게임메이커 스튜디오(스크립트) (신)

□ 스프라이트

○ create 또는 step 이벤트

* sprite_index = 스프라이트 이미지; : 스프라이트의 이미지를 변경
* image_index = 숫자; : 지정한 스프라이트 내의 이미지 중 선택
  - 0 : 0번째 이미지 선택
  - -1 : 재생된 이미지를 출력
* image_speed = 숫자(0~1); : 이미지의 스피드를 설정
  - room에서 설정한 room speed 가 30일 경우, 0.1를 선택하면 3 speed가 되는 것.
* image_blend = 색상(c_color); : 해당 스프라이트에 색상을 입힌다.
  - c_white : 처음의 색상으로 설정한다.(투명한 색)
  - c_yellow, c_blue, c_aqua 등등
* image.weight : 해당 스프라이트의 넓이를 반환
* image.height : 해당 스프라이트의 높이를 반환

○ draw 이벤트

* draw_sprite_ext(스프라이트, x좌표, y좌표, 가로사이즈, 세로사이즈, angle, color, alpha);
  - 해당 좌표에 오브젝트가 아닌 설정한 스프라이트를 그려준다.
* draw_text(x, y, '텍스트'); : 해당 좌표에 텍스트를 draw한다.
* draw_text_ext(x, y, '텍스트', weight, alpha); : 해당 좌표에 텍스트를 draw한다. // 크기, 투명도 조절 추가
* draw_set_color(color); : draw_*함수에서 사용될 색 지정
* draw_set_alpha(alpha); : draw_*함수에서 사용될 불투명도 지정

□ 사운드

* audio_master_gain() : (0~1) 사이의 음량 조절이 가능하다.
* audio_stop(사운드); : (사운드)를 중지한다.
  - audio_stop_all(); : 현재 모든 사운드를 중지한다. // 룸 변경 시 사용
* sound_play(사운드, 순서, 루프(1,0)); : 해당 사운드를 플레이, 중요순서, 루프 (한 번만 플레이할 것 인지 설정)

□ 오브젝트 /// 오브젝트 상속 가능 ///

* instance_create(x, y, 오브젝트); : 해당 좌표에 오브젝트를 생성한다.
* instance_create_ext(x, y, 오브젝트, weight, height, angle, color, alpha);
  : 해당 좌표에 오브젝트를 생성한다. // (x좌표, y좌표, 오브젝트, 넓이, 높이, 회전률, 색상, 투명도)
* instance_destroy(); : 현재 인스턴스를 삭제한다.

* 변수 = instance_find(오브젝트, i); : i번 째 오브젝트의 지정번호를 변수에 집어넣음
* 변수 = instance_number(오브젝트); : 현재 room안의 해당 오브젝트 총 개수를 변수에 집어넣음

```
if ( instance_number(O_Item) == 0 )
    instance_create(x, y, O_Item2);
```

* instance_exists(오브젝트) : 해당 오브젝트가 존재하는지 확인
* instance_nearest(x, y, 오브젝트); : 현재 위치에서 해당 오브젝트 중 가장 가까이 있는 인스턴스의 위치를 반환
* motion_set(180,10); : 해당 오브젝트를 회전 시킨다. (회전각, 회전스피드)
* depth : 오브젝트의 깊이(앞/뒤 설정)

□ 룸

* room_goto(룸 명); : 해당 룸으로 이동한다.
* room_goto_next(); : 다음 룸으로 이동한다.
* room_restart(); : 해당 룸을 재시작 한다.
* room_height; : 룸의 높이를 리턴
* room_weight; : 룸의 넓이를 리턴

□ 스크립트

* 스크립트(); : 해당 스크립트를 불러온다.
* 스크립트 변수 사용법

```
/// script_a 스크립트
if ( keyboard_check(vk_down) ) {
    y += argument0;
}
if ( keyboard_check(vk_up) ) {
    y -= argument0;
}
if ( place_free(x, y+3) ) {
    vspeed = argument1;
}
```

```
/// 스크립트를 사용할 오브젝트
script_a(3, 15); // 각각 argument 0, 1에 넣어진다.
```

* global.변수 : 전역 변수로 설정 // 모든 인스턴스(오브젝트)에 적용 가능
* 오브젝트.변수 : 해당 오브젝트의 변수를 변경한다. // ex) O_Player.x = 3 : 플레이어의 x값을 3으로 지정
* exclusive(^) : 전등 스위치와 같은 원리 // ex) 임의의 bool 변수(tf) 일 경우 tf가 0일 때 tf ^= tf을 하면 1이 됨

