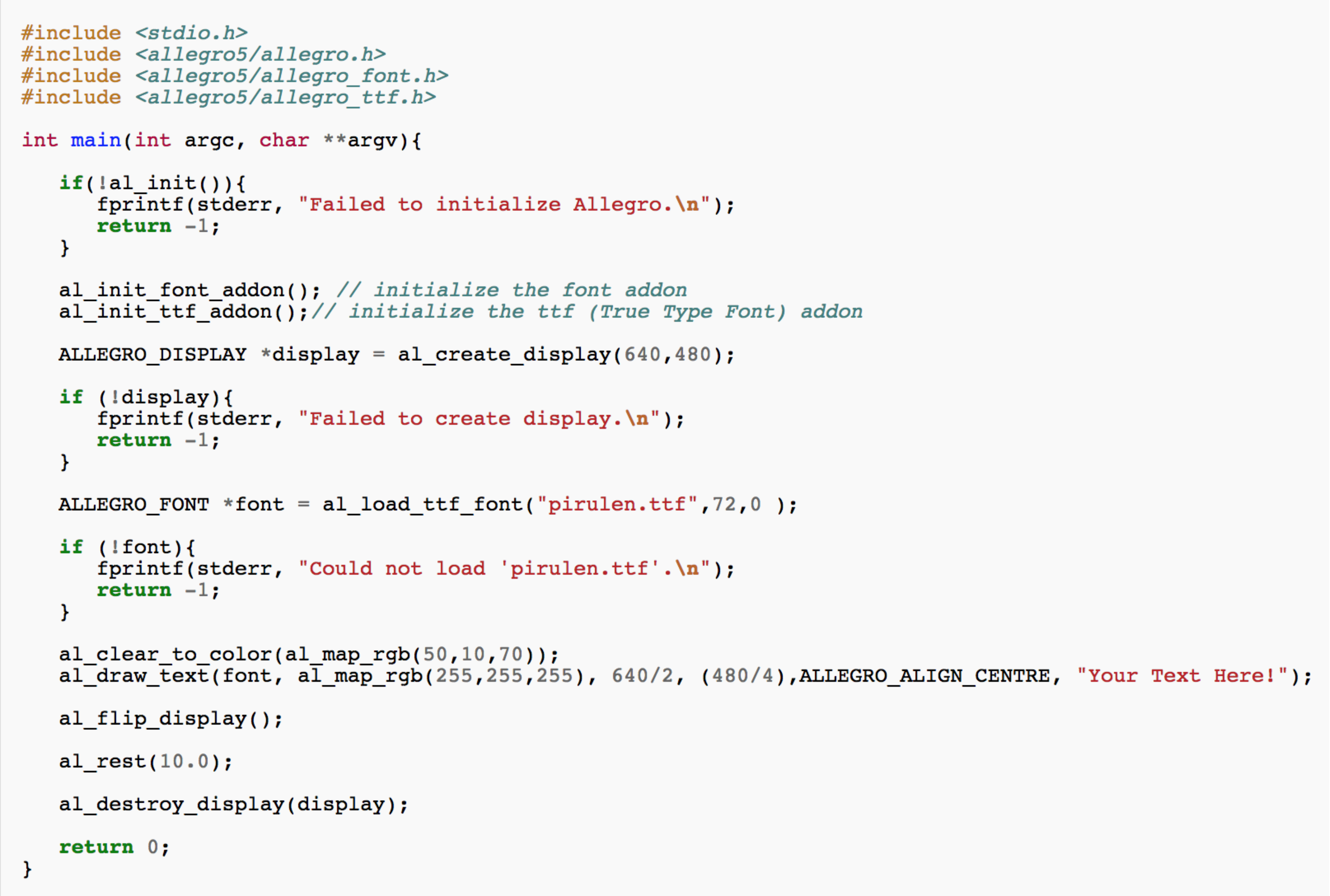
- 1. draw text on the window:
  - *void al_draw_text(const ALLEGRO_FONT *font,   ALLEGRO_COLOR color, float x, float y, int flags, char const *text)*

  - ALLEGRO_COLOR color: The same as we mentioned before.
    - *ALLEGRO_COLOR al_map_rgb(unsigned char r, unsigned char g, unsigned char b)*
    - r, g, b are the integers belonging to [0, 255]

Draw something!!! – cont.

