Fun with project

Outline

Outline

Allegro install

- Windows:
  - https://drive.google.com/file/d/1OQBL_ChhbuOfK_qGZtIvIzfh11LlxWz4/view?usp=sharing
- Mac:
  - https://goo.gl/vpTA9t
  - https://hackmd.io/@kerwintsai/SkRTk6kCS
- Allegro download:
  - https://github.com/liballeg/allegro5/releases
- Allegra reference:
  - https://liballeg.org/a5docs/trunk/

Outline

Main.cpp

```
GameWindow *TowerGame= new GameWindow();

TowerGame->game_play();
```

Initial the game

Start the game

GameWindow.cpp

~GameWindow()

```
al_init_primitives_addon();
al_init_font_addon(); // initialize the font addon
al_init_ttf_addon(); // initialize the ttf (True Type Font) addon
al_init_image_addon(); // initialize the image addon
al_init_acodec_addon(); // initialize acodec addon

al_install_keyboard(); // install keyboard event
al_install_mouse();    // install mouse event
al_install_audio();    // install audio event
```

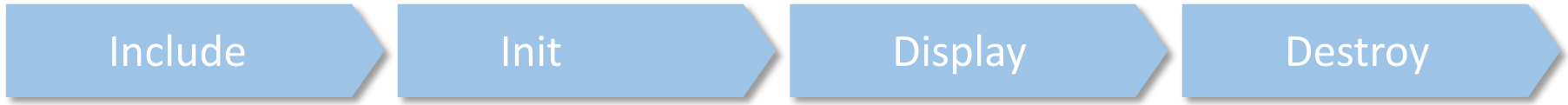
Initialize the basic function of allegro, install event and register event.

```
al_register_event_source(event_queue, al_get_display_event_source(display));
al_register_event_source(event_queue, al_get_keyboard_event_source());
al_register_event_source(event_queue, al_get_mouse_event_source());

al_register_event_source(event_queue, al_get_timer_event_source(timer));
al_register_event_source(event_queue, al_get_timer_event_source(monster_pro));
```

GameWindow.cpp

~initial the game

Four steps to use the function in allegro



GameWindow.cpp

~start the game

```
GameWindow::game_play()
{
    int msg;

    srand(time(NULL));

    msg = -1;
    game_reset();
    game_begin();

    while(msg != GAME_EXIT)
    {
        msg = game_run();
    }

    show_err_msg(msg);
}
```

Clear components you display or used.
Stop the sound and timer

GameWindow.cpp

~start the game

```
GameWindow::game_play()
{
    int msg;

    srand(time(NULL));

    msg = -1;
    game_reset();
    game_begin();

    while(msg != GAME_EXIT)
    {
        msg = game_run();
    }

    show_err_msg(msg);
}
```

Start timer, disply sound and draw picture on screen

GameWindow.cpp

~start the game

```
GameWindow::game_play()
{
    int msg;

    srand(time(NULL));

    msg = -1;
    game_reset();
    game_begin();

    while(msg != GAME_EXIT)
    {
        msg = game_run();
    }

    show_err_msg(msg);
}
```

Use while loop to keep running the game.
Process the event, update game component, and draw picture

GameWindow.cpp

~game_reset()

```
GameWindow::game_reset() {
    // reset game and begin
    for(auto&& child : towerSet) {
        delete child;
    }
    towerSet.clear();
    monsterSet.clear();
    selectedTower = -1;
    lastClicked = -1;
    Coin_Inc_Count = 0;
    Monster_Pro_Count = 0;
    mute = false;
    redraw = false;
    menu->Reset();
    // stop sample instance
    al_stop_sample_instance(backgroundSound);
    al_stop_sample_instance(startSound);
    // stop timer
    al_stop_timer(timer);
    al_stop_timer(monster_pro);
}
```

You can clear all the component in your game in this function.
If you want your game can restart then this function will be important

GameWindow.cpp

~game_begin()

```
GameWindow::game_begin() {
    printf(">>> Start Level[%d]\n", level->getLevel());
    draw_running_map();

    al_play_sample_instance(startSound);
    while(al_get_sample_instance_playing(startSound));
    al_play_sample_instance(backgroundSound);

    al_start_timer(timer);
    al_start_timer(monster_pro);
}
```

Start your game here, you can draw the basic item first like start menu.
Start timer then you can run your game.

