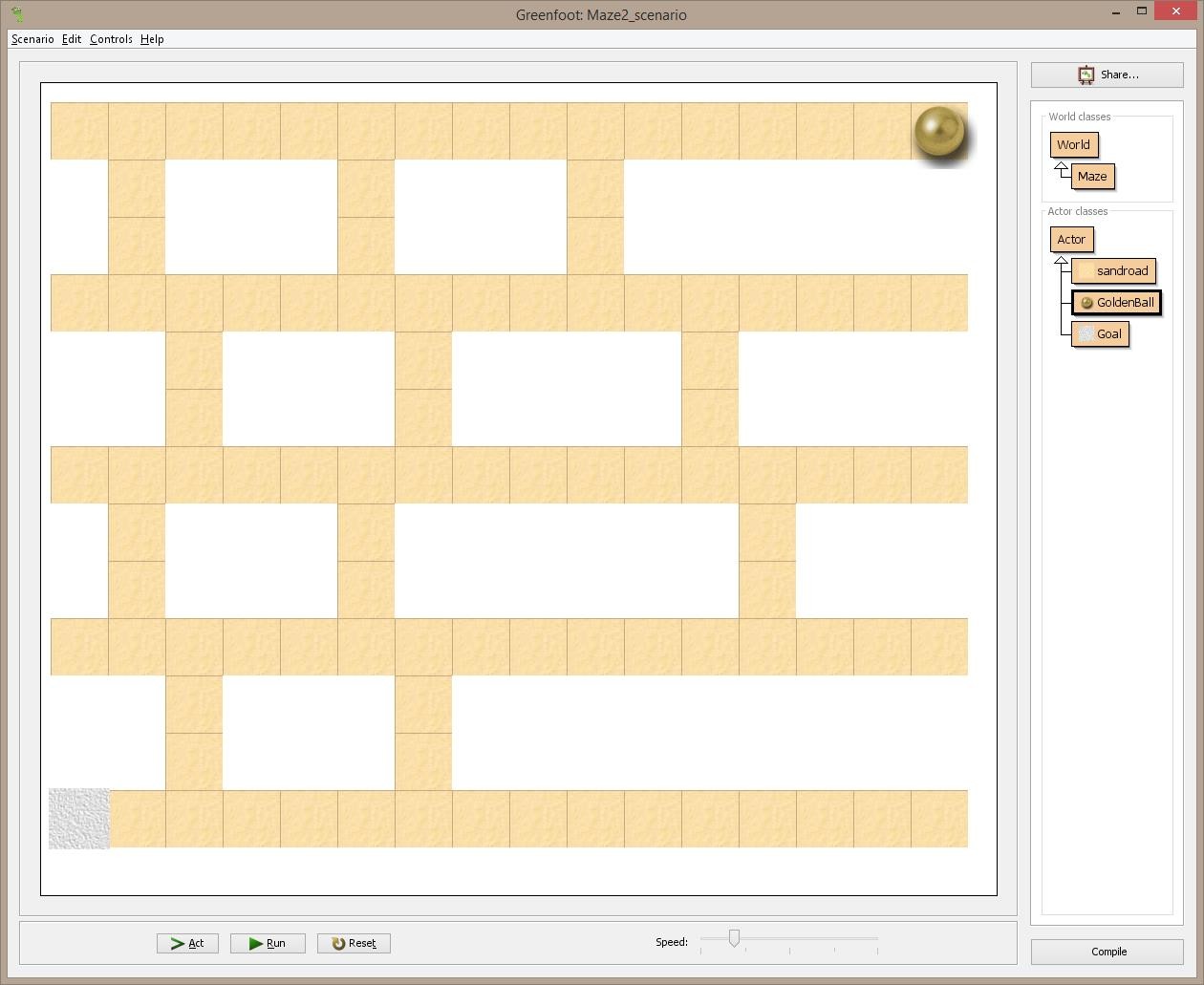
|  |  |  |  |
| --- | --- | --- | --- |
| **CSY1020: Problem Solving & Programming** | | | |
| Agreed Date for submission: | **22 April 2018** | Module Tutor: | Scott Turner |
| Student Name: | Saimon Karki | | |
| Student ID: | 18418326 | | |
| Video Link: | https://youtu.be/ieop6mxKlQo | | |

The University of Northampton Policy on Plagiarism & Mitigating Circumstances will be strictly implemented. By submitting this assignment you are asserting that this submission is entirely your own/individual work.

