

Bangabandhu Sheikh Mujibur Rahman Science and Technology University, Gopalganj-8100.



Box Shooter Game

Submitted to the Department of Computer Science and Engineering Bangabandhu Sheikh Mujibur Rahman Science and Technology University in partial fulfillment to the requirements for the degree of B.Sc. Engineering.

**Course Code : CSE178
30 November, 2020**

By:

Tushar Sarkar
ID:18CSE035

Supervised By:

Md. Nesarul Hoque
Assistant Professor

**Department of Computer Science & Engineering
Bangabandhu Sheikh Mujibur Rahman Science and Technology University**

ABSTRACT

A box shooter is one of the simplest physics simulations yet provides a novice graphics programmer with a host of useful experience. The student creates a ball that bounces off the walls of a box, adds a bat, then gamifies the whole experience into a simple block-out game. The assignment is designed as the first significant piece of programming on a 2D computer graphics course. It is designed to be accessible to students who have taken an introductory programming course and who have physics and algebra to the level of a high-school graduate.

This project is developed using C, C++ language.

Windows comes with built in box shooter that you can select. If you purchase your computer with the operating system already installed, a box shooter may have been selected for you. You can set the box shooter using the task bar by selecting start, then setting, then Control Panel, then Display, and the box shooter, which lets you select one of the provided box shooter. Our analysis was focused primarily on the theoretical and practical subject matter that is taught in game programming classes and secondarily, in how game based programming projects and assignment can help students learn core computer science concept and coding skills.

ACKNOWLEDGEMENTS

Firstly I would like to express my deprest sense of gratitude to my reverend teacher and supervisors **MD Nesarul Hoque**, Assistant Professor of Department of Computer Science & Engineering (CSE), Bangabandhu Sheikh Mujibur Rahman Science and Technology University. And for her untiring guidance, constant supervision, enthusiastic encouragement, sagacious advice and an effective surveillance throughout the entire period of my project and preparation of the manuscript. I greatly sy thank you. Wish to express my heart full thanks to all of my honorable teachers of the department of Computer Science and Engineering(CSE), Bangabandhu Sheikh Mujibur Rahman Science and Technology University.

Secondly I would like to thank my department senior brother **MD Fahim Hossain** who helped me a lot in finalizing this project within a limited time frame.

PROJECT ON

“Box Shooter Game”

DECLARATION

I, **Tushar Sarkar** , **ID : 18CSE035** declare that the project consideration of degree of Bachelor of Computer Science & Engineering (CSE) embodies our own work with suggestion received during the work, which have been suitably acknowledge.

Tushar Sarkar

ID:18CSE035

.....Tushar..Sarkar.....

Signature

.....30/11/2020.....

Date

APPROVAL

I certify that this project “Box Shooter” is the original work of the above named candidate and has been done under my supervision. To the best of my knowledge and belief, this work which embodies the work of candidates themselves, has been duly completed, fulfills the requirement of the ordinance relating to the first year of Bangabandhu Sheikh Mujibur Rahman Science and Technology University and is up to standard in respect of content, presentation and language for being referred to the examiner. The work has never been submitted anywhere. It's only submitted to Bangabandhu Sheikh Mujibur Rahman Science and Technology University.

Project Supervisor :

MD Nesarul Hoque

Assistant Professor

Department of Computer Science and Engineering

Bangabandhu Sheikh Mujibur Rahman Science and Technology University, Gopalganj.

Signature:.....

Date:.....

TABLE OF CONTENT

Title.....	1
Abstract	2
Acknowledgement	3
Chapter 1 :	
1.1 Introduction.....	8
1.2 Aim of the project	8
1.3 Main purpose	9
1.4 What to expect.....	9
1.5 Features of this project	10
Chapter 2 :	
2.1 Some Snapshot of my project.....	11
Chapter 3 :	
3.1 Implementation.....	14
3.1.1 Below is a sample of the graphics.....	15
3.2 Result & Evualations.....	16
Chapter 4 :	
4.1 Future Look	17
4.2 Conclusion.....	18
Reference	18

CHAPTER-1

1.1 Introduction:

Game Programming has been a popular topic in computer science education for many years. Well, you all have heard about the term **Game**. A Game can be defined as an activity enrolled for amusement as a prepared form of play that basically undertakes for enjoyment and/or as an educational tool.

