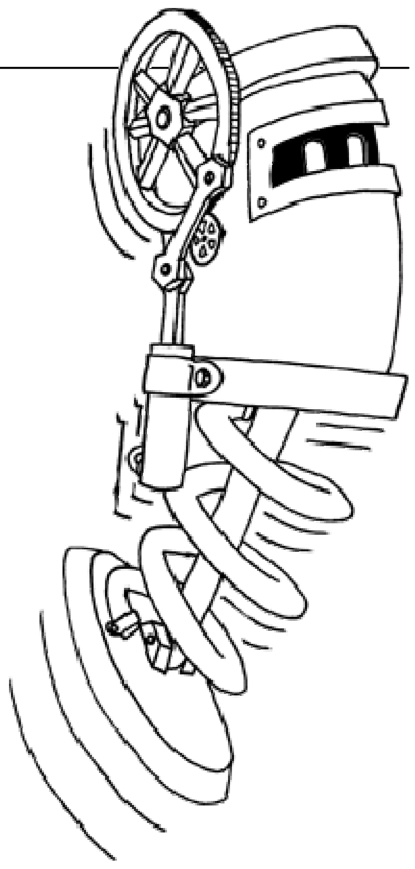
Endless Tower

Jump, jump, jump. Up you go as the platforms fall below you. If you fall off the bottom of the screen, you lose.

Endless Tower is another simple game that can be limitless fun. Move with the left and right arrow keys, jump with the space. Keep a falling platform under you at all times, but keep looking for the next jump. You can jump through platforms, which can help. You can jump off the top of the screen, but that doesn't help you as much since most of the time you end up dropping where you can't see.

Endless Tower is by Jacob Charles, AKA Rio Kikaru, submitted for the MinorHack at October 14th, 2006 at 6:00 pm UTC.

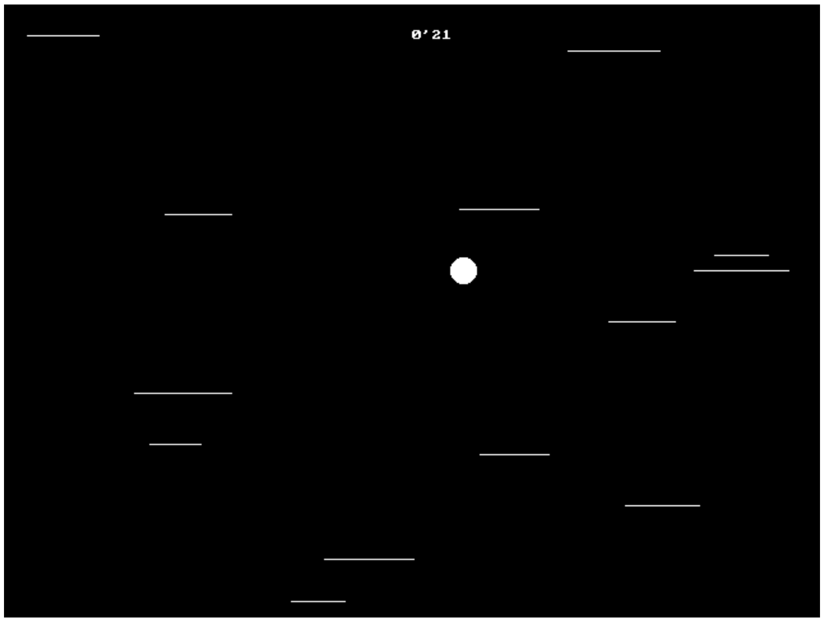


// endlesstower.cpp listing begins:

```
#include <allegro.h>

#define MODE GFX_AUTODETECT_WINDOWED
#define WIDTH 640
#define HEIGHT 480
#define WHITE makecol(255, 255, 255)
#define BLACK makecol(0, 0, 0)

class platform {
public:
int x, y, w;
};

BITMAP *buffer;
long start_time;
int jumpforce, x = 320, y = 20;
bool jump;
platform land[14];
int on_land, game_time = 0, speed = 3;

//platform spawning function
void add_platform() {
    for (int i = 0; i < 13; i++) {
      if (land[i].y > HEIGHT) {
        land[i].y = 0;
        land[i].x = ((rand()%300)*2) + 20;
        land[i].w = (rand()%20) + 20;
        return;
      }
    }
    return;
}

void main() {
    allegro_init();
    install_keyboard();
    install_mouse();
    install_timer();
    set_color_depth(32);
    set_gfx_mode(MODE, WIDTH, HEIGHT, 0, 0);
    srand(time(NULL)+clock());
    text_mode(-1);

    buffer = create_bitmap(WIDTH, HEIGHT);
    textprintf_centre(screen, font, 320, 200, WHITE, "-Endless Tower-");
    textprintf_centre(screen, font, 320, 240, WHITE, "press enter to start");
    while (!key[KEY_ENTER]) {}

    for(int i = 0; i < 13; i++) {
      land[i].y = rand() % HEIGHT;
      land[i].x = ((rand()%300)*2) + 20;
      land[i].w = (rand()%20) + 20;
    }
    land[13].y = y + 10;
    land[13].x = x;
    land[13].w = 45;
    rest(50);
    while (!key[KEY_ESC]) {
      start_time = clock();

    /*game goes here*/
```

// Listing continued on next page…

// Listing continued from previous page



```
    //move left
    if (key[KEY_LEFT]) x -= 10;
    //move right
    if (key[KEY_RIGHT]) x += 10;
    //jump
    if ((!jump)&&(key[KEY_SPACE])) {
      jump = true;
      jumpforce = 60;
    }
    if ((jump)&&(key[KEY_SPACE])) {
      jumpforce += 1;
    }
    //speed up!
    speed = 3 + (game_time/20000);
    //fall
    if (jump) jumpforce -= 3;
    if (!jump) jumpforce = -12;
    y -= jumpforce/4;
    //drop platforms down
    add_platform();
    for (int i = 0; i < 14; i++) land[i].y += speed;

    //collision
    on_land = 0;
    for (int i = 0; i < 14; i++) {
      if ((y > land[i].y - 20)&&(y < land[i].y)) {
        if ((x < land[i].x + land[i].w)&&(x > land[i].x - land[i].w)) {
          if (jumpforce < 5) jump = false;
          jumpforce = -speed*3;
          on_land += 1;
        }
      }
    }
    if (on_land == 0) jump = true;
    //warp
    if (x < 0) x = 0;
    if (x > WIDTH) x = WIDTH;
    //die
    if (y > HEIGHT) return;

    /*drawing goes here*/
    rectfill(buffer, 0, 0, WIDTH, HEIGHT, BLACK);
    circlefill(buffer, x, y, 10, WHITE);
    textprintf(buffer, font, 320, 20, WHITE, "%i'%i"
      , game_time/60000, (game_time/1000)%60);

    for (int i = 0; i < 14; i++) {
      hline(buffer, land[i].x - land[i].w, land[i].y,
        land[i].x + land[i].w, WHITE);
    }

    blit(buffer, screen, 0, 0, 0, 0, WIDTH, HEIGHT);

    game_time += 20;
    while (clock() < start_time + 20) {}
  }
  destroy_bitmap(buffer);
  return;
}
END_OF_MAIN()
```