

Witam!

Wiele osób lubi grać w gry komputerowe. Niektórzy wolą platformówki, inni bijaktyki, FPS'y, MMO ...

Ale niewielu tworzy gry – wydaje się to być trudne, czasochłonne, nawet trochę nerdowskie ;)

Ja postaram się pokazać wam, że tworzenie gier, to dość prosta (choć czasem czasochłonna) zabawa, w której „wcielamy się” w Boga :P Jak inaczej nazwać tworzenie czegoś według własnej koncepcji? Oczywiście mocy nie mamy, ale coś stworzymy.

Najpierw powinniśmy się zaopatrzyć w coś w czym można grę stworzyć, ogólnie znam 2 programy „jako tako” (tzn. testowałem), a trzeciego nie zrozumiałem. Ale też polecę :D

- Game Editor

Game Editor jest prostym klikaczem, w którym przede wszystkim trzeba znać język angielski. O ile większość rzeczy da się wyklikać, to trzeba wiedzieć co wyklikać. Sam projekt jest na licencji GPL i umożliwia konwersję na wiele platform (głównie starych telefonów i podstawowych 3 systemów operacyjnych (choć można spróbować przekompilować na BSD czy inne systemy)

- Enigma-dev (Będę tego używać)

Enigma to średnio-zaawansowany program (jeśli można tak nazwać) – ogólnie pozwala tworzyć gry przez amatorów, jak i przez zaawansowanych użytkowników (oczywiście im więcej umiemy, tym większe cuda tworzymy), opiera się na EDL (Enigma Development Language). Ogólnie można nazwać program klonem Game Makera (ogólnie Panowie starają się osiągnąć kompatybilność)

- SandBox Game Maker (wkraczamy na głębokie wody, uciekać)

Mimo, że próbowałem go opanować – poległem. SBGM służy do tworzenia gier RPG, FPS i Platformówek w 3D (na silniku Cube 2), ogólnie wszystko wygląda genialnie i jak ktoś go opanuje, to musi tłuc dużo gier Open Source (i wysokiej jakości, w coś muszę grać :P)

1) Planowanie

Pierwsze co powinniśmy zrobić to zaplanować sobie co nieco o grze. Należy tutaj zwrócić na parę rzeczy uwagę (o których napiszę). Podstawowa zasada – nie wymyślajmy gry 3D z kampanią na 5000h i cudami na kiju. W zasadzie o ile enigma jest w stanie takie cuda tworzyć, to zrobienia takiej gry potrwałoby pewnie kilka lat. Jako, że jesteśmy początkującymi twórcami, powinniśmy tworzyć gry z jak najmniejszą złożonością, głównie dlatego, że takie rzeczy szybko się tworzy (tzn. szybko widać efekty) i można szybko skończyć.

Druga rzecz to grafika, jeśli nie robimy gry tekstowej, powinniśmy o tym pomyśleć. Problemem jest tu to, że albo ją stworzymy w paint (lub czymś innym), albo kogoś załatwimy. Problem w tym, że jeśli nie umiemy rysować w paint'cie to lepiej się nie zabierać. A osoba, która nam będzie tworzyć – zazwyczaj olewa to (smutna prawda) i projekt sam umiera. Najlepiej tu wcześniej sobie obejrzyć tileset'y (grafiki) i pod to dobierać grę. Bardzo warto przeglądać: <http://opengameart.org> za nimi.

Muzyka i dźwięki – z tym ogólnie jest spory kłopot. Można oczywiście coś wyklikać lub zabrać z innych projektów (czemu nie?), ale nie zawsze wszystko pasuje. Warto też zauważyć, że jak coś bierzecie z innego projektu, to na jakiej licencji owa rzecz jest.