GameWindow.cpp

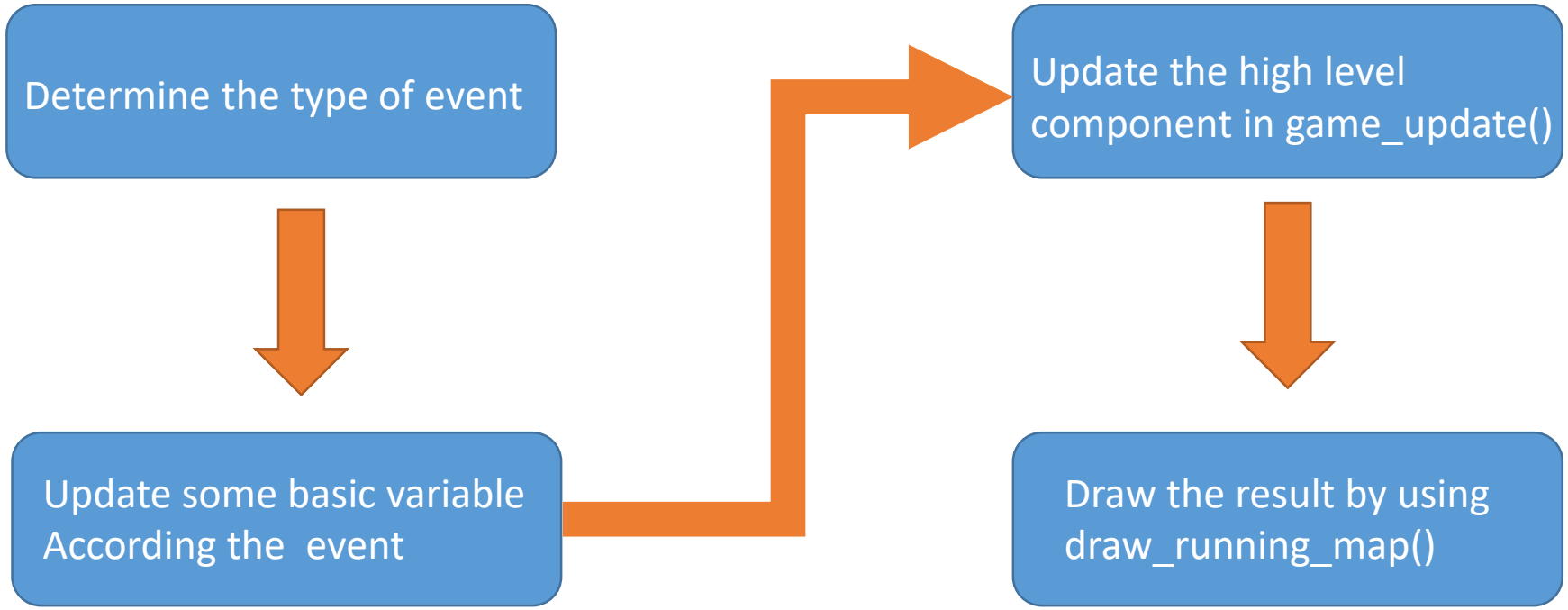
~game_run()

```
GameWindow::game_run() {
    int error = GAME_CONTINUE;

    if (!al_is_event_queue_empty(event_queue)) {

        error = process_event();
    }
    return error;
}
```

Process the event in event queue

GameWindow.cpp

~process_event ()



Outline

Todo 1

- In GameWindow.cpp
- In GameWindow::game_update() function

```
/*TODO:*/
/*Allow towers to detect if monster enters its range*/
/*Hint: Tower::DetectAttack*/
```

Todo 1

- In GameWindow.cpp
- In GameWindow::game_update() function

```
for( /* go through towerSet */ ) {
    for(/* go through monster Set */) {
        /* DetectAttack */
    }
}
```

Todo 2

- In GameWindow.cpp
- In GameWindow::game_update() function

```
/*TODO:*/
/*1. For each tower, traverse its attack set*/
/*2. If the monster collide with any attack, reduce the HP of the monster*/
/*3. Remember to set isDestroyed to "true" if monster is killed*/
/*Hint: Tower::TriggerAttack*/
```

Todo 2

- In GameWindow.cpp
- In GameWindow::game_update() function

```
for(/*go through towerSet*/){
    isDestroyed = /*TriggerAttack*/
}
```

Todo 3

- In GameWindow.cpp
- In GameWindow::game_update() function

```
/*TODO:*/
/*1. Update the attack set of each tower*/
/*Hint: Tower::UpdateAttack*/
```

