Project
Predator & Prey Game

By Students

Name :Sameh mohamed mohamed salama
Number ID : 959

Name : Ibrahim Bassiouni ibrahim beltagy
ID Number : 642

Introduction :

This report will discuss the main parts in our project, the project is a 2D board game and the suitable platform for the game is Windows

The Game design :

the main classes in the game is the " GameSystem " class and the " Creature " class

then two classes are inherited from the "Creature " class and they are "Predator " and " Prey " and from the predator class inherited the Snack class and the scorpion class , and from the prey class inherited the Ant class , the worm class and the bug class .

Data Structures :

In this project we are using 2D array of characters, in this array we are storing the the place of every creature and the type of the creature(predator or prey)

Libraries Used:

<Allegro.h>

tutorials that been watched and learn:

http://www.loomsoft.net/resources/alltut/alltut_index.htm

http://www.cppgameprogramming.com/cgi/nav.cgi?page=allegio

and so many others.....

pseudo code for the all project :

class GameSystem : \\this class intializes allegro and contains the menus and the background and the score and the turn

the private varibles:

int score;

int turn;

SAMPLE *Song;

char map[14][9];

the public ones:

void set(int x,int y,char c)

setter function that set the x , y and c to their values.

char get(int indexx,int indexy)

getter function that return the character that stored in the

GameSystem()

all the following functions will begin allegro video mode

```
allegro_init();
install_keyboard();
set_color_depth(24);
set_gfx_mode( GFX_AUTODETECT_WINDOWED, 800, 600, 0, 0);
install_sound(DIGI_AUTODETECT, MIDI_AUTODETECT, 0);
```

the pseudo-random number generator that generate a different succession of results in the subsequent

calls to rand function.

srand(time(NULL));

this function initialize the main menu image and the choices in it(1- new game , 2 About , 3- Exit)

void menuinit();

this functions load the music that runs with the begining of the game

Song = load_sample("BGM.wav");

play_sample(Song, 50, 128,1000, true);

construct the intger turn and score to zero

```
turn=0;
score=0;
```

the following FOR loop sets the all elements of the array "map" to character f means free

```
for (int i=0;i<9;i++)
{ for (int j=0;j<14;j++)
   { set(i,j,'f');  //resets array to free
   }}}
```

this fuction will declare the ' about ' image and the key that return the screen to the main menu

void aboutit()

this function will begin the game after choosing the ' New Game ' option .

void bginit()

this two functions will show the score on the screen and update it every time you eat a prey

void showscore()

void updatescore()

this two functions will show the number of the current turn on the screen and update it every time you move your predator

void showturn()

void updateturn()

this function will show the choosing screen between the predators (Scorpion or Snack) and let you decide which one you choose by pressing 1 , 2 keys

int choose()

this function will show the message "You Win " if your score get more than 6 and will show "GameOver " message if the number of turns get more than 50 turn

void WinLose()

class Creature :

the protected varibles:

```
int x;
int y;
int tempx;
int tempy;
int moveCounter;
```

int dead;

BITMAP *buffer;

BITMAP *bg;

the public ones:

this two functions will return the two coordinates (that are in the screen) of the object

int getx()

int gety()

the following function is the constructor which will set the variables x ,y and movecounter to zero and make tempx ,tempy equal them ,then it initialize the buffer and the background image of the game

Creature()

setter function the set the intger named dead to the value x and the getter function after it

void setdead(int x)

int getdead()

this two functions will generate the x coordinate (the range from 50 to 700) and the y coordinate(the range from 50 to 500)

int generate_x()

int generate_y()

this function will generate the number which control the directions(the range from 1 to 4)

int generate_random()

this boolean functio will check if the place generated is suitable for the creature to move or not

bool check(GameSystem &g , int a , int b , char c)

this function will show the entire buffer to the screen."in this game we will draw every thing to the buffer then draw the buffer to the screen once to prevent the flickering "

void show()

this two functions will control the collisions between the creatures

Rectangle RegionCoordinates()

bool IsCollision(Rectangle A, Rectangle B)

void putrandom()

void moverandom(GameSystem &g)

class Predator:

protected variables

SAMPLE *EAT ;

pointer to the WAV file (the sound to eat)