2) Tworzenie właściwe

Warto zwrócić uwagę, że „moja gra” będzie tylko przykładem, więc nie napalajcie się na pełnowymiarową platformówkę (a co najwyżej jej załączek)

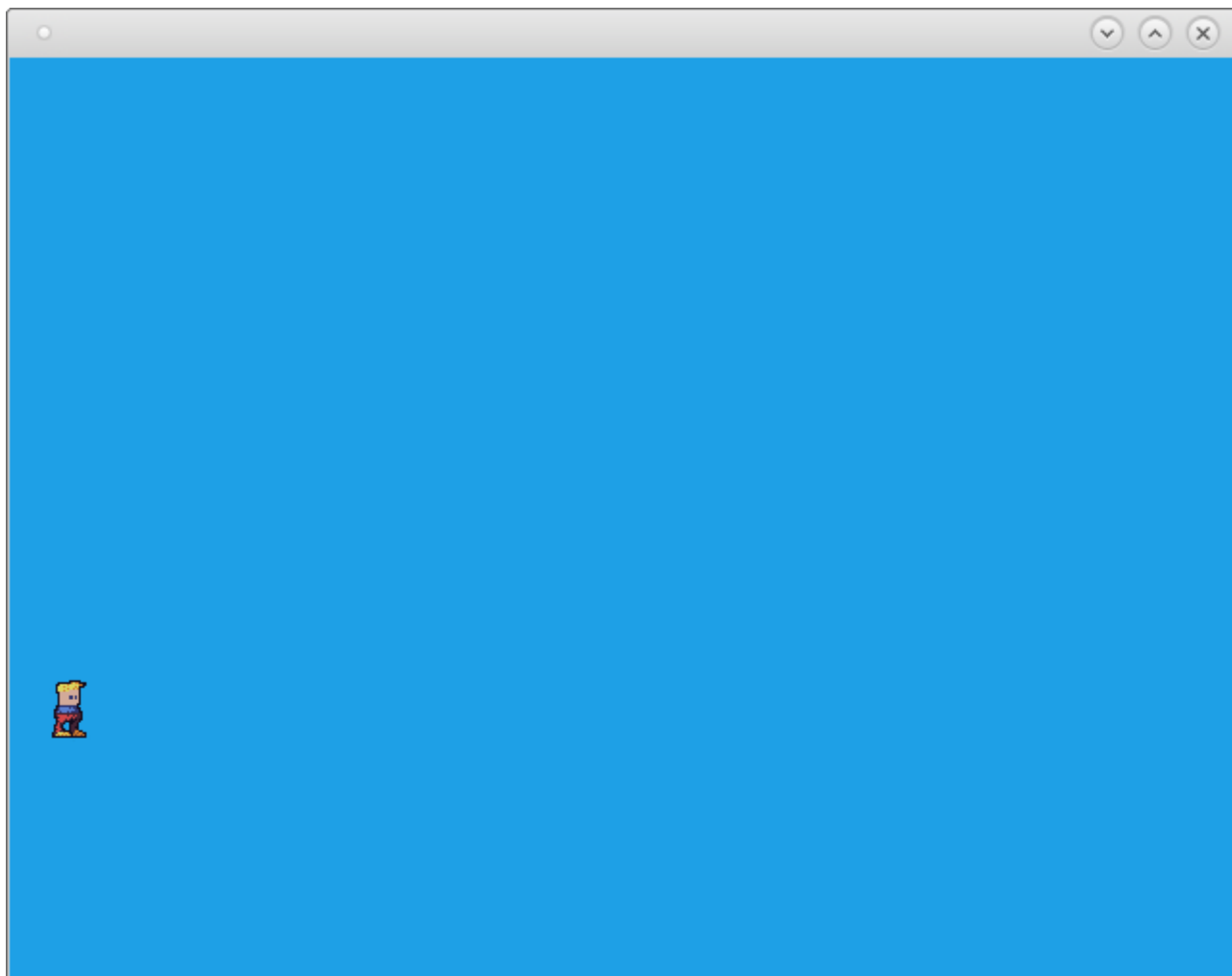
<http://opengameart.org/content/outside-tileset>

Wybrałem ten tileset, jest prosty i pozwoli szybko wykorzystać go.

Najpierw powinniśmy zacząć od głównego bohatera, wycinamy go w Paint (ja używam Pinty, ale Kolourpaint jest równie dobry), tworzymy nowy Sprite, wklejamy go i nazywamy grafikę odpowiednio (ja zalecam stosowanie nazw spr_nazwa, z powodu, że obiekty (i inne rzeczy) nie mogą się nazywać tak samo, a spr_hero (na przykład) można łatwo powiązać myślowo z obj_hero (itd.).



Teraz warto stworzyć obiekt (obj_hero) i przypisać mu grafikę (spr_hero), stworzyć nowy room i ustawić tam postać.



