

Universidade Federal do Rio Grande do Sul
UFRGS

Bow And Arrow

João Afonso Sieben
236712
Robson Mior
230048

Dezembro — 2013
Porto Alegre - RS

Introdução.

O trabalho foi o desenvolvimento de uma versão do jogo “*Bow And Arrow*”, que seria programada na linguagem C. Tendo como base as diretrizes disponibilizadas pelos professores, e um vídeo (<http://www.youtube.com/watch?v=0sue0jkOKTo>), a proposta é que o jogo fosse inteiramente desenvolvido pelos alunos.

Desenvolvimento.

Como proposto, o jogo foi inteiramente desenvolvido na linguagem C. Foi decidido que o jogo seria feito para Windows. Inicialmente, foi feita a divisão de funções básicas para o funcionamento do jogo. Ao final do trabalho, as funções desenvolvidas foram as seguintes:

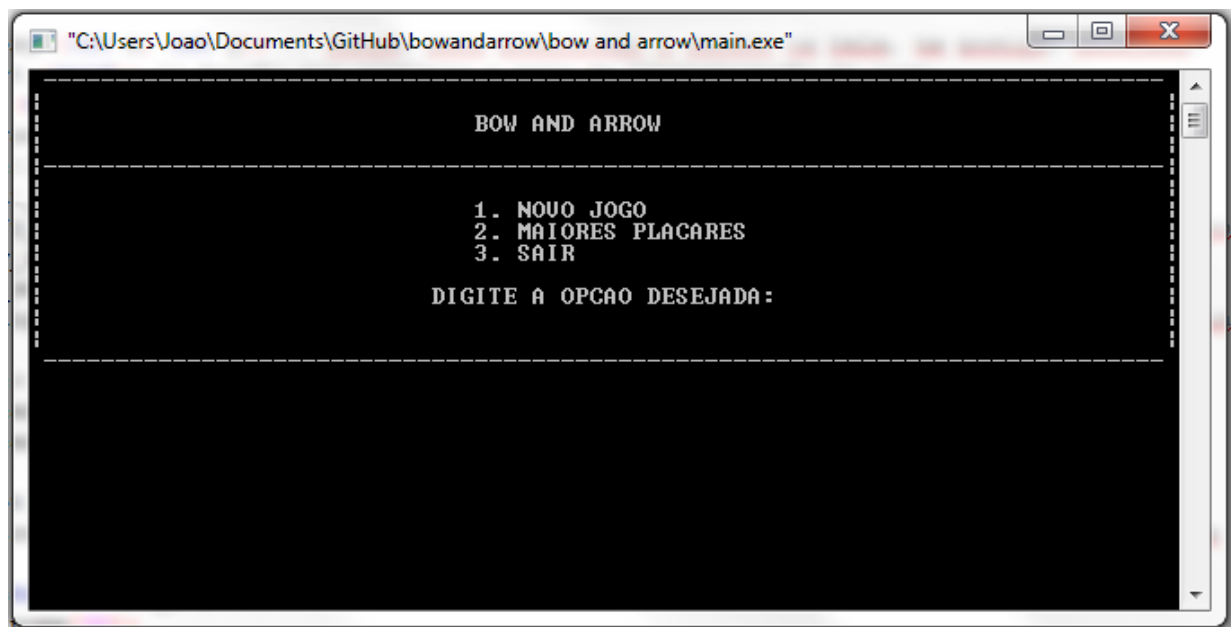
```
void gotoxy (int x, int y); // função para posicionar o cursor na tela, na posição desejada.
void start (void); // função que controla a tela de inicialização do jogo.
