

Samouczek do programu Game Maker

Twoja Pierwsza Gra

Napisane przez : Mark Overmars (Copyright © 2007-2009 YoYo Games Ltd), 23.12.2009

Tłumaczenie : Grzegorz Madejski

Korzysta z: Game Maker 8.0, Edycja Lite lub Pro, Tryb: Simple Mode

Poziom : początkujący

Pomimo tego, że *Game Maker* jest łatwy w obsłudze, to początkowo może sprawiać trochę trudności. Ten samouczek jest przeznaczony dla tych, którzy mają początkowe trudności z *Game Makerem*. Przeprowadzi Cię on krok po kroku poprzez proces tworzenia Twojej pierwszej gry, która z reguły jest najtrudniejsza do zrobienia. By ją stworzyć, musisz zrobić pewną liczbę aspektów programu *Game Maker*. Przeczytaj ten samouczek uważnie i spróbuj zrozumieć wszystkie kroki. Gdy skończysz pierwszą grę, druga będzie o wiele łatwiejsza.

Pomysł na grę

Ważne jest, by na początku zrobić krótki opis gry, którą będziemy tworzyć. Ponieważ będzie to nasza pierwsza gra, zrobimy coś prostego, co powinno zainteresować gracza chociaż na krótki okres czasu. Zrobimy grę akcji pod tytułem *Złap klauna* (zawsze spróbuj wymyślić ciekawy tytuł dla swojej gry). Oto opis gry:

Złap Klauna

Złap Klauna jest niewielką grą akcji. W tej grze klaun porusza się po polu gry. Zadaniem gracza jest złapanie klauna poprzez kliknięcie na niego. Wraz z postępem gry, klaun staje się coraz szybszy, a złapanie go coraz trudniejsze. Za każde złapanie dostaje się punkty, a celem jest zgromadzenie jak największej liczby punktów. Oczekiwany czas gry to kilka minut.

Oczywiście gra taka nie będzie bardzo atrakcyjna, ale trzeba zacząć od rzeczy prostych. Potem można wzbogacić grę o szczegóły, które uczynią ją bardziej interesującą.

Dokument projektu gry

Mając pomysł na grę, należy zastanowić się nad dokładniejszym jej działaniem. Drugim krokiem w tworzeniu gry, jest więc napisanie dokumentu z projektem gry, który może być naprawdę prosty. Poniżej przedstawiamy nasz pomysł na taki dokument (z pominięciem opisu, który daliśmy u góry).

Złap Klauna – dokument projektu gry

Obiekty w grze

W grze będą tylko dwa obiekty : klaun i ściana. Obiekt Ściana to prostokątny obrazek. Plansza gry będzie otoczona takimi blokami. Ściana nic nie robi, powstrzymuje jedynie klauna przed opuszczeniem planszy. Obiekt Klaun ma obrazek głowy klauna, porusza się ze stałą prędkością, a kiedy dotknie ściany – odbija się od niej. Gdy gracz naciśnie na klauna, punkty wzrastają o 10. Klaun przesuwany jest w losowe miejsce a jego szybkość wzrasta.

Dźwięki

Będziemy używać dwóch dźwięków w grze. Dźwięk odbijania zastosujemy podczas uderzania klauna o ścianę, a dźwięk kliknięcia, gdy uda nam się nacisnąć kursorem na klauna.

Kontrola gry

Gracz kontroluje grę tylko poprzez myszkę. Kliknięcie lewym przyciskiem myszy na klauna powoduje jego złapanie.

Przebieg gry

Na początku gry, wynik ustawiany jest na 0. Pokój z klaunem jest załadowany, i gra od razu się rozpoczyna. Po naciśnięciu klawisza <Esc>, gra się kończy.

Poziomy



Jest tylko jeden poziom gry. Trudność gry wzrasta z każdym kolejnym złapaniem klauna.

Tyle powinno na razie starczyć. Możemy zacząć tworzenie gry. Włącz Game Makera. Ten samouczek opiera się na Game Makerze 8.0, jeśli używasz innej wersji obrazki mogą wyglądać inaczej. Zakładamy też, że program działa w trybie: Simple Mode (Prosty tryb bez zbędnych narzędzi). Możesz przełączyć na tryb zaawansowany: **Advanced Mode** w menu **File** (plik), który dostarcza wielu różnych opcji, ale na nie jest potrzebny w tworzeniu naszej prostej gry.

Materiały do tego samouczka są załączone w folderze *Resources* (Materiały). Są to obrazki i dźwięki, których będziemy używać.

Dodawanie obrazków i dźwięków

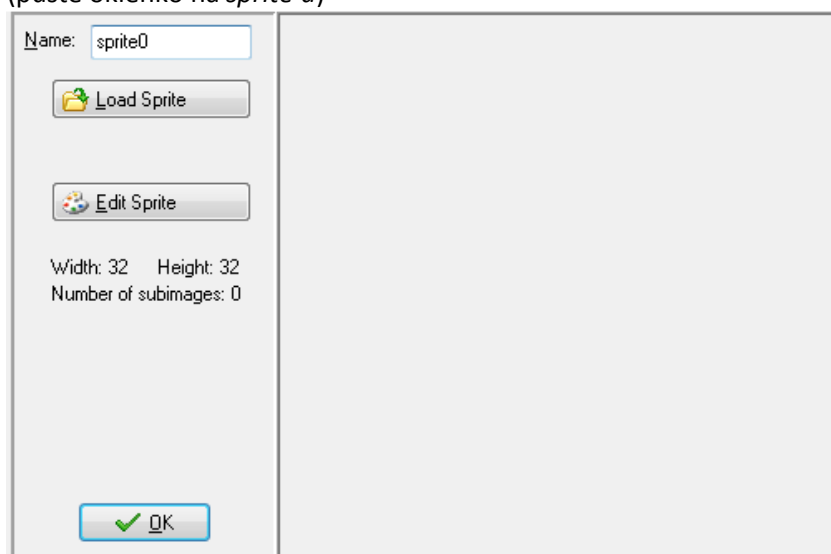
Tak jak opisaliśmy w naszym projekcie, będziemy potrzebować dwóch obrazków dla dwóch obiektów w grze. Takie obrazki nazywamy *sprites* w *GameMakerze*. Dla ułatwienia będziemy spolszczać wymowę (jeden sprite = jeden sprajt, dwa sprite'y = dwa sprajty). Można o nich powiedzieć wiele dodatkowych rzeczy, ale na razie możemy o nich myśleć po prostu jako o małych obrazkach, z których często montuje się coś większego umieszczając je na scenie. Do tworzenia obrazków można wykorzystać dowolny edytor grafiki, choćby dostępny w każdym Windowsie program *Paint*, ale *GameMaker* też ma wbudowany program do tego celu. Tworzenie *sprite'ów* jest sztuką samą w sobie i wymaga wiele czasu, by ją opanować. Na szczęście w sieci można znaleźć wiele obrazków udostępnionych za darmo i jeśli dobrze poszukasz to na pewno znajdziesz ich sporo. *GameMaker* również zawiera znaczną ich ilość, znajdziesz ich także więcej dostępnych na stronie *YoYo Games* (www.yoyogames.com). Jednak nam wystarczą dwa załączone *sprite'y* z folderu *Resources*.

