**Computer Science**

**Snake Game**

**Project Report By**

**Poorab Gangwani**

**Group Members:**

Name Poorab Gangwani

Table of Contents

[1. Introduction 3](#_Toc511892420)

[2. Scope 3](#_Toc511892421)

[3. Module Description 3](#_Toc511892423)

[4. Hardware/Software Requirements 3](#_Toc511892424)

[4.1 Libraries , IDE 3](#_Toc511892426)

[5. Screen Shots 3](#_Toc511892427)

# Introduction

**My name is Poorab Gangwani and I am a student of DHA Suffa University studing Computer Science (2nd semester) and I have constructed an OOP(Object Oriented Programming) program which mimics the behaviour and functions of a simple snake game and have built on C++ programming language. Along with my primary subject which is a snake game ,I have coded additional programs and minor programs in OOP such as a guessing game which has also been created with the basic OOP functions and concepts and secondly I have developed a game that mimics the functionality of a pong game which is comprised of C++ and Objected Oriented Programming concepts which are the following**

1. **Enum**
2. **Dynamic Memory Allocation**
3. **Operator Overloading**

# Scope

**For the implementation of this program I made use of the many object oriented programming concepts and techniques such as this pointers, Mutators and Getters ,friend functions ,constructors, copy constructors, Separate Header file implementation, Constant functions, Constant Integer Variables, Inheritance and additional C++ features are also implemented such as switch case, system Operations for effects and char-codes.**

**Additional Functions that were applied in the making of auxiliary features are Enum , Dynamic Memory Allocation , Operator Overloading and Bool Datatype.**

# 3. Module Description

#include"border.h"

#ifndef TARGET\_H

#define TARGET\_H

class target{

private:

int x;

int y;

public:

target();

int getx()const;

int gety()const;

void myfriend(border& b);

void getcoordinates();

friend class border;

};

#endif

#include"target.h"

#ifndef SNAKE\_H

#define SNAKE\_H

class snake{

private:

int a;

int b;

public:

snake();

void place(border& b);

};

#endif

#include"snake.h"

#ifndef DIRECTION\_H

#define DIRECTION\_H

class direction{

private:

char dir;

public:

void move();

void movement(border& bor);

};

#endif

#include<iostream>

#include<conio.h>

#include<stdlib.h>

#ifndef BORDER\_H

#define BORDER\_H

class border{

private:

private:

int x;

int y;

int t1;

int t2;

int s1;

int s2;

public:

border();

border(const int,const int);

void settarget(const int,const int);

void setsnake(const int,const int);

int getx()const;

int gety()const;

void setx(const int);

void sety(const int);

void getcoordinates();

void create();

friend class target;

friend class snake;

friend class direction;

};

#endif

***Guess Game Code***

#include<iostream>

#include<stdlib.h>

#ifndef GUESS1\_H

#define GUESS1\_H

class guess{

private:

char word1[5]={'t','i','g','e','r'};

char random[15]={'a','g','z','d','c','t','i','r','e','f','k','s','x','p','q'};

char guess\_word[5]={' ',' ',' ',' ',' '};

int guess\_count;

int help\_count;

char answer;

bool decide;

int help\_number;

int win;

public:

guess();

void draw();

void select();

void Verify();

void help\_end();

void guess\_end();

void help();

void run();

};

#endif

#include"guess1.h"

guess::guess()

{

this->guess\_count=7;

this->decide=false;

this->help\_count=3;

}

void guess::draw()

{

std::cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<std::endl;

std::cout<<"| HINTS |"<<std::endl;

std::cout<<"|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|"<<std::endl;

std::cout<<"| Part of the largest extant cat |"<<std::endl;

std::cout<<"| species |"<<std::endl;

std::cout<<"|-----------------------------------|"<<std::endl;

std::cout<<"| Recognisable for it's Stripes |"<<std::endl;

std::cout<<"|-----------------------------------|"<<std::endl;

std::cout<<"| Preys on Ungulates |"<<std::endl;

std::cout<<"|-----------------------------------|"<<std::endl;

std::cout<<"| powerful Swimmers |"<<std::endl;

std::cout<<"|-----------------------------------|"<<std::endl;

std::cout<<"|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|"<<std::endl;

std::cout<<"|"<<random[0]<<" "<<random[1]<<" "<<random[2]<<" "<<random[3]<<" "<<random[4]<<" "<<random[5]<<" "<<random[6]<<" "<<random[7]<<" "<<random[8]<<" "<<random[9]<<" "<<random[10]<<" "<<random[11]<<" "<<random[12]<<" "<<random[13]<<" "<<random[14]<<" |"<<std::endl;

std::cout<<"|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|"<<std::endl;

std::cout<<"| | | | | |"<<std::endl;

std::cout<<"| "<<guess\_word[0]<<" | "<<guess\_word[1]<<" | "<<guess\_word[2]<<" | "<<guess\_word[3]<<" | "<<guess\_word[4]<<" |"<<std::endl;

std::cout<<"|\_\_\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_\_\_\_\_|"<<std::endl;

