

Back Ground



2024 Indie Dev Gallery
Summer 게임잼 우수상

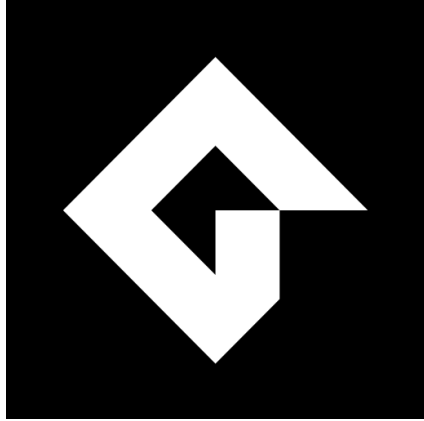
게임 플레이

01 기획

주제 : “한번도 보지 못한 참신한 게임”

**플레이어가 아니라 배경이 움직이는 플랫폼머 게임은 어떨까?
하는 생각으로 제작을 시작 하였습니다.**

엔진 & 장르



게임메이커2

1주일의 짧은 과제시간, 2D 플랫폼머 라는 장르를 고려하여, 사용에 익숙하고 2D게임에 강점을 가지는 게임메이커 엔진을 선정



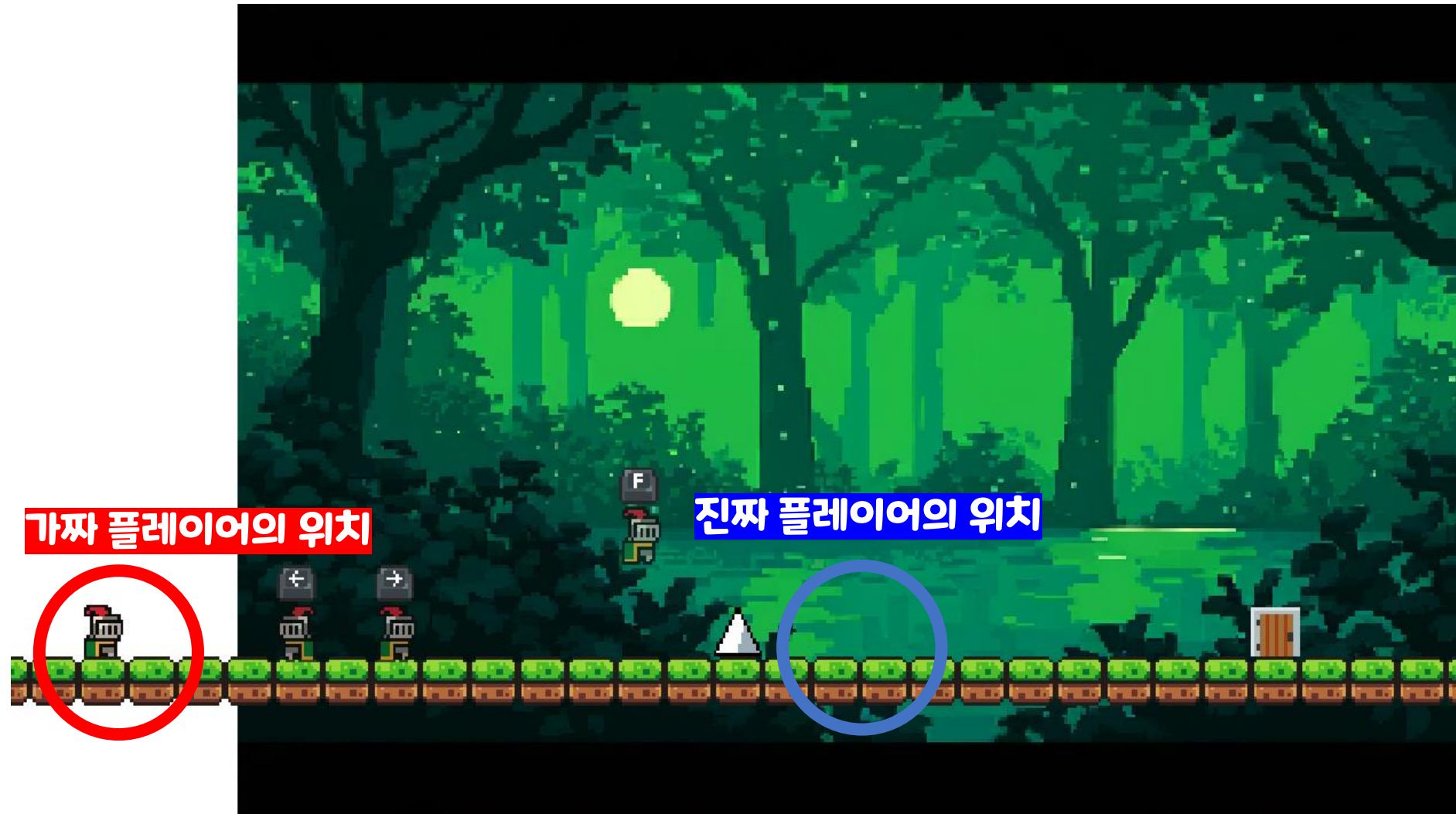