○ 랜덤 및 계산 함수

* randomize(); : 랜덤 값을 일정하지 않게 바꿈 (시드설정)
* random_range(0,100); : 0~100 사이의 숫자중 하나를 랜덤으로 선택
* random(100); : 0~100 사이의 숫자중 하나를 랜덤으로 선택
* choose(0,1,2,3,4); : 0, 1, 2, 3, 4 중 하나를 랜덤으로 선택
* abs(x); : x의 절대값
* floor(x) : x의 소숫점 제거
* round(x) : x의 반올림
* ceil(x) : x의 올림
* clamp(val, min, max) : val이 min과 max범위에 넘어가지 않도록 값 반환
* min(x1, x2, ...) : 최소 값 반환
* max(x1, x2, ...) : 최대 값 반환

○ 키보드 입력 체크

* keyboard_check(키보드); : ()안의 키가 눌렸는지 체크.
  - (키보드) 중요키 : vk_space, vk_up, vk_left 등
  - (키보드) 영어키 : ord('X'), ord('Z') 등
* keyboard_check_pressed(키보드); : ()안의 해당 키를 눌렀을 때 체크.
* keyboard_check_released(키보드); : ()안의 해당 키를 뗐을 때 체크.

○ 마우스 입력 체크

* mouse_check(mb_left); : 마우스 왼쪽 키를 눌렀는지 체크
* mouse_wheel_up(); : 마우스 휠을 위로 올렸는지 체크
* mouse_wheel_down(); : 마우스 휠을 아래로 내렸는지 체크

○ 이동 구현

* move_towards_point(대상오브젝트.x, 대상오브젝트/y, 스피드); : 해당 오브젝트를 유도탄처럼 날아감
* x += 1; or x -= 1; // 좌표 이동
* y += 1; or y -= 1; // 좌표 이동
* friction : 마찰력(0~1 사이의 실수형 값)

○ 중력 구현

* gravity
  - direction : 방향(270도)
* vspeed : 수직 속력
  - vspeed > 0; : 오브젝트 중력 y++
  - vspeed = 0; : 오브젝트 중력의 값이 0 (정지 상태)
  - vspeed < 0; : 오브젝트 중력 y-- (하늘로 올라감)
  - 점프 지형 예제 (if문, 충돌, 다른 오브젝트의 변수 불러옴(vspeed))

```
if (place_meeting(x, y, O_Player)) { // 지금 지형이 플레이어와 만나면
    O_Player.vspeed = -5;          // 플레이어의 vspeed를 -5로 변경한다.
}
```

○ 오브젝트 충돌 체크

* place_empty(); : 빈 공간일 경우(룸 밖)
* place_free(x, y); : 현재 오브젝트의 x, y 좌표가 다른 오브젝트와 충돌하고 있지 않을 경우
* place_meeting(x, y, 오브젝트); : 현재 오브젝트의 x, y 좌표가 지정한 오브젝트와 충돌할 경우
* 몇 번째 오브젝트와 충돌했는지 알아보는 법

```
n = instance_number(오브젝트);
for (i = 0; i <= n; i++) { //게임메이커에서는 변수 생성 시 인자 타입을 설정 하지 않는다.
    crash = instance_find(오브젝트, I);
     // 충돌한 i번째 인스턴스를 임의의 변수 crash에 넣어줌
    switch (crash) {
        case 0 : 스크립트내용; break;
    }
}
```

○ 상대 오브젝트와의 거리 구하기

* 임의의 변수(distance) = sqrt(sqr(x-O_Player.x)*sqr(y-O_Player.y));
  // sqrt 또는 sqr : 제곱근(square root)

```
distance = sqrt(sqrt(x-O_Player.x)*sqrt(y-O_Player.y));
if (distance <=50 && O_player.x>x) {
    image_blend = c_aqua;
} // 현재 오브젝트와 대상 오브젝트와의 거리가 50보다 작을 경우
```

○ 타겟 오브젝트를 바라보는 화살표(8방향)

