

게이미메이커 스튜디오(스크립트)

게이미메이커 스크립트 사이트 : http://www.gmlscripts.com/script/sprite_desaturate

□ 스프라이트 (Sprite)

○ 함수(function)

* :

```
function sprite_add (fname,imgnumb,removeback,smooth,xorig,yorig)
function sprite_add_from_surface (ind,id,x,y,w,h,removeback,smooth)
function sprite_assign (ind,source)
function sprite_collision_mask (ind,sepmasks,bboxmode,bbleft,bbright,bbtop,bbbottm,kind,tolerance)
function sprite_create_from_surface (id,x,y,w,h,removeback,smooth,xorig,yorig)
function sprite_delete (ind)
function sprite_duplicate (ind)
function sprite_exists (ind)
function sprite_flush (ind)
function sprite_flush_multi (indarray)
function sprite_get_bbox_bottom (ind)
function sprite_get_bbox_left (ind)
function sprite_get_bbox_right (ind)
function sprite_get_bbox_top (ind)
function sprite_get_height (ind)
function sprite_get_name (ind)
function sprite_get_number (ind)
function sprite_get_texture (spr,subimg)
function sprite_get_tpe (index,subindex)
function sprite_get_uvs (spr,subimg)
function sprite_get_width (ind)
function sprite_get_xoffset (ind)
function sprite_get_yoffset (ind)
function sprite_merge (ind1,ind2)
function sprite_prefetch (ind)
function sprite_prefetch_multi (indarray)
function sprite_replace (ind,fname,imgnumb,removeback,smooth,xorig,yorig)
function sprite_save (ind,subimg,fname)
function sprite_save_strip (ind,fname)
function sprite_set_alpha_from_sprite (ind,spr)
function sprite_set_cache_size (ind, max)
function sprite_set_cache_size_ext (ind, image, max)
function sprite_set_offset (ind,xoff,yoff)
```

○ 변수(variable)

- * `sprite_index` = 스프라이트 이미지; : 스프라이트의 이미지를 변경
- * `sprite_width` : 스프라이트의 가로 크기를 반환 및 설정
- * `sprite_height` : 스프라이트의 세로 크기를 반환 및 설정
- * `sprite_xoffset` :
- * `sprite_yoffset` :
- * `image_alpha` : 해당 이미지의 불투명도를 반환 및 설정
- * `image_angle` : 해당 이미지의 각도(angle)를 반환 및 설정
- * `image_blend` = `c_color`; : 해당 스프라이트에 색상을 입힌다.
 - `c_white` : 처음의 색상으로 설정한다.(투명한 색)
 - `c_yellow`, `c_blue`, `c_aqua` 등등
- * `image_index` = 숫자; : 지정한 스프라이트 내의 이미지 중 선택
 - 0 : 0번째 이미지 선택 / -1 : 재생된 이미지를 출력
- * `image_number` : 해당 이미지의 번호를 반환
- * `image_speed` = 숫자(0~1); : 이미지의 스피드를 설정
 - room에서 설정한 room speed 가 30일 경우, 0.1를 선택하면 3 speed가 되는 것.
- * `image_xscale` : 해당 스프라이트의 가로 비율을 반환
- * `image_yscale` : 해당 스프라이트의 세로 비율을 반환

□ 사운드 (Sound)

○ 함수(function)