장르

2D 플랫폼머 퍼즐/추리

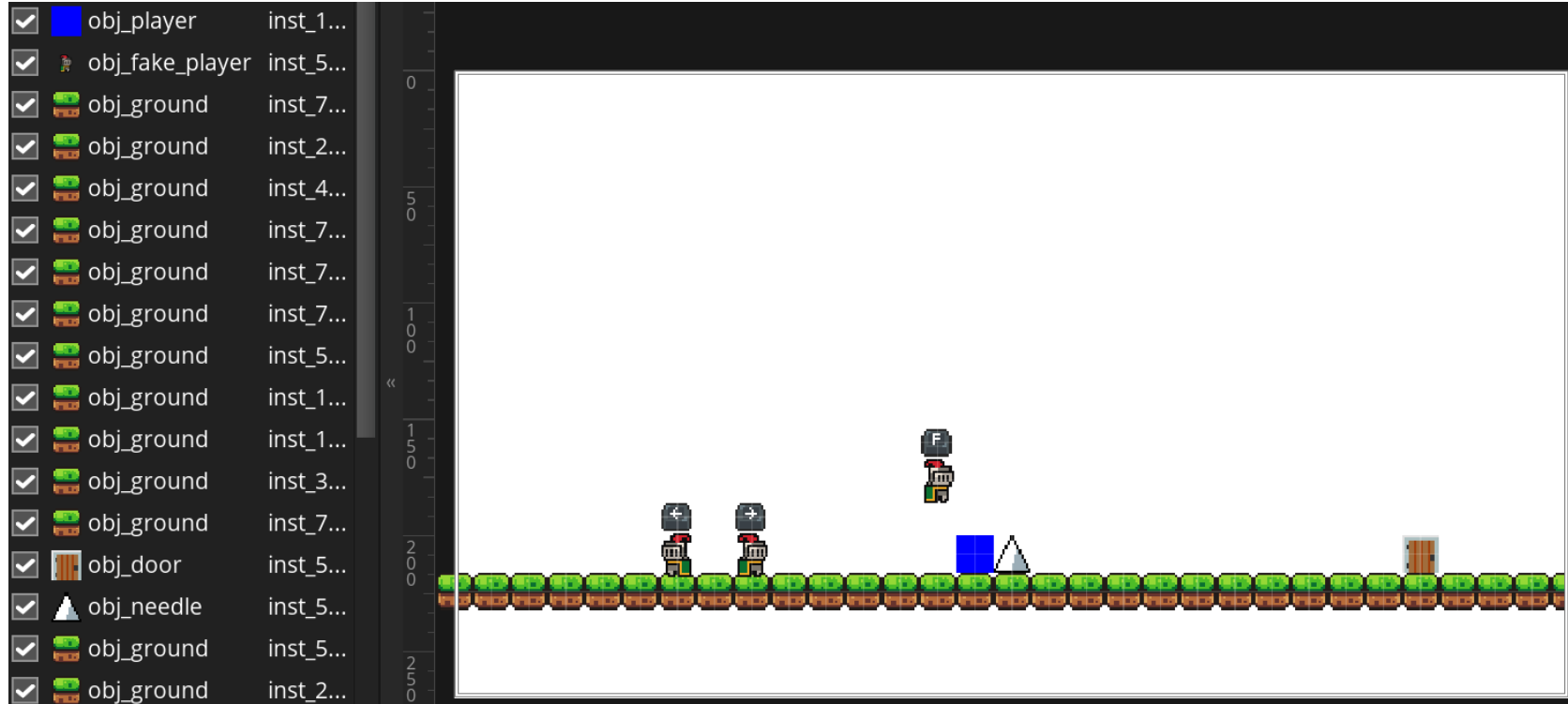
핵심 메카닉



핵심 메카닉

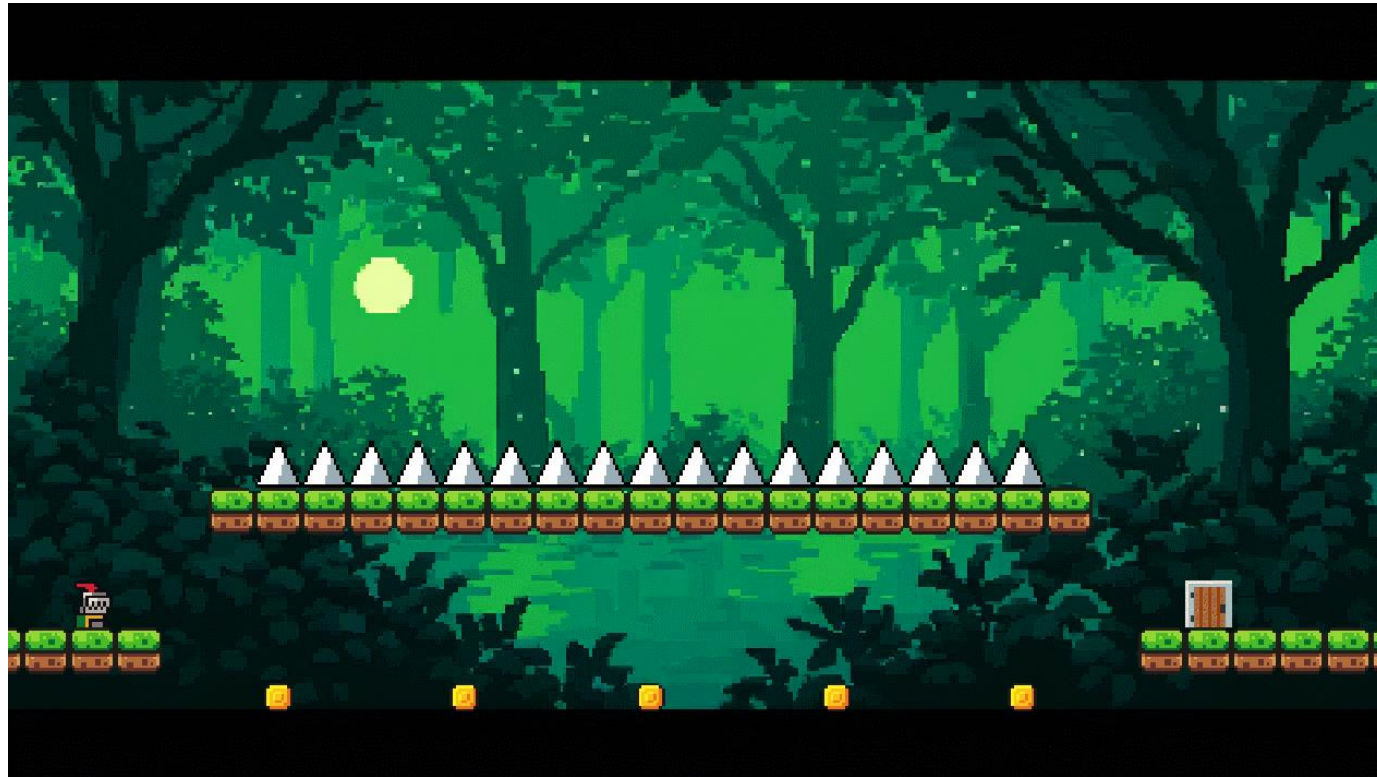


핵심 메카닉



