HACETTEPE UNIVERSITY COMPUTER SCİENCE ENGINEERING DEPARTMENT

BIL-341 SOFTWARE LABORATORY
4. EXPERIMENT REPORT

Name:
Serdar GÜL

Number:
20421689

Subject:
Operating Systems – PC Interrupt Handling

Development Environment:
cmd.exe, TCC.exe, PSPad

System:
Windows

Advisors:
R.A. Kaan KILIÇDOĞAN
Dr. Ahmet Burak CAN

Due Date:
28.11.2008

SOFTWARE USAGE:

Compilation, building and running:

This program has been developed via *tcc* compiler and window's *command.exe* application. So source files must be compiled via *tcc* in *MS-DOS* environment. So i've written two batch files to achieve this. First one is for builder and the other is for running the generated program via builder. This batches' contents are like these;

build.bat (path is tc's path, compiles and builds main.cpp, generates main.exe)

SET PATH=%PATH%;C:\TC\BIN
TCC.EXE main.cpp
---

run.bat (calls main.exe with parameter "1")

%comspec% /k main.exe 1

This is the way i used for testing rapidly. To compile and build from command line this command must be written (and TCC must be in the path variable of windows);

TCC main.cpp

And to run program a command like this;

main.exe [level-argument]

must be written.

Arguments:

Only parameter of the program is level argument. This argument must be in the interval of [1,3] With this argument, program will decide the difficulty level of the program. (Initial numbers of enemies and time frequency-so speed of the enemies)

After compilation, building and running; player (represented with character 'A') can be controlled via "arrow keys" to move related direction and "M" to fire enemies. Aim of game is killing all enemies by this fire (represented with character '-') without meeting any enemies (represented with character 'E') or any enemy missiles. (represented with character '*')

Movements:

Enemy misseles move three cells per unit time to the right without changing their row. Player misseles are similar to enemies' ones, but they move to the left.

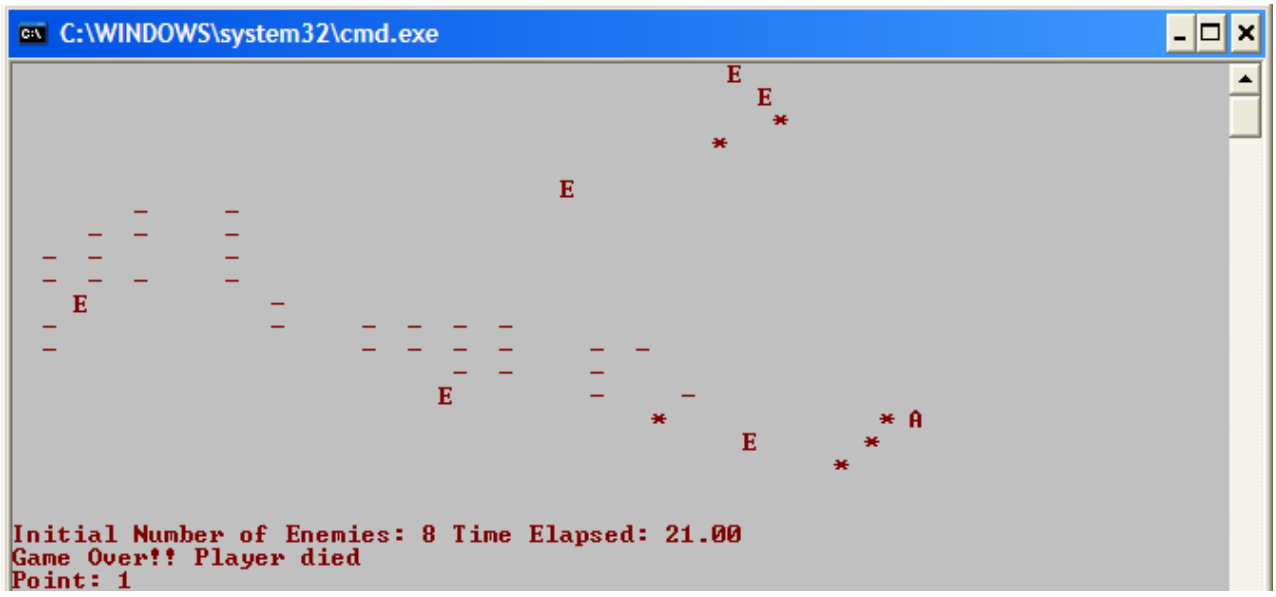
Enemies comes from left side of game board and move one cell right in a unit time. Beside this, they can move up/down or fire randomly. Or they only moves to the right. If player can not kill an enemy until it arrives rightest column of the board, new enemy is created in the left side of the board. But if any missele comes to the boundries of the board, they only dissappears.

Player starts game in the last row (at right) of the gameboard randomly. It can move all directions according to pressed key and it can fire with "m" character.