According to **Game Theory**, which is a study of mathematical models, while playing a game, there is either a win or a loss or a tie among the candidates participating in the game and that is based on the actions performed or the decisions taken during the gameplay.

The key components almost every game is having are its objectives, regulations and conventions, challenges, and interaction. While a player plays any game, it usually involves mental or physical stimulation and frequently both. This eventually boosts up human senses and reacting and thinking capability also (which we will discuss in the **Advantages of Game Engineering and playing** part). Many games help develop practical skills (like simulation games), while other serve as a form of doing exercises (VR and X-box oriented Games), while some other games help developing educational or psychological potential.

1.2 Aim of the project:

The goal of this project is to investigate the effects of timing interval and step size on perceived animation smoothness. You will write a simple JavaScript program that will move an object around on a Web page. This project will help you create more advanced Game that could have any type of animation.

1.3 Main Purpose:

Let's start with the basics. We have a circle on the screen. We want it to move in straight lines at a constant speed. We want the screen edges to act as “walls,” reflecting the circle's motion. To make all this happen, there are a few things we need to keep track of. The circle (our low-tech ball) has a position and a velocity. The screen has a width and height. We'll ignore complications like friction, elasticity, and ball diameter for now (but reserve the right to add them later, if we feel ambitious).

1.4 What to expect

The hopes are given below:-

- You will first be given a frame in the shape of a quadrangle with 50 boxes at the top and a stage and a ball at the bottom.
- Now clicking on space will start the game, clicking on the left key of the special keyboard will move 10 houses to the left and will continue to move 90 degrees and hit the box. If you can eat one box, it will be 5 marks.
- The game will end when the ball goes off the stage.
- If you can finish the whole game you will be successful and you will be congratulated.

1.5 Features of this project

The features are given below:-

- The frame for this game will be $1200 * 75$ in size.
- The 50 boxes at the top will each be $100 * 20$ in size.
- The stage below will be $200 * 10$ in size.
- The radius of the ball on the stage will be 10.
- The song will play in the background as soon as the game starts. Songs can be started and stopped at will.
- There are 5 pictures set in the background of the frame, you can change the picture as you wish.
- If you play, you will be notified of the current score and signal and if you finish the game, you will be congratulated and the signal will be there.