Klaun:  (clown.jpg) Ściana:  (wall.jpg)

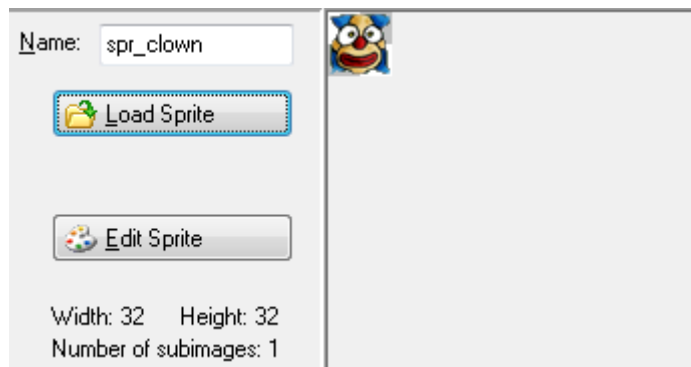
By dodać te obrazki musimy postępować wg poniższych kroków:

Tworzenie *sprite'a* z klauna:

1. Z menu **Resources** (Zasoby) wybierz **Create Sprite** (Stwórz *Sprite'a*). Pojawi się okno takie jak poniżej (puste okienko na *sprite'a*)



2. Naciśnij na pole **Name** (Nazwa), które narazie zawiera napis `sprite0`. Jest to domyślna nazwa *sprite'a*. Zmień nazwę na `spr_clown` (spr to skrót to *sprite'a*, clown = klaun).
3. Naciśnij guzik **Load Sprite** (Załaduj *Sprite*). Pokaże nam się okno z prośbą o znalezienie pliku.
4. Znajdź folder *Resources* i wybierz obrazek `clown.png`. Teraz okno właściwości *Sprite* powinno wyglądać następująco:



5. Naciśnij guzik **OK** by zamknąć formatkę.

Tworzenie sprite'a ze ściany:

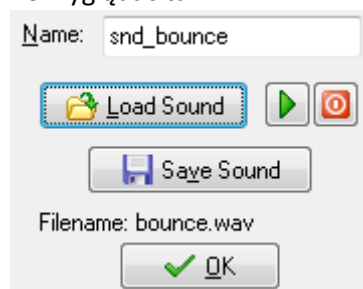
1. Postępuj podobnie, jednak nadaj ścianie nazwę: `spr_wall`.
2. Po naciśnięciu **Load Sprite** musisz oczywiście wybrać plik `wall.png`.
3. Naciśnij **OK** by zamknąć formularz.

Jak pewnie zauważyłeś, klaun i ściana pojawiły się na liście zasobów po lewej stronie okna programu *GameMaker*. Tu znajdziesz wszystkie sprite'y, dźwięki (sounds), obiekty, pokoje itd. , które stworzyłeś podczas tej gry. To wszystko nazywane jest zasobami (resources) gry. Możesz wybrać dany zasób klikając na jego nazwę, następnie za pomocą menu **Edit** (Edytuj) zmienić go, powielić lub skasować. Kliknięcie prawym przyciskiem myszy na nazwę pokaże to samo menu. Podgląd wszystkich zasobów jest ważny, zwłaszcza podczas tworzenia skomplikowanych gier.

Teraz, po stworzeniu sprite'ów, zrobimy dwa efekty dźwiękowe. Jeden odpowiada za uderzenie klauna o ścianę, a drugi za kliknięcie myszą na klauna. Użyjemy do tego dwóch plików dźwiękowych z rozszerzeniem `.wav`. Takie pliki są bardzo dobre do tworzenia efektów dźwiękowych. Różne inne dźwięki można znaleźć w programie *GameMaker*, a jeszcze więcej w sieci.

Tworzenie zasobów dźwiękowych:

1. Z menu **Resources** wybierz **Create Sound** (Stwórz dźwięk). Wskoczy okienko właściwości dźwięku. Naciśnij na pole **Name** i zmień nazwę na `snd_bounce` (snd = sound czyli dźwięk, bounce = odbicie).
2. Naciśnij na guzik **Load Sound** (załaduj dźwięk), przejdź do folderu `Resources` i wybierz plik `bounce.wav`. Okienko powinno wyglądać tak:



3. Naciśnij **OK** by zamknąć formatkę.
4. W podobny sposób jak powyżej stwórz dźwięk o nazwie `snd_click` z pliku `click.wav` z folderu **Resources**.

Zauważ, że w okienku właściwości dźwięku znajdują się dwa przyciski do jego odtwarzania. Zielona strzałka odtwarza dźwięk (bez końca, w pętli), a czerwony guzik zatrzymuje odtwarzania. Zauważ również, że dźwięki pojawiły się na liście wszystkich zasobów z lewej strony.

Obiekty i akcje

Stworzenie sprite'ów i dźwięków jeszcze nie oznacza, że cokolwiek będzie się działo. Sprite'y to tylko obrazki dla pewnych obiektów w grze, których jeszcze nie zdefiniowaliśmy. Podobnie, dźwięki nie będą odtwarzane jeśli im nie każemy. Musimy więc stworzyć dwa obiekty.

Zanim to zrobimy, musisz zrozumieć sposób w jaki program *GameMaker* działa. Jak już wspomnieliśmy, w grze masz pewną liczbę różnych obiektów (game objects). W czasie tworzenia gry będziemy korzystać z obiektu umieszczając go w grze jednokrotnie (klaun) albo będziemy korzystać z wielu kopii tego samego obiektu (ściana). Stworzenie obiektu nie oznacza więc, że możemy skorzystać z niego tylko raz. Możemy zrobić jego wiele kopii, które nazywamy *instancjami*. Jeśli obiekt ściana siedzi w zasobach, ale go nie wykorzystaliśmy to można powiedzieć, że nie stworzyliśmy żadnej instancji tego obiektu (obiekt niby istnieje, ale tylko w sferze modelu). Jeśli umieścimy dwa klocki ściany na planszy, to mówimy, że stworzyliśmy dwie instancje obiektu ściana (instancje to jakby żywe, stworzone jednostki na podstawie modelu obiektu z zasobów).

Ponieważ rozróżnienie obiektu (w zasobach) i instancji obiektu jest ważne to zastanówmy się nad przykładem z życia wziętym. Gdy mówisz „Fiat 126 p”, przed oczami stoi Ci maluch. Niektórzy eksperci wiedzą co taki samochód ma w środku, znają każdy jego szczegół techniczny, mają go te szczegóły zapisane w planach itp. Jest to jednak tylko idea Fiata 126 p lub projekt. Instancją Fiata 126 p byłby konkretny samochód który wyjdzie z fabryki, którego można dotknąć i nim pojeździć. Takich instancji fabryka potrafi wyprodukować wiele, mając do dyspozycji tylko jeden obiekt (model).

Instancje obiektów gry nie będą nic robiły, dopóki im nie każesz. Robisz to poprzez wskazywanie jak mają zachowywać się instancje obiektów po zajściu pewnych zdarzeń (*events*). Zdarzenia mogą być różne. Pierwszą zdarzeniem w życiu każdej instancji obiektu jest jej utworzenie czyli **Create Event**. Te zdarzenie może wymagać pewnej akcji np. po stworzeniu klauna (zdarzenie) konieczne jest rozpoczęcie jego ruchu w określonym kierunku (akcja). Kolejnym ważnym zdarzeniem jest zdarzenie kolizji **Colission Event**, które ma miejsce wtedy gdy stykają się dwa obiekty. Przykładowo, gdy klaun dotknie ściany (zdarzenie) musi nastąpić zmiana jego kierunku ruchu (akcja). Jeszcze innym przykładem do zdarzenia związane z użyciem myszki **Mouse Event**. Gdy gracz klika myszką na klauna (zdarzenie) ilość punktów jest zwiększana (akcja).