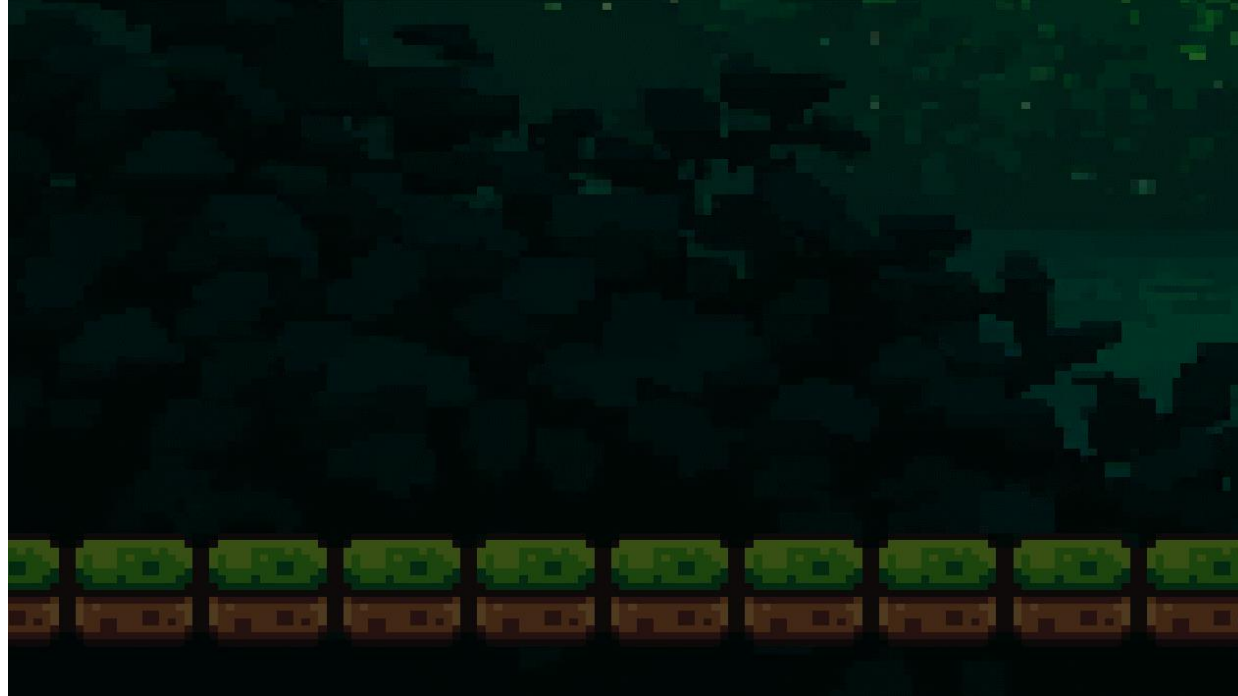
**배경은 눈속임이며, 실제 플레이어는 파란색 네모 박스로
파란색 박스만큼의 히트박스를 가지고 있습니다.
플레이어는 처음 Stage를 시작할때
좌우로 살짝 씩 움직여 보며 자신의 진짜 위치를 추리 해야 합니다.**