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

```
function sound_delete(ind)
function sound_discard(ind)
function sound_exists(ind)
function sound_fade(index,value,time)
function sound_get_kind(ind)
function sound_get_name(ind)
function sound_get_preload(ind)
function sound_global_volume(value)
function sound_isplaying(index)
function sound_loop(index)
function sound_play(index)
function sound_restore(ind)
function sound_stop(index)
function sound_stop_all()
function sound_volume(index,value)
```

```
function audio_channel_num(numchannels)
function audio_create_buffer_sound ( bufferId, format, rate, offset, length, channels )
function audio_create_play_queue (bufferFormat, sampleRate, channels)
function audio_create_stream(filename)
function audio_create_sync_group(looping)
function audio_debug(enable)
function audio_destroy_stream(stream_sound_id)
function audio_destroy_sync_group(sync_group_id)
function audio_emitter_create()
function audio_emitter_exists(emitterid);
function audio_emitter_falloff(emitterid, falloff_ref_dist,falloff_max_dist,falloff_factor)
function audio_emitter_free(emitterid)
function audio_emitter_gain(emitterid,gain)
function audio_emitter_get_gain(emitterid)
function audio_emitter_get_listener_mask(emitterid)
function audio_emitter_get_pitch(emitterid)
function audio_emitter_get_vx(emitterid)
function audio_emitter_get_vy(emitterid)
function audio_emitter_get_vz(emitterid)
function audio_emitter_get_x(emitterid)
function audio_emitter_get_y(emitterid)
function audio_emitter_get_z(emitterid)
function audio_emitter_pitch(emitterid,pitch)
function audio_emitter_position(emitterid,x,y,z)
function audio_emitter_set_listener_mask(emitterid,mask)
function audio_emitter_velocity(emitterid,vx,vy,vz)
function audio_exists(soundid)
function audio_falloff_set_model(falloffmode)
function audio_free_buffer_sound ( soundId )
function audio_free_play_queue(queueId)
function audio_get_listener_count()
function audio_get_listener_info(index)
function audio_get_listener_mask()
function audio_get_master_gain(listenerIndex)
```

```

function audio_get_name(index)
function audio_get_recorder_count()
function audio_get_recorder_info(recorder_num)
function audio_get_type(soundid)
function audio_group_is_loaded(groupId)
function audio_group_load(groupId)
function audio_group_load_progress(groupId)
function audio_group_name(groupId)
function audio_group_set_gain(groupId, volume, time)
function audio_group_stop_all(groupId)
function audio_group_unload(groupId)
function audio_is_paused(soundid)
function audio_is_playing(soundid)
function audio_listener_get_data(index)
function audio_listener_orientation(lookat_x,lookat_y,lookat_z,up_x,up_y,up_z)
function audio_listener_position(x,y,z)
function audio_listener_set_orientation(index, lookat_x,lookat_y,lookat_z,up_x,up_y,up_z)
function audio_listener_set_position(index, x,y,z)
function audio_listener_set_velocity(index, vx,vy,vz)
function audio_listener_velocity(vx,vy,vz)
function audio_master_gain(gain)
function audio_pause_all()
function audio_pause_sound(soundid)
function audio_pause_sync_group(sync_group_id)
function audio_play_in_sync_group(sync_group_id,soundid)
function audio_play_sound(soundid,priority,loops)
function audio_play_sound_at(soundid,x,y,z, falloff_ref_dist,falloff_max_dist,falloff_factor,loops, priority)
function audio_play_sound_on(emitterid,soundid,loops,priority)
function audio_queue_sound(queueId, buffer_id, offset, length)
function audio_resume_all()
function audio_resume_sound(soundid)
function audio_resume_sync_group(sync_group_id)
function audio_set_listener_mask(mask)
function audio_set_master_gain(listenerIndex, gain)

```

```

function audio_sound_gain(index,volume,time)
function audio_sound_get_gain(index)
function audio_sound_get_listener_mask(soundid)
function audio_sound_get_pitch(index)
function audio_sound_get_track_position(index)
function audio_sound_length(soundid)
function audio_sound_pitch(index,pitch)
function audio_sound_set_listener_mask(soundid,mask)
function audio_sound_set_track_position(index, time)
function audio_start_recording(recorder_num)
function audio_start_sync_group(sync_group_id)
function audio_stop_all()
function audio_stop_recording(channel_index)
function audio_stop_sound(soundid)
function audio_stop_sync_group(sync_group_id)
function audio_sync_group_debug(sync_group_id)
function audio_sync_group_get_track_pos(sync_group_id)
function audio_sync_group_is_playing(sync_group_id)
function audio_system()

```

○ 변수(variable)

```

constant audio_3d
constant audio_falloff_exponent_distance
constant audio_falloff_exponent_distance_clamped
constant audio_falloff_inverse_distance
constant audio_falloff_inverse_distance_clamped
constant audio_falloff_linear_distance
constant audio_falloff_linear_distance_clamped
constant audio_falloff_none
constant audio_mono
constant audio_new_system
constant audio_old_system
constant audio_stereo
audiogroup audiogroup_default