int menu (void); // função para inicializar o menu principal do jogo.
void newgame (void); // função para iniciar um novo jogo.
void level1 (void); // função para iniciar a primeira fase do jogo.
void print_borders (void); // função para imprimir as bordas do mapa atual.
int verify_score (void); // função que verifica se o score do jogador está entre os 5 primeiros.
void show_scores (void); // função que mostra na tela os 5 maiores placares do jogo.
void add_score (void); // função que adiciona o score do jogador ao arquivo highscores.bin.
int get_higher_score (void); // função que retorna o score mais alto do arquivo highscores.bin.

// funções de movimento.
void movearcher_up (void);
void movearcher_down (void);

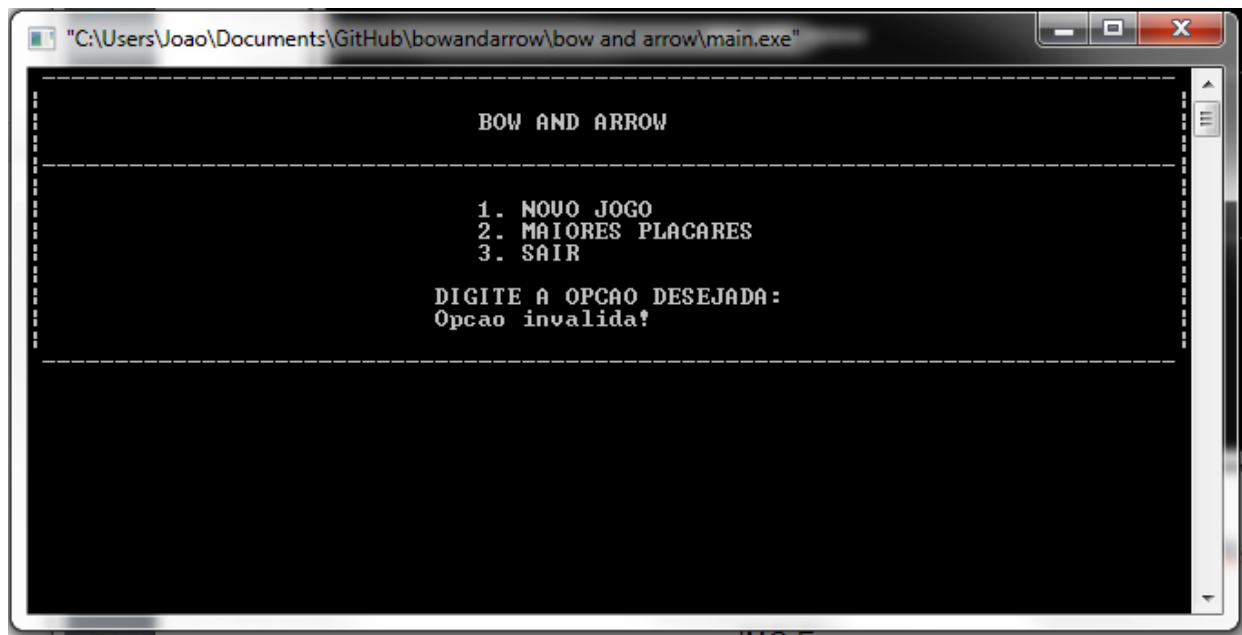
void shoot (void); // função de disparo de flecha.
void destroy_balloon (int arrow_x, int arrow_y); // função que trata a colisão entre a flecha e o balão.
```