Todo 3

- In GameWindow.cpp
- In GameWindow::game_update() function

```
for( /* go through towerSet */ ) {
    /* UpdateAttack */
}
```

Todo 4

- In GameWindow.cpp
- In GameWindow::process_event() function

```
case ALLEGRO_KEY_P:
    /*TODO: handle pause event here*/
    break;
```

Todo 4

- In GameWindow.cpp
- In GameWindow::process_event() function

```
al_get_timer_started(timer)
al_stop_timer(timer)
al_start_timer(timer)
```

Check if “timer” is started.
If yes return true otherwise return false

Stop timer

Start timer

Todo 5

- In Tower.cpp
- In Tower::TriggerAttack(Monster *monster) function

```
/*TODO:*/
/*1. Reduce the monster HP by the harm point*/
/*2. Erase and delete the attack from attack set*/
/*3. Return true if the monster's HP is reduced to zero*/
```

Todo 5

- In Tower.cpp
- In Tower::TriggerAttack(Monster *monster) function

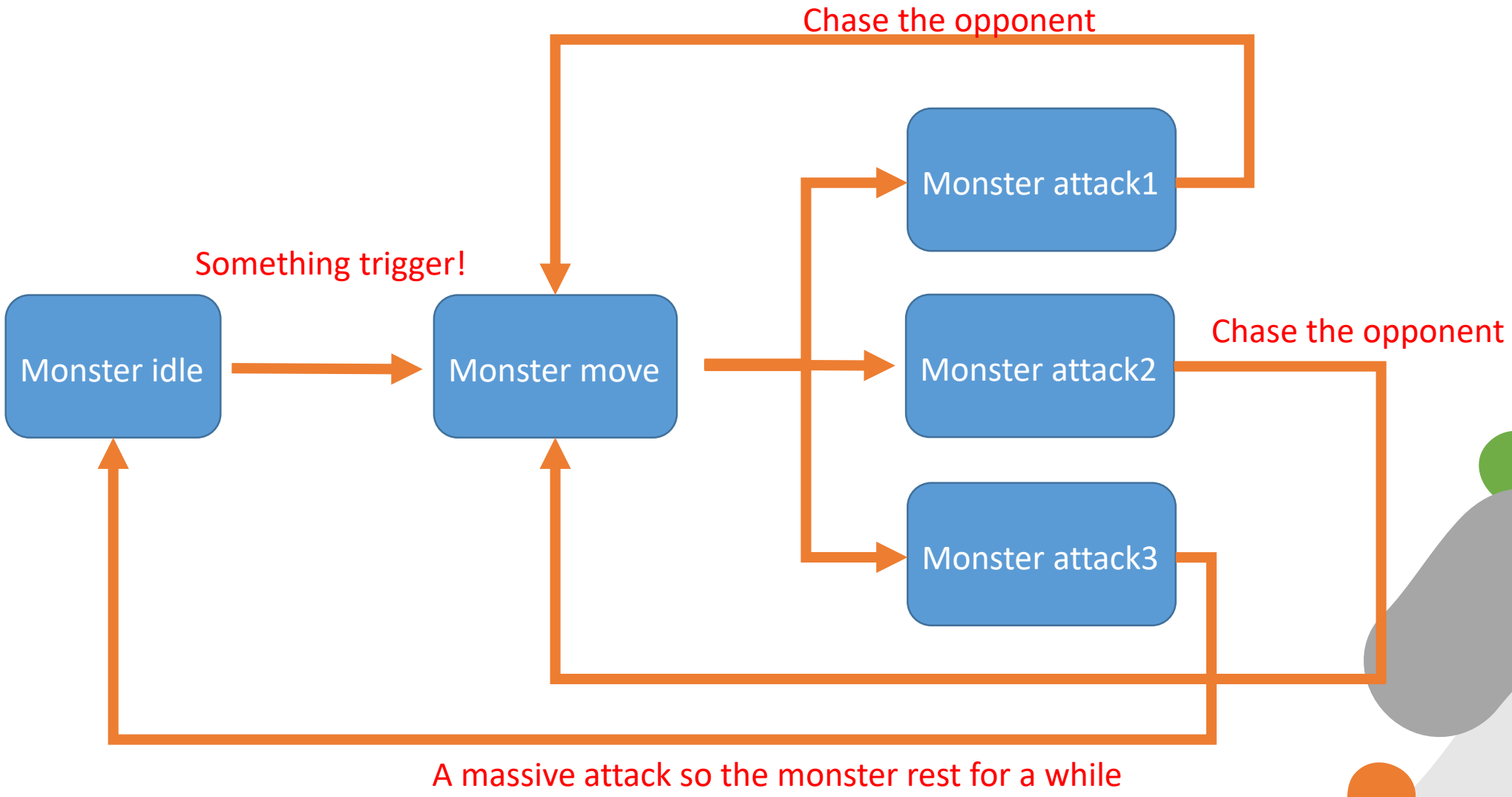
Use something like this…..

```
monster->Subtract_HP( attack_set[i]->getHarmPoint() )
```