}

void guess::select()

{

std::cout<<"Select a Character("<<this->guess\_count<<") or press 'q' for help guess("<<this->help\_count<<"):"<<std::endl;

std::cin>>this->answer;

}

void guess::Verify()

{

if(this->answer=='t')

{

guess\_word[0]='t';

}

else if(this->answer=='i')

{

guess\_word[1]='i';

}

else if(this->answer=='g')

{

guess\_word[2]='g';

}

else if(this->answer=='e')

{

guess\_word[3]='e';

}

else if(this->answer=='r')

{

guess\_word[4]='r';

}

else

{

this->guess\_count--;

}

}

void guess::guess\_end()

{

if(this->guess\_count==0)

{

this->decide=true;

}

else if(guess\_word[0]=='t' && guess\_word[1]=='i' && guess\_word[2]=='g' && guess\_word[3]=='e' && guess\_word[4]=='r')

{

this->decide=true;

}

}

void guess::help()

{

if(this->answer=='q' && this->help\_count>0)

{

if(guess\_word[0]==' ')

{

guess\_word[0]='t';0

}

else if(guess\_word[1]==' ')

{

guess\_word[1]='i';

}

else if(guess\_word[2]==' ')

{

guess\_word[2]='g';

}

else if(guess\_word[3]==' ')

{

guess\_word[3]='e';

}

else if(guess\_word[4]==' ')

{

guess\_word[4]='r';

}

this->help\_count--;

}

if(guess\_word[0]=='t' && guess\_word[1]=='i' && guess\_word[2]=='g' && guess\_word[3]=='e' && guess\_word[4]=='r')

{

this->decide=true;

this->win=1;

}

}

void guess::help\_end()

{

if(this->help\_count==0)

{

std::cout<<"No More Help Guess"<<std::endl;

}

}

void guess::run()

{

do

{

draw();

select();

Verify();

guess\_end();

help();

system("cls");

}

while(this->decide==false);

if(this->win==1)

{

std::cout<<"You guessed the word"<<std::endl;

std::cout<<std::endl<<std::endl;

}

}

***Pong Game Code***

#include<iostream>

#include<stdlib.h>

#include<conio.h>

enum eDir{

stop=0,

left=1,

upleft=2,

downleft=3,

right=4,

upright=5,

downright=6,

};

#ifndef BALL\_H

#define BALL\_H

class ball{

private:

int x;

int y;

int originalX;

int originalY;

eDir direction;

public:

ball(const int,const int);

void reset();

void changeDirection(const eDir);

int getX()const;

int getY()const;

eDir getDirection()const;

void randomDirection();

void move();

friend std::ostream& operator <<(std::ostream&,const ball&);

};

#endif

#include"ball.h"

ball::ball(const int x,const int y)

{

this->originalX=x;

this->originalY=y;

this->x=x;

this->y=y;

this->direction=stop;

}

void ball::reset()

{

this->x=this->originalX;

this->y=this->originalY;

this->direction=stop;

}

void ball::changeDirection(const eDir direction)

{

this->direction=direction;

}

int ball::getX()const

{

return this->x;

}

int ball::getY()const

{

return this->y;

}

eDir ball::getDirection()const

{

return this->direction;

}

void ball::randomDirection()

{

this->direction=(eDir)((rand()%6)+1);

}

void ball::move()

{

switch(this->direction)

{

case left:x--;

break;

case right:x++;

break;

case upleft:

x--;

y--;

break;

case upright:

y--;

x++;

break;

case downleft:

y++;

x--;

break;

case downright:

y++;

x++;

}

}

std::ostream& operator <<(std::ostream& out,const ball& b)

{

out<<"["<<b.x<<","<<b.y<<"]"<<"["<<b.direction<<"]"<<std::endl;

return out;