* angle : 회전 각도
* point_distance(x1, y1, x2, y2) : (x1, y1)과 (x2, y2) 사이의 거리 반환
* point_direction(x1, y1, x2, y2) : (x1, y1)과 (x2, y2) 사이의 각도 반환
* lengthdir_x(lengh, direction) : direction방향으로 length 만큼 이동했을 때 x축 변화량
* lengthdir_y(lengh, direction) : direction방향으로 length 만큼 이동했을 때 y축 변화량
* angle_difference(angle1, angle2) : angle1과 angle2의 각도 차이

```
/// 해당 오브젝트를 바라보고 이동
direction = point_distance(x1, y1, x2, y2);
speed = 5;
```

○ 메시지 대화상자

* Display_message(); : 디스플레이 메시지 창을 생성

○ 알람

* alarm[0] = 30; : room speed가 30일 경우, 알람0을 1초 후 실행한다.
  - 주의 : if문 안에 넣을 경우 그 if문이 계속 참이면 alarm이 계속 초기화 돼서 실행이 되지 않는다.

○ 폰트 설정

* setfont = ; : 폰트 설정

```
/// 미는 벽돌
if ( !place_meeting(x+2, y, O_Wall) or !place_meeting(x-2, y, O_Wall) ) {
// 벽과 충돌 시 멈춰야하기 때문
        if ( place_meeting(x-1, y, O_Player) ) {
                x+=2; // 짝수, 가로 길이 32에 맞추기 위함
        } else if ( place_meeting(x+1, y, O_Player) ) {
                x-=2;
        }
}

if ( place_free(x, y+3) ) { // 미는 벽이기 때문에 중력이 있어야 떨어진다.
        vspeed = 15;
} else {
        vspeed = 0;
}
```

```
/// 점프 벽돌
if (place_meeting(x, y, O_Player)) { // 지금 지형이 플레이어와 만나면
    O_Player.vspeed = -5;          // 플레이어의 vspeed를 -5로 변경한다.
}
```

```
/// 랜덤 벽돌
if ( place_meeting( x, y, O_Play_Bullet ) ) {
        instance_create(x, y, choose(O_Wall, O_Wall2, O_Castle_Wall, O_Castle_Wall_item, O_Magma_Wall) )
        instance_destroy();
}
```

○ 일시정지 기능

* pause = abs(pause-1) : anti-lock break system, 일시정지 역할을 함
* instance_deactivate_all(1) : 자기자신을 제외한 모든 인스턴스를 비활성화 시킴
* surface_create(room_width, room_height); : 서피스는 일종의 도화지
* surface_copy(paste, x, y, copy); : 서피스의 그림들을 paste서피스에 붙여넣는 역할을 함
  - surface_copy(suf, 0, 0, application_surface); : copy 서피스의 그림을 paste에 그려주고 비활성화
    - application_surface : 게임메이커의 기본적인 도화지
* draw_surface(suf, 0, 0); : suf서피스의 화면을 드로우해준다.

```
if ( !place_meeting(x+2, y, O_Wall) or !place_meeting(x-2, y, O_Wall) ) {
// 벽과 충돌 시 멈춰야하기 때문
        if ( place_meeting(x-1, y, O_Player) ) {
                x+=2; // 짝수, 가로 길이 32에 맞추기 위함
        } else if ( place_meeting(x+1, y, O_Player) ) {
                x-=2;
        }
}

if ( place_free(x, y+3) ) { // 미는 벽이기 때문에 중력이 있어야 떨어진다.
        vspeed = 15;
} else {
        vspeed = 0;
}
```

```
/// 점프 벽돌
if (place_meeting(x, y, O_Player)) { // 지금 지형이 플레이어와 만나면
    O_Player.vspeed = -5;          // 플레이어의 vspeed를 -5로 변경한다.
}
```

```
/// 랜덤 벽돌
if ( place_meeting( x, y, O_Play_Bullet ) ) {
        instance_create(x, y, choose(O_Wall, O_Wall2, O_Castle_Wall, O_Castle_Wall_item, O_Magma_Wall) )
        instance_destroy();
}
```

○ 문자열

* ord(str) : (str)의 문자를 ASCII 코드치로 반환
* string(val) : val를 문자열로 변환해 반환, 소수점 이하 2자리수까지 유효
* string_length(str) : str문자의 문자수를 반환

