

Dokumentacja Projektu

Bartosz Kaaz, 106085

Temat:

-Tematem mojego projektu było napisanie gry sterowanej za pomocą akcelerometru. Gra jaką napisałem to remake gry „Battle City”. Jest to gra zręcznościowa typu akcja wyprodukowana i wydana na konsole NES w 1985r. przez japońską firmę Namco. Jest to kolejna część gry Tank Battalion z 1980 roku. Battle City jest również znane pod nazwą Tank 1990 oraz wydane pięć lat później po grze Tank Battalion przez tego samego producenta.

Starałem się aby moja wersja gry nie odbiegała zbyt od oryginału więc:

Zasady gry są proste. Gracz steruje czołgiem, którym niszczy inne czołgi za pomocą rakiet.

Rozgrywka polega głównie na tym, aby zniszczyć wszystkie czołgi wroga o liczbie 20, przy czym należy chronić orzełka znajdującego się na dole planszy. Po zniszczeniu wszystkich czołgów gracz przechodzi do następnego poziomu.

Jeżeli gracz straci wszystkie punkty życia, lub on bądź wrogowie zniszczy orła na dolnym ekranie, następuje koniec gry.

W grze istnieją 2 tryby rozgrywki: 1 gracz i 2 graczy. W trybie gry jednoosobowej na planszy pojawia się jeden czołg o kolorze żółtym sterowany przez gracza. Podczas rozgrywki z dwoma graczami na ekranie pojawiają się dwa czołgi różniące się między sobą kolorami, gdzie jeden gracz steruje czołgiem żółtym, a drugi zielonym. Jedynym przeciwnikiem z jakim gracz może stawić czoło, są czołgi o srebrnym kolorze (lub innym w zależności od stopnia ochrony czołgu), które (podobnie jak u gracza) potrafią się poruszać i strzelać, co daje większą przewagę nad oponentem (czołg gracza). Ich głównym celem, jaki próbują wykonać na każdym poziomie, jest przebicie się przez mur orła oraz pokonanie jego.

Każde z poniższych czołgów różni się wyglądem oraz szybkością poruszania się i amunicji, natomiast w większości nie posiadają odporności na strzały. Jeśli będą atakowane przez przyjaciół (wrogów gracza), nie otrzymają żadnych obrażeń.

W każdym etapie czołgów wroga jest 20, gdzie na planszy pojawia się maksymalnie 4. Jeśli gracz pokona jednego

z obecnych na planszy, to po chwili powstaje następny, którego należy także wyeliminować i rozgrywka toczy się do momentu, gdy gracz zastrzeli wszystkie wrogie czołgi (20). Liczba czołgów zniszczonych przez gracza jest wyświetlana na pasku po prawej stronie. W grze występują 3 typy czołgów:

1 typ: średnia prędkość, słabe strzały

2 typ: duża prędkość, słabe strzały

3 typ: mała prędkość, strzały przebijają stal

Dokumentacja programistyczna

Środowisko w jakim pracowałem to CoCo IDE oraz Microsoft Visual 2012. Skorzystałem również z biblioteki SDL, która ułatwiła moją pracę. Dzięki niej mogłem obsługiwać grafikę oraz dźwięki.

Sterowanie odbywa się za pomocą połączenia akcelerometru oraz połączenia USB.

Jednym z problemów jakiego musiałem podołać było sprawdzanie kolizji czołgów z innymi obiektami na planszy. Inne, pomniejsze problemy polegały na wyświetlaniu animacji w odpowiednich odstępach czasu, odpowiednie oddziaływanie na siebie obiektów na planszy (np. pocisk niszczy cegłę, ale przelatuje przez wodę i las). Innym problemem było losowanie czołgów-przeciwników w odpowiedniej kolejności, oraz sprawdzanie czy w jednym miejscu i w tym samym czasie nie powstaną dwa czołgi.

Struktury danych jakich użyłem w moim projekcie to wektory z biblioteki STL.

Wszystkie obiekty jakie poruszały się na planszy były umieszczone w jednym dużym wektorze i porozumiewały się ze sobą przez funkcje wirtualne.

Każdy obiekt który znajdował się na planszy był potomkiem klasy GAME_Object w której znajdują się funkcje wirtualne, które wyglądały inaczej przy każdej z klas potomnych (np. takich jak Player czy Brick)

Wybrane fragmenty kodu

Przykładowa funkcja check_collision, która sprawdza kolizje z innym obiektami.

```
bool Tank_source::check_collision(GAME_Object *check){
if(
check!=this && up<=check->bottom && left<=check->right && right>=check->left &&
bottom>=check->up
)
return true;
return false;
}
```

