
TYTUŁ: ESCAPE FROM ALCATRAZ

AUTOR: Mikołaj Szotowicz

AGENDA:

- 1) Temat projektu
- 2) Realizacja wymagań
- 3) Zasada działania programu
- 4) Metody i funkcje
- 5) Film prezentacyjny (<https://youtu.be/0JuQk017qZA>)

1. TEMAT PROJEKTU:

"Escape from Alcatraz" to gra logiczna, w której wcielamy się w postać więźnia. Naszym celem jest przesuwanie kluczy, w taki sposób, aby wszystkie znajdowały się na kłodce. Dopiero wtedy otwarte zostają drzwi do kolejnego pomieszczenia.

2. REALIZACJA WYMAGAŃ:

Wszystkie zamierzone wymagania zostały spełnione:

- 30 zróżnicowanych pomieszczeń (pod względem poziomu trudności i liczby kluczy),
- łatwa edycja istniejących pomieszczeń,
- odblokowywanie kolejnych pomieszczeń,
- możliwość ręcznego wyboru pomieszczeń (spośród wcześniej odblokowanych),
- zapamiętywanie aktualnego rozgrywanego pomieszczenia i łatwy powrót do rozgrywki,
- responsywny układ elementów graficznych pozwalający na prawidłowe wyświetlanie, w każdych warunkach

3. ZASADA DZIAŁANIA PROGRAMU:

a) Gra korzysta z plików tekstowych (katalog "Lib"), w których:

- przechowuje układ schematyczny poszczególnym poziomów/pomieszczeń ("`<nr>.txt`"),
- przechowuje informację o aktualnie rozgrywanym pomieszczeniu ("`level.txt`"),
- przechowuje informację o ilości odblokowanych poziomów ("`maxLevel.txt`").

b) Wybranie poziomu z menu i nowa gra:

Wybranie pomieszczenia z poziomu menu (zakładka „select level”) odbywa się za pomocą zapisu wartości tekstowej z przycisku do pliku „level.txt” i uruchomienie gry. Przykładowo przycisk z wartością „2”, zapisuje „2” do pliku, w efekcie uruchamia się pomieszczenie o schemacie 2.txt. Tak proste rozwiązanie pozwala na zarządzanie wieloma przyciskami dzięki jednej, wspólnej funkcji. Podobnie działa opcja „nowej gry”, która zapisuje do ww. pliku wartość „1”.

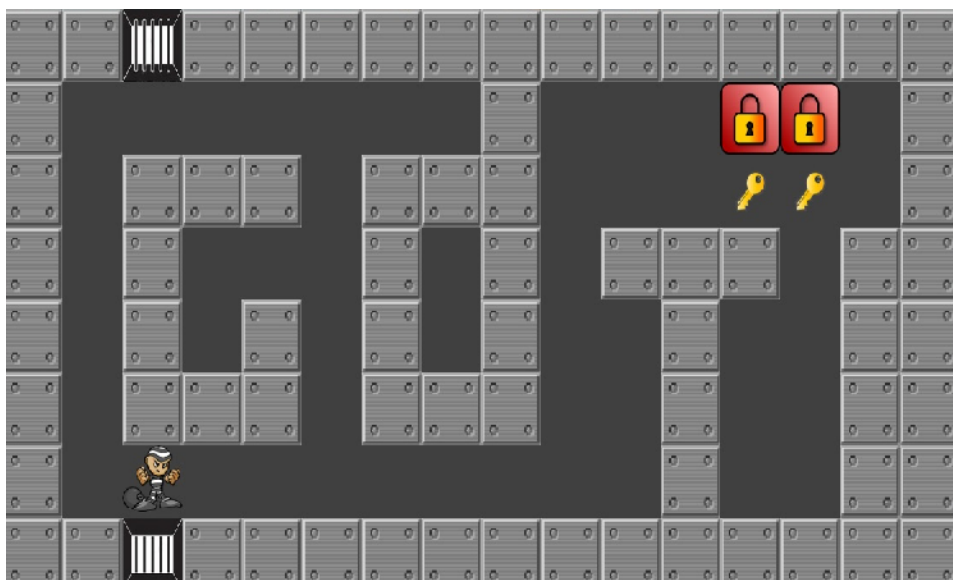
c) Kontynuowanie rozrywki

Opcja „resume” pozwala na kontynuowanie rozgrywki w ostatnio rozgrywanym pomieszczeniu. Z punktu technicznego polega to na odczycie pliku „level.txt” i załadowaniu odpowiedniej planszy.

d) Plansza

Przykładowy układ planszy wraz z schematem w formie tekstowej (rys. 1).