- 1. draw text on the window:
  - *void al_draw_text(const ALLEGRO_FONT *font,   ALLEGRO_COLOR color, float x, float y, int flags,   char const *text)*

  - Float x and y: the coordiate of our text.
  - Flags: The flags parameter can be 0 or one of the following flags:
    - ALLEGRO_ALIGN_LEFT - Draw the text left-aligned (same as 0).
    - ALLEGRO_ALIGN_CENTRE - Draw the text centered around the given position.
    - ALLEGRO_ALIGN_RIGHT - Draw the text right-aligned to the given position.
  - Text: The words you want to draw on the window.



```
#include <stdio.h>
#include <allegro5/allegro.h>
#include <allegro5/allegro_font.h>
#include <allegro5/allegro_ttf.h>

int main(int argc, char **argv){

   if(!al_init()){
      fprintf(stderr, "Failed to initialize Allegro.\n");
      return -1;
   }

   al_init_font_addon(); // initialize the font addon
   al_init_ttf_addon();// initialize the ttf (True Type Font) addon

   ALLEGRO_DISPLAY *display = al_create_display(640,480);

   if (!display){
      fprintf(stderr, "Failed to create display.\n");
      return -1;
   }

   ALLEGRO_FONT *font = al_load_ttf_font("pirulen.ttf",72,0 );

   if (!font){
      fprintf(stderr, "Could not load 'pirulen.ttf'.\n");
      return -1;
   }

   al_clear_to_color(al_map_rgb(50,10,70));
   al_draw_text(font, al_map_rgb(255,255,255), 640/2, (480/4),ALLEGRO_ALIGN_CENTRE, "Your Text Here!");

   al_flip_display();

   al_rest(10.0);

   al_destroy_display(display);

   return 0;
}
```

Draw something!!! – cont.

- 2. draw graphic on window:
  - Step 1: *#include <allegro5/allegro_primitives.h>*
  - Step 2: call the following function to initialize the library:
    - *bool al_init_primitives_addon(void)*
  - Step 3: call the function corresponding to the graphic you want to draw
    - Using rectangle as an example:
    - *void al_draw_rectangle(float x1, float y1, float x2, float y2,   ALLEGRO_COLOR color, float thickness)*
      - x1, y1, x2, y2 - Upper left and lower right points of the rectangle
      - color - Color of the rectangle
      - thickness - Thickness of the lines, pass <= 0 to draw hairline lines
    - There’s bunch of functions drawing different graphics filled or not.
    - Using Google to explore if you need them!

Draw something!!! – cont.

- 3. draw a image
  - Step 1: including the following head file:
    - *#include <allegro5/allegro_image.h>*
  - Step 2: call the following function to initialize the library:
    - *al_init_image_addon();*
  - Step 3: declare a variable in ALLEGRO_BITMAP* type to contain our im age:
    - *ALLEGRO_BITMAP *img=NULL;*

Draw something!!! – cont.

- 3. draw a image
  - Step 4: loading the image into our variable (e.g. img):
    - *ALLEGRO_BITMAP *al_load_bitmap(**const char** *filename)*
      - Filename: the address of your image file.
    - E.g:
      - ALLEGRO_BITMAP * img = NULL;
      - img = al_load_bitmap(“image.jpg”);
    - We can judge whether the variable img is NULL to determine whether the loading is success.