Reakcja pocisku na kontakt z czołgiem:

```
void Shot::movecontact(int type){
if(type!=TANK_SOURCE && type!=WATER && type!=FOREST && type!=EXPLODE &&
Helper==NULL)
{
Helper=new Explode(x,y,5,150);
my_helper=Helper;
Dead=true;
}
}
```

funkcja zapewniająca odbiór danych z akcelerometru i sterowanie czołgiem

```
void Tank_Game::stmControl(int bufferX, int bufferY, int bufferB){
```

```
    EVENTS[K_RIGHT]=false;
    EVENTS[K_LEFT]=false;
    EVENTS[K_UP]=false;
    EVENTS[K_DOWN]=false;
    EVENTS[K_SPACE]=false;

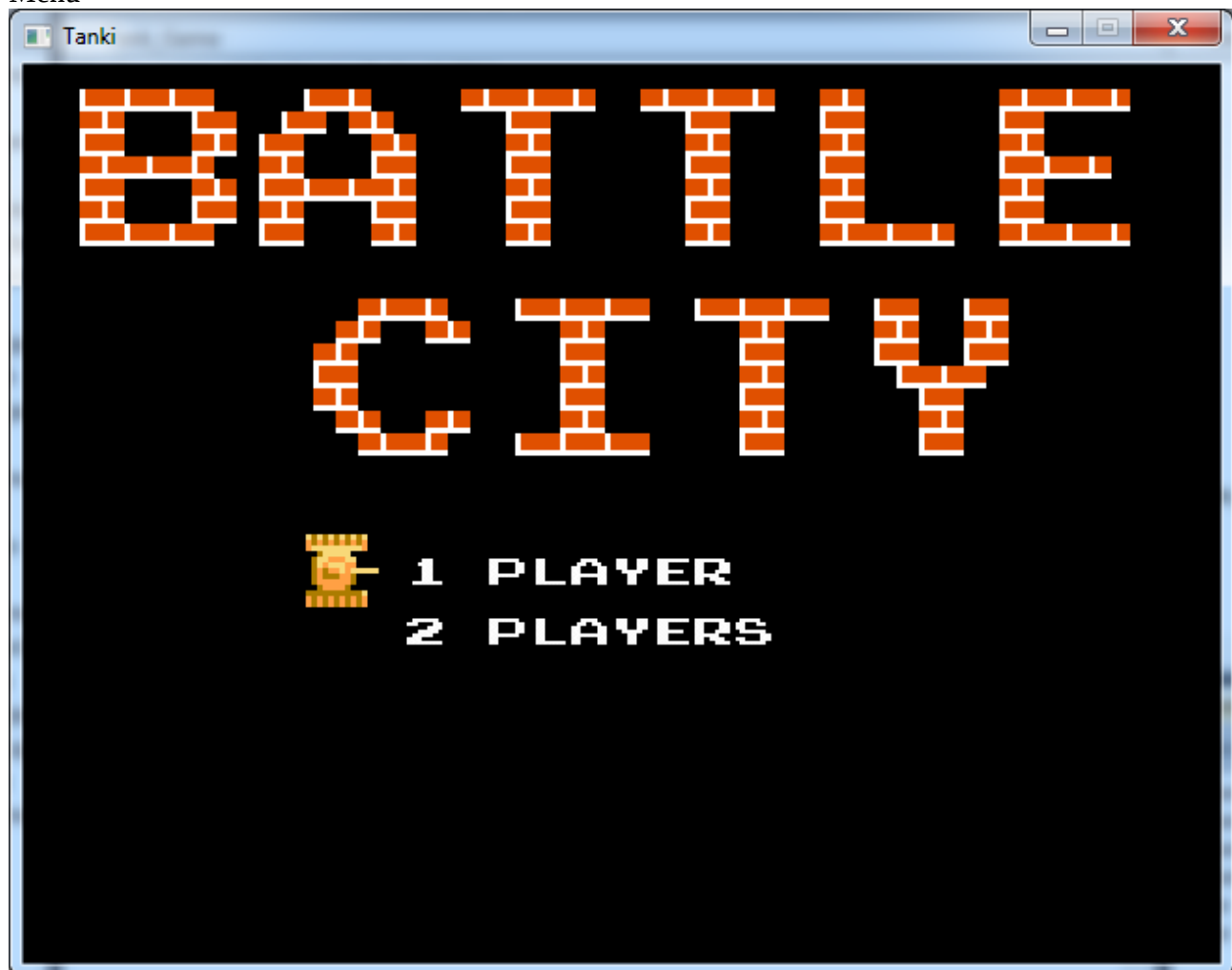
    if(abs(bufferX) > 10 || abs(bufferY) >10)
    {

        if(abs(bufferX) > abs(bufferY)){
            if(bufferX < 0)
                EVENTS[K_RIGHT] = true;
            else
                EVENTS[K_LEFT] = true;
        }
        else{
            if(bufferY < 0)
                EVENTS[K_DOWN] = true;
```

```
        else
            EVENTS[K_UP] = true;
        }
    }
    if(bufferB)
        EVENTS[K_SPACE]=true;
}
```