Public ones:

in this constructor we will just load the Wav file

Predator()

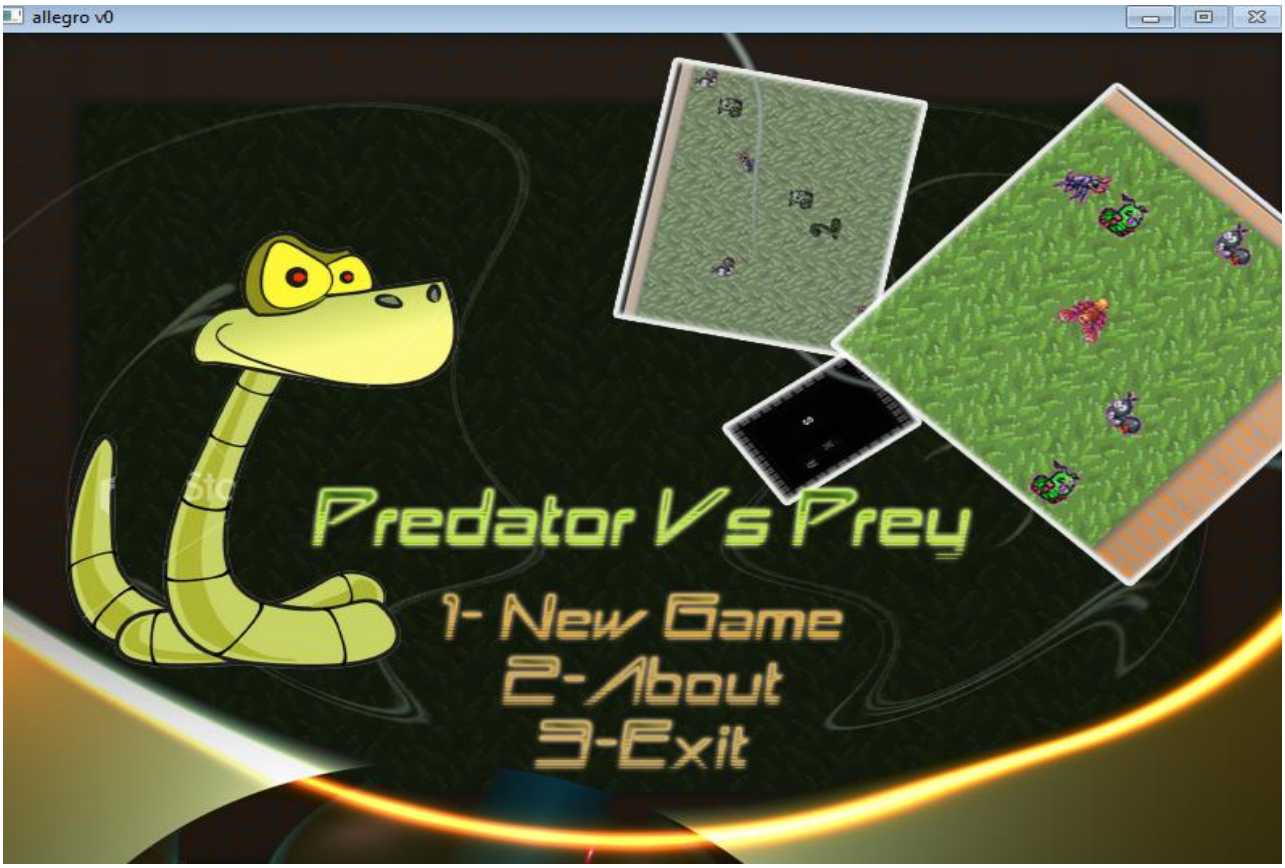
This function will determine if the predator is alive or dead(after 12 move without eating)

void alive(GameSystem &g,char c)