○ 이펙트 기능

* effect_create_below(kind, x, y, size, colour);
  - kind : ef_cloud / ef_ellipse / ef_explosion / ef_firework / ef_flare / ef_rain / ef_rain / ef_ring / ef_smoke / ef_smokeup / ef_snow / ef_spark / ef_star
  - size : 0~1
  - colour : c_color

image_speed = 수 -> (수)의 속도로 스프라이트가 변화.
ex - > image_speed = 30 -> 30의 속도로 스프라이트가 변화

image_angle = 각도 또는 오브젝트 -> (등식 뒤의 내용)으로 스프라이트의 각도가 바뀜
ex - > image_angle = 30 -> 30도로 스프라이트 각도가 바뀜
image_angle=point_direction(x,y,object.x,object.y) - 루나매직님의 댓글. 감사합니다. (이미지)가 (오브젝트)를 바라보게 한다
image_index = 스프라이트 안의 설정하고 싶은 사진의 번호 (?) = (등식 뒤의 내용)으로 스프라이트 모양이 바뀜

if keyboard_check_pressed(키) {내용} = (키)가 눌렸다면 {내용}해라. anykey를 넣으면 아무 키나.
if 변수 이름 <'=' 숫자 {내용} - (변수)가 (숫자) 보다 크면(크거나 같다면) {내용}해라.
>'=' 숫자 {내용} - (변수)가 (숫자) 보다 작으면(크거나 같다면) {내용}해라.
= 숫자 {내용} - (변수)가 (숫자) 와 같으면 {내용}해라.

if instance_number(오브젝트 이름) <'=' 숫자 {내용} - (오브젝트)가 (숫자) 보다 크면(크거나 같다면) {내용}해라.
>'=' 숫자 {내용} - (오브젝트)가 (숫자) 보다 작으면(크거나 같다면) {내용}해라.
= 숫자 {내용} - (오브젝트)가 (숫자) 와 같으면 {내용}해라.

keyboard_lastkey = 키보드 마지막으로 입력된 키.

스크립트 이름(); = (스크립트 이름)을 실행한다.

instance_destroy(); - 이거 제가 destory라고 쳐서 화났었죠 매우. 자신을 없앤다.
instance_create(x,y,오브젝트 이름) - (오브젝트)를 (x)와 (y)에 만든다.

effect_create_above(이펙트 번호, x, y, 숫자, 숫자2) - (이펙트 번호)를 (x)와 (y)앞에 (숫자)의 크기로 (숫자2)의 색으로 만든다.
이 이펙트 번호는 draw의 create effect에 있습니다. 세 보시고 쓰세요
effect_create_below(이펙트 번호, x, y, 숫자, 숫자2) - (이펙트 번호)를 (x)와 (y)뒤에 (숫자)의 크기로 (숫자2)의 색으로 만든다.

중심 오브젝트 주위를 계속 빙글빙글 돌게하는 그러한 방법. 누군지 모르겠지만 일단 강좌를 따옴. 문제되면 지우지워
이거 사용한게... 보시에 슈퍼소닉 소나이퍼패턴 다음에 보시 주위에 도는 그거..라고..해야....되..나 | ?
스텝에 사용

direction -= 숫자 - (숫자)만큼 반시계 방향으로 돈다. (숫자)가 크면 클 수록 많이많이 움직임
= 숫자 - (숫자)만큼 시계 방향으로 돈다. (숫자)가 크면 클 수록 많이많이 움직임
이건 뭔가 약간 다를 수 있어요? 반시계인지 헷갈려서
뭐.. 등등등등 많은 코드는 있지만..
설명하기 귀찮으므로 패스 나중에 다시 추가할 예정임다. 근데 솔직히 주로 쓰는거 빼고는 안쓰잖아요 instance나 image나.
..나만인가? 뭐 많이 다른거 쓰겠죠 뭐. 제 게임은 너무 간단해서

오브젝트를 보면요, 왼쪽에 깊이라는게 있잖아요. 그 깊이는 음수가 되면 앞에 나오고 양수가 되면 뒤로 갑니다. 이건 무슨 반대로 가는지.
하지만 음수도 순서가 있어요.
-100이 클까요 -10이 클까요? 물론 -10이 더 크죠. 그러니까 더 작은 -100이 앞에 나옵니다. 작으니까요. 키 순으로 생각하면 되요. 비유보소 키 작은 분들 죄송합니다.