By wskazać, co ma się dziać po zaistnieniu zdarzenia, musisz określić akcje (*actions*). Mamy ich wiele do wyboru np. akcja wprawiająca instancję w ruch w konkretnym kierunku, akcja zmiany punktów, akcja odtworzenia dźwięku. Zatem zdefiniowanie obiektu składa się z kilku aspektów:

- Dajemy obiektowi sprite'a jako obrazek,
- Ustawiamy jego własności (np. nazwę),
- Ustalamy na jakie *zdarzenia* ma obiekt reagować, i jaką wtedy *akcję* podejmuje.

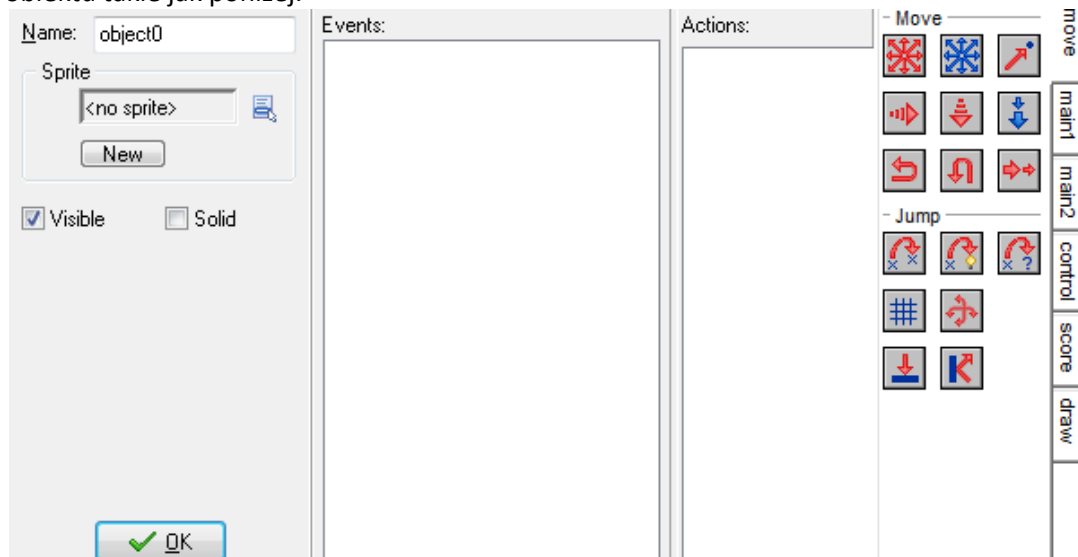
particular direction, there is an action to change the score, and there is an action to play sounds. So defining a game object consists of a few aspects: we give the object a sprite as an image, we can set some properties, and we can indicate to which events instances of the object must react and what actions they must perform.

I jeszcze raz podkreślamy, że obiekt i instancja tego obiektu to dwie różne rzeczy. Obiekt to coś abstrakcyjnego (model, projekt siedzący w zasobach) a instancja to konkretny egzemplarz tego modelu będący w użyciu. Zauważ, że zdarzenia i akcje projektujemy wraz obiektem i każda instancja obiektu będzie się tak zachowywała jak zaprojektowaliśmy. Wracając do naszego fiata. Powinniśmy mieć w planie, że gdy kluczyk zostanie przekręcony w stacyjce (zdarzenie) zostanie odpalony silnik (akcja). Każda instancja fiata wyprodukowana w fabryce będzie działała tak, że kluczyk odpali silnik.

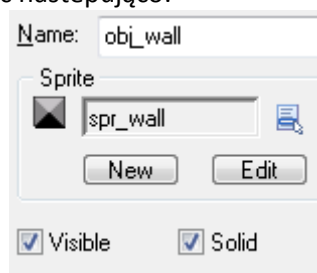
Co zatem z naszą grą? Potrzebujemy dwóch obiektów. Stwórzmy najpierw obiekt ściana, który jest o tyle prosty, że nie musi reagować na żadne zdarzenia i wykonywać żadnych akcji.

Create the wall object:

1. Z menu **Resources** wybierz **Create object** (Stwórz obiekt). Pojawi się okienko właściwości obiektu takie jak poniżej.



2. W polu **Name** wpisz nazwę obiektu: `obj_wall`.
3. Kliknij na ikonkę na prawo od pola **Sprite** (niebieski dokument z strzałką) i z listy dostępnych sprite'ów wybierz `spr_wall`.
4. Instancje obiektu ściana muszą być widoczne (Visible) oraz stałe (Solid), dzięki czemu inne obiekty (np. Klaun) nie przejdą przez nie jak przez powietrze. By obiekt był stały, zaznacz pole **Solid**. Okienko powinno wyglądać następująco:



5. Naciśnij **OK** by zamknąć formularz.

Z klaunem postąpimy na początku tak samo.

1. Stwórz obiekt **Create Object**.
2. Nazwij go `obj_clown`.
3. Wybierz dla niego sprite'a `spr_clown`, ale nie zaznaczaj tym razem opcji **Solid**.

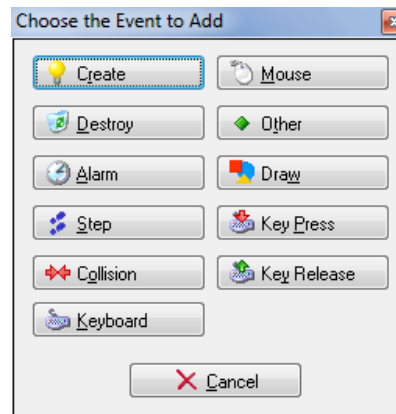
Te trzy kroki to dopiero początek. Klaun jest obiektem interaktywnym, więc musimy określić jego zachowanie. Do tego posłuży nam cała reszta okienka właściwości obiektu. Po środku widać pustą listę z nagłówkiem *Events* (zdarzenia), na które ma reagować klaun. Pod nią są trzy przyciski: Add Event (Dodaj zdarzenie), Delete (Usuń zdarzenie), Change (Zmień zdarzenie). Lista możliwych zdarzeń jest ogromna, ale w naszej grze wykorzystamy tylko kilka.


Po prawej stronie widzimy pustą listę akcji *Actions*, które ma wykonać obiekt w odpowiedzi na zajście określonego zdarzenia. Dla ułatwienia wszystkie możliwe akcje są pogrupowane w zakładki (prawy brzeg okna). Mamy na przykład zdarzenia związane z ruchem (Move) lub kontrolą (Control). Każda zakładka zawiera pewną ilość ikon, które reprezentują akcje (jest ich razem ponad 100!). By dowiedzieć się czegoś więcej o akcji można najechać na nią kursorem myszki (pojawi się podpowiedź). Aby dodać akcję do listy, wystarczy chwycić ją myszką i przeciągnąć do białego pola *Actions* (dla danego zdarzenia).

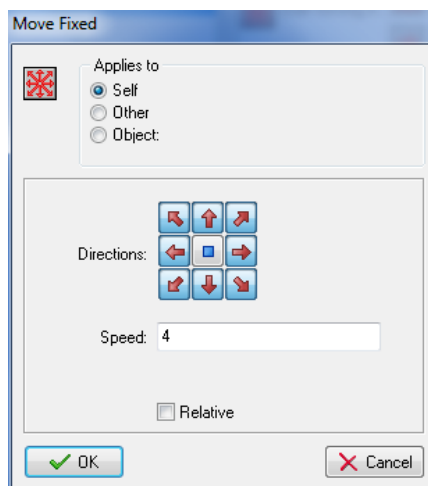
Najpierw zdefiniujemy co powinno się dziać, gdy zostanie stworzony klaun. W tym przypadku, chcemy aby klaun zaczął poruszać się w dowolnym kierunku.