2) Ruch postaci

Każda postać w grze powinna się ruszać (chyba, że gramy zwłoki, albo postaci nie ma). Nasza będzie się ruszać w prawo, lewo i górę (w dół będzie spadać co najwyżej)

Warto tutaj znać podstawy matematyki – X i Y (czyli układ kartezjański).

Ale na razie do obiektu obj_hero musimy dodać eventy (zdarzenia), w którym będzie wciśnięcie klawisza (Keyboard) → Prawo, lewo i góra (po kolei dodajemy)

Do chodzenia w prawo używamy kodu:

```
x=x+3
```

Oznacza to dodanie do X'a liczby 3 (gdzie X to pozycja postaci) i zapisania jako X (czyli nowy X będzie X+3)

Podobnie robimy w lewo (tylko, że odejmując)

Żeby zrobić skok nie wystarczy zrobić $y=y+3$ z prostego powodu, zmiana pozycji Y to zrobienie „wspinania się po drabinie” (lub czymś takim), gdzie nie ma grawitacji, my użyjemy

```
if !place_free(x,y+1)
vspeed=-8;
```

Działa to tak, że jeśli nic nad sobą nie mamy (żadnego obiektu) to prędkość pionowa (Vertical Speed) wynosi -8.

Przy okazji dodajmy Grawitację (która nas będzie ściągała w dół), używamy do tego Event „Step” (taka pętla, powtarza cały czas kod) i wpisujemy

```
if place_free(x+0,y+3)
{
gravity=0.25
}
else
{
gravity=0
}

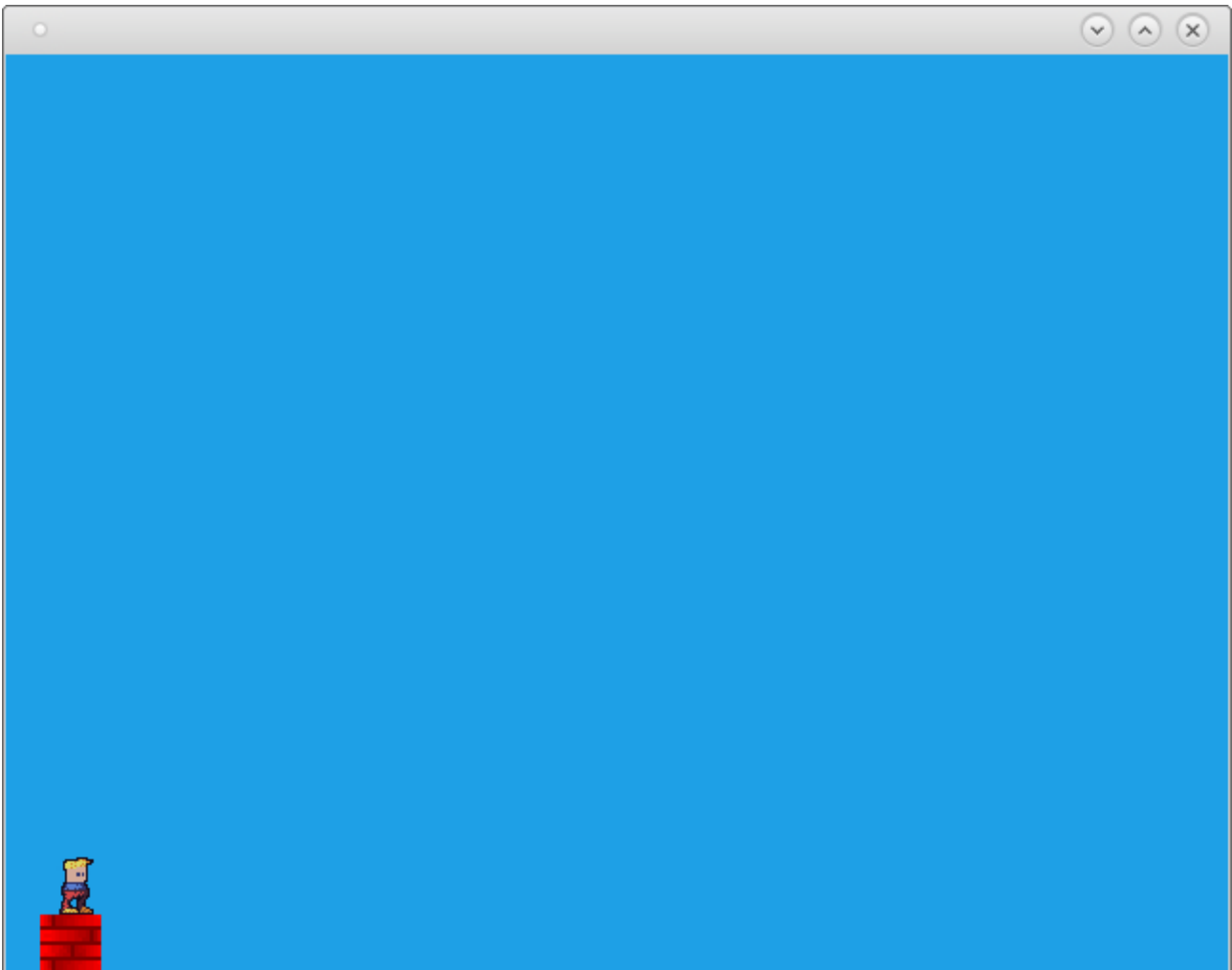
if(vspeed>8)
{
vspeed=8
}
```