패런츠라는게 있죠? 한글번역을 이렇게 해 놓으면 어떻게해. parents 뭐 이렇게 써있을거 같은데 부모님이라고 생각하면 편해요. 이 부모님
을 따라하기 때문에

==========

게임메이커 스튜디오(스크립트) (구)

❑ 스크립트

○ 스크립트 설정(이동 스크립트)

* 클래스처럼 불러올 수 있게 스크립트를 따로 생성한다.
* 키 인식 스크립트

스크립트 명	스크립트

scr_get_input	right_key = keyboard_check(vk_right);
	left_key = keyboard_check(vk_left);
	up_key = keyboard_check(vk_up);
	down_key = keyboard_check(vk_down);

* 이동 스크립트

스크립트 명	스크립트

scr_move_state	scr_get_input(); // 키 인식 스크립트를 불러옴
	if(running == true) {
	  if (right_key) {
	  phy_position_x += spd; // x좌표(오른쪽) 이동
	  sprite_index = spr_player_right; // 오른쪽 스프라이트를 사용
	  image_speed = .2; // 이미지의 모션의 변화 속도는 0.2
	  }
	  if (left_key) {
	  phy_position_x -= spd;
	  sprite_index = spr_player_left;
	  image_speed = .2;
	  }
	  if (!right_key and !left_key) { // 키를 누르고 있지 않을 때
	  image_speed = 0; // 이미지의 스피드를 0로 설정
	  image_index = 0; // 현재 스프라이트의 0번째 이미지로 설정
	  }
	}

* 이동 스크립트2 + 공격, 점프, 폭탄

스크립트 명	스크립트

scr_move_state	scr_get_input(); // 키 인식 스크립트를 불러옴
	if (right_key) {
	  if hspeed < maxspd+0.1{
	  hspeed += 2;
	  }
	  sprite_index = S_Player_move_right; // 오른쪽 스프라이트를 사용
	  image_speed = .2; // 이미지의 모션의 변화 속도는 0.2
	}
	if (left_key) {
	  if hspeed >-maxspd-0.1{
	  hspeed += -2;
	  }
	  sprite_index = S_Player_move_left;
	  image_speed = .2;
	}
	
	if (jump_key and jump<3) {
	  vspeed = -jh;
	  jump+=1;
	}
	
	if (down_key) {
	  sprite_index = S_Player_std_right;
	}
	
	if (!right_key and !left_key) { // 키를 누르고 있지 않을 때
	  image_speed = 0;
	  image_index = 0;
	}

* 서있는 상태 스크립트

스크립트 명	스크립트
scr_frozen_state	image_blend = c_aqua;

* 중력 스크립트

스크립트 명	스크립트

scr_gravity	/// scr_set_gravity
	if (place_free(x,y+5)){
	gravity = gra;
	}
	if (!place_free(x-hspeed-0.1,y)){ // left
	hspeed = -rc;
	friction = 0;
	}
	if (!place_free(x+hspeed+0.1,y)){ //right
	hspeed = rc;
	friction = 0;
	}
	x = round(x);
	y = round(y);
	friction = 0.7; //마찰력

○ 생성(O_플레이어) - Create ()

* 플레이어 오브젝트에서 일어날 이벤트의 초기 설정

스크립트

/// Set fixed rotation
jump=0;
scr_arguments(1,10,0.3,3,8,1); // gra,jh,spd,maxspd,sjh,rc
image_speed = 0;
scr_get_input();
state = scr_move_state; // 이동 상태를 넣어줌 (중요)

○ 이동(O_플레이어) - Step ()

* 키 인식

스크립트

/// state
scr_set_gravity(); // 중력 구현
script_execute(state); // 뛸 때의 상태

○ 충돌(O_플레이어) - Collision ()

스크립트

move_contact_solid(direction,0.1);
vspeed = 0; // 충돌 시 vspeed를 0
jump=0; // 땅과 충돌 시 점프를 0으로 만들어줌

○ 그림자 설정(O_플레이어) - Draw ()

* 스프라이트 Draw

기능	스크립트
그림자 스프라이트 설정	draw_sprite(spr_player_shadow, image_imdex, x, y); // (스프라이트,서브이미지,x,y)
플레이어의 스프라이트를 불러옴	draw_self(); // 플레이어 스프라이트