핵심 메카닉



**캐릭터가 죽게되면 배경이 히트 박스에 맞게 줄어들며
플레이어 위치에 대한 힌트를 주게 됩니다.**

튜토리얼



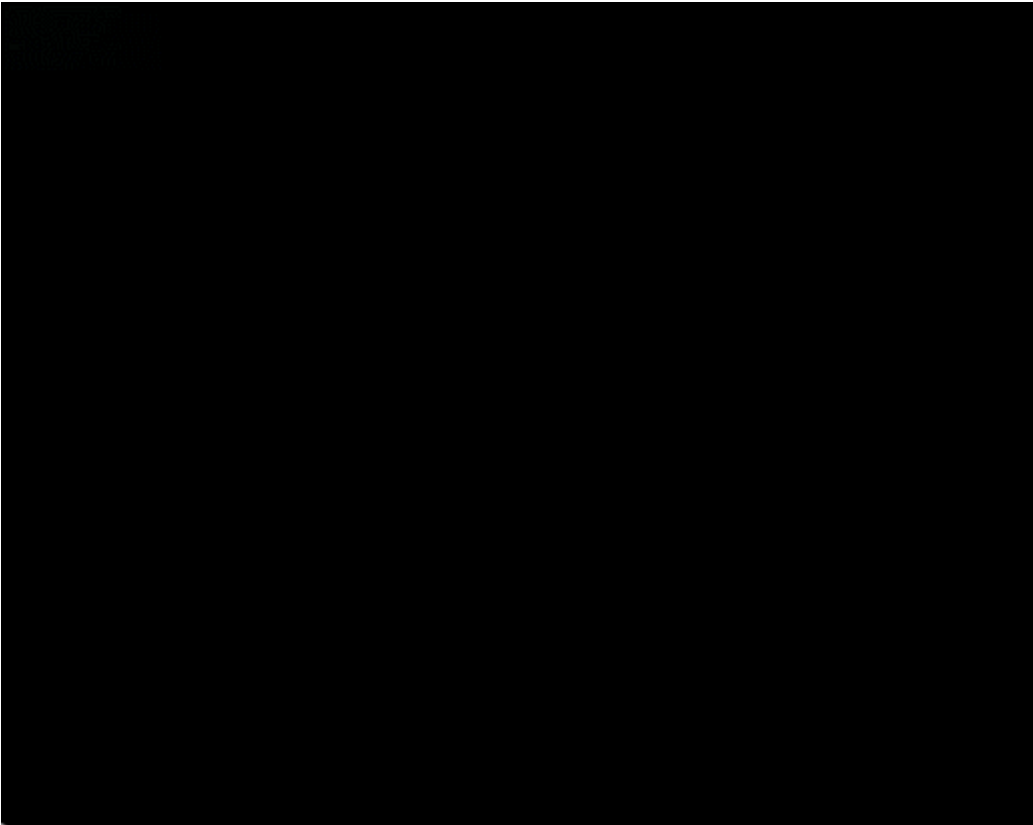
**게임을 켜게 되면 플레이어, 가시, 출입문을 차례로 비추며
일반적인 플랫폼어 게임임을 암시합니다.
Press 화살표로 이동을 유도 하고, 플레이어가 R을 누르면
가시 바로 옆에 있던 진짜 플레이어가 죽게 되면서
이 게임의 진짜 플레이어는 배경이라는 점을 알려줍니다.**

혼란을 주기위한 트릭



**가짜 플레이어 이미지를 사용해 튜토리얼 안내를 만들어서
무의식에 내가 조종하는 캐릭터가 가짜 플레이어 인 것 처
럼 생각하도록 유도하였습니다.**

혼란을 주기위한 트릭



**Stage를 시작할때 좌측 구석에서
걷는 효과음과 함께 가짜 플레이어가 튀어나옵니다.**

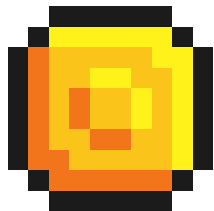
**순간적으로 내가 컨트롤 해야 할 캐릭터가 가짜 플레이
어인 것처럼 착각하게 만드는 것을 목표하였고
실제로 저 또한 스테이지를 테스트 하며 여러 번 트릭
에 당했습니다.**