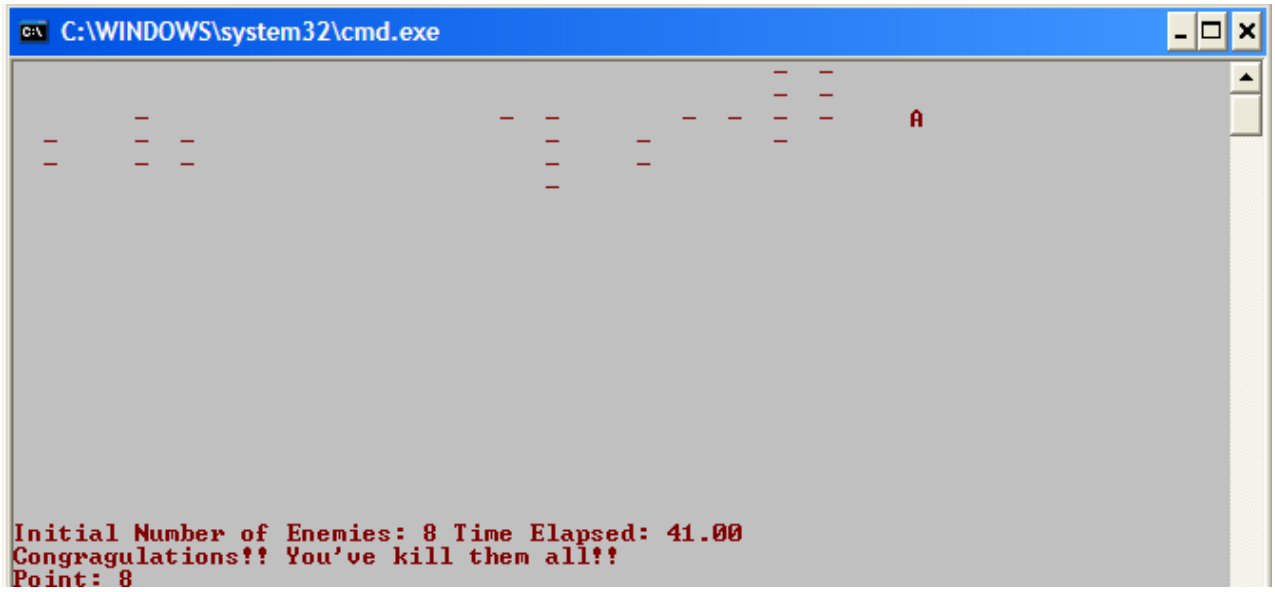
All enemies and player is put to the board randomly at the beginning of the game.

Terminations:

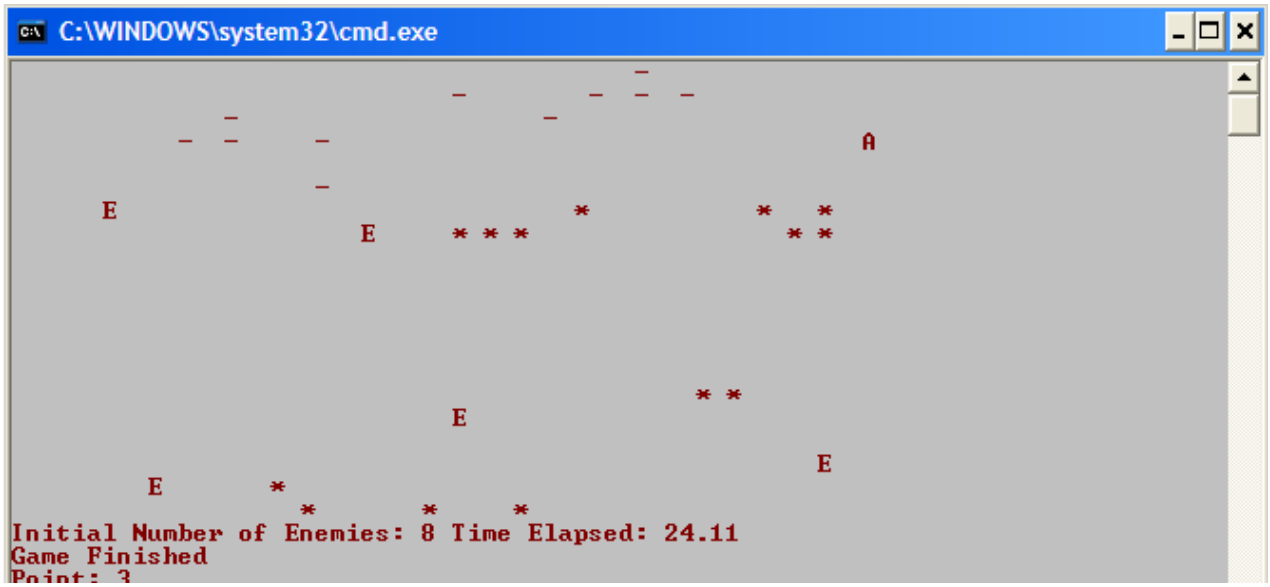
If player is hit by any enemies or any misseles, game ends. Point is calculated with formula like "*level*killedEnemy"* and it is printed on screen with elapsed time.