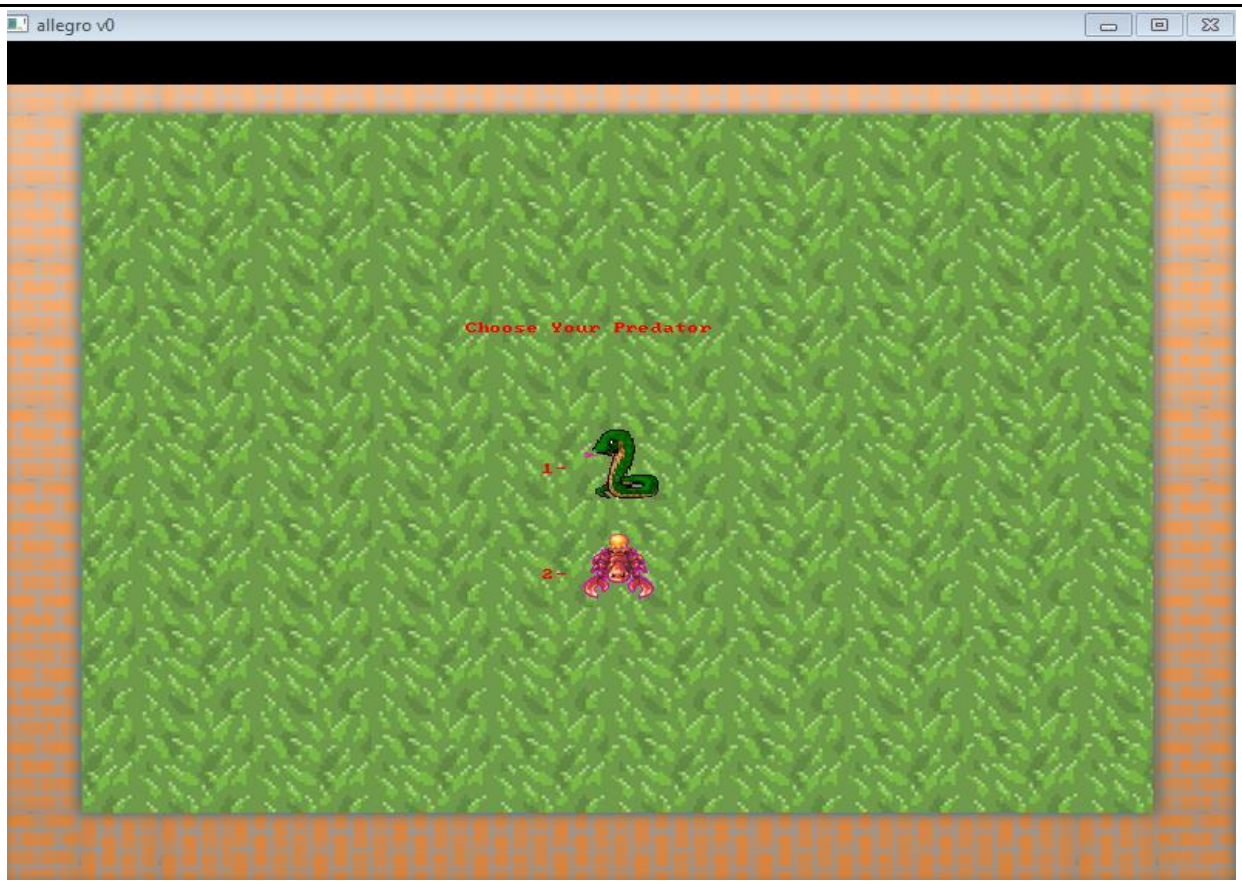
this function will make movecounter to zero
void resetCounter()

