

Menu kontekstowe obiektów w room editor.

Delete / delete all:

Dwie pierwsze pozycje są jak nazwa wskazuje do usuwania wskazanego obiektu (pozycja delete) i usuwania wszystkich obiektów które są pod kursorem w momencie rozwinięcia menu (pozycja delete all), te opcje są mało ważne bo łatwiej i szybciej jest po prostu kliknąć prawym przyciskiem myszy na któryś z obiektów.

Change position of the instance / snap to grid:

Obie pozycje służą do zmiany pozycji klikniętego obiektu, jeżeli musimy przenieść dany obiekt w konkretną pozycje to można skorzystać z „Change position...” wystarczy wpisać pozycje ‘x’ oraz ‘y’. Snap to grid służy do przyciągnięcia obiektu do siatki, jest to mało pożyteczna opcja ponieważ każdy obiekt jest przyciągany automatycznie no chyba że ktoś ustawiał obiekt za pomocą altu i lewego przycisku myszy.

Send to back / bring to front:

Jeżeli stawiamy obiekt i na niego nałożymy inny obiekt o tym samym “depth” czyli głębi to ten drugi przykryje pierwszego, zamiast usuwać oba obiekty i zmienić ich kolejność stawiania można skorzystać z tych dwóch opcji. „Send to back” przesuwca wskazany obiekt a „bring to front „ wypycha go na wierzch.

Locked:

Podczas tworzenia gry wiele razy potrzebujemy jakiegoś obiektu kontrolującego odpowiadającego za np. wyświetlanie interfejsu czy też spałnowania nowych potworków. W tym celu stawiamy gdzieś taki obiekt i zapominamy o nim. Kiedy przyjdzie moment gdzie cały level zawalimy obiektami i one zakryją kontroler to czasem podczas usuwania niepotrzebnych elementów np. sceny usuwamy przypadkowo ten kontroler. Wtedy po odpaleniu gry powstaje zdziwienie bo wyskakują błędy których wcześniej nie było. Szybko kapujemy że brakuje jednego z kontrolerów i stawiamy go. Ale jeżeli ten kontroler musi być postawiony jako pierwszy obiekt bo np. tworzy on zmienne globalne których domagają się inne obiekty to po uruchomieniu gry okazuje się że mimo postawienia brakującego kontrolera błędy wciąż istnieją. Zmuszeni jesteśmy pousuwać wszystkie obiekty poczym postawić kontroler.

Żeby zapobiec takiej sytuacji warto ważne obiekty blokować, w tym celu trzymamy ctrl i prawym przyciskiem myszy klikamy na obiekt po czym wybieramy pozycje „Locked”. Aby odblokować obiekt robimy to powtórnie.

Object Creation Code:

Pewnie wielu z was spotkało się z problemem stawiania tych samych obiektów pod różnymi kątami, lub innymi startowymi wartościami. Przykład?

Powiedzmy że mamy `obj_drzewo` i żeby na planszy wszystkie postawione drzewa nie wyglądały tak samo to zwykle w „create” piszemy:

```
image_index = random(ilość klatek sprita);  
image_angle = random(360);
```

Jest to w pewnym sensie jakieś rozwiązanie ale tylko dla drzew i np. krzaków. Jeżeli jednak musimy postawić na levelu jakiś element scenarii np. krzesło, stół. To powyższy sposób jest bardzo niekorzystny, choćby dlatego że w domu meble nie mogą być stawiane losowo, bo jak by to wyglądało, z pewnością kiepsko. Zapewne wielu z was robi duplikaty takich obiektów np.

`obj_stol0`, `obj_stol90`, `obj_stol45`, `obj_stol225`, nazwa symbolizuje kąt pod jakim jest ustawiony stół. Robienie takich duplikatów to błąd.

Można zrobić własny level editor ale po co się nie potrzebnie męczyć.

Zrób nowy obiekt: `obj_sofa` w „create” ustaw mu początkowe wartości np.:

```
image_angle = 0;  
image_index = random(ilość klatek);  
HP=100;
```

I postaw go na levelu w room editorze, teraz trzymając ctrl kliknij na niego prawym przyciskiem myszki. Wyświetli się menu kontekstowe, interesuje nas ostatnia pozycja czyli „Creation code”.

Można tu ustalać inne początkowe wartości niż te standardowe z eventa „create” dla każdego obiektu osobno. No więc ustalmy coś innego np.:

```
image_angle = 0;  
image_index = 5;  
HP=200;
```

odpalamy grę i co? ku naszemu zdziwieniu wartości wcale się nie zmieniły są z powrotem takie jakie ustaliliśmy w eventie „Create”. Pytanie dlaczego?

Ano dlatego że „Creation code” wykonuje się przed eventem „Create” to znaczy że na nic ustalanie wartości które potem event „Create” z powrotem zmienia na swoje.

Żeby wszystko zadziało musimy dopisać i usunąć coś z eventu „Create”:

dopisujemy:

```
if !variable_local_exists("nazwa zmiennej");
```

a usuwamy:

```
image_angle = 0;  
image_index = random(ilość klatek);
```

Teraz event „Create” powinien wyglądać tak:

```
if !variable_local_exists("HP") HP=100;
```

Funkcja ta sprawdza czy zmienna o nazwie ‘HP’ już istnieje jeżeli nie istnieje to tworzy ją ze standardową wartością 100. Ale dlaczego usunęliśmy `niebieskie` zmienne?

Dlatego że `niebieskie` zmienne są zmiennymi wbudowanymi w Game Makera i chcąc czy nie chcąc i tak będą one tworzone z wartościami 0. No więc po co tworzyć je jeszcze raz.

Co więcej jeżeli byśmy zostawili je w spokoju to “creation code” znowu by nic nie wskórał.

Wróćmy do room editora i zmienimy “Creation code” jakiegoś obiektu **obj_sofa**.

Np. na taki:

```
image_angle = 0;  
image_index = 7;  
HP=222;
```

Jak widać dopiero tutaj pojawiają się **niebieskie** zmienne, tym razem ustalenie ich zmieni je. Odpalamy grę i działa.

Room Creation Code:

Istnieje jeszcze inny Creation Code w zakładce settings, kod który tu zostanie napisany wykona się po stworzeniu wszystkich obiektów. Jest to użyteczne np. do zwiększenia poziomu trudności danego levelu. Powiedzmy że **obj_potwor** ma na poziomie trudności „normal” standardową ilość HP taką jaką ustaliliśmy w evencie „Create” a chcemy żeby ten potwór na poziomie trudności „easy” miał mniej HP o 20 ale na poziomie trudności „hard” miał więcej. Zamiast tworzyć kolejnych 3 duplikatów tego obiektu to można w „room creation code” napisać:

```
obj_potwor.HP+=20;  
lub  
obj_potwor.HP-=20;
```

W zależności od poziomu trudność.

Miłego kodzenia życzy †-Huderlord-†