```


□ 패스 (Path)

○ 함수(function)

- * `path_add();` : 패스를 추가한다.
- * `path_add_point(ind, x, y, speed);` : 해당 패스(ind)에 해당 좌표(x, y)에 speed를 가진 패스를 추가
- * `path_append(ind, path);` : 해당 패스(ind)에 새로운 패스(path)를 추가한다.
- * `path_assign(ind, path);` :
- * `path_change_point(ind, n, x, y, speed);` : 해당 패스(ind)의 n번째 패스를 x, y, speed로 변경한다.
- * `path_clear_points(ind);` : 해당 패스(ind)를 초기화 시킨다.
- * `path_delete(ind);` : 해당 패스(ind)를 삭제시킨다.
- * `path_delete_point(ind, n);` : 해당 패스(ind)의 n번째 패스를 삭제시킨다.
- * `path_duplicate(ind);` : 해당 패스(ind)를 복제한다.
- * `path_end();` : 패스를 종료한다.
- * `path_exists(ind);` : 해당 패스(ind)가 존재하는지 확인한다.
- * `path_flip(ind);` : 해당 패스(ind)를 반전한다.
- * `path_get_closed(ind);` : 해당 패스(ind)의 종료지점을 반환한다.
- * `path_get_kind(ind);` : 해당 패스(ind)의 kind(Straight lines(0) 또는 Smooth curve(1))를 반환한다.
- * `path_get_length(ind);` : 해당 패스(ind)의 길이를 반환한다.
- * `path_get_name(ind);` : 해당 패스(ind)의 이름을 반환한다.
- * `path_get_number(ind);` : 해당 패스(ind)의 개수를 반환한다.
- * `path_get_point_speed(ind, n);` : 해당 패스(ind)의 스피드(speed)를 반환한다.
- * `path_get_point_x(ind, n);` : 해당 패스(ind)의 n번째 점의 x좌표를 반환한다.
- * `path_get_point_y(ind, n);` : 해당 패스(ind)의 n번째 점의 y좌표를 반환한다.
- * `path_get_precision(ind);` : 해당 패스(ind)의 정확도(precision)를 반환한다.
// precision은 패스에 딱 맞춰서 이동하는지, 스무디하게 이동하는지 결정
- * `path_get_speed(ind, pos);` : 해당 패스(ind)에서 해당 포지션(pos)의 스피드를 반환
- * `path_get_time(ind, speed);` :
- * `path_get_x(ind, pos);` : 해당 패스(ind)에서 해당 포지션(pos)의 x좌표를 반환
- * `path_get_y(ind, pos);` : 해당 패스(ind)에서 해당 포지션(pos)의 y좌표를 반환
- * `path_insert_point(ind, n, x, y, speed);` : 해당 패스(ind)의 n번째에 x, y, speed의 패스 점을 추가
- * `path_mirror(ind);` : 해당 패스(ind)를 반전
- * `path_rescale(ind, xscale, yscale);` : 해당 패스(ind)의 가로(xscale), 세로(yscale) 비율을 재설정
- * `path_reverse(ind);` :
- * `path_rotate(ind, angle);` : 해당 패스(ind)를 angle만큼 회전
- * `path_set_closed(ind, closed);` : 해당 패스(ind)의 끝 지점(closed)을 설정
- * `path_set_kind(ind, kind);` : 해당 패스(ind)의 kind(Straight lines(0) 또는 Smooth curve(1))를 설정
- * `path_set_precision(ind, prec);` : 해당 패스(ind)의 정확도(precision)를 설정
// precision은 패스에 딱 맞춰서 이동하는지, 스무디하게 이동하는지 결정
- * `path_shift(ind, xshift, yshift);` :
- * `path_start(path, speed, endaction, absolute);` : 해당 패스(ind)를 시작
 - speed : 속도
 - endaction : 종료 시 행동, 0 : 종료 / 1 : 반복 / 2 : 반대로 돌아감

○ 변수(variable)

- * `path_endaction` : 종료 액션을 반환
- * `path_index` :
- * `path_orientation` :
- * `path_position` :
- * `path_positionprevious` :