}

#include<iostream>

#ifndef PADDLE\_H

#define PADDLE\_H

class paddle{

private:

int x;

int y;

int a;

int originalX;

int originalY;

public:

paddle();

paddle(const int,const int);

int getA()const;

void setA(const int);

void reset();

int getX()const;

int getY()const;

void movement();

friend std::ostream& operator <<(std::ostream&,paddle&);

};

#endif

paddle::paddle()

{

this->x=0;

this->y=0;

this->originalX=0;

this->originalY=0;

this->a=0;

}

paddle::paddle(const int x,const int y)

{

this->x=x;

this->y=y;

this->originalX=x;

this->originalY=y;

}

int paddle::getA()const

{

return this->a;

}

void paddle::setA(const int a)

{

this->a=a;

}

void paddle::reset()

{

this->x=this->originalX;

this->y=this->originalY;

}

int paddle::getX()const

{

return this->x;

}

int paddle::getY()const

{

return this->y;

}

#include"ball.h"

#include"paddle.h"

#ifndef GAME\_H

#define GAME\_H

class game{

private:

int height;

int width;

int score1;

int score2;

char up1;

char up2;

char down1;

char down2;

bool status;

ball \*b;

paddle \*player1;

paddle \*player2;

public:

game(const int width,const int height);

~game();

void scoreUp(const paddle \*p);

void draw();

void input();

void logic();

void ball\_Move();

void run();

};

#endif

#include"game.h"

game::game(const int width,const int height)

{

this->height=height;

this->width=width;

up1='o';

down1='l';

score1=0;

score2=0;

this->status=false;

b=new ball(width/2,height/2);

player1=new paddle(1,(height/2)-3);

}

game::~game()

{

delete b;

delete player1;

delete player2;

}

void game::scoreUp(const paddle \*p)

{

if(p==player1)

{

score1++;

}

else if(p==player2)

{

score2++;

}

b->reset();

player1->reset();

player2->reset();

}

# 4. Hardware/Software Requirement

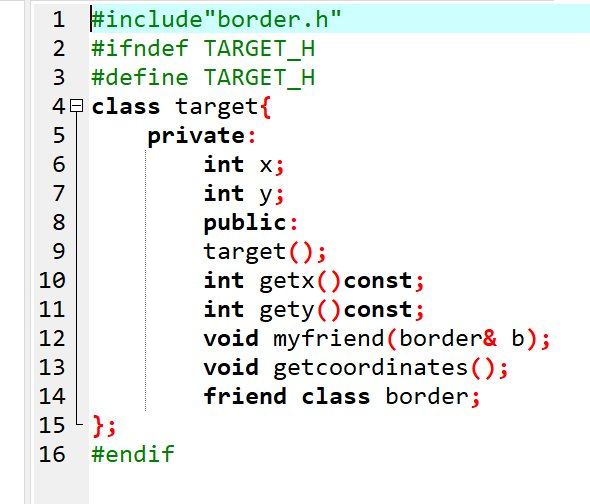
## 4.1 Libraries , IDE

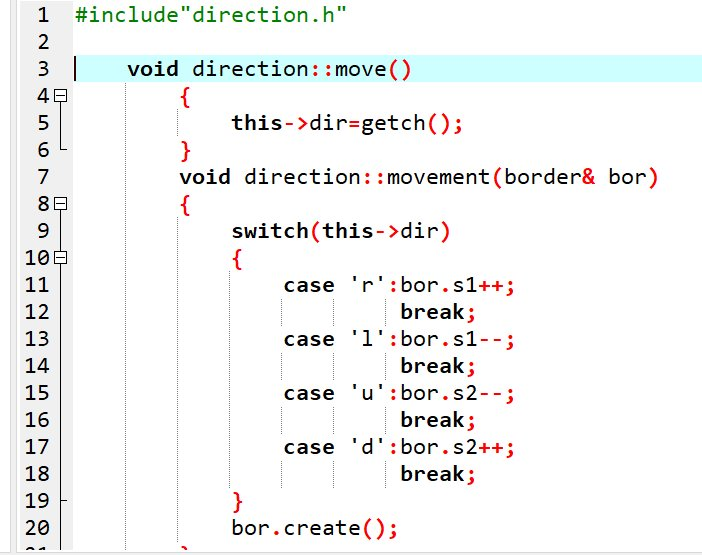
**Built-In functions from multiple different libraries were implemented into the program to accommodate the effective working of some of the features of the program.**

