

게임메이커 스튜디오(스크립트)

□ 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

□ 사운드

- * `audio_master_gain(0~1)` : (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 ///

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

□ 타임라인 : 스토리 이벤트, 리듬 게임, 화면 흔들림 등을 만들 때 사용

- * `timeline_index = TL_Window_Shake`; : 불러올 타임라인을 선택
- * `timeline_position = 0`; : 타임라인의 포지션(번호) 선택
- * `timeline_running = true`; : 타임라인을 활성화 시킴
- * `timeline_loop = false`; : 타임라인을 반복할 것인지 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

□ 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 : 마지막에 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(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
```

○ 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);` : solid오브젝트만 체크하며, 해당 방향으로 해당 속도로 날아가서 붙음
- * `move_contact_all(dir, maxdist);` : 모든 오브젝트를 체크하며, 해당 방향으로 maxdist속도로 날아감
- * `move_outside_solid(dir, maxdist);` : solid오브젝트만 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 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;
    }
}
```

○ 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;
```

○ 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_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

○ 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
- * `draw_surface(suf, 0, 0);` : suf서피스의 화면을 드로우해준다.

○ 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_xview[0]` : 뷰의 x위치를 반환
- * `view_yview[0]` : 뷰의 y위치를 반환
- * `view_wview[0]` : 뷰의 가로 길이를 반환
- * `view_hview[0]` : 뷰의 세로 길이를 반환
- * `view_object[0]` : 뷰가 따라다닐 오브젝트를 설정
- * `view_hborder[0]` : 뷰에서 따라다닐 오브젝트의 가로 위치 설정
- * `view_vborder[0]` : 뷰에서 따라다닐 오브젝트의 세로 위치 설정

○ 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 읽어들이
- * `ini_read_real(section, key, default);` : 문자열을 ini 파일을 읽어들이
 - 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.