Testy

Menu



Początek rozgrywki, 1 Player



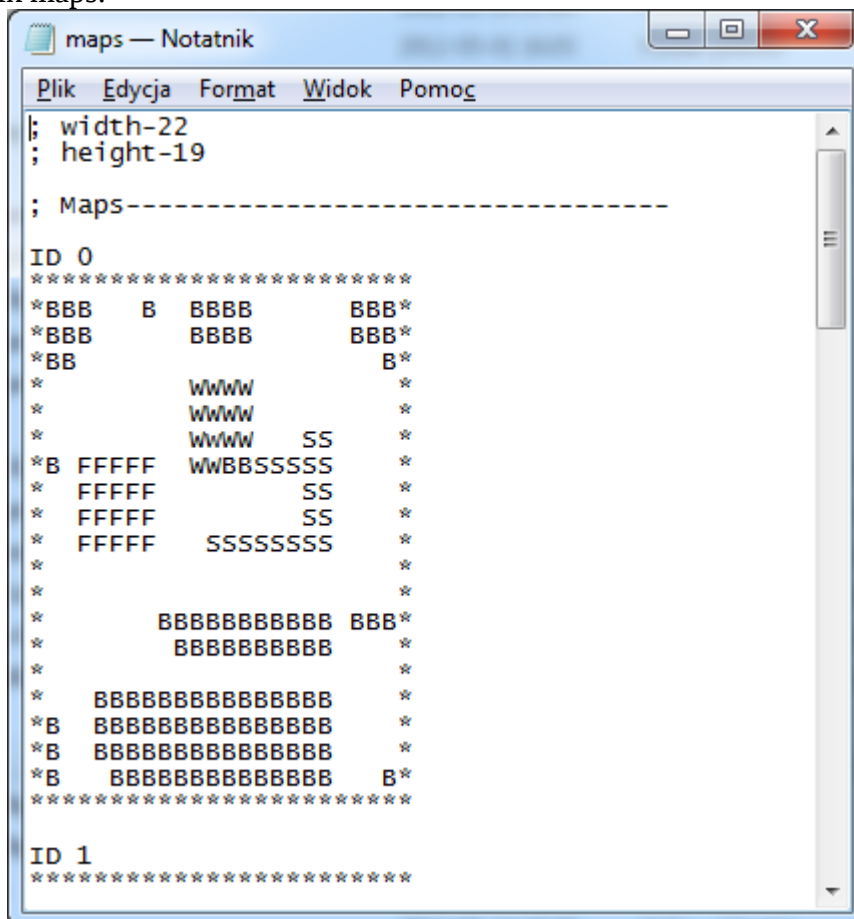
Początek rozgrywki, 2 Player



Dokumentacja użytkowa.

Będąc w menu poruszamy się między opcjami za pomocą akcelerometru (góra, dół), wybór zatwierdzamy buttonem. W grze Player1 steruje swoim żółtym czołgiem również za pomocą akcelerometru oraz buttona. Player2 sterowanie ustawione ma na WASD oraz strzał na P. Za każdym razem gdy przegramy gra automatycznie wraca do menu, jeśli uda nam się pokonać wszystkich przeciwników przechodzimy na kolejną planszę. Plansze wczytywane są z pliku maps.txt, przez edycję tego pliku możemy edytować mapy według własnego uznania, pamiętając tylko o zasadach, że każda mapa musi mieć 19 pól wysokości i 22 pola szerokości. Jeden akapit w pliku txt to szerokość jednego pola w grze.

Przykładowy plik maps:



```
maps — Notatnik
Plik  Edycja  Format  Widok  Pomoc
; width-22
; height-19

; Maps-----

ID 0
*****
*BBB  B  BBBB      BBB*
*BBB      BBBB      BBB*
*BB                      B*
*                      WWWW      *
*                      WWWW      *
*                      WWWW      SS  *
*B  FFFFF  WWBSSSSSS  *
*  FFFFF      SS      *
*  FFFFF      SS      *
*  FFFFF      SSSSSSSS  *
*
*                      *
*         BBBBBBBBBBB BBB*
*         BBBBBBBBBB      *
*
*  BBBBBBBBBBBBBBBBBB      *
*B  BBBBBBBBBBBBBBBBBB      *
*B  BBBBBBBBBBBBBBBBBB      *
*B  BBBBBBBBBBBBBBBBBB      B*
*****

ID 1
*****
```

Definicja każdej mapy musi stać pod linią zaczynającą się od ID, a następnie numer mapy. Po instrukcji „ID” gra wczyta kolejno 21 linii. Znak po znaku. Za każdym razem kiedy wystąpi jedna z kluczowych liter (B- cegła, F- las, W- woda, S- stal) w grze zostanie utworzony odpowiedni obiekt, każdy inny znak będzie ignorowany. Obiekty nie będą stawiane w miejscach gdzie pojawiać się będą czołgi lub gdzie stoi orzeł.

Github: github.com/bartek47/Tank