Jak poruszyć klauna:

4. Naciśnij na guzik **Add Event** (Dodaj zdarzenie). Pojawi się okno wyboru zdarzenia, takie jak niżej:

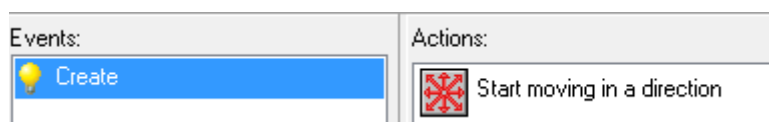


5. Naciśnij na guzik **Create** (Tworzenie). Zdarzenie dotyczące tworzenia instancji obiektu zostało dodane.
6. Mamy zdarzenie, teraz musimy dodać akcję. Wybierz akcję **Move Fixed** (Poruszaj się w wyznaczonym kierunku) i wrzuć ją do listy akcji. By to zrobić przeciągnij ikonkę  do pola *Actions*, chwytając ją myszką. Pojawi się formularz proszący o dodatkowe informacje związane z akcją.
7. W okienku akcji **Move Fixed** możesz wybrać kierunek, w którym zaczniesz poruszać się klaun. Zaznacz wszystkie osiem strzałek (bez środkowego przycisku, który oznacza brak ruchu). Zaznaczone kierunki podświetlają się na czerwono. Gdy zaznaczysz więcej niż jeden, wybierany jest jeden z nich losowo. Ustaw *Speed* (Szybkość) na 4. Okienko powinno wyglądać następująco:



Naciśnij **OK**.


Ustawiłeś w ten sposób zachowanie obiektu Klaun, gdy jest on tworzony. Dodałeś zdarzenie, akcje w odpowiedzi na to zdarzenie oraz ustawiłeś pewne właściwości samej akcji. W konsekwencji otrzymujemy następujący wynik:

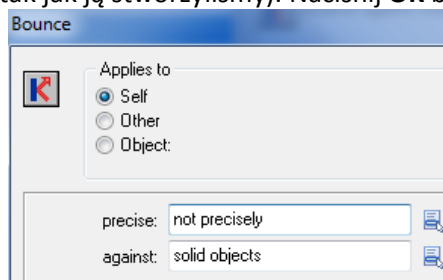



Zdarzenie: Stwórz, Akcje: Zaczynj poruszać się w określonym kierunku

Następnym zdarzeniem będzie zderzenie klauna ze ścianą. Na to zdarzenie odpowiemy dwoma akcjami: zmianą kierunku i włączeniem efektu dźwiękowego.

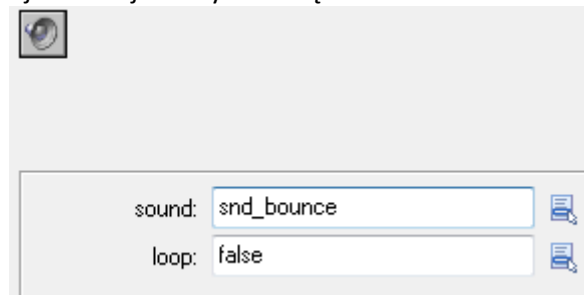
Obsługa zdarzenia zderzenia ze ścianą:

1. Znow naciśnij **Add Event**. W oknie wyboru zdarzeń wybierz kolizję: **Collision** i z wysuwanego menu wybierz obiekt ścianę: `obj_wall`. Kolizja została dodana do listy.
2. Dodaj akcję **Bounce** (Odbicie) poprzez przeciągnięcie jej ikony  do pola z akcjami. Pojawi się okienko takie jak poniżej, gdzie możemy określić precyzyjnie kąt odbicia lub ustalić o jakich obiektów można się odbijać. Obecny wybór jest dobry (klaun ma się odbijać od stałych *Solid* obiektów czyli ściany, tak jak ją stworzyliśmy). Naciśnij **OK** by zamknąć okienko.

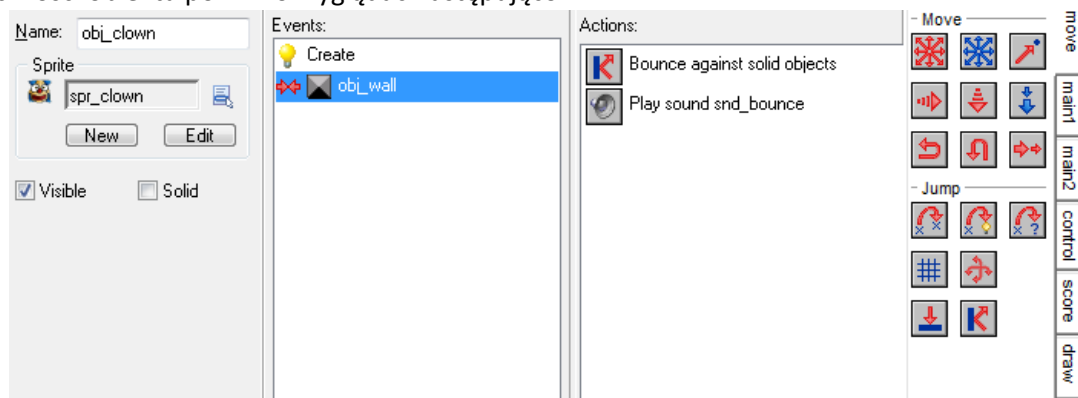


3. Dodamy teraz dźwięk odbicia od ściany. Naciśnij na zakładkę **main1** (po prawej stronie) i z dostępnych akcji wybierz **Play Sound** (Odtwarzaj dźwięk). Przeciągnij jej ikonę  do pola akcji pod akcją odbijania. W okienku akcji, na prawo od pola **sound** (dźwięk) znajduje się

ikona. Kliknij na nią i z listy wybierz dźwięk `snd_bounce`. Chcemy by dźwięk był odtworzony tylko raz więc zostawiamy opcję **Loop** (Pętla) na **false** (Nieprawda/Brak). Całość powinna wyglądać tak jak poniżej. Naciśnij **OK** by zamknąć okno.



To kończy naszą pracę nad zderzeniem klauna ze ścianą i odpowiadającym mu akcją. Okno właściwości obiektu powinno wyglądać następująco:




Gdy kolizja nastąpi, wykonywane są dwie akcje w podanej kolejności. Kolejność ma znaczenie, więc jeśli masz w innej kolejności możesz to naprawić, po prostu chwytając za daną akcję i przeciągając ją kursorem w inne miejsce w kolejce. Można akcje również kasować przyciskiem **Delete** w okienku lub na klawiaturze. Akcje można też zmieniać naciskając przycisk **Edit Values** (Edytuj wartości) lub klikając na nią podwójnie.

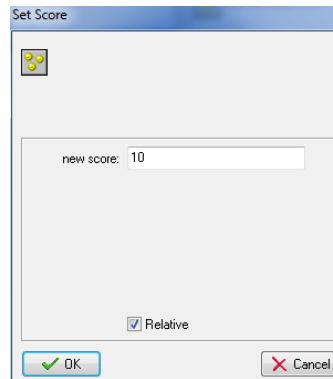
Wreszcie przechodzimy do tego, co się dzieje podczas kliknięcia lewym przyciskiem myszy na klauna. Takie zdarzenie będzie miało reakcję w postaci aż 4 akcji:

- Dodanie 10 punktów do wyniku (jest to łatwe, bo program zapamiętuje wynik),
- Odtworzenie przycisku kliknięcia,
- Przesunięcie klauna w losową pozycję,
- Wyznaczenie nowego kierunku i szybszego poruszania się.