- * path_scale :
- * path_speed :
- * path_action_continue :
- * path_action_restart :
- * path_action_reverse :
- * path_action_stop :

□ 스크립트 (Script)

- * 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

□ 타임라인 (Time Lines) : 스토리 이벤트, 리듬 게임, 화면 흔들림 등을 만들 때 사용

○ 함수(function)

- * timeline_add(); : 타임라인을 추가한다.
- * timeline_clear(ind); : 해당 타임라인(ind)을 비운다.
- * timeline_delete(ind); : 해당 타임라인(ind)를 삭제한다.
- * timeline_exists(ind); : 해당 타임라인(ind)가 존재하는지 확인한다.
- * timeline_get_name(ind); : 해당 타임라인(ind)의 이름을 반환한다.
- * timeline_max_moment(ind); : 해당 타임라인(ind)의 마지막 스텝(moment)을 반환한다.
- * timeline_moment_add_script(ind, step, script); : 해당 타임라인(ind)의 시간(step)에 스크립트를 추가
- * timeline_moment_clear(ind, step); : 해당 타임라인(ind)의 시간(step)의 코드들을 비워준다.
- * timeline_size(ind); : 해당 타임라인(ind)의 크기를 반환한다.

○ 변수(variable)

- * timeline_index = 타임라인 이름; : 타임라인을 선택하여 불러온다.
- * timeline_loop = false; : 타임라인을 반복할 것인지 설정
- * timeline_position = 0; : 타임라인의 포지션(번호) 선택
- * timeline_running = true; : 타임라인을 활성화 시킨다.
- * timeline_speed; : 타임라인의 스피드를 설정

□ 오브젝트 (Object) /// 오브젝트 상속 가능 ///

○ 기본 설정

- * depth : 오브젝트의 깊이(앞/뒤 설정)
- * Persistent : 룸을 이동하여도 오브젝트가 삭제되지 않는다.
- * Parent : 해당 오브젝트에 상속을 받는다.
- * Children : 현재 오브젝트에 상속받고 있는 오브젝트를 나타낸다.

○ 함수(function)

- * `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
- * `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)

```
function instance_activate_all()
function instance_activate_object(obj)
function instance_activate_region(left,top,width,height,inside)
function instance_change(obj,performevents)
function instance_copy(performevent)
function instance_create(x,y,obj)
function instance_deactivate_all(notme)
function instance_deactivate_object(obj)
function instance_deactivate_region(left,top,width,height,inside,notme)
function instance_destroy()
function instance_exists(obj)
function instance_find(obj,n)
function instance_furthest(x,y,obj)
function instance_nearest(x,y,obj)
function instance_number(obj)
function instance_place(x,y,obj)
function instance_position(x,y,obj)
```

```
function object_exists(ind)
function object_get_depth(ind)
function object_get_mask(ind)
function object_get_name(ind)
function object_get_parent(ind)
function object_get_persistent(ind)
function object_get_physics(ind)
function object_get_solid(ind)
function object_get_sprite(ind)
function object_get_visible(ind)
function object_is_ancestor(ind_child,ind_parent)
function object_set_depth(ind,depth)
function object_set_mask(ind,spr)
function object_set_persistent(ind,pers)
function object_set_solid(ind,solid)
function object_set_sprite(ind,spr)
function object_set_visible(ind,vis)
```

○ 변수(variable)

- * `instance_count`
- * `instance_id`
- * `object_index`

□ 룸(Room)

○ 함수(function)

- * `room_add()`; : 룸을 추가한다.
- * `room_assign(ind, source)`; :
- * `room_duplicate(ind)`; : 해당 룸(ind)을 복사한다.
- * `room_exists(ind)`; : 해당 룸(ind)가 존재하는지 확인한다.
- * `room_get_name(ind)`; : 해당 룸(ind)의 이름을 반환한다.
- * `room_goto(numb)`; : 해당 룸(numb)으로 이동한다.
- * `room_goto_next()`; : 다음 룸으로 이동한다.