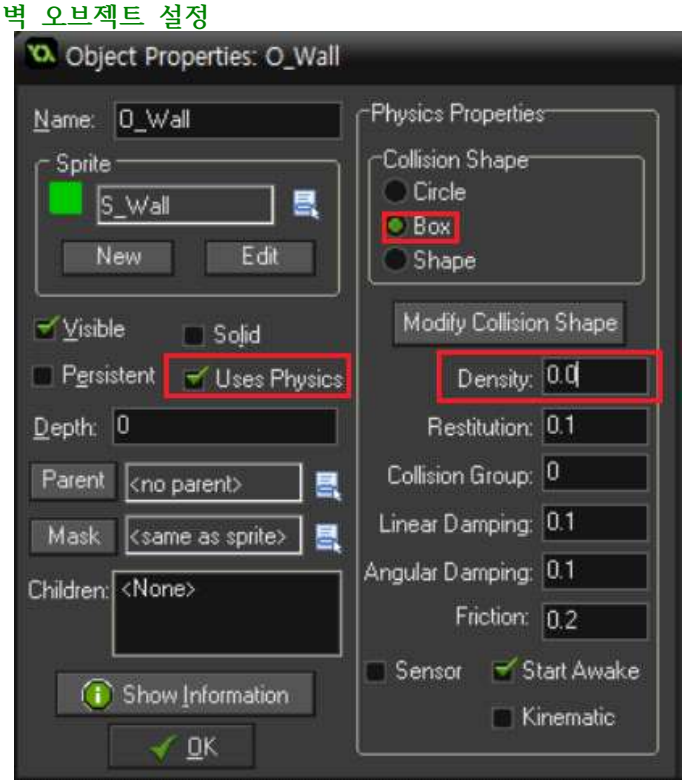
* 코드

기능	스크립트
맵 변경	room_goto(room이름);
인스턴스 생성	instance_create(x,y,O_bullet);
인스턴스 삭제	instance_destroy();
스프라이트 생성	sprite_index = 스프라이트;
스프라이트 속도	image_speed = .2;
총알 방향	else if(O_Player.x>O_eneymy.x)//주인공이 오른쪽에 있으면 오른쪽으로 발사 { x+=10 }
알람	alarm[0] = 30; // alarm[알람번호] = 속도
뷰 좌표	view_xview[0]; , view_yview[0]; // 인터페이스 시 사용
랜덤 함수	변수 = random_range(0,100); // 0~100 사이 숫자 랜덤
선택 함수	변수 = choose(0, 5, 10, 15); // ()안의 숫자 중 하나 랜덤

○ 스크립트 모음(여러개 사용 시)

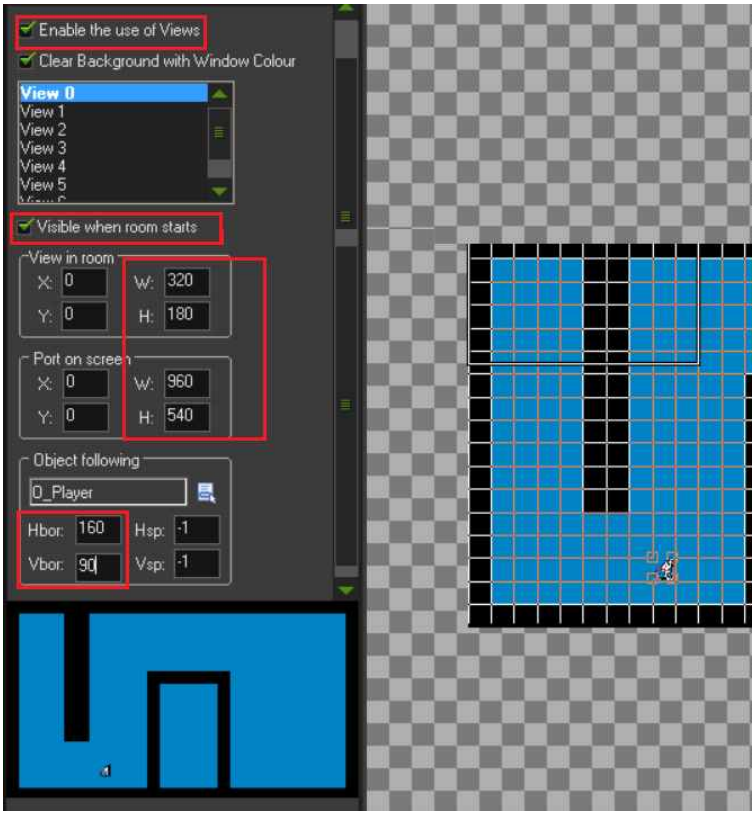
스크립트
변수 = argument숫자;

* 벽 오브젝트 설정



◒ Vew 탭 - 설정

* Object following의 Hboc과 Vboc의 크기는 뷰 크기의 절반으로 설정.



* 게임메이커 설정 변경