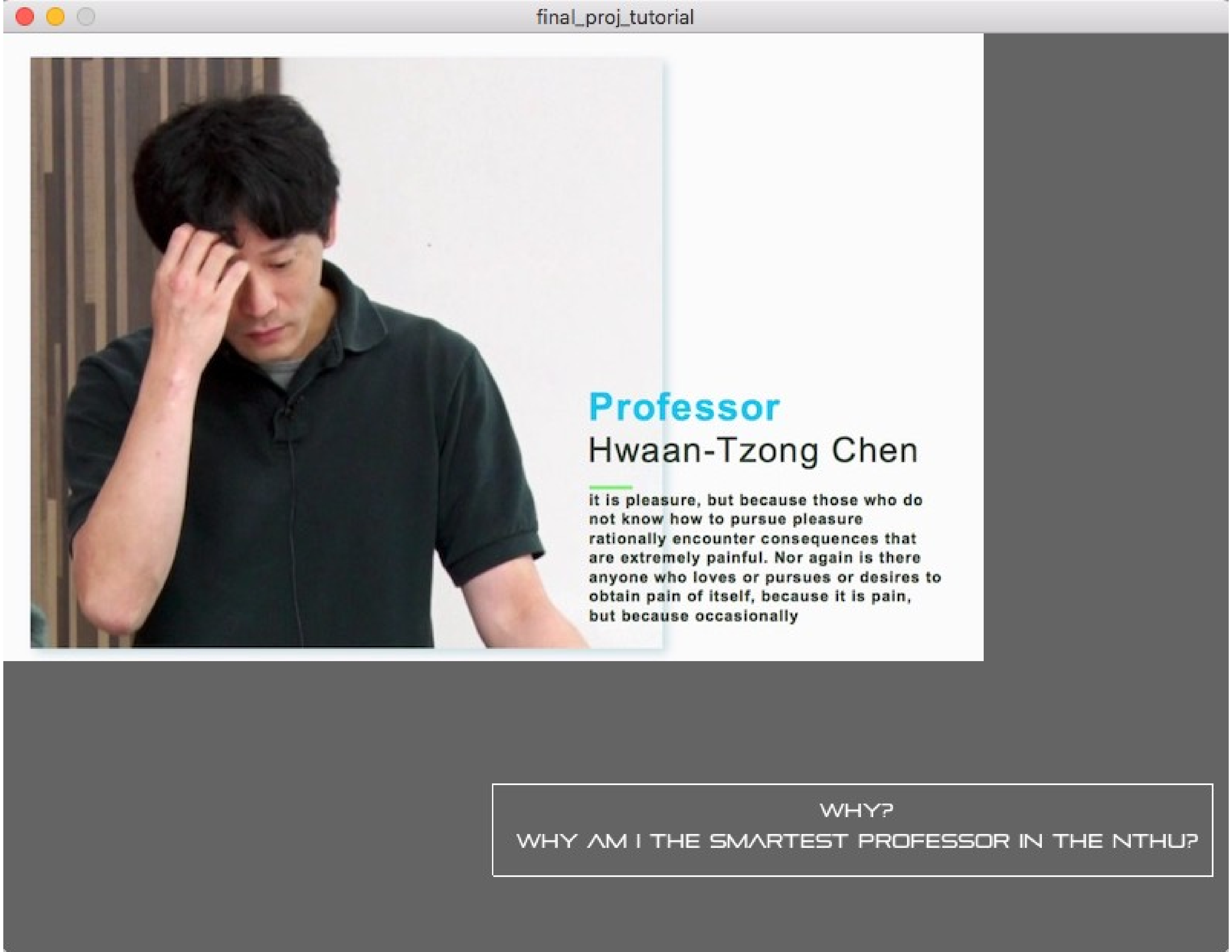
Draw something!!! – cont.

- 3. draw a image
  - Step 5: display the image
    - *void al_draw_bitmap(ALLEGRO_BITMAP *bitmap, float dx, float dy, int flags)*
    - bitmap: the variable containing the image
    - dx dy: the location of the image
    - flags can be:
      - ALLEGRO_FLIP_HORIZONTAL - flip the bitmap about the y-axis
      - ALLEGRO_FLIP_VERTICAL - flip the bitmap about the x-axis
      - 0 – do nothing

  - Step 6: don’t forget to destroy all things you create and load
    - ***void** al_destroy_bitmap(ALLEGRO_BITMAP *bitmap)*

Task 2 – Just Draw!!

- In this task, you are asked to draw something, including at least some words, one graphic and one image.
- A sample font and image has already provided for you.
- We also provided a sample code for you, filled with hint. You can either use it to design your own program, or coding from scratch.
- Feel free to ask us if there’s any question!!!!
- You should show something like this:



Professor

Hwaan-Tzong Chen

it is pleasure, but because those who do not know how to pursue pleasure rationally encounter consequences that are extremely painful. Nor again is there anyone who loves or pursues or desires to obtain pain of itself, because it is pain, but because occasionally

WHY?
WHY AM I THE SMARTEST PROFESSOR IN THE NTHU?