Funções utilizadas no programa, com suas respectivas especificações.

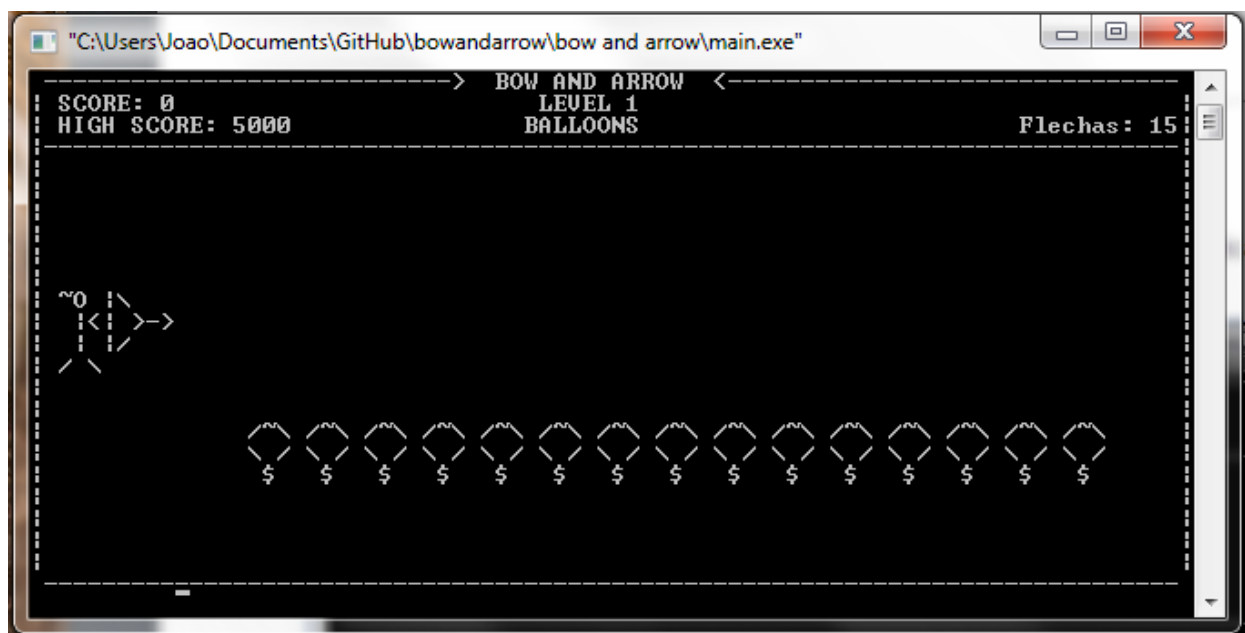
O jogo é iniciado através do menu principal, que imprime na tela as opções disponíveis para o usuário:



Caso o jogador escolha uma opção diferente das disponíveis, o programa indica que a entrada não é válida:



Quando a opção 1 é selecionada, o jogo inicia.



Os balões movem-se verticalmente, de baixo para cima, conforme especificado no enunciado do trabalho. O jogador possui de início 15 flechas, e com estas deve tentar acertar os 15 balões em movimento na tela. O arqueiro move-se para cima e para baixo (com as teclas W e S, respectivamente), e o disparo da flecha é feito ao apertar a tecla de barra de espaço.



Cada balão acertado adiciona em 100 pontos o score do jogador. O jogo encerra-se caso o jogador chegue ao final de suas flechas (“arrows == 0”, como feito no programa) ou caso os balões se esgotem (“balloons == 0”). Se uma destas condições se cumprir, o jogo encerra-se (“endgame = 1”). O controle do *status* do jogo é feito através da variável endgame, e portanto o “laço principal” é um “do {...} while (!endgame)”.

```
void level1 (void) {
    int cont1, cont2, cont3, archer_x, archer_y, x, y, arrow_x, arrow_y, flag_final_balao = 0;
    char archer[4][8], arrow[] = "-->", balloon[3][3], key;
    score = 0;
    arrows = 15;
    balloons = 15;
    shoot_flag = 0;
    endgame = 0;
}
```

Variáveis de controle, que são inicializadas a cada vez que o jogo se inicia.

Com exceção do arqueiro, todos os componentes do jogo (balões e flechas) estão na matriz principal `map[80][25]`. Essa matriz é constantemente atualizada e reimpressa na tela, dando a noção de movimento aos elementos contidos nela.

Quando alguma flecha é disparada, a *flag* `shoot_flag` é ativada, fazendo o controle para que apenas uma flecha possa ser disparada por vez. A função de disparo da flecha (`shoot`) apenas cria uma flecha na posição da ponta da flecha do arqueiro. O “movimento” da flecha, em si, é feito no *loop* principal. Ao encontrar a “ponta” da flecha, o programa a move para quatro colunas à direita e uma linha abaixo (por causa do movimento dos

