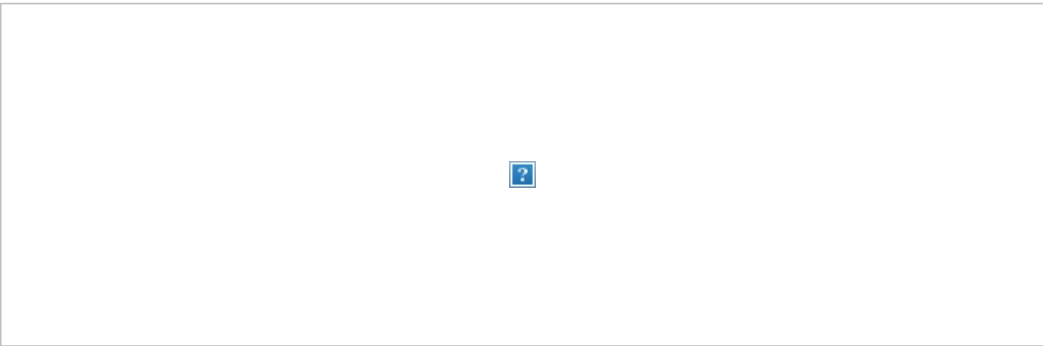


# [SFML] Wczytywanie dużych poziomów (przewijana mapa) (sheadovas/poradniki/goto/sfml-wczytywanie-duzych-poziomow-przewijana-mapa/)

Wrz 15, 2014 / goto (sheadovas/category/poradniki/goto/)

Dzisiaj dowiesz się w jaki sposób można wczytywać poziomy zajmujące dużą ilość kafli jednak w bardziej oszczędny dla pamięci sposób, tak aby nie potrzebnych kafli po prostu nie wczytywać. W przykładzie poniżej korzystam z SFML'a w wersji 2.0, jednak kod powinien być łatwy w konwersji na inne biblioteki.

Tak jak wspomniałem zajmiemy się problemem wczytywania zbyt dużych poziomów, aby zmieściły się na ekranie oraz tak dużych że mogą spowolnić działanie gry, zobrazujmy sobie ten problem:



([https://i2.wp.com/sites.google.com/site/sheadovasdatabase/home/images/schemat\\_wczytywania.png?ssl=1](https://i2.wp.com/sites.google.com/site/sheadovasdatabase/home/images/schemat_wczytywania.png?ssl=1))

(kliknij aby powiększyć)

Przy wczytywaniu wszystkich kafli naraz (i rysowanej tablicy o tych samych rozmiarach) posiadamy jedynie 2 z 3 obszarów wyróżnionych na schemacie powyżej:

- obszar niewidoczny dla użytkownika, na którym znajduje się cały wczytany poziom, tzn nawet te fragmenty, których nie widzimy
- Scena gry, która wyświetla ten fragment poziomu na który zwrócona jest kamera.

Weźmy na ten przykład poziom który ma wymiary w kaflach 100 x 30, a ekran może wyświetlić 50x30 kafli. Tworzenie i wyświetlanie tablicy sprite'ów o tych samych rozmiarach co poziom jest bezużyteczne bo 49-50 kafli nie zostanie zobaczone przez gracza mimo że są rysowane. Olbrzymia strata pamięci dla komputera.

Rozwiązanie zbyt dużej ilości kafli na ekranie przychodzi za pomocą 3 pośredniej warstwy: obszar wczytywany do pamięci to obszar, który jest nieco większy niż scena (okno) gry o parę kafli więcej niż ekran po to aby przy ruchu gracza kamera mogła wyświetlić kafle dotychczas schowane za sceną.

Wracając do przykładu: jeżeli przy takich samych rozmiarach poziomu i ekranie mieszczących tyle samo kafli co wcześniej utworzymy tablicę do wyświetlania sprite'ów o nieco większych wymiarach (52 x 30) to zaoszczędzimy bardzo dużo pamięci, bo nie będziemy wczytywać całego poziomu lecz potrzebny nam fragment, w poziomie wczytujemy kafli nieco więcej bo musimy być gotowi do wyświetlenia kolejnych kafli gdyby gracz poszedł w prawo albo lewo (kamera nie musi obejmować całych kafli lecz np. połowę tego co wcześniej w całości za sceną).

Poniżej zamieszczam przykładową implementację takiego wczytywania poziomu jedynie fragmentami, zapraszam do samodzielnej implementacji, a w razie problemów, sugestii, wątpliwości zapraszam do systemu komentarzy (pamiętajmy że wszelkie komentarze zawsze są miło widziane).

Ostrzegam, że minimapa się przycina z tego powodu, że poruszamy się tam o 4 piksele, stąd to złudzenie (trzeba by odpowiednio prędkość gracza dopasować do delta time, wtedy powinno wszystko być ok).

C++

```

1  #include <cstdlib>
2  #include <ctime>
3  #include <iostream>
4  #include <SFMLGraphics.hpp>
5
6  using namespace sf;
7
8  const int WIDTH = 100; // wysokosc i szerokosc poziomu w kafkach
9  const int HEIGHT = 12;
10
11 short lPoziom[HEIGHT][WIDTH]; // logiczna reprezentacja poziomu (za pomoca cyfr)
12 Texture kafle[3]; // tekstury kafli, skorzytsamy tylko z 3 na 6 mozliwych
13
14 void ustaw_kafle(float,Sprite[12][22]); // funkcja ustawiajaca, ktore kafle maja byc wyswietlane
15
16 int main()
17 {
18     for(int i=0;i<3;i++)
19     {
20         // wczytywanie kafli
21         kafle[i].loadFromFile("_data/tileset.png", IntRect(i*64,0,64,64));
22     }
23
24     // generowanie poziomu
25     srand(time(NULL));
26
27     for(int i=0;i<HEIGHT;i++)
28     for(int j=0;j<WIDTH;j++)
29     lPoziom[i][j] = rand()%3; // losowanie wartosci: 0 || 1 || 2
30
31     // stworzenie okna z zablockowana mozliwoscia zmieniania wielkosci ekranu
32     RenderWindow okno(VideoMode(1280,720),"Przewijany poziom",Style::Close);
33

```

Kod jest również dostępny w wersji na GitHub:



([https://github.com/sheadovas/Source-code/blob/master/sfml\\_scrollingmap.cpp#L81](https://github.com/sheadovas/Source-code/blob/master/sfml_scrollingmap.cpp#L81))