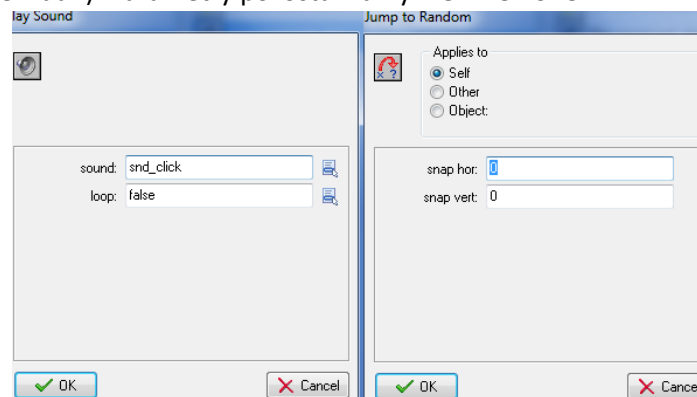
Ostatnie dwie akcje służą do stopniowego zwiększania trudności gry.

Obsługa naciśnięcia myszki:

1. Najpierw dodajemy nowe zdarzenie: **Add Event**. W okienku wyboru zdarzenia wybieramy myszkę: **Mouse**, a z wyskakującego menu wybieramy: **Left Pressed** (Lewy przycisk). Zdarzenie to można opisać następującymi słowami: gracz nacisnął lewy przycisk myszy będąc kursorem na klaunie. Dlaczego na klaunie? Bo wszystko to robimy wewnątrz obiektu Klaun.
2. Teraz pora na dodanie akcji. Wybieramy po prawej stronie zakładkę **score** (wynik), a z niej akcję  **Set Score** (Ustaw wynik). Jako **new score** (nowy wynik) wybierz wartość 10. Zaznacz też pole **Relative** (zależne). Jego zaznaczenie powoduje, że 10 punktów będzie się dodawało do obecnego wyniku, a nie zamieniało obecny wynik na 10.

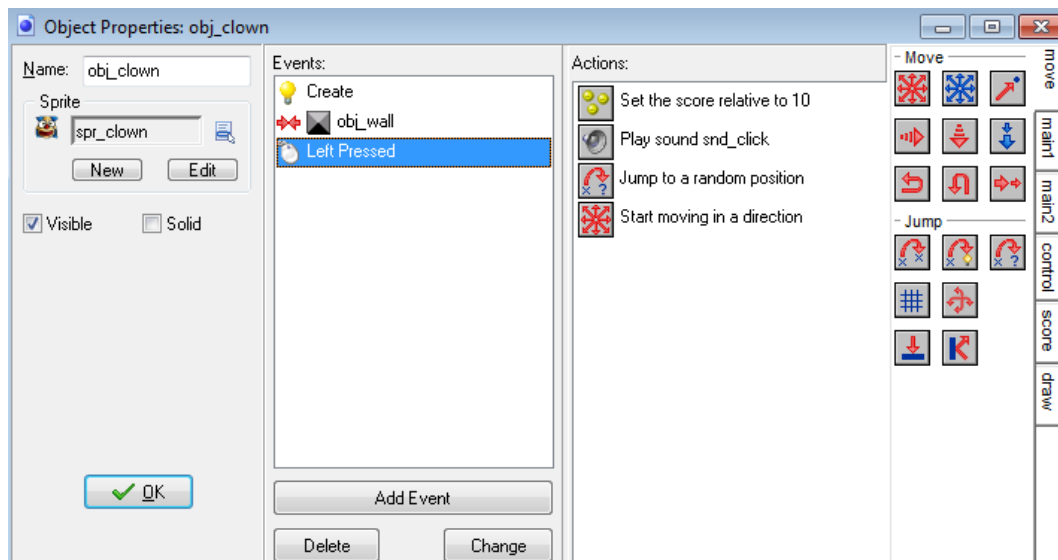


3. Druga akcja dla kliknięcia: z zakładki **main1** wybierz akcję **Play Sound**. Jako dźwięk **Sound** wybierz **snd_click**. Zostaw zapętlenie **Loop** na wartość **false** (Nieprawda/brak).
4. Trzecia akcja: z zakładki **move** wybierz akcję **Jump to Random** (Losowy skok). Ta akcja umieszcza instancję w miejscu wolnym od kolizji (ponieważ jesteśmy w środku klauna, to skakać będzie klaun). Parametry pozostawiamy niezmiennione.



5. Ostatnia, czwarta akcja z zakładki **move** to **Move Fixed**. Zaznacz wszystkie osiem strzałek, tak jak poprzednio. Ustaw szybkość **Speed** na 0.5 i włącz opcję **Relative**. Opcja **Relative** spowoduje, że prędkość nie zamieni się na 0.5 tylko zwiększy o 0.5 (podobnie było z punktami).

Skończyliśmy tworzenie obiektu Klaun. Wszystkie zdarzenia i akcje zostały dodane. Możesz nacisnąć **OK** by zamknąć okienko właściwości obiektu Klaun.

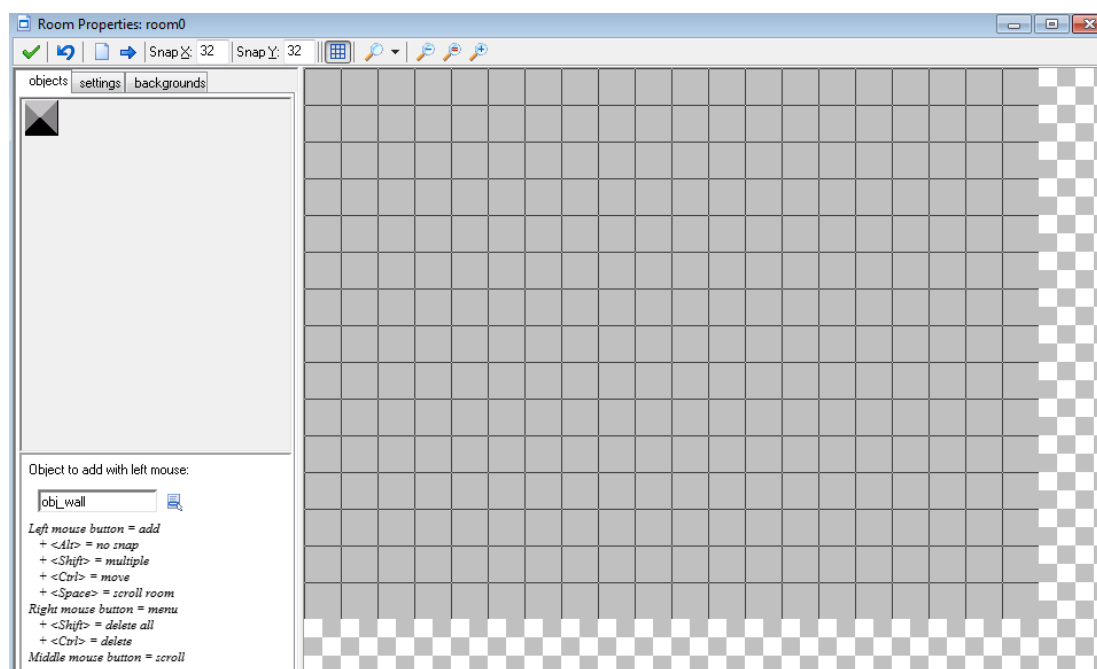


Tworzenie pokoju

Stworzyliśmy już obiekty w grze, ale pozostała nam jeszcze jedna ważna rzecz do zrobienia. Musimy stworzyć pokój, w którym będzie rozgrywała się gra. Dla większości gier projektowanie odpowiednich pokoi (które częściej nazywa się poziomami) jest bardzo czasochłonne, bo trzeba dobrze zbalansować grę – by nie była zbyt szybka, ani zbyt powolna, zbyt łatwa ani zbyt trudna. Na szczęście nasza gra jest bardzo prosta, pokój ogrodzony ścianą z klaunem po nim się poruszającym jest prosty do zrobienia.