10 스테이지 마다 새로운 기믹

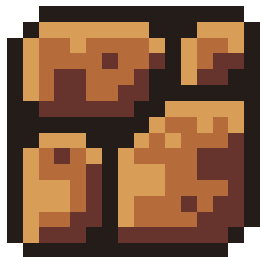


10스테이지 마다 짧은 개그성 만담을 집어넣어
앞으로 새로운 기믹이 등장 할 것을 예고 하였습니다.

기믹들



**일반적인 플랫폼머 게임에서 자주 볼 수 있는 점프횃수 추
가 입니다.**



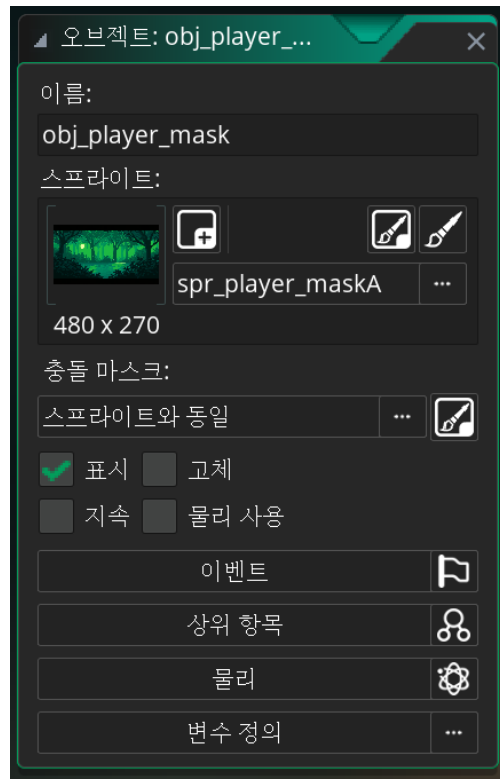
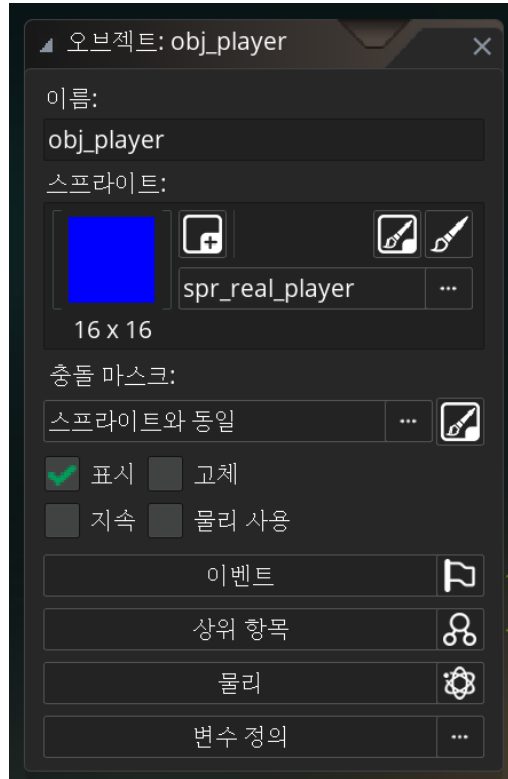
**일반적인 플랫폼머 게임에서 자주 볼 수 있는 밟으면 부서
지는 블록 입니다.**



배경처럼 플레이어를 따라다니는 가짜 블록들 입니다.

02 구현

플레이어



```
Create::
player_ask = instance_create_layer(room_width/2,room_height/2,"Instances",obj_player_mask);
mask_x = x - player_mask.x;
mask_y = y - player_mask.y;

Step::
if (instance_exists(player_mask)){
player_mask.x = x - mask_x;
player_mask.y = y - mask_y;
}
```

플레이어는 플레이어의 히트박스과 플레이어 마스크 두개의 오브젝트로 구현됩니다.