- * `room_goto_previous()`; : 이전 룸으로 이동한다.
- * `room_next(numb)`; : 해당 룸(numb)의 다음 룸을 반환한다.
- * `room_previous(numb)`; : 해당 룸(numb)의 이전 룸을 반환한다.
- * `room_restart()`; : 해당 룸을 재시작 한다.
- * `room_instance_add(ind, x, y, obj)`; : 해당 룸(ind)의 x, y좌표에 오브젝트(obj)를 추가한다.
- * `room_instance_clear(ind)`; : 해당 룸(ind)의 모든 오브젝트를 없앤다.
- * `room_set_background(ind, bind, vis, fore, back, x, y, htil, vtil, hspeed, vspeed, alpha)`;
:
- * `room_set_background_colour(ind, col, show)`; :
- * `room_set_height(ind, h)`; : 해당 룸(ind)의 높이(h)를 지정한다.
- * `room_set_width(ind, w)`; : 해당 룸(ind)의 너비(w)를 지정한다.
- * `room_set_persistent(ind, pers)`; :
- * `room_set_view(ind, vind, vis, xview, yview, wview, hview, xport, yport, wport, hport, hborder, vborder, hspeed, vspeed, object)`;
: 해당 룸(ind)에 뷰를 지정한다.
- * `room_set_view_enabled(ind, val)`; : 해당 룸(ind)에 해당 view[번호](val)가 존재하는지 확인한다.
- * `room_tile_add(ind, back, left, top, width, height, x, y, depth)`; :
- * `room_tile_add_ext(ind, back, left, top, width, height, x, y, depth, xscale, yscale, alpha)`; :
- * `room_tile_clear(ind)`; : 해당 룸(ind)의 타일을 제거한다.

○ 변수(variable)

- * `room_height`; : 룸의 높이를 리턴
- * `room_weight`; : 룸의 넓이를 리턴

```
variable room_caption
variable room_first
variable room_height
variable room_last
variable room_persistent
variable room_speed
variable room_width
```

□ 게임메이커 함수

○ 랜덤 및 계산 함수

- * `randomize()`; : 랜덤 값을 일정하지 않게 바꿈 (시드설정)
- * `random_range(0,100)`; : 0~100 사이의 숫자중 하나를 랜덤으로 선택(소수 포함)
- * `random(100)`; : 0~100 사이의 숫자중 하나를 랜덤으로 선택(소수 포함)
- * `random_range(0,100)`; : 0~100 사이의 숫자중 하나를 랜덤으로 선택(정수만)
- * `irandom(100)`; : 0~100 사이의 숫자중 하나를 랜덤으로 선택(정수만)
- * `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.
- * `keyboard_lastkey` : 마지막으로 누른 키를 반환

○ 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

○ 상대 오브젝트와의 거리 구하기

- * 임의의 변수(`distance`) = `sqrt(sqrt(x-O_Player.x)*sqrt(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
```

○ 이동 구현

- * `x += 1`; or `x -= 1`; // 좌표 이동 * `y += 1`; or `y -= 1`; // 좌표 이동
- * `hspeed += 1`; or `hspeed -= 1`; // 가로의 스피드를 주어 이동
- * `vspeed += 1`; or `vspeed -= 1`; // 세로의 스피드를 주어 이동

- * `move_towards_point(대상오브젝트.x, 대상오브젝트.y, 스피드)`: : 해당 오브젝트를 유도탄처럼 날아감
- * `mp_linear_step(x, y, speed, checkall)`: : 해당 좌표로 날아가지만 장애물에 부딪히면 멈춤
- * `mp_linear_step_object(x, y, speed, obj)`: : 해당 좌표로 날아가지만 특정 장애물에 부딪히면 멈춤
- * `mp_potential_step(x, y, speed, checkall)`: : 해당 좌표로 날아가지만 장애물을 인지하고 피해서 감
- * `mp_potential_step_object(x, y, speed, obj)`: : 해당 좌표로 날아가지만 특정 장애물을 인지하고 피해서 감
- * `move_contact_solid(dir, maxdist)`: : `solid`오브젝트만 체크하며, 해당 방향으로 해당 속도로 날아가서 붙음
- * `move_contact_all(dir, maxdist)`: : 모든 오브젝트를 체크하며, 해당 방향으로 `maxdist`속도로 날아감
- * `move_outside_solid(dir, maxdist)`: : `solid`오브젝트만 체크하며, 장애물 외곽으로 이동시킬 수 있다.
- * `move_outside_all(dir, maxdist)`: : 모든 오브젝트를 체크하며, 장애물 외곽으로 이동시킬 수 있다.

○ 중력 구현

- * **friction** : 마찰력(0~1 사이의 실수형 값)
- * **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`))
// `O_Player.vspeed = -5`; // 플레이어의 `vspeed`를 -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
- * place_snapped(hsnap, vsnap);
- * 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;
    }
}
```