CHAPTER-2

2.1 Some Snapshots of my project:

First Look of Program-

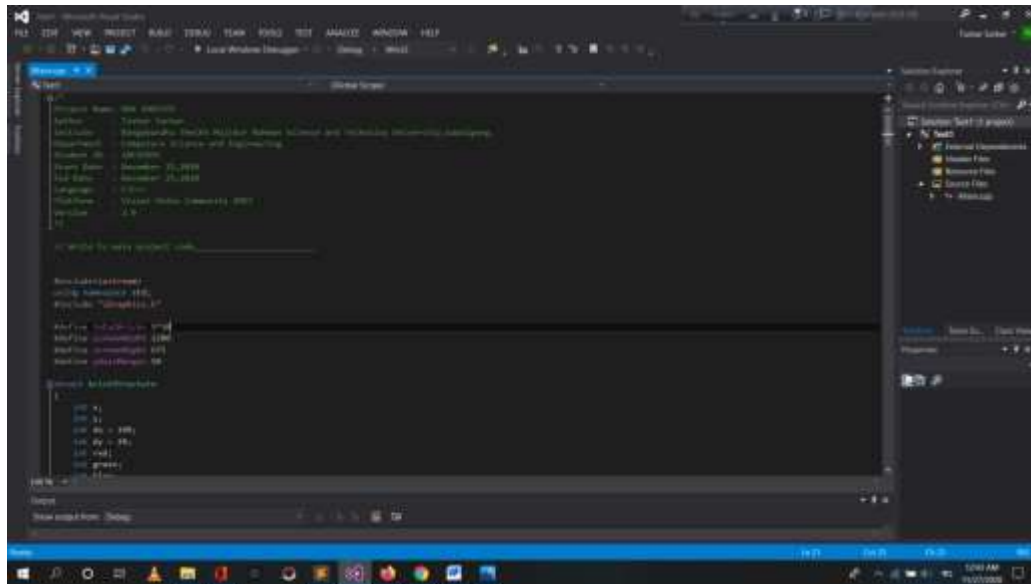


Figure-2.1.1

First Look when Program Run-

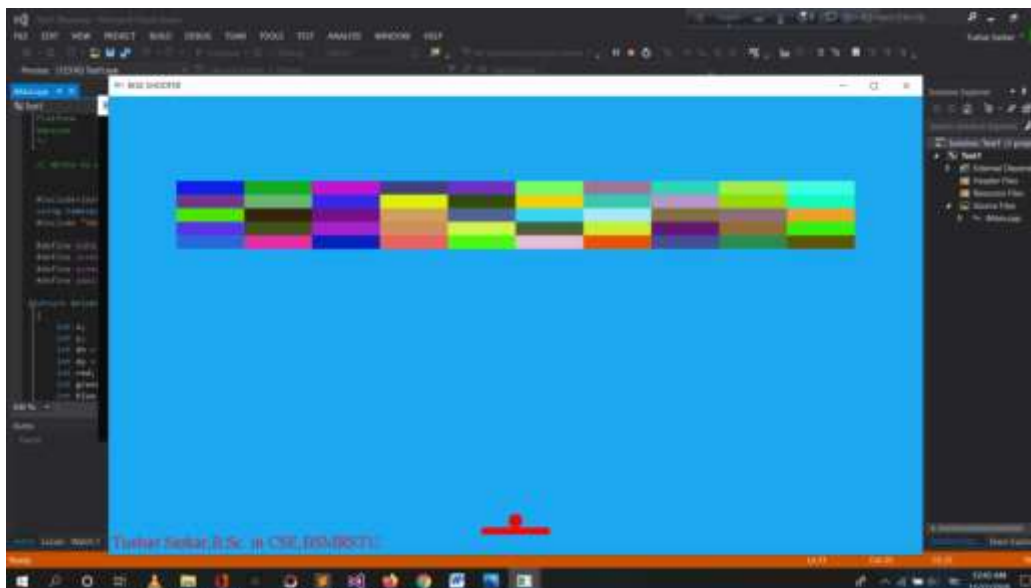


Figure-2.1.2

While the game is going on

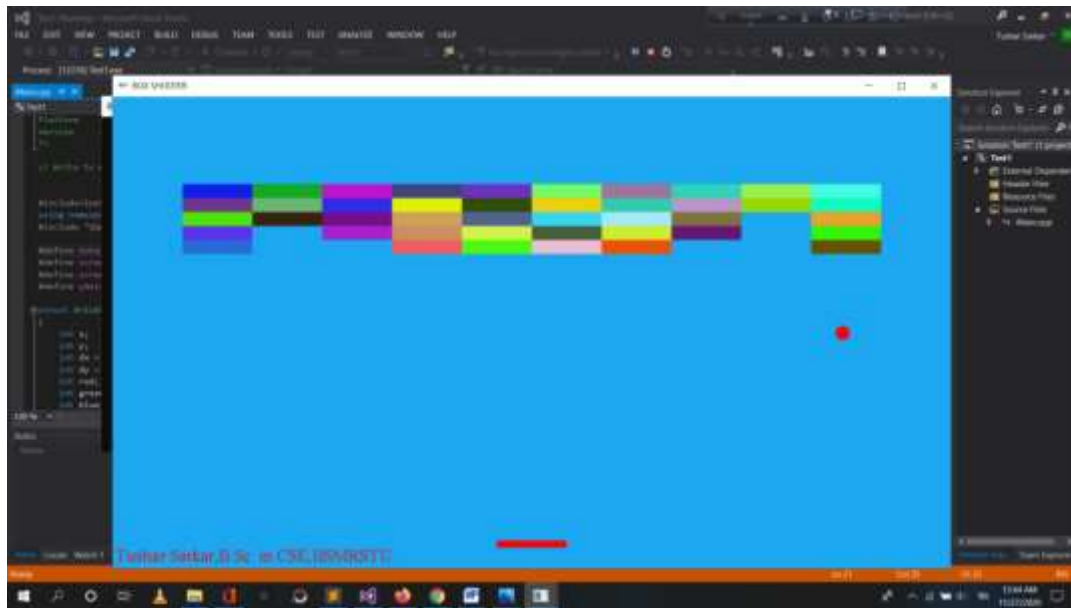


Figure-2.1.3

While the game is loss

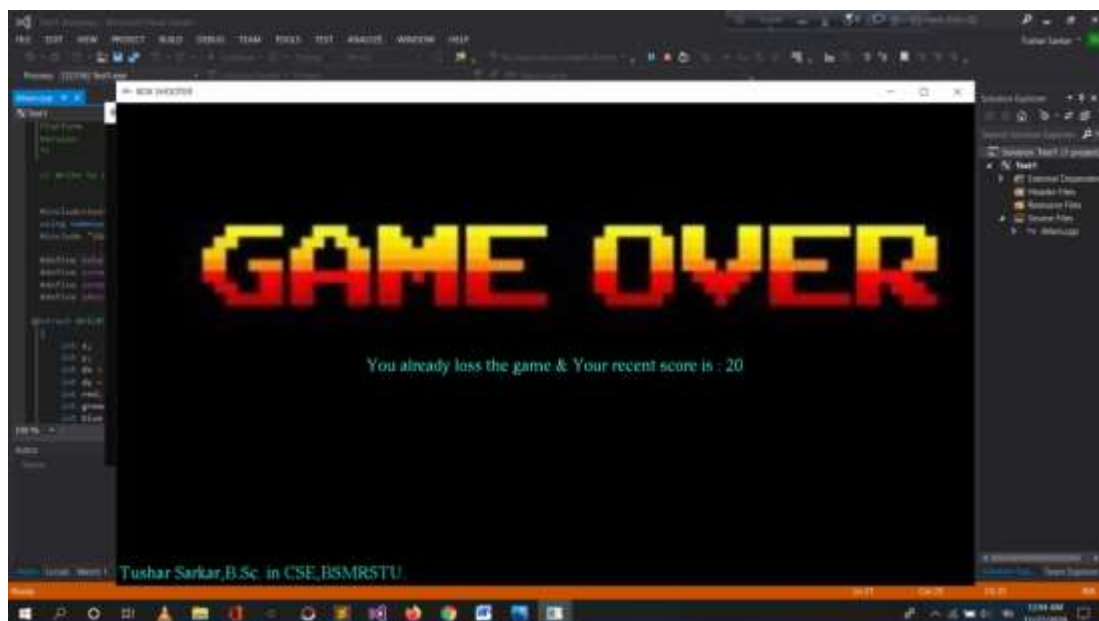


Figure-2.1.4

When all the boxes have been eaten and success



CHAPTER-3

3.1 Implementation

The game is mainly about Cartesian multiplication, vectors, trigonometry and general mathematics. The game has been completed using C, C ++, graphics using the community version of Visual Studio 2013 to create the game. It takes a wave file of a song to play a song in the background and 10-12 pictures which have to be done in a BMP file. First you have to open the new project and open the MPT file and give the name. Since the game added animation, a graphics file was added to the main project and it was originally created.

In this implementation, we have kept track of the direction that each ball is moving, up or down, in order to execute the correct change to y. This requires us to add another variable to the Ball class. You could also choose to make the 'stime' variable positive or negative to simplify the logic but the 'direction' approach makes it easier for students to work on a falling ball first and then extend their logic to the other direction. The following code shows the entire implementation. This is one way to solve the problem but there are many other.