플레이어 마스크는 항상 플레이어의 좌표를 따라다니며 자신이 플레이어 인척 연기합니다.

플레이어에게 필요한 이동 등의 로직은 obj_real_player가 수행합니다.

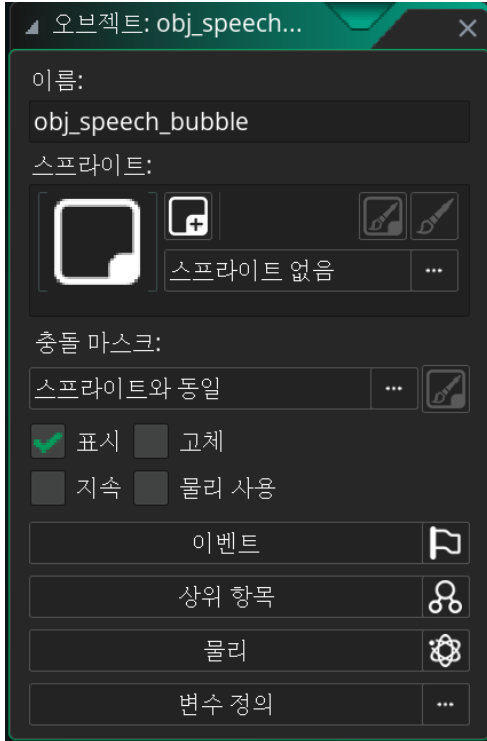
유저 친화적 컨트롤

```
if (_h_input != 0){
if (!out_of_control) {hspeed = player_speed * _h_input;}
if (control_recovery_time > 10) {
out_of_control = false;
control_recovery_time = 0;
}
}else{
if (!out_of_control){
hspeed = 0;
}
}
if (place_meeting(x, y,obj_ground)){
gravity = 0;
vspeed = 0;
var _target = instance_place(x,y,obj_ground);
if (_target.bbox_top + 1 > y || _target.bbox_bottom - 1 < y){
if (_target.bbox_top > y) {
y = _target.bbox_top - (sprite_height div 2) + 1;
out_of_control = false;
jump_count = 0;
} else {
y = _target.bbox_bottom + (sprite_height div 2);
gravity = gravity_rate;
vspeed = 0.1;
}
}
}else{
gravity = gravity_rate;
}
```

```
var _nearest_wall = instance_nearest(x,y,obj_ground);
if (point_distance(x,y,_nearest_wall.x,_nearest_wall.y) > 20) {
if (hold_wall_time < hold_wall_time_max){
hold_wall_time++;
}
wall_jump_rebound = 0;
wall_jump = true;
}
if (keyboard_check_pressed(ord("F"))){
if (jump_count == 0){
audio_play_sound(snd_jump,1001,false);
vspeed = -jump_rate;
hspeed = wall_jump_rebound;
if (wall_jump_rebound != 0) {
out_of_control = true;
}
hold_wall_time = 0;
jump_count += 1;
}
}
if (out_of_control){
control_recovery_time++;
}
```