Legenda: ‘s’ – start, ‘e’ – wyjście (exit), ‘w’ – ściana (wall), ‘k’ – klucz (key), ‘l’ – kłódka (lock), ‘p’ – więzień (prisoner), ‘g’ – grunt (ground).



```
wwewwwwwwwwwwwwwww
wggggggggwgggllgw
wgwwwgwwwgggkkgw
wgwgggwgwgwwwgww
wgwggwgwgwgwgww
wgwwwgwwwggwgww
wgpggggggggwgww
wwswwwwwwwwwwwwww
```

Rys. 1 Przykładowa plansz

e) Rozgrywka

Plansza składa się ze 128 pól (8x16). Podczas wczytania elementów graficznych poziomu, wczytywany jest również układ planszy do macierzy o ww. rozmiarze. W ten sposób łatwo zlokalizować, czy stojącym obok więźnia (‘p’) elementem jest ściana (‘w’), czy np. klucz (‘k’).

Gracz porusza się więźniem wyłącznie po pustych polach (‘g’) lub kłódkach (‘l’), a przesuwanie się na klucz powoduje również jego przesunięcie o jedno pole w kierunku, w którym poruszył się bohater gry. Przedstawiona wyżej sytuacja ma miejsce wyłącznie wówczas, gdy miejsce, które ma zająć klucz, jest puste lub znajduje się na nim kłódka. Sytuacja, w której dowolny klucz znajduje się na dowolnej kłódce jest prezentowana przed odmienną grafiką (rys. 2).



Rys. 2 Ikona kluczy. Po prawej klucz znajdujący się na kłódce, po lewej pojedynczy klucz bez kłódki

Po każdym ruchu następuje sprawdzenie planszy, czy w wyniku interakcji nie zostały umieszczone wszystkie klucze w kłódkach. Jeśli tak, otwierane są drzwi (zmiana ikony na pusty grunt) i dopiero po przekroczeniu strefy planszy rozgrywany poziom jest zakończony i ładowany jest następny (zmiana wartości level.txt) – wyjątek stanowi ostatni poziom, po którym zamiast kolejnego poziomu gracz przekierowany zostaje do menu głównego gry.

4. METODY I FUNKCJE

Nazewnictwo metod i funkcji jest intuicyjna i jednoznacznie świadcząca o czynności wykonywanej przez nią. Zatem CheckWin() sprawdza, czy wszystkie klucze znajdują się na kłódkach, NextLevel() informuje o ukończeniu poziomu i ładuje nową planszę, a chociażby GetIdKey(Key[] keys, int y, int x) zwraca identyfikator klucza znajdującego się w podanej lokalizacji. W związku z powyższym nie ma sensu tłumaczenia każdej funkcji osobno, pozwolę sobie wybrać kilka z nich i wyjaśnić intencje ich użycia.

- a) DisplayPrison(PictureBox[] boxes) to procedura, która interpretuje macierz elementów planszy i umieszcza odpowiednią grafikę w elementach przekazanych jako parametr, na które składają się pola graficzne planszy. Przykładowo, jeśli na danym miejscu znajduje się symbol 'w' w odpowiadającym mu elemencie pola umieszcza ikonę ściany. Różnica, w porównaniu do ładowanej interpretacji z plików tekstowych (<nr>.txt) jest tylko jedna, macierz może zawierać dodatkowo symbol 'K' będący oznaczeniem klucza znajdującego się na kłódce.
- b) MoveItems(char direction) to procedura do przesuwania elementów po planszy więzienia. Przykładowo, jeśli wciśniemy klawisz 'A' (oznaczający iście w lewo) i nasz więzień stoi na współrzędnych ikony wyjścia, a dodatkowo współrzędne X są równe 0 oznacza to, że więzień opuścił pomieszczenie i tym samym gracz odblokował kolejny poziom. Lub jeżeli po lewej od gracza znajduje się klucz, a po lewej od tegoż klucza jest puste miejsce/kłódka to należy zmienić lokalizację zarówno klucza, jak i więźnia. Ostatnią możliwością jest możliwość przesunięcia się na miejsce nie zajmowane przez nic (symbol 'g'), w tym wypadku następuje zmiana lokalizacji wyłącznie więźnia. Inne ruchy (W/S/D) są sprawdzane w podobny sposób.
- c) CheckWin() sprawdza ile kluczy znajduje się w tych samych lokalizacjach co kłódki, jeśli liczba ta równa się dostępnej ilości kluczy na dane pomieszczenie otwierane są drzwi do kolejnego poziomu. Każdorazowo w sytuacji kiedy klucz pokrywa kłódkę, dany symbol zastępowany jest symbolem 'K'.

Ponadto projekt korzysta z kilku klas pomagających sprawnemu zarządzaniu widokami lub będącą instancją rzeczywistego obiektu (np. klasa Key). Projekt korzysta także z wyrażenia lambda, w celu prostego wyszukiwania określonych wartości (np. identyfikator klucza znajdującego się w podanej lokalizacji).

5. FILM PREZENTACYJNY:

<https://youtu.be/0JuQk017qZA>