Deal with event

- In this section, we will learn how to set up a basic interactive m echanism and deal with some interaction which is called event in the allegro.

- For reach our goal, we need to **initial event queue** to record the event, and then **process the event** caught from event queue.

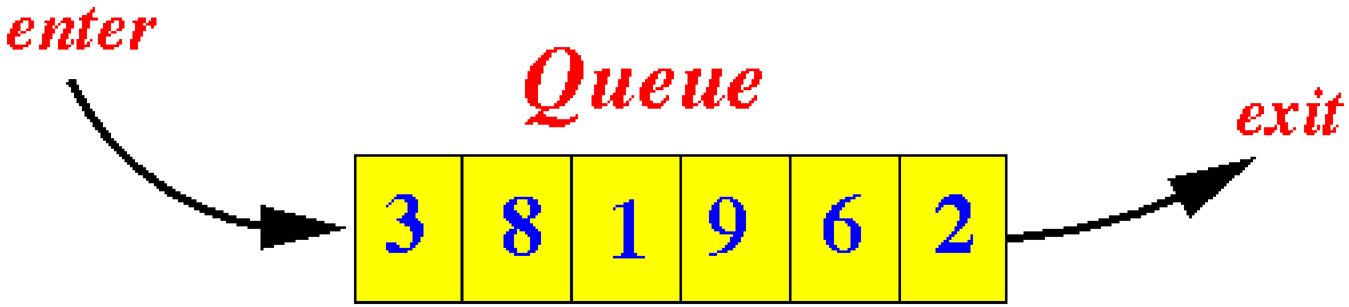
- Now we share the method for you step by step.

Deal with event – cont.

- 0. Queue

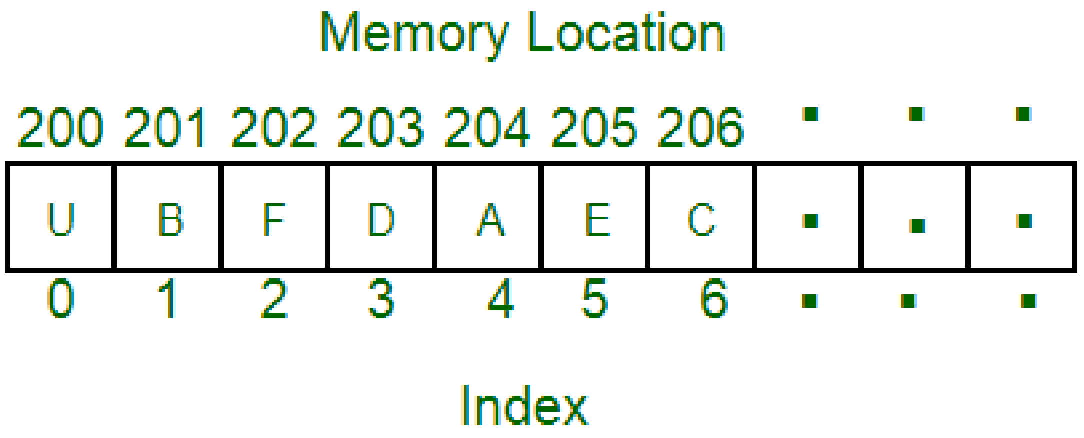
Deal with event – cont.