**플레이어가 벽 점프를 할때 좀더 매끄러운 조작을 위해서
20 스텝 (대략 0.3초)간 플레이어가 벽을 붙잡고 있도록
설계 하였습니다.**

대화 시스템



```
Create::
depth = -100;
is_dead = false;
str_buffer = ds_queue_create();
alarm[0] = 5;
str = "";
function str_buffer_create(_str){
    for(var _i = 1; _i <= string_length(_str); _i++){
        ds_queue_enqueue(str_buffer,string_char_at(_str,_i));
    }
}
function str_clear(){
    str = "";
}
Alarm[0]::
depth = -100;
is_dead = false;
str_buffer = ds_queue_create();
alarm[0] = 5;
str = "";
function str_buffer_create(_str){
    for(var _i = 1; _i <= string_length(_str); _i++){
        ds_queue_enqueue(str_buffer,string_char_at(_str,_i));
    }
}
function str_clear(){
    str = "";
}
```

```
Draw::
draw_set_alpha(image_alpha);
draw_sprite(spr_speechbubble_L,0,x-string_width(str)/2,y-12);
for(var _i = -string_width(str)/2; _i < string_width(str)/2; _i++){
    draw_sprite(spr_speechbubble,0,x+_i,y-12);
}
draw_sprite(spr_speechbubble_R,0,x+string_width(str)/2,y-12);
draw_set_font(fnt_GCJ_8);
draw_set_halign(fa_center);
draw_set_valign(fa_middle);
draw_set_color(c_gray);
draw_text(x + 1,y -11, str);
draw_set_color(c_black);
draw_text(x,y-12, str);
draw_set_alpha(1);
```

**말풍선은 입력받은 String을 큐에 한글자씩 삽입합니다.
Alarm[0]는 5스텝마다 반복하며 큐가 빌 때까지 문자를
하나씩 삽입합니다.**

**Draw에선 그렇게 늘어난 str의 크기에 맞춰 말풍선의 크
기를 늘립니다.**

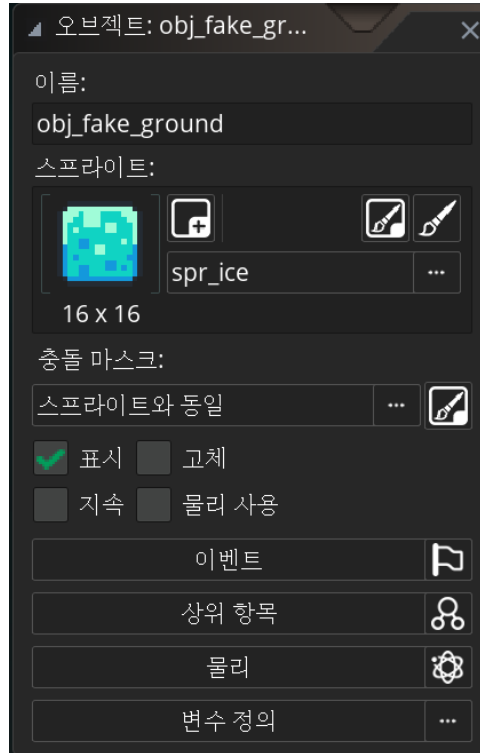
기믹 – 가짜 지형



**배경이 사실은 플레이어다, 라는 생각에서 출발해
지형지물도 사실은 플레이어다! 라는 컨셉의 기믹입니다.**

Stage를 시작할때 가시나 벽 처럼 자신을 속이지만 사실 플레이어의 움직임과 함께 움직이는 배경 입니다.

기믹 – 가짜 지형



```
ds_list_add(obj_player.fake_grounds,id);
mask_x = obj_player.x - x;
mask_y = obj_player.y - y;
if (global.stage_number < 10) {
    sprite_index = spr_ground1;
}else if (global.stage_number >= 10 && global.stage_number < 20){
    sprite_index = spr_ground2;
}else if (global.stage_number >= 20 && global.stage_number < 30){
    sprite_index = spr_ground3;
}else if (global.stage_number >= 30 && global.stage_number < 40){
    sprite_index = spr_ground4;
}
```

가짜 지형은 생성될때 플레이어의 List에 등록됩니다.

```
if (!ds_list_empty(fake_grounds)){
    for(var _i = 0; _i < ds_list_size(fake_grounds); _i++){
        fake_grounds[_i].x = x - fake_grounds[_i].mask_x;
        fake_grounds[_i].y = y - fake_grounds[_i].mask_y;
    }
}
```

플레이어는 움직일때마다 가짜 땅들의 위치를 같이 최신화 해 줍니다.