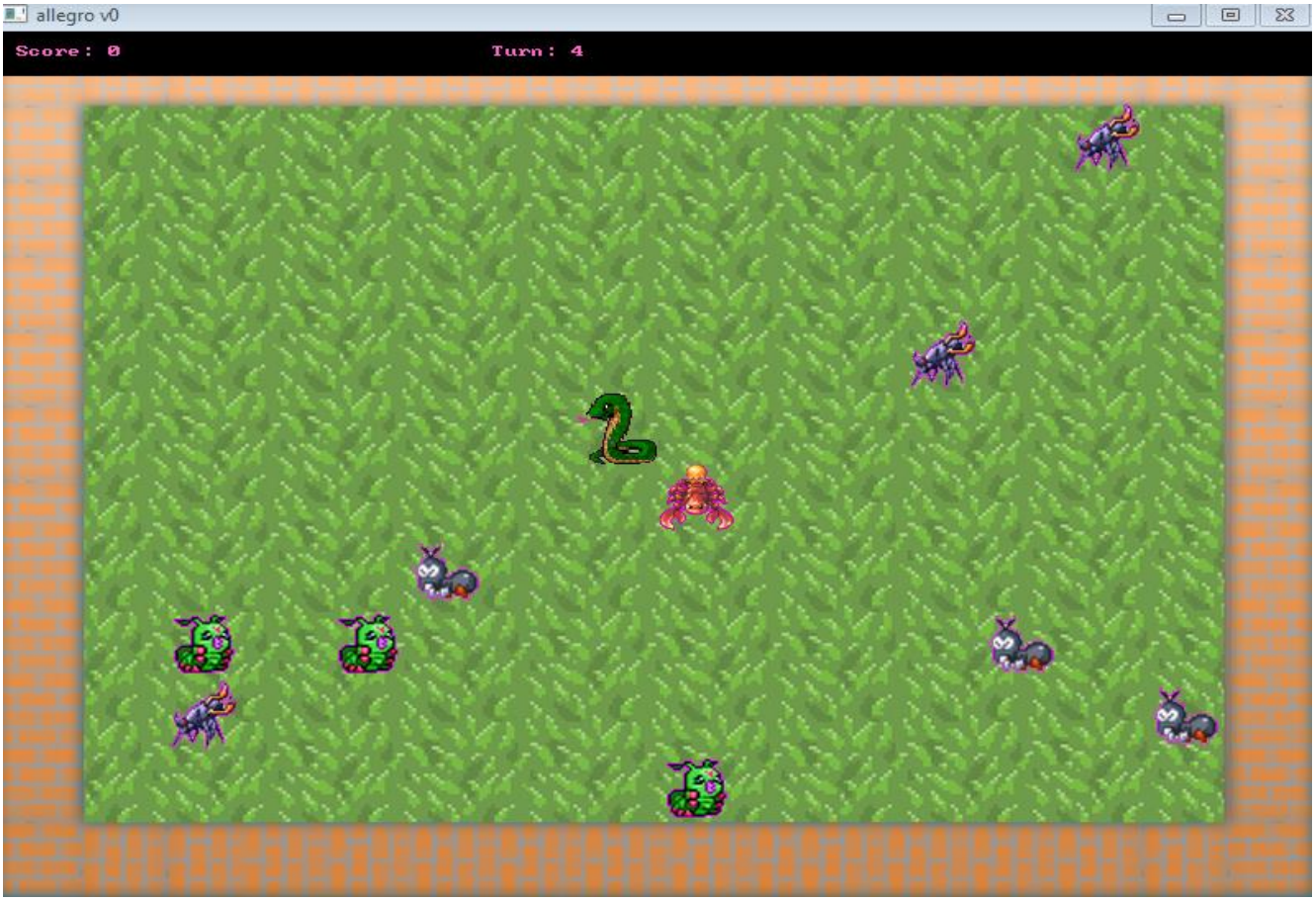
this function will delete the prey object when its eaten
void eat(Prey &b,GameSystem &g)
this function will control the moving of the selected predator
void movep(GameSystem &g)

screen shots from the game :
Main Menu



The choosing step :





Playing...!!

<u>Member contribution:</u>

-the parent classes were made with both of us together

-the member functions of the classes were written by either of us by himself

-in breed function we both worked on it together ,but unfortunately we didn't have enough time to make it work correctly .

The UML :