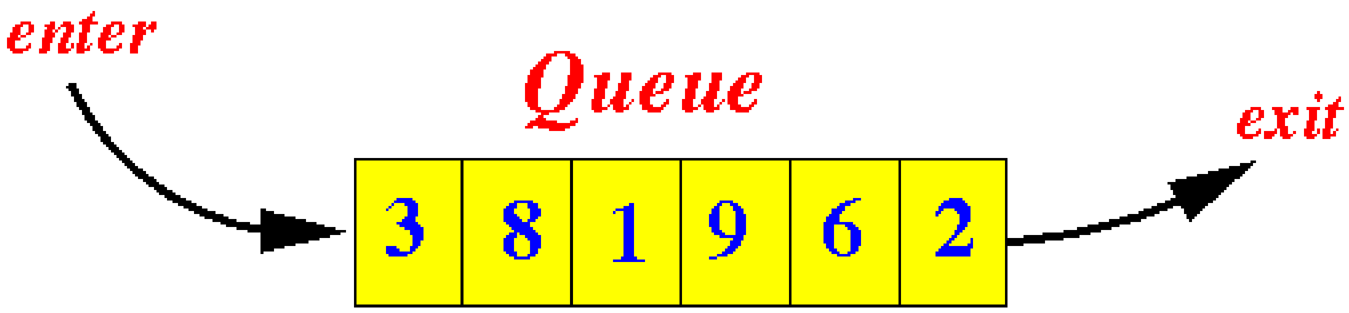
- 0. Queue
  - The one that **enters** the **queue first** will **also exit first (because he/she gets serviced first**)



Deal with event – cont.

- 0. Queue vs Array





Deal with event – cont.

- 1. Event
  - Event is something happened during the game running
  - Like keyboard input, click mouse, close display and so on.

- 2. Event_queue
  - Event_queue is a queue containing all event, and is the only source we catch any event.
  - Therefore, to let the code know what has happened, we must create a event queue first:
    - *ALLEGRO_EVENT_QUEUE *event_queue = NULL;*
    - *event_queue = al_create_event_queue();*

Deal with event – cont.

- 2. Event_queue – cont.
  - To set the type of event (keyboard, mouse, display, etc) recorded:
    - *void al_register_event_source (ALLEGRO_EVENT_QUEUE *queue, ALLEGRO_EVENT_SOURCE *source)*
    - E.g. *al_register_event_source(event_queue, al_get_display_event_source(display));*
  - Then we can get event from it.
    - More specific, the event we get from it must belongs to one of the source we have already registered.
    - If you can not get the event you expect (e.g. keyboard input), please check whether you register your keyboard as one of your event source before smashing your keyboard or your laptop.

Deal with event – cont.

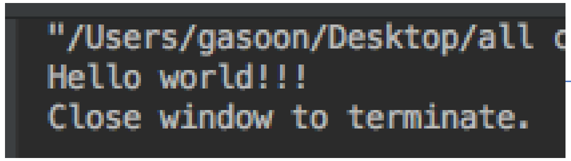
- 3. Get event.
  - To know whether there is any event in event_queue:
    - *bool al_is_event_queue_empty(ALLEGRO_EVENT_QUEUE *)*
    - It will return a bool value:
      - 1 (true) means the queue is empty;
      - 0 (false) means there's some event in the queue.
  - To get the event out of the queue:
    - *ALLEGRO_EVENT event;* (declare a variable to record the event)
    - *void al_wait_for_event(ALLEGRO_EVENT_QUEUE * queue*, *ALLEGRO_EVENT* event*);
      The program will be blocked here until the data of the earliest event have been duplicated into the allegro event we give.

Deal with event – cont.

- 4. Event type
  - How can we know what the event is (keyboard pressed, mouse clicke d..)
  - Use *event.type*
  - E.g. *event.type == ALLEGRO_EVENT_DISPLAY_CLOSE*
    - The function is used to find whether the display is closed just now.

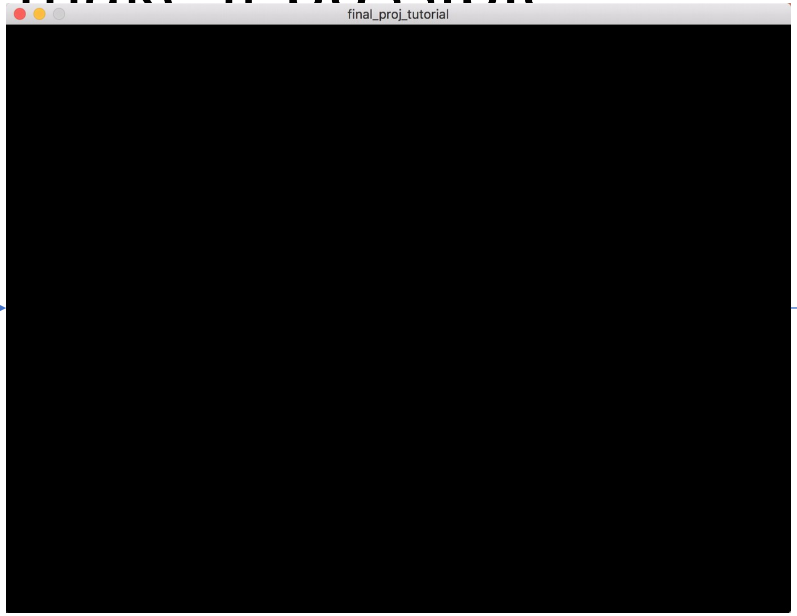
Task 3 – Close the window!!