3.1.1 Below is a sample of the graphics.

```
#include<iostream>
using namespace std;
#include "iGraphics.h"

#define screenWidth 1200
#define screenHight 675

int mposx,  mposy;
bool musicOn = true;
int x, y;
int r,g,b;
int dx,dy;

void change();
void iDraw()
{
    iClear();
}
void iMouseMove(int mx, int my)
{
    //place your codes here
}
void iMouse(int button, int state, int mx, int my)
{
    if (button == GLUT_LEFT_BUTTON && state == GLUT_DOWN){}
}
void iPassiveMouseMove(int mx,int my)
{
    mposx = mx;
    mposy = my;
    if(mx== 2){}      /*Something to do with mx*/
    else if(my== 2){} /*Something to do with my*/
}
void iKeyboard(unsigned char key)
{
    if(key == 'w'){}
}
void iSpecialKeyboard(unsigned char key)
{
    if(key == GLUT_KEY_END){}
}
int main()
{
    iInitialize(screenWidth , screenHight , "Box Shooter");
    return 0;
}
```

3.2 Results & Evaluations :

How can we test this code?

The functional requirements that the student developed turn, almost immediately, into a checklist that can be handed to another student. Let's look at an example. The first thing we want to do is to make sure that we can draw at least one ball and see it fall on the screen. So our testing for correct function will be:

- Can I display the ball?
- Does it move towards the bottom of the screen?
- Does it appear to speed up?
- What happens when it hits the bottom?

If we identify an error in the Ball, because it's a Class, we will go into the Ball class and fix it there. The next step for the Ball will be checking what happens when the Ball collides with the bottom of the screen. Does it change direction? Does it slow down as it climbs? Does it reach the same height as before? Can we produce multiple balls at different fall rates with different colours? We'd then continue to test the program until we've tested all of the individual elements and their interactions together. One useful test case is to see if everything is being drawn where you expect. Are the balls are starting at the same point and then falling the correct way? Testing the non-functional requirements often falls into the realm of playability. We noted before that we can ask students whether the controls are obvious and responsive, and whether the animation delay changes slow the game down.

CHAPTER-4

4.1 Future Look:

Although the game is simple for now, a lot of work has to be done. Here are the functions of C, C ++, graphics and many functions have been used. But later on, there is a desire to make this game suitable for world class play. Now only those who have a programming platform and have added iGraphics can play with PC. It needs to be made more workable in the form of applications and there is a desire to create separate versions for mobile and PC and that will be. If you work harder, you will have to upload it to the Play Store separately for Windows, Linux, Mac. Then let's eat; and in the inner workings: now there are only 50 boxes in the game and if you can eat them all, then the game is over, that is, there is only one level and the time is always the same. But later, when one level is finished at different times, boxes of different sizes will come and reducing the time means that the game will be different level wise.

Above all, the game should be played on any operating system and in any version. We have to work with unwavering faith so that this game becomes very popular with everyone. I will take care that the game can gain far-reaching popularity in all respects.

4.2 Conclusion:

It may seem hard to understand at first, but as you're working on more complex animations you can always go back to what you learned on your ball bounce exercise to get you out of a road block you've hit. For instance, a walk cycle is actually prime place to use the ball bounce to help you work through it. The hips should move basically the exact same way that a ball bounces on the ground, of course, it's not as exaggerated but you're essentially trying to capture the exact same pattern that a ball travels. In this project, students will undertake an extensive design and implementation process. The project is well suited to be taught over a number of sessions, and using group work. While we have not included explicit assessment information here, we have provided a discussion of how these sessions might be structured, and how students might be able to demonstrate their learning.

Reference:

1. <https://developer.amazon.com/blogs/appstore/post/a07ab562-0609-4519-a2ba-9b15d69ea62b/introduction-to-game-math-raw-and-cooked>
2. https://www.youtube.com/watch?v=FxCC9Ces1Yg&list=PLSPw4ASQYyymu3PfG9gxywSPghnSMiOAW&ab_channel=MakingGamesWithBen

The End