Należy też na szybko stworzyć jakiś murek, bo grawitacja nas pochłonie w dół, poza ekran :) Czasowo pożyczyłem sobie z mojego innego projektu ścianę, dodałem obiekt i ustawiłem opcję solid (jest ona stosowana przy takich ścianach, oznacza zazwyczaj obiekty nieruchome)

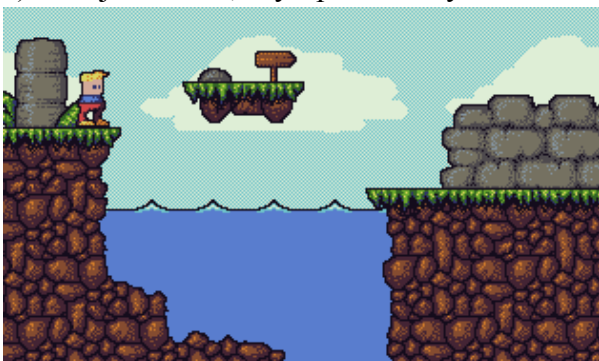
W obj_hero dodajemy event Collision z obj_wall i wpisujemy kod:

```
if(vspeed>0){  
  move_contact_solid(270,vspeed);  
}  
vspeed=0;
```

Dzięki czemu jeśli się spotka nasza postać ze ścianą, nie będzie spadać



3) Zabójcza woda, czyli przeszkody



Na rysunku (proponowanym przez autora tilesetu) mamy morderczą wodę. Jako, że my lubimy mieć mordercze wody, wycinamy wodę z tilesetu i dodajemy do sprite i tworzymy obiekt obj_water

Niestety z powodu problemów technicznych wykorzystam wodę ciemniejszą (bo jaśniejszą wyciąłem, a nie mogę cofnąć tego wyboru :D)