Outline

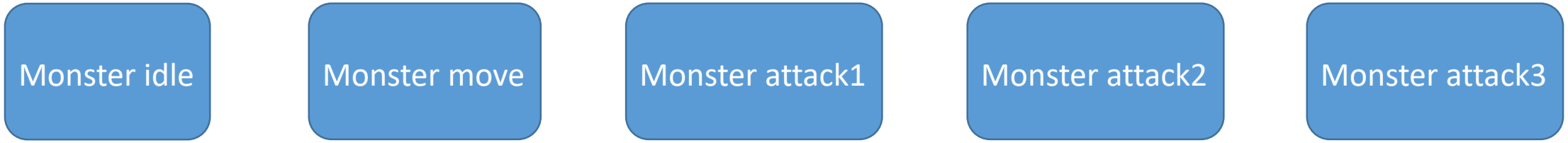
AI

~Use finite state machine!!!



AI

~Use finite state machine!!!



Draw the stop monster

Draw the moving animation

Draw the attack1 animation

Draw the attack2 animation

Draw the attack3 animation

By using states, we can easy draw the animation of monster

Special effect sound

~Use finite state machine!!!

Monster idle	Monster move	Monster attack1	Monster attack2	Monster attack3
Display the sound of panting	Display the sound of moving	Display the sound of attack1	Display the sound of attack2	Display the sound of attack3

Just judge the state then use
al_play_sample(*ALLEGRO_SAMPLE)

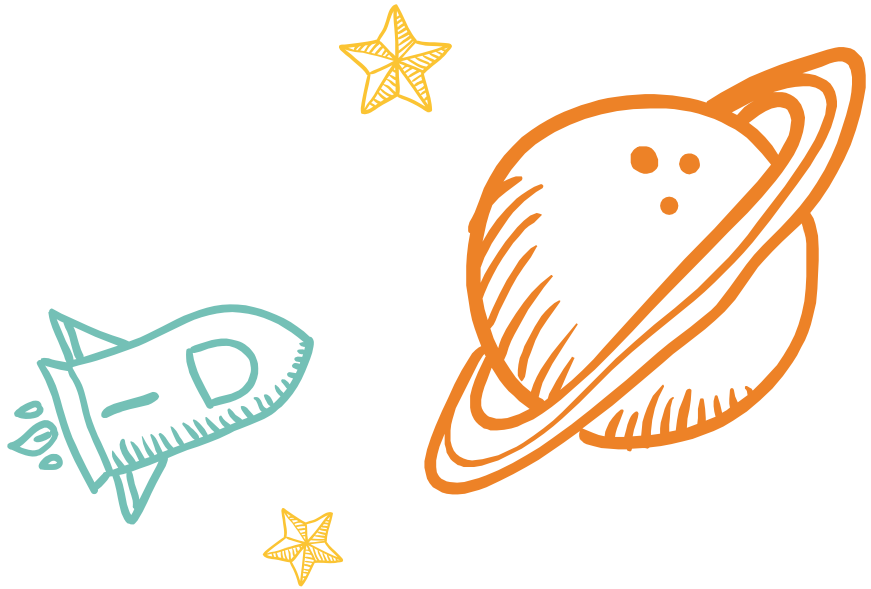
Display video

- Use something like mp4 to jpg to transfer the video into images.
- Load the images into an array. You can use sprintf to manipulate the path of image
- Set a timer as the fps of your video.
- Set a event queue to get the timer event.
- Each time the timer trigger display the image on the screen
- Plus the index of array by 1 to display the next image.
- Then you get the effect of display video!

Allegro reference

~find useful function here!!!

- al_set_window_position()
- al_resize_display()
- ……



The End~