Tworzenie pokoju:

1. Z menu **Resources** wybierz **Create Room** (Twórz pokój). Otworzy się okienko właściwości pokoju. Jeśli odpowiednio rozciągniesz te okienko oraz lewy pasek to powinno to wyglądać mniej więcej tak:



- Po lewej stronie znajduje się okienko z trzema zakładkami: objects (obiekty), settings (ustawienia) i backgrounds (tła). Wybierz środkową zakładkę **settings** i w polu **Name** wpisz `rm_main` (RM = skrót od room, czyli pokój). W polu **Caption for the room** wpisz 'Złap Klauna'.
- Wybierz teraz zakładkę **objects**. Powiększ okno z pokojem tak, aby był widoczny cały pokój (wszystkie szare kratki). Szarobiała szachownica oznacza zwykle tło przezroczyste, czyli tam gdzie się znajduje – nic nie ma, pokój jest tylko tam gdzie są szare kratki. Upewnij się, że wartości **Snap X** i **Snap Y** wynoszą 32 (jeśli nie to zmień). Liczby te ustawiają czarne linie siatki w pokoju, które pomagają przy jego projektowaniu. Ponieważ zarówno ściana jak i klaun mają rozmiary 32x32, jest to najlepsze rozwiązanie. Zaznacz też opcję **Delete Underlying** w lewym dolnym rogu. Jej ustawienie powoduje, że w jednej kratce nie można wstawić dwóch obiektów jeden na drugim.
- Po lewej powinieneś widzieć aktualnie zaznaczony obiekt: albo klauna, albo ścianę - w zależności od tego co kliknąłeś ostatnio. Jeśli trzymasz się cały czas bez przerwy tego samouczka, to najprawdopodobniej jest tam klaun. Poniżej powinno być widoczne pole wyboru obiektu (jeśli nie to rozciągnij to okienko). Wybierz klauna:

Object to add with left mouse:



Left mouse button = add
+ <Alt> = no snap

Kliknij na kratkę gdzieś w środku pokoju by go tam umieścić.

- Kliknij na ikonę klauna lub na okienko z nazwą `obj_clown` i z menu wybierz ścianę `obj_wall`:

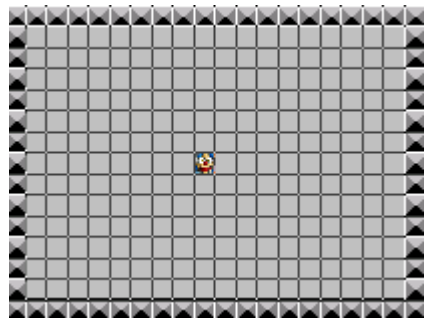
Object to add with left mouse:




Left mouse button = add
+ <Alt> = no snap


Teraz kliknij na wszystkie kratki graniczne pokoju aby utworzyć ścianę wokół niego. Dla ułatwienia można przytrzymać klawisz Shift na klawiaturze i przytrzymując klawisz myszki przejechać po wybranych polach.

Inne przydatne komendy do obsługi pokoju: prawy przycisk myszy i wybranie **Delete** to usunięcie obiektu z pokoju (prawy przycisk myszy z naciśniętym CTRL to też usunięcie). Chwyatanie i przesuwanie obiektów działa podobnie: przytrzymaj CTRL i lewym przyciskiem myszy przesuń obiekt w wybrane miejsce.



Rolką na myszce możesz pomniejszać lub powiększać pokój by mieć na niego lepszy ogląd.

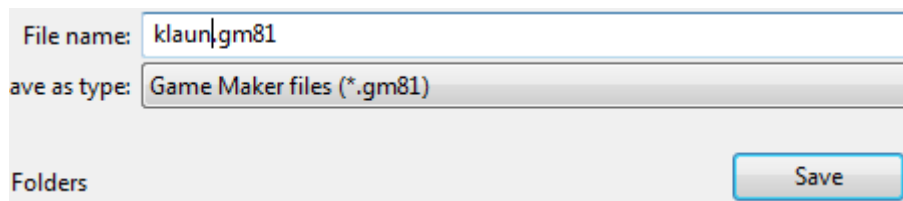
Możesz też w razie błędu cofnąć wykonanie ostatniej akcji strzałką  w górnym pasku.

- Naciśnij zielony  w lewym górnym rogu, by zakończyć projektowanie pokoju.

Zapisanie i testowanie

Może nie zdajesz sobie sprawy, ale właściwie Twoja gra jest już skończona. Dodałeś sprite'y i dźwięki, stworzyłeś na ich podstawie obiekty. Wreszcie, stworzyłeś pierwszy (i jedyny) pokój, w którym będzie rozgrywała się akcja gry. Teraz czas na zapisanie gry.

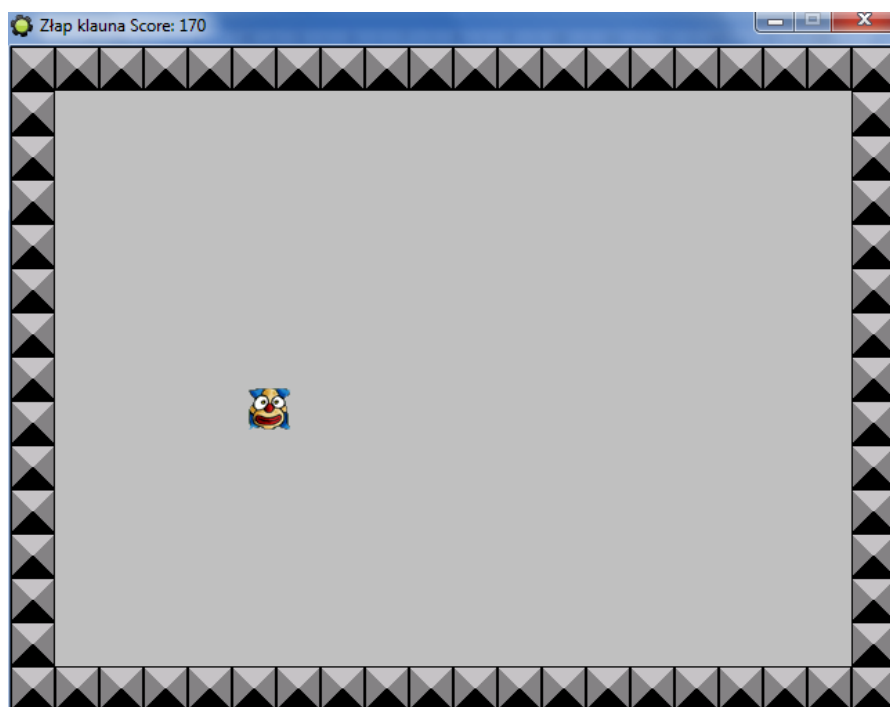
Jak w większości programów windowsowych, wybierz menu **File** (Plik) i opcję **Save** (Zapisz). Wybierz lokalację i wpisz nazwę dla pliku:



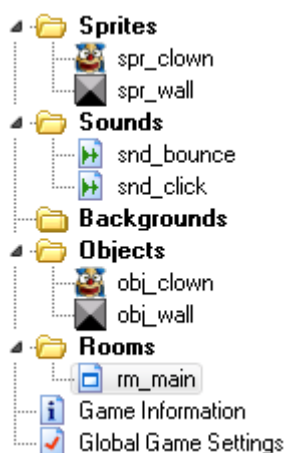
Zauważ, że rozszerzenie pliku dla programu Game Maker 8.1 to **.gm81**. W przeszłości były stosowane też inne rozszerzenia np. **.gmk**, na które być może natkniesz się w sieci. Pliki z takimi rozszerzeniami to jednak nie jest ostateczna wersja gry, gdyż można je odpalić tylko poprzez program GameMaker (a przecież nie każdy go ma). Później dowiemy się, jak stworzyć pliki wykonawcze z rozszerzeniem **.exe**, które każdy może odpalić.