○ 좌표의 거리 / 각도 측정

- * 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(length, direction) : direction방향으로 length 만큼 이동했을 때 x축 변화량
- * lengthdir_y(length, direction) : direction방향으로 length 만큼 이동했을 때 y축 변화량
- * angle_difference(angle1, angle2) : angle1과 angle2의 각도 차이

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

```
function point_direction(x1,y1,x2,y2)
function point_distance(x1,y1,x2,y2)
function point_distance_3d(x1,y1,z1,x2,y2,z2)
function point_in_circle(px,py,cx,cy,rad)
function point_in_rectangle(px,py,x1,y1,x2,y2)
function point_in_triangle(px,py,x1,y1,x2,y2,x3,y3)
```

○ 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_halign(fa_center); : 텍스트 가운데 정렬(가로)
- * draw_set_valign(fa_center); : 텍스트 가운데 정렬(세로)
- * draw_set_font(font) : 텍스트 폰트 지정 (폰트이름)
- * draw_set_color(color); : draw_*함수에서 사용될 색 지정
- * draw_set_alpha(alpha); : draw_*함수에서 사용될 불투명도 지정
- * draw_rectangle_colour(x, y, width, height, color, color, color, color, outline); : 사각형을 그려줌

```

function draw_arrow(x1,y1,x2,y2,size)
function draw_background(back,x,y)
function draw_background_ext(back,x,y,xscale,yscale,rot,colour,alpha)
function draw_background_general(back,left,top,width,height,x,y,xscale,yscale,rot,c1,c2,c3,c4,alpha)
function draw_background_part(back,left,top,width,height,x,y)
function draw_background_part_ext(back,left,top,width,height,x,y,xscale,yscale,colour,alpha)
function draw_background_stretched(back,x,y,w,h)
function draw_background_stretched_ext(back,x,y,w,h,colour,alpha)
function draw_background_tiled(back,x,y)
function draw_background_tiled_ext(back,x,y,xscale,yscale,colour,alpha)
function draw_button(x1,y1,x2,y2,up)
function draw_circle(x,y,r,outline)
function draw_circle_colour(x,y,r,col1,col2,outline)
function draw_clear(col)
function draw_clear_alpha(col,alpha)
function draw_ellipse(x1,y1,x2,y2,outline)
function draw_ellipse_colour(x1,y1,x2,y2,col1,col2,outline)
function draw_enable_alphablend(enable)
function draw_enable_drawevent(enable)
function draw_enable_swf_aa(enable)
function draw_flush()
function draw_get_alpha()
function draw_get_alpha_test()
function draw_get_alpha_test_ref_value()
function draw_get_colour()
function draw_get_swf_aa_level()
function draw_getpixel(x,y)
function draw_getpixel_ext(x,y)
function draw_healthbar(x1,y1,x2,y2,amount,backcol,mincol,maxcol,direction,showback,showborder)
function draw_highscore(x1,y1,x2,y2)
function draw_line(x1,y1,x2,y2)
function draw_line_colour(x1,y1,x2,y2,col1,col2)
function draw_line_width(x1,y1,x2,y2,w)
function draw_line_width_colour(x1,y1,x2,y2,w,col1,col2)