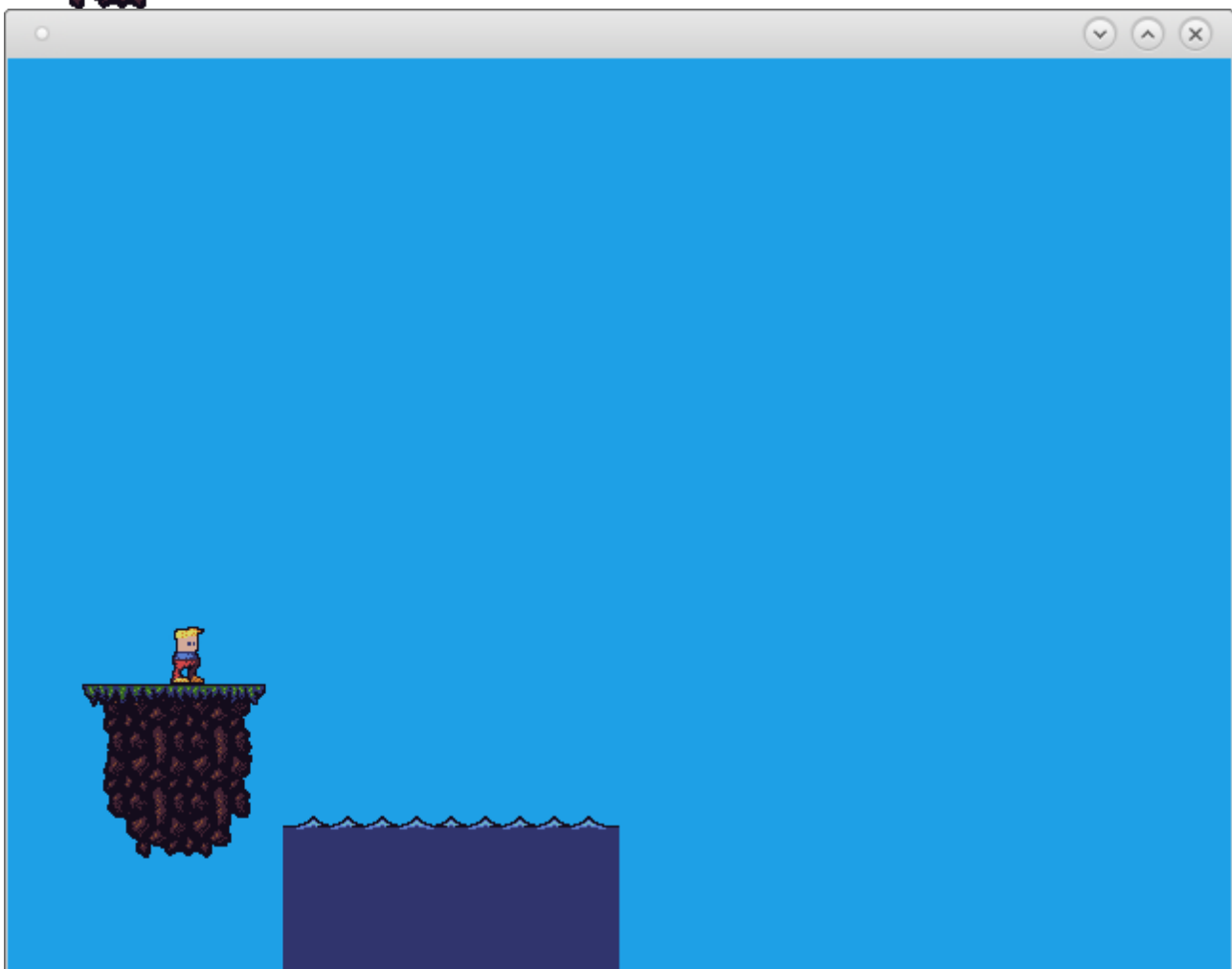
Teraz tworzymy kolizję z obj_water i ustawiamy kod

```
game_restart();
```

Oznacza to tyle, że gra (w tym nasza pozycja, zebrane punkty [jakby były]) się zrestartuje, prawie jakbyśmy dopiero ją odpalili

4) Ziemia

To jest ten moment, w którym obj_wall (i spr_wall) wywalamy i zastępujemy ziemią. Konkretnie to wyrzucamy spr_wall, dodajemy spr_ground, w obj_wall zmieniamy sprite na spr_ground, a samą nazwę obiektu zmieniamy. Uff