balões, que imprimem a matriz com os valores uma linha abaixo do que estavam anteriormente).

```
for (cont1=4; cont1<23; cont1++) {
    for (cont2=10; cont2<78; cont2++) {
        if (map[cont1][cont2] == '>') {
            arrow_x = cont2;
            arrow_y = cont1;
        }
        printf("%c", map[cont1][cont2]);
    }
    y++;
    gotoxy(x, y);
}
if (shoot_flag) {
    if (map[arrow_y+1][arrow_x+4] == '/' || map[arrow_y+1][arrow_x+4] == '\\') {
        destroy_balloon(arrow_x+4, arrow_y);
    }
    for (cont1 = 1; cont1 <= 4; cont1++) {
        if (map[arrow_y+1][arrow_x+cont1] == '$')
            flag_final_balao = cont1;
    }
    map[arrow_y][arrow_x] = ' ';
    map[arrow_y][arrow_x-1] = ' ';
    map[arrow_y][arrow_x-2] = ' ';
    map[arrow_y+1][arrow_x+2] = '-';
    map[arrow_y+1][arrow_x+3] = '-';
    map[arrow_y+1][arrow_x+4] = '>';
    if (flag_final_balao) {
        map[arrow_y+1][arrow_x+flag_final_balao] = '$';

        flag_final_balao = 0;
        map[arrow_y+1][arrow_x+2] = ' ';
        map[arrow_y+1][arrow_x+4] = '-';
        map[arrow_y+1][arrow_x+5] = '-';
        map[arrow_y+1][arrow_x+6] = '>';
    }
    if (arrow_x+1 > 75) {
        map[arrow_y][arrow_x] = ' ';
        map[arrow_y][arrow_x-1] = ' ';
        map[arrow_y][arrow_x-2] = ' ';
        shoot_flag = 0;

        for (cont1 = 4; cont1 < 23; cont1++) {
            if (map[cont1][77] == '-')
                map[cont1][77] = ' ';
        }
        if (arrows == 0)
            endgame = 1;
    }
}
```

Trecho de código no qual a ponta da flecha é encontrada durante o loop que percorre a matriz principal; Abaixo, pode-se encontrar o caso especial em que a flecha “colide” com algum balão. Caso isso aconteça, a função destroy_balloon é chamada.

```

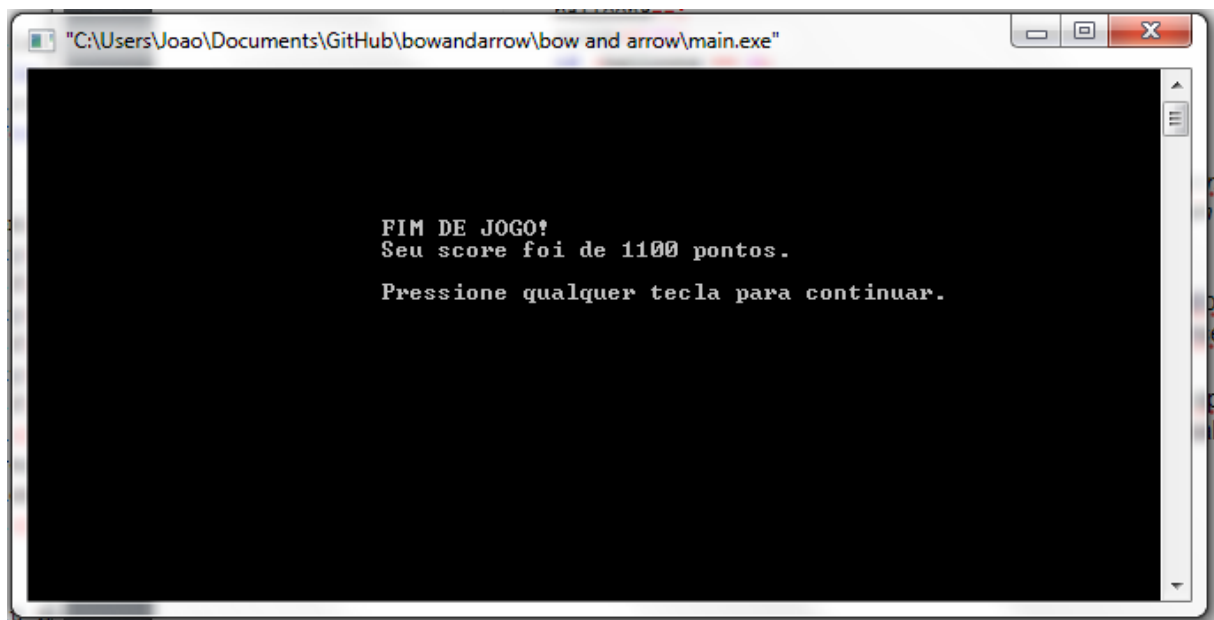
void destroy_balloon (int arrow_x, int arrow_y) {
    int cont1, cont2;
    if (map[arrow_y][arrow_x] == '\\')
        arrow_y--;
    map[arrow_y+1][arrow_x-2] = ' ';
    map[arrow_y+1][arrow_x-1] = ' ';
    map[arrow_y+1][arrow_x] = ' ';
    map[arrow_y+2][arrow_x-2] = ' ';
    map[arrow_y+2][arrow_x-1] = ' ';
    map[arrow_y+2][arrow_x] = ' ';
    map[arrow_y+3][arrow_x-2] = ' ';
    map[arrow_y+3][arrow_x-1] = ' ';
    map[arrow_y+3][arrow_x] = ' ';
    if (arrow_y+3 > 22)
        firstline[arrow_x-1] = ' ';
    balloons--;
    score+=100;
    if (balloons == 0)
        endgame = 1;
}