And if player kills all enemies, a message is printed on screen with point and elapsed time too.



If player presses "*ESC*" at any time of the game, program terminates and results are printed like the other terminations.



- If parameter number is wrong, program gives an error message like;

SOFTWARE DESIGN NOTES

PROBLEMs and SOLUTIONs

In this project we are suppose to develop a program with using interrupts and by handling them. This program is a popular game, "space impact". When doing this, i have met with some problems;

- Updating game screen

There is a function called "writeChar(char col, char row, char c)□which goes to (col,row) position of the board and writes character c into it. So when a movement happens, this function is called and game screen is updated. Beside this, my main data structure □oard□ must be updated too.

- Managing interrupts

There is two interrupts in this program, they are about "timer" and "keyboard". And they are added to interrupt vector and taken from this with functions "set_vect" and "get_vect". And there is a function called "game" which handles this interrupts and controls if program is finished for any reason. (ESC, killing all enemies, killing player..)

- Managing movements

Especially i thought about creation of enemies from left side of the board consistantly. It must not be a constant number of enemies, or infinite. So i decided to create from left side when an enemy has pass through the player to the right boundary of game screen.

Beside this, random movements of the items were another problem. Misseles must move 3 cells in a unit time and enemies must one cell right and randomly up or down. Only moving right has the biggest probability of them. Firing, moving up-down has same probability according to my randomizing.

- Managing game rules

I controlled all board cell by cell after each turn, according to player's keyboard presses and enemies' random actions. (fire or up-down) According to their positions, i control if player is dead or any enemy or missele is down.

MAIN DATA STRUCTURES

char board[ROW][COLUMN]; // -> keeps the gameboard's situation.

Definitions:

Keyboard Scan Codes:

```
#define MAKE_RIGHT  77
#define MAKE_LEFT   75
#define MAKE_UP     72
#define MAKE_DOWN   80
#define MAKE_M      50
#define ESC         0x01
```

Interrupt definitions:

```
#define KEYBOARD_INTERRUPT    0x09
#define TIMER_INTERRUPT       0x1C
```

Interrupt control variables

```
#define KEY_BUFFER      0x60
#define KEY_CONTROL     0x61
#define INT_CONTROL     0x20
```

ALGORITHM

Take arguments
  Take difficulty level (arg[1])
  Calculate initial enemy number (level * 8)

Initialize timer frequency

Initialize board
  Find players "row" randomly from 20 (column is 59)
  Find enemies places randomly (according to enemy number)

Set Interrupts
  Set timer interrupt
    For each time interval
      Print Board
      Move enemies randomly
        Move one cell right
          If random result is up, move one cell up
            If there is player on its way, set end game, set player killed
          If random result is down, move one cell up
            If there is player on its way, set end game, set player killed
          If random result is fire, fire missele to the right
            If there is player on its way, set end game, set player killed
      Move enemy misseles to the right, 3 cells
        If there is player on its way, set end game, set player killed

Move player misseles to the left, 3 cells
If there is an enemy on its way, dissappear both

Set keyboard interrupt
If key pressed is ESC
Set end game
If key pressed is an arrow key
Move player according to direction
Control moving cell
If there is an enemy or missele
Set end game, set player killed

If key pressed is 'M' fire to the left from player
If there is an enemy on its way, dissappear both

Wait for end game
Print ending message
Print point and time elapsed

Update interrupt vector

EXECUTION FLOW BETWEEN SUBPROGRAMS

Board Generation
----
void initializeBoard(int eNum); // Generate board randomly according
void randomBoard(int eNum); // to eNum

Movements
----
void moveEnemies(); // move enemies
void moveMisseles(); // move both enemy and player misseles
void move(int dir); // move player according to direction(dir)
void fire(); // fire from player

Run Time
---
void game(); // starts the game and handles the interrupts
void printBoard(); // prints boards situation

_asm functions:
---
void writeChar(char col, char row, char c); // moves corsor to (col,row) to change its content
void interrupt Timer(); // timer interrupt
void interrupt KeyBoard(); // keyboard interrupt
void interrupt (*get_vect(int intNumber))(); //function instead of _dos_getvect
void set_vect(int intNumber, void interrupt(*interruptPointer)()); // function instead of _dos_setvect

SOFTWARE TESTING NOTES

SOFTWARE RELIABILITY AND BUGS

- Player and enemies must be represented with three characters, but in my program they covers only one cell-one character.

- Sometimes enemies are dead by their own missiles.

SOFTWARE EXTENDIBILITY AND UPRADIBILITY

- Program can be made more visual by fixing bugs.

- Program can be seperated into levels, so for example after finishing 20 enemies, this level finishes, difficulty increases and another level starts like the original space impact.

- New enemy and misseles types can be created. (faster, bigger..)

RESOURCES

- Bilgisayar İşletim Sistemleri, Ali SAATÇİ
http://www.frontiernet.net/~fys/demoisr.htm
http://en.wikipedia.org/wiki/Scan_codes
http://en.wikipedia.org/wiki/Interrupt