Zajmijmy się jednak przetestowaniem gry, gdyż jest to bardzo ważna rzecz. Czasem w grze mogą być zakamuflowane błędy, których nawet Ty nie wyłapiesz. Weź więc pod uwagę pomoc innych osób. Daj do testowania grę kolegom, koleżankom czy rodzinie. ;)

By uruchomić grę z poziomu GameMakera wystarczy wybrać menu **Run** (Uruchom) i opcję **Run normally** (Uruchom normalnie), lub zieloną strzałkę ▶ na górnym pasku z narzędziami. Zniknie okno do projektowania gry, zaś pojawi się sama gra (o ile nie zrobiłeś błędów) z pokojem i poruszającym się po nim klaunem.



Poklikaj na klauna i zobacz czy gra działa tak, jak powinna. Powinieneś słyszeć dźwięki, a szybkość klauna powinna rosnąć. Wynik powinien pojawiać się na górnym pasku obok tytułu jaki nadaliśmy pokojowi. By wyjść z gry zamknij okno lub wciśnij Esc. Okno doprojektowania gry pojawi się z powrotem.



Teraz jest czas by podrasować grę. Powinieneś się zastanowić nad takimi pytaniami: czy początkowa prędkość jest dobra? Czy przyspieszanie jest dobre? Czy wielkość pokoju jest dobra? Czy obrazki są dobre, a dźwięki nie zbyt rażące? Jeśli nie jesteś zadowolony z jakiegokolwiek rzeczy zmień ją i przetestuj grę jeszcze raz. Pamiętaj, że wszystkie rzeczy, które stworzyłeś są dostępne w oknie zasobów. Wystarczy kliknąć na nie dwukrotnie myszką.

Pamiętaj też, że dobrym pomysłem jest zachęcanie innych ludzi do testowania. Tobie na pewno będzie grało się inaczej, bo sam tę grę tworzyłeś.

Gdy już jesteś zadowolony z gry możesz stworzyć plik wykonawczy, dzięki któremu każdy będzie mógł zagrać w Twoją grę bez konieczności instalowania *GameMakera*. Procedura jest prosta: wejdź do menu **File** i wybierz opcję **Create Executable** (Twórz plik wykonawczy), następnie wybierz nazwę i miejsce dla pliku. Możesz teraz zamknąć GameMakera i włączyć grę po prostu poprzez podwójne kliknięcie na plik .exe. Ten plik wykonawczy możesz również przekazać znajomym, a nawet udostępnić w sieci na oficjalnej stronie GameMakera: <http://www.yoyogames.com> lub innych np. <http://gmclan.org/>. Oczywiście „Złap klauna” to niezbyt oryginalna gra, lepiej zrobić własną i dopiero nią się chwalić. By rozreklamować grę w sieci warto zrobić jej parę screenshotów i sympatyczny opis.

Prace wykończeniowe

Nasza pierwsza gra jest gotowa, ale potrzebuje jeszcze kilku zabiegów upiększających. Po pierwsze, dodamy muzykę leącą w tle. Jest to bardzo proste.

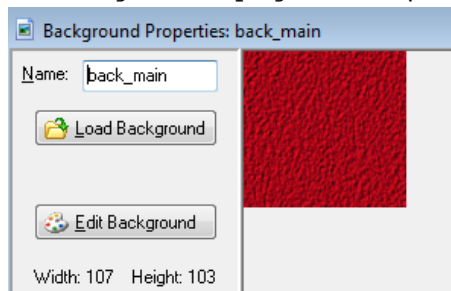
Stwórz muzykę w tle:

1. Z menu **Resources** wybierz **Create Sound**. W okienku właściwości dźwięku kliknij na pole **Name** nadaj nazwę dźwiękowi: `snd_music`.
2. Naciśnij na guzik **Load Sound**, wejdź do folderu `Resources` i wybierz plik muzyczny `music.mid`. Zauważ, że to plik midi. Ten rodzaj pliku muzycznego oferuje proste melodie, ale jest przy tym bardzo oszczędny jeśli chodzi o ilość zużywanej pamięci komputerowej. Bardziej nadaje się zatem niż popularne lecz zasobożerne pliki mp3.
3. Naciśnij **OK** by zamknąć okno.
4. Otwórz ponownie obiekt `obj_clown` poprzez naciśnięcie na niego podwójnie w spisie zasobów gry po lewej stronie.
5. Kliknij na zdarzenie **Create** (tzn. Zdarzenie utworzenia klauna) w spisie zdarzeń w środkowej części okna. Temu zdarzeniu została przyporządkowana już jedna akcja rozpoczynająca ruch. Dodamy drugą. Z zakładki **main1** dla akcji wybierz akcję **Play Sound**. Jako dźwięk **sound** ustaw `snd_music` i tym razem zaznacz opcję zapętlenia **Loop** na wartość `true` ponieważ chcemy by muzyka grała w kółko bez końca. Na koniec, naciśnij **OK** by wyjść z okna edycji klauna.

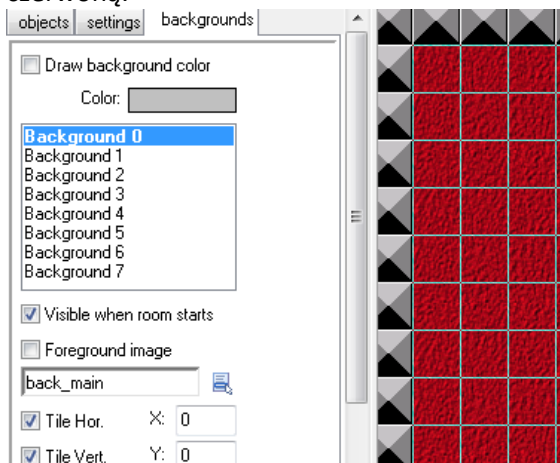
Po drugie, dodamy obrazek tła. Szare tło w pokoju jest trochę nudne. By dodać tło użyjemy nowego typu zasobu: zasób tła (Background resource).

Dodawanie obrazka tła:

1. Z menu resources **Resources** wybierz **Create Background** (Twórz tło). Pojawi się okienko właściwości tła. Kliknij na pole nazwy **Name** i zmień ją na `back_main`.
2. Kliknij na guzik **Load Background** (załaduj tło) i przejdź do folderu `Resources`, następnie wybierz obrazek o nazwie `background.png`. Okienko powinno wyglądać następująco:



3. Naciśnij **OK** by zamknąć formatkę.
4. Otwórz okienko właściwości pokoju `rm_main`, poprzez naciśnięcie na niego dwukrotnie w liście zasobów.
5. Kliknij na zakładkę **backgrounds** (tła) w lewej części okna. Odznacz opcję **Draw background color** (Pomaluj kolorem tła).
6. W polu znajdującym się pod napisem „Foreground image” wstaw nasze tło `back_main`. Pokój zmieni barwę na czerwoną.