```

```

function draw_path(path,x,y,absolute)
function draw_point(x,y)
function draw_point_colour(x,y,col1)
function draw_primitive_begin(kind)
function draw_primitive_begin_texture(kind,texid)
function draw_primitive_end()
function draw_rectangle(x1,y1,x2,y2,outline)
function draw_rectangle_colour(x1,y1,x2,y2,col1,col2,col3,col4,outline)
function draw_roundrect(x1,y1,x2,y2,outline)
function draw_roundrect_colour(x1,y1,x2,y2,col1,col2,outline)
function draw_roundrect_colour_ext(x1,y1,x2,y2,radiusx,radiusy,col1,col2,outline)
function draw_roundrect_ext(x1,y1,x2,y2,radiusx,radiusy,outline)
function draw_self()
function draw_set_alpha(alpha)
function draw_set_alpha_test(enable)
function draw_set_alpha_test_ref_value(value)
function draw_set_blend_mode(mode)
function draw_set_blend_mode_ext(src,dest)
function draw_set_circle_precision(precision)
function draw_set_colour(col)
function draw_set_colour_write_enable(red,green,blue,alpha)
function draw_set_font(font)
function draw_set_halign(halign)
function draw_set_swf_aa_level(aa_level)
function draw_set_valign(valign)
function draw_skeleton(sprite,animname,skinname,frame,x,y,xscale,yscale,rot,colour,alpha)
function draw_skeleton_collision(sprite,animname,frame,x,y,xscale,yscale,rot,colour)
function draw_skeleton_time(sprite,animname,skinname,time,x,y,xscale,yscale,rot,colour,alpha)
function draw_sprite(sprite,subimg,x,y)
function draw_sprite_ext(sprite,subimg,x,y,xscale,yscale,rot,colour,alpha)
function draw_sprite_general(sprite,subimg,left,top,width,height,x,y,xscale,yscale,rot,c1,c2,c3,c4,alpha)
function draw_sprite_part(sprite,subimg,left,top,width,height,x,y)
function draw_sprite_part_ext(sprite,subimg,left,top,width,height,x,y,xscale,yscale,colour,alpha)
function draw_sprite_pos(sprite,subimg,x1,y1,x2,y2,x3,y3,x4,y4,alpha)

```

```

function draw_sprite_stretched(sprite, subimg, x, y, w, h)
function draw_sprite_stretched_ext(sprite, subimg, x, y, w, h, colour, alpha)
function draw_sprite_tiled(sprite, subimg, x, y)
function draw_sprite_tiled_ext(sprite, subimg, x, y, xscale, yscale, colour, alpha)
function draw_surface(id, x, y)
function draw_surface_ext(id, x, y, xscale, yscale, rot, colour, alpha)
function draw_surface_general(id, left, top, width, height, x, y, xscale, yscale, rot, c1, c2, c3, c4, alpha)
function draw_surface_part(id, left, top, width, height, x, y)
function draw_surface_part_ext(id, left, top, width, height, x, y, xscale, yscale, colour, alpha)
function draw_surface_stretched(id, x, y, w, h)
function draw_surface_stretched_ext(id, x, y, w, h, colour, alpha)
function draw_surface_tiled(id, x, y)
function draw_surface_tiled_ext(id, x, y, xscale, yscale, colour, alpha)
function draw_text(x, y, string)
function draw_text_colour(x, y, string, c1, c2, c3, c4, alpha)
function draw_text_ext(x, y, string, sep, w)
function draw_text_ext_colour(x, y, string, sep, w, c1, c2, c3, c4, alpha)
function draw_text_ext_transformed(x, y, string, sep, w, xscale, yscale, angle)
function draw_text_ext_transformed_colour(x, y, string, sep, w, xscale, yscale, angle, c1, c2, c3, c4, alpha)
function draw_text_transformed(x, y, string, xscale, yscale, angle)
function draw_text_transformed_colour(x, y, string, xscale, yscale, angle, c1, c2, c3, c4, alpha)
function draw_texture_flush()
function draw_triangle(x1, y1, x2, y2, x3, y3, outline)
function draw_triangle_colour(x1, y1, x2, y2, x3, y3, col1, col2, col3, outline)
function draw_vertex(x, y)
function draw_vertex_colour(x, y, col, alpha)
function draw_vertex_texture(x, y, xtex, ytex)
function draw_vertex_texture_colour(x, y, xtex, ytex, col, alpha)

```

○ 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
- * `string_char_at(str, n)` : n번째의 문자를 반환