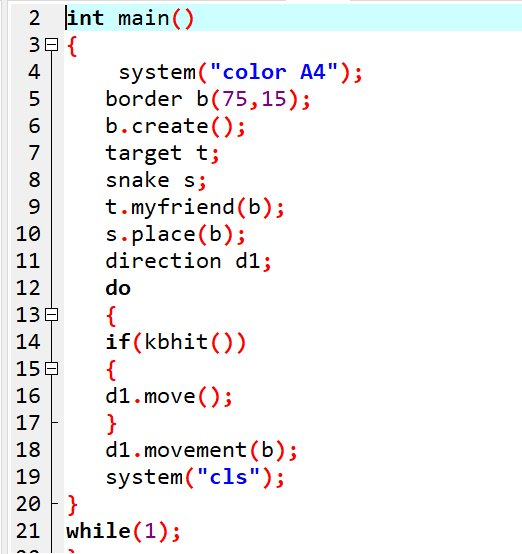
1. **iostream : used for system operations such as clearing screen , background coloring and multitude of functions.**
2. **Conio.h : this header file has been used for working of functions such as detection of key pressing for input ( kbhit() ) and for the taking of user-inputted value during execution (getch()).**
3. **Stdlib.h : this Function library has been used to provide function for the random selection of coordinates for the target of snake using rand() and srand() functions. The abort() function has also been used to end the program when needed.**

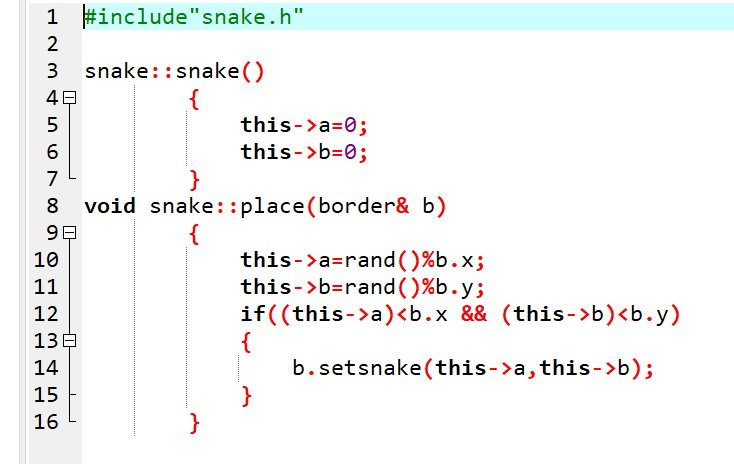
# 5. Screen Shots

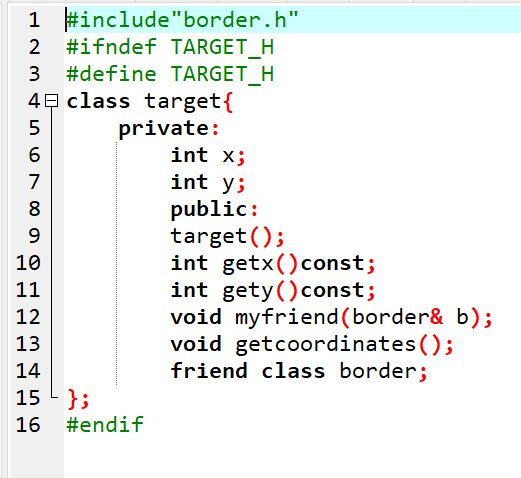
***Snake Game***





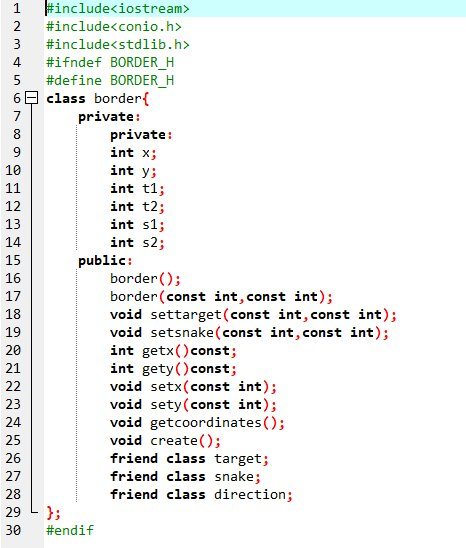








***Guess Game***



# 



***Pong Game***