- Have you ever noticed that in the 2 tasks before, the window can not be closed in normal way (click the cross symbol at the corner)?
- In this task, we will make it possible !!!

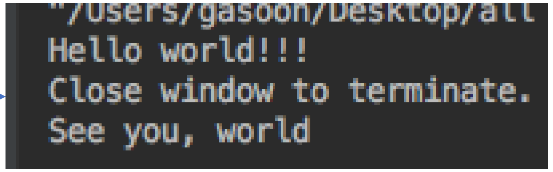


```
"/Users/gasoon/Desktop/all c
Hello world!!!
Close window to terminate.
```

Wait …



Click the cross at the corner



```
"/Users/gasoon/Desktop/all
Hello world!!!
Close window to terminate.
See you, world
```

Keyboard interaction

- In this section, we will learn how to use keyboard and mouse a s input source to interact with our program.

Keyboard interaction– cont.

- Step 1: declare and create the event_queue and event as before.

- Step 2: call *bool al_init_primitives_addon(void)* first then *bool al_install_keyboard(void)* to initial the keyboard

- Step 3: register keyboard as event source to the event queue
  - Using *ALLEGRO_EVENT_SOURCE *al_get_keyboard_event_source(void)* to get the event source of keyboard
  - Using *void al_register_event_source(ALLEGRO_EVENT_QUEUE *queue, ALLEGRO_EVENT_SOURCE *source)* to register the source to event queue.

Keyboard interaction – cont.

- Step 4: To find if there's some key being pressed or bouncing up:
  - Compare the event.type with keyboard type.
  - There are 2 event types for the keyboard:
    - ALLEGRO_EVENT_KEY_DOWN – indicating some keys being pressed.
    - ALLEGRO_EVENT_KEY_UP – indicating some keys bouncing up just now.
    - Using event.type to judge the type of event.
- Step 4: get the specific key:
  - keycode to each button of keyboard: https://www.allegro.cc/manual/5/keyboard.html (e.g. A = ALLEGRO_KEY_A)
  - *event.keyboard.keycode* capsules the keycode of the key causing the event.
  - Comparing it with the keycode you design to get the result.

Task 4 – control a box with keyboard!!!

- Using up, down, left, right button to control the box move in the same way.

- The box should move to the direction whenever the corresponding button is pressed, and stop whenever the corresponding button is released.

- Press ESC to terminate the program

Mouse interaction

- In this section, we will learn how to use mouse as input source to interact with our program.

Mouse interaction– cont.

- Register the source of mouse to the event_queue
  - Using *ALLEGRO_EVENT_SOURCE *al_get_mouse_event_source()* to get the source of mouse.
- Call *bool al_init_primitives_addon(void)* first then *bool al_install_mouse(void)* to initial the mouse.
- To find whether the mouse is clicked just now
  - Compare the event.type with ALLEGRO_EVENT_MOUSE_BUTTON_DOWN
  - event.mouse.button capsules the information of the pressed button
    - 1 for left button. 2 for right one, and so on so forth
- To find whether the mouse is moving:
  - Compare the event.type with ALLEGRO_EVENT_MOUSE_AXES
  - Using event.mouse.x and event.mouse.y to get the position of cursor.

Task 5 – control a box with mouse!!!

- Using mouse to determinate the location of the box.
- The box should be the same place as the cursor whenever the cursor is in the window.
- Press left button to make the box disappear and press it again to make it visible.
- Press right button to terminate the program