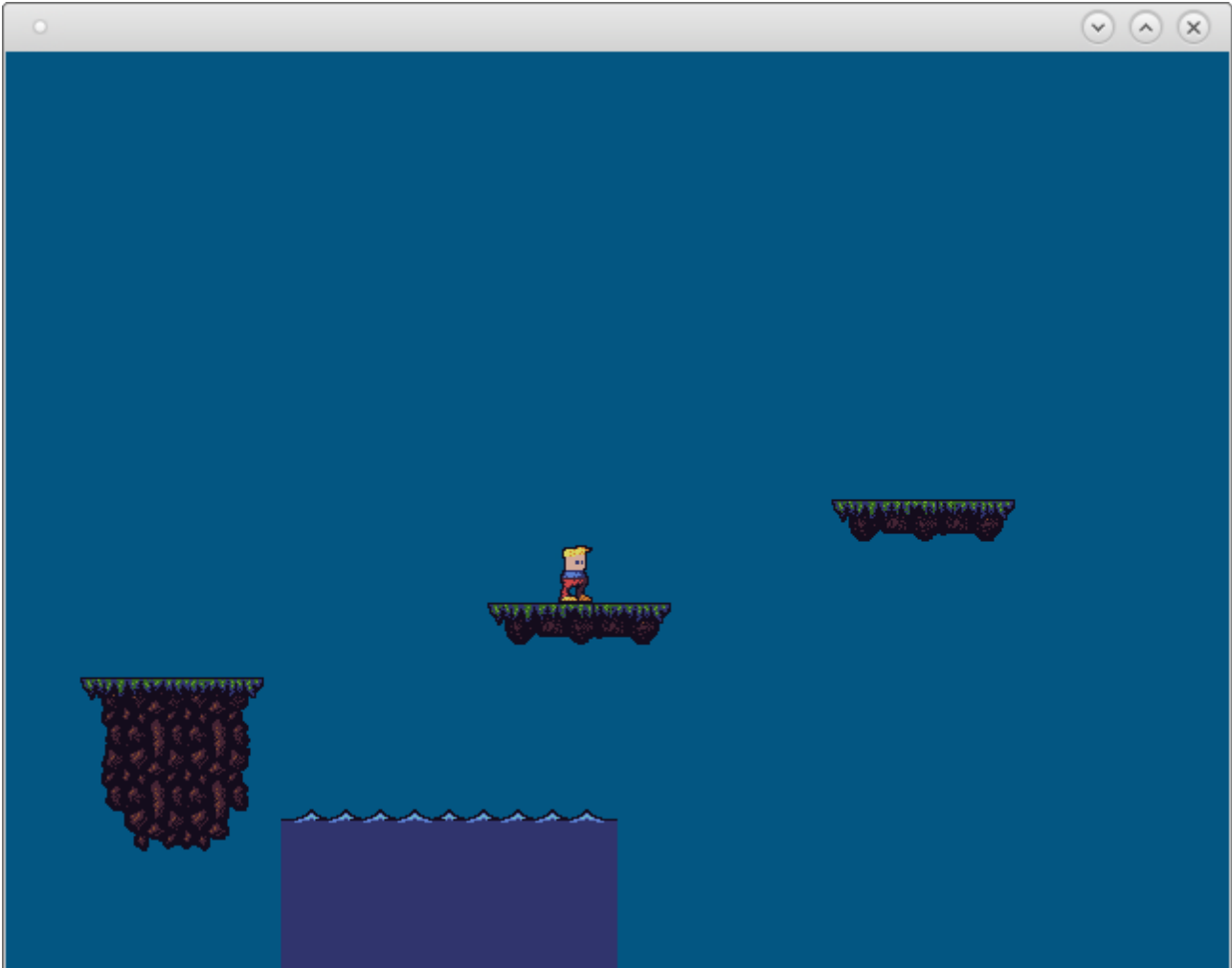


4) Tło

Autor Tilesetu nie dał nam tła, więc albo sami je zdobędziemy, albo wykorzystamy jednolity kolor. Mi osobiście tak jaskrawy niebieski się nie podoba, to w ustawieniach rooma (w zakładce Background) zmieniłem kolor na ciemniejszy.

5) Mała ziemia

Dodajemy jeszcze obiekt „mała ziemia”, która będzie platformą. Jest to nic innego jak ziemia. (tyle, że mała)



6) Koniec gry

Wszystko ma swój początek i koniec – my musimy dodać koniec. Wykorzystamy tu grafikę znaku (ze strzałką, ale co tam), jak nasz cel.

Tutaj robimy podobnie, dodajemy znak, dodajemy obiekt z grafiką znaku, w postaci dodajemy kolizję ze znakiem. Tym razem nasz kod:

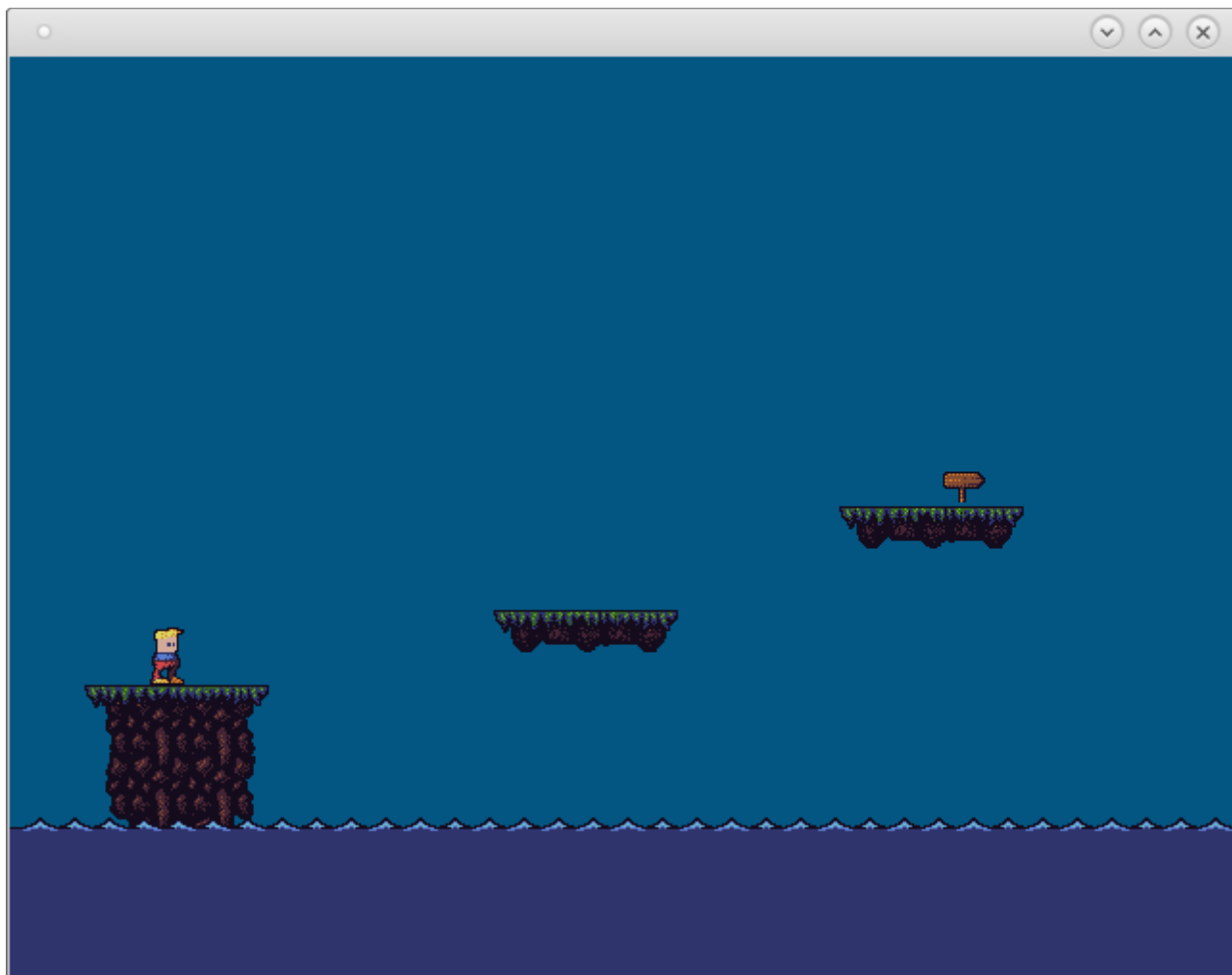
```
game_end();
```

Wyłączy grę

7) Poprawki wizualne

Żeby nasza gra ładnie wyglądała, musimy poprawić wizualnie ją – woda niech obejmuje całe terytorium itd.

Voila! Gra skończona!



Oczywiście możemy jeszcze wiele mechanizmów wprowadzić w grze, jak np. zapobieganiu możliwości ucieczce poza grę. Ale to już w waszych rękach zależy co zrobicie. Moja rola dobiegła końca, mogę umrzeć w spokoju :)

Przydatne linki:

<http://enigma-dev.org/>

Strona enigma-dev

<http://gmclan.org/>

Polska strona dotycząca Game Makera (i po części enigmy)

<http://gmclan.org/index.php?wiki>

Polska dokumentacja Game Makera, pozwoli nam nauczyć się jak tworzyć gry.

<http://forum.gmclan.org/>

Forum Gmclan (dotyczące Game Makera), tu można zadawać pytania

http://gmclan.org/up6184_3_gra.html

Kod źródłowy naszej gry