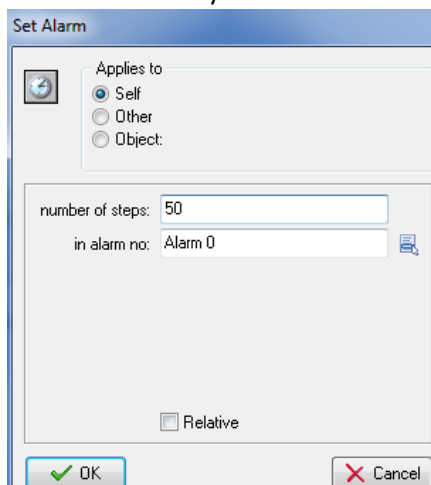
Zwróć też uwagę na dwie opcje: **Tile Hor.** i **Tile Vert.** Ich zaznaczenie powoduje, że kafelki tła są układane w cały pokój wzdłuż i w szer. Żeby to dobrze działało, należy dobrze dobrać obrazek tła by nie było widać przerw i szczelin między kafelkami.


Jeśli pograsz w grę wystarczająco długo, zauważysz że jest ona zbyt prosta, ponieważ wiesz dokładnie, w którą stronę idzie klaun. By ją trochę utrudnić, zmienimy naszego klauna w ten sposób, że będzie on sam zmieniał co jakiś czas swój kierunek ruchu. Użyjemy do tego narzędzi, które nazywa się **Alarm clock** (budzik). Każda instancja obiektu może mieć kilka budzików. Zegarki odliczają czas, a

gdy dojdą do 0 następuje zdarzenie o nazwie **Alarm**. W zdarzeniu tworzenia klauna (**Create**) ustawimy nasz budzik, a dla zdarzenia Alarm, damy akcję zmiany ruchu i ponownie ustawimy budzik.

Dodawanie budzika:

1. Otwórz obiekt Klaun, poprzez podwójne kliknięcie na niego w liście zasobów.
2. Wybierz z listy zdarzeń (Events): Create. Ma ono dwie przyporządkowane mu akcje (rozpoczęcie ruchu i rozpoczęcie muzyki), my dodamy trzecią akcję. Z zakładki **main2** wybierz akcję **Set Alarm** (Ustaw alarm) . Jako **Number of steps** (Ilość kroków) wybierz 50. Numer w którym budzik rozpocznie alarm zostawiamy na zero - Alarm 0. Naciśnij OK.



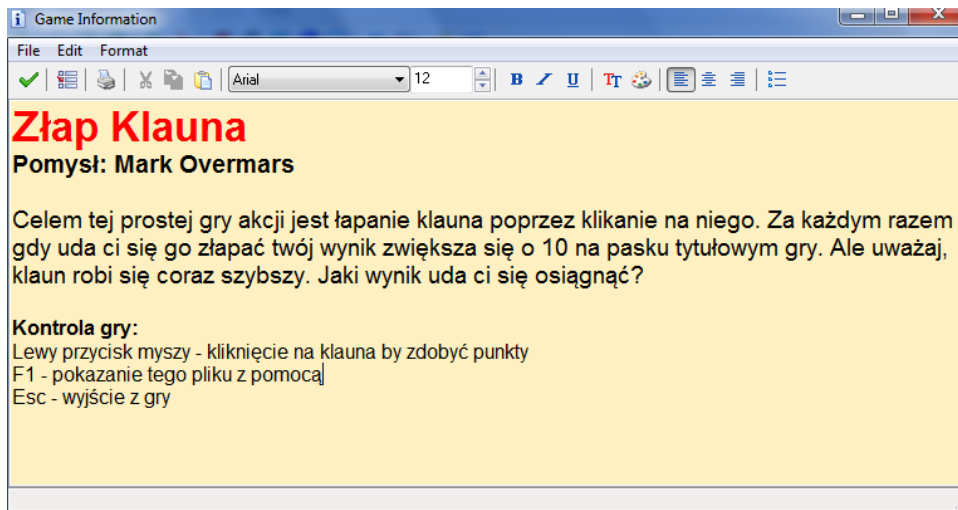
3. Wraz z postawieniem klauna w pokoju nakręcamy budzik na 50 kroków (to przed chwilą zrobiliśmy). Co się stanie jak budzik zejdzie do 0? Musimy dodać nowe zdarzenie guzikiem **Add Event**. Następnie wybrać **Alarm** a z menu które wyskoczy: **Alarm 0**.
4. W akcjach dla zdarzenia Alarm 0 musi znaleźć się zmiana ruchu czyli znane nam już **Move Fixed**  (z zakładki **move**). Znow zaznacz wszystkie 8 strzałek i ustaw szybkość (Speed) na 0 oraz wybierz opcję **Relative**. To znaczy, że dodasz do szybkości 0, czyli właściwie szybkość się nie zmieni.
5. Drugą akcją dla zdarzenia Alarm 0 powinno być ponowne nakręcenie budzika do 50 jednostek. Dodaj zdarzenie **Set Alarm** i ustaw liczbę kroków **Number of steps** by znow wskazywała 50.

Warto też zastanowić się ile trwa te tajemnicze 50 kroków. Domyślnie w programie *GameMaker* 30 kroków to sekunda (ma to związek z ilością klatek na sekundę). 50 kroków to zatem około 1.5 sekundy. (Możesz zmienić prędkość gry w zakładce **settings** we właściwościach pokoju).

Ostatnią rzeczą, którą zrobimy jest dodanie pomocy do naszej gry. Czasem gracze mogą nie wiedzieć i nie domyślić się o co chodzi w naszej grze. Dobrze jest zatem dodać tę opcję do gry. W *GameMakerze* jest na to prosty sposób:

Dodawanie tekstu pomocy:

1. Z menu **Resources** wybierz opcję **Change Game Information** (Zmień informację o grze). Pojawi się prosty edytor tekstowy.
2. Wpisz w nim jakiegokolwiek informacje, które mogą przydać się graczowi, zwłaszcza cel gry i sposób na kontrolowanie jej. Możesz używać różnych czcionek, wielkości i kolorów:



Gdy skończysz naciśnij zielony dziubek w lewym górnym rogu.

W trakcie gry ta pomoc będzie pokazana graczowi gdy naciśniesz klawisz F1 na klawiaturze (który zwyczajowo odpowiada za wyświetlenie pomocy).
Pozostaje zapisać grę i wygenerować plik exe.

Twoja pierwsza gra jest gotowa

Gratulacje! Zakończyłeś tworzenie swojej pierwszej gry, która z reguły jest najbardziej kłopotliwą. W toku tworzenia nauczyłeś się wielu ważnych aspektów programu *GameMaker* takich jak: sprite'y, obrazki, dźwięki, obiekty i ich instancje, zdarzenia i akcje oraz pokoje. Jest jednak wiele innych rzeczy do poznania co zostało zawarte w dużej ilości samouczków na stronach <http://www.yoyogames.com> lub polskiej <http://gmclan.org/>. Istnieje też angielska książka dla szczególnie zainteresowanych tym programem:

Jacob Habgood, Mark Overmars, *The Game Maker's Apprentice: Game Development for Beginners*, Apress, 2006, ISBN 1-59059-615-3.