○ 폰트 설정

- * `setfont = ;` : 폰트 설정

○ 서피스 - 일시정지 기능

- * `pause = 1;` : 일시정지 역할을 함
- * `instance_deactivate_all(1)` : 자기자신을 제외한 모든 인스턴스를 비활성화 시킴
- * `suf = surface_create(room_width, room_height);`
: 서피스는 일종의 도화지 // (0, 0)부터 (룸의 크기까지) 서피스를 만들어 줌
- * `surface_copy(suf, x, y, copy);` : 서피스의 그림들을 paste 서피스에 붙여넣는 역할을 함
- `surface_copy(suf, view_xview[0], view_yview[0], application_surface);`
: 만든 서피스를 현재 view의 위치에 붙여줌
- `application_surface` : 게임메이커의 기본적인 도화지
- * `surface_resize(suf, view_wview[0], view_hview[0]);` : 서피스의 사이즈를 재조정한다.
// 룸 시작시 System에 넣어준다.
- * `draw_surface(suf, 0, 0);` : suf서피스의 화면을 드로우해준다.

```

function surface_copy(destination, x, y, source)
function surface_copy_part(destination, x, y, source, xs, ys, ws, hs)
function surface_create(w, h)
function surface_create_ext(name, w, h)
function surface_exists(id)
function surface_free(id)
function surface_get_height(id)
function surface_get_texture(id)
function surface_get_width(id)
function surface_getpixel(id, x, y)
function surface_getpixel_ext(id, x, y)
function surface_reset_target()
function surface_resize(id, width, height)
function surface_save(id, fname)
function surface_save_part(id, fname, x, y, w, h)
function surface_set_target(id)
function surface_set_target_ext(index, id)

```

○ 이펙트 기능

- * `effect_create_below(kind, x, y, size, colour);`
- * `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

○ View(뷰)

- * `view_visible[0]` :
- * `view_enable[0]` :
- * `view_angle[0]` : 뷰의 각도를 반환 또는 조정
- * `view_surface_id[0]` :

- * `view_xview[0]` : 뷰의 x위치를 반환 또는 조정
- * `view_yview[0]` : 뷰의 y위치를 반환 또는 조정
- * `view_wview[0]` : 뷰의 가로 길이를 반환 또는 조정
- * `view_hview[0]` : 뷰의 세로 길이를 반환 또는 조정

- * `view_xport[0]` :
- * `view_yport[0]` :
- * `view_hport[0]` :
- * `view_vport[0]` :

- * `view_object[0]` : 뷰가 따라다닐 오브젝트를 반환 또는 조정
- * `view_hborder[0]` : 뷰에서 따라다닐 오브젝트의 가로 위치 반환 또는 조정
- * `view_vborder[0]` : 뷰에서 따라다닐 오브젝트의 세로 위치 반환 또는 조정
- * `view_hspeed[0]` :
- * `view_vspeed[0]` :

○ 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.

○ ini파일 작성

- * ini파일의 구조 : `[section] key = value` // [섹션] 아래에 키와 그 값으로 구성
- * ex) `[option] data_01 = "1" data_02 = "hello, world!"`

- * `ini_open(name);` : ini파일 호출
- * `ini_close();` : ini파일을 닫음(반드시 닫아야함)

- * `ini_write_string(section, key, value);` : 문자열을 ini 파일에 기록
 - ex) `ini_write_string("option", "data_02", "Hello, World!");`
- * `ini_write_real(section, key, value);` : 숫자를 ini 파일에 기록
 - ex) `ini_write_real("option", "data_01", 5);`

- * `ini_read_string(section, key, default)` : 문자열을 ini 파일을 읽어들이м
- * `ini_read_real(section, key, default)` : 문자열을 ini 파일을 읽어들이м
 - default : 값을 읽는데 실패했을 때 반환하는 값 지정
- * `ini_section_exists(section)` : 지정한 section이 존재하는지 체크
- * `ini_key_exists(section, key)` : 지정한 section의 key가 존재하는지 체크
 - 키가 없으면 false(0), 키가 있으면 true(1)을 반환
- * `ini_section_delete(section)` : 지정한 section을 삭제 (하위 키들도 삭제 됨)
- * `ini_key_delete(section, key)` : 지정한 section의 key를 삭제

🔴 View 탭 - 설정

- * Object following의 Hboc과 Vboc의 크기는 뷰 크기의 절반으로 설정.