```

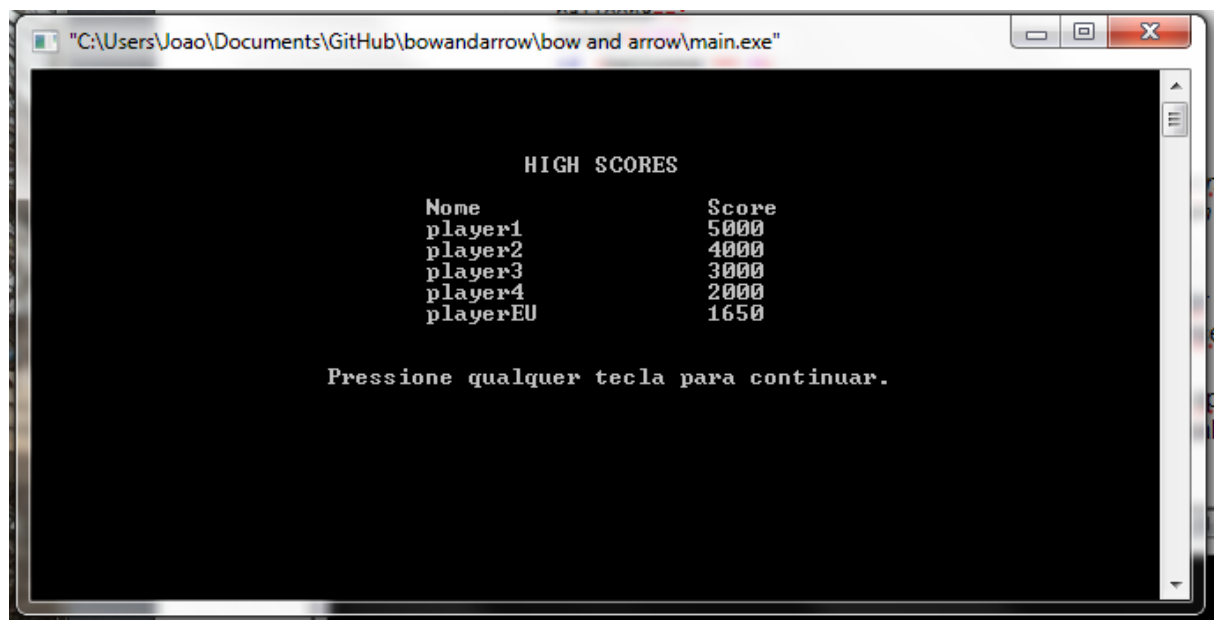
Função destroy_balloon, que, ao encontrar um balão na posição da flecha, preenche as 3 próximas colunas e linhas com espaços em branco, incrementa a pontuação em 100 e decrementa o número total de balões em 1.

Ao chegar no final do jogo, o programa verifica se a pontuação do jogador está entre as cinco primeiras (chamando a função verify_score), e, caso esteja, insere o score na posição correta (chamando a função add_score).

Com o fim do jogo, o programa retorna à tela inicial, mostrando as três opções disponíveis. Caso o usuário teclasse 2, o top 5 de maiores pontuações que está salvo no arquivo highscores.bin é exibido na tela.



Tela que aparece ao final do jogo.



Tela que exibe os cinco primeiros scores do arquivo highscores.bin

Ao apertar qualquer tecla, o programa volta ao menu principal. Ao apertar a tecla 3, o programa se encerra.