```
void load_images(); // declaration of \verb|load_images|
void animate(void*); // declaration of our movement


#include <iostream>
#include <sstream>
#include <cstdlib>
#include <ctime>
#include "balloon.h"
#include "balloonheader.h"
int y_speed = 10; //

int main (){

    make_window()->show();
    load_images();
    Fl::add_timeout(1.0/y_speed, animate); // 10 frames per second
    Fl::run();
    return 0;
}

Fl_GIF_Image* balloon_flying;
Fl_GIF_Image* balloon_hit;
Fl_GIF_Image* blower_static;
Fl_GIF_Image* balloon_fall;

void load_images(){
    background->image(new Fl_GIF_Image("background.gif")); // background with mountain
    flag1->image(new Fl_GIF_Image("finish.gif"));
    flag2->image(new Fl_GIF_Image("finish.gif"));
    blower_static = new Fl_GIF_Image("blower.gif"); // blower
    blower1->image(blower_static);
    blower2->image(blower_static);
    blower3->image(blower_static);
    blower4->image(blower_static);
    blower5->image(blower_static);
    balloon_flying = new Fl_GIF_Image("balloon.gif"); // going up image
    // balloon_move = new Fl_GIF_Image(); // moving right image
    balloon_hit = new Fl_GIF_Image("balloonhit.gif"); // collision image
    balloon_fall = new Fl_GIF_Image("balloonfall.gif"); // falling down image

    // blower_air = new Fl_GIF_Image(); // blower wind
    // finish_left = new Fl_GIF_Image(); // left side of finish
    // finish_right = new Fl_GIF_Image(); // right side of finish
    return;
}

bool collision(int bx, int by){
    int b_left = balloon->x();
    int b_right = balloon->x() + balloon->w();
    int b_top = balloon->y();
    int b_bottom = balloon->y() + balloon->h();

    int i = balloon->parent()->children();

    for (int j = 0; j < i; j++){

        int m_left = balloon->parent()->child(j)->x();
        int m_right = balloon->parent()->child(j)->x() + balloon->parent()->child(j)->w();
        int m_top = balloon->parent()->child(j)->y();
        int m_bottom = balloon->parent()->child(j)->y() + balloon->parent()->child(j)->h();

        if (balloon->parent()->child(j) != balloon and
            balloon->parent()->child(j) != background and
            balloon->parent()->child(j) != flag1 and
            balloon->parent()->child(j) != flag2 and
            balloon->parent()->child(j) != lose and
            balloon->parent()->child(j) != restart and
            balloon->parent()->child(j) != win)

        if (b_left < m_right and
            b_right > m_left and
            b_top < m_bottom and
            b_bottom > m_top)
        return true;
    }
    return false;
}

/* bool finish (int bx, int by){
} */


void animate (void*pointer){



    int bx = balloon->x(); // x position of balloon
    int by = balloon->y(); // y position of balloon
```

```
 int wh = balloon->parent()->h(); // height of window
 int bh = balloon->h(); // height of hummingbird

 balloon->image(balloon_flying);


 Fl::repeat_timeout (1.0/y_speed, animate);


 if (not collision(bx, by)){
     by--; // moves balloon up
 }
 else if (by + bh < wh){
         lose->activate();
         lose->show();
         restart->show();
         restart->activate();
         balloon->image(balloon_hit);
         by+=10; // if collision, move down until it touches the bottom window
     }
 if (by + bh >= wh - 1){
     balloon->image(balloon_fall);
 }
 if (by == 0){
     lose->activate();
     lose->show();
     restart->show();
     restart->activate();
 }
/*  if (finish){
        win->show();
        y_speed+=5
    }
    */

    balloon->position (bx, by);
    balloon->parent()->redraw();

    return;
}
```

Click here to view the video in your browser.