Marking Scheme

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | A to A+ | A- To A | B | C | D | F+ | F or Below |
| Introduction (10%) | Adequately introduces the **whole** report – not just the task. | | | | | Introduces the task only or insufficiently introduces the report | Task only  introduction |
| Solution (20%) | Has the features of B but includes some appropriate extra features. | | As in solution C but meets the requires as well for the intermediate solution. | Solution meets the basic solution but with a minor improvement. For example appropriate  sound effects or visual effects. | Solution meets the rules for the basic solution only an all the errors are spotted. | Solution has met all but one of the basic rules successfully. | Solution has not met most of the basic rulessuccessfully. |
| Group Work Evidence (20%) | Features of B with excellent evidence of working with others, including well- constructed reflection on the process of team working. Ideally backed up with evidence of reading around and implementing practices in working with others. Please remember this is still an individual piece of work. | | Features of C with strong evidence of understanding the issuesaround working with others. | All the  requirements under “**Evidence of working in a group.**” how been met successful  and clearly  written with satisfactory evidence. | All the  requirements under “**Evidence of working in a group.**” how been met successful, but the limited evidence provided. | All but one the requirements list under “**Evidence of working in a group.**” have been met | More the one of the requirements under “**Evidence of working in a group.**” |
| Testing (20%) | As B but also testing advanced features of solution (i.e. feature for an A in thesolution) | Testing shows what does and does not work. Designed to test wide range of conditions | Testing shows what does and does not work. | Test is  satisfactory | Testing is just about satisfactory. | Testing is  unsatisfactory | Poor or No Testing |
| Conclusions (20%) | All the features of A- but we critically applied to advanced features in both tasks. | Conclusions discuss what was observed in an appropriate manner Showing evidence of being able to critically evaluate the work with some reference to future directions of thework | Conclusions discuss what was observed in an  appropriate manner with some reference to future directions of the work | Conclusions discuss what was observed in an appropriate manner with some reference to future directions of the work | Conclusions discuss what was observed in a satisfactory manner. | Conclusions discuss what was observed in less than satisfactory manner. | Conclusions are too limited |
| References (10%) | Good, appropriate referencing with no faults | | Good, appropriate referencing with only slightfaults | Good, appropriate referencing with only minor faults | Appropriate referencing with some evidence of not fully  understanding the reference process. | Evidence of not fully understanding the reference process. | |

Please note it is expected that you will be discussing ideas with other colleagues and this will be reflected in the report you produce (20% of the marks are allocated to your discussion of group work) but the report and the solution (and code) you produce must be your own. You must include the front two sheets of the brief – the marking scheme or lose 10% of the overall marks.

**TABLE OF CONTENTS**

University of Northampton……………………………………………………………

1. INTRODUCTION…………………………………………………………………1
   1. Project Background…………………………………………………………1
   2. About Greenfoot……………………………………………………………1
2. Evidence of working in a group…………………………………………………….1
3. Analysis & Design………………………………………………………………….2  
   3.1 Algorithm………………………………………………………………………..2  
    3.1.1 Algorithm for Basic Level…………………………………………………..2  
    3.1.2 Algorithm for Intermediate Level………………………………………….3  
    3.1.3 Algorithm for Advance Level………………………………………………3

3.2 Design …………………………………………………………………………

1. Coding……………………………………………………………………………
2. Testing……………………………………………………………………………  
   5.1 Walkthroughs…………………………………………………………………
3. CONCLUSION…………………………………………………………………..  
   6.1 Limitation……………………………………………………………………..
4. Referencing…………………………………………………………………………

**1 INTRODUCTION**

Problem Solving and Programming is a course assigned for us to solve different problems by ourselves. It helps to improve our problem solving skills which will make us able to apply an appropriate way to solve different problems. Similarly, it makes us familiar to the basic programming concept and also the relationship between programming, problem solving and algorithms.

* 1. **Project Background**

For the first assignment of problem solving and programming, we were provided a maze scenario which contained Goldenball, sandroad and Goal as its actor classes. Our task was to solve the given scenario which had some errors. We needed to solve it by using three levels which were Basic, Intermediate and Advance. The rule for each level was provided in the assignment brief. For Basic level, we need to make the ball travel in the path of sandroad and reach to the Goal by using the arrow keys, and if it’s not in the sandroad, it shouldn’t move. For Intermediate level, we were told to make the ball fall in the blocks below the block area producing sound. Furthermore, for Advance level, we need to follow the rules of Basic and Intermediate level and make the scenario more appropriate and attractive and this will finish our assignment. The scenario given to us should be solved in Greenfoot.

**1.2. About Greenfoot**

Greenfoot is a useful educational tool used for educational purpose which makes learning of programming easier. It is a free global software used to develop and understand interactive games. We can create a world of our choices and insert different actor classes in Greenfoot. It supports java programming language. We can also add appropriate program to it, so that it makes a complete scenario. There is an Act, Run and Reset button to execute the program and scenario.

**2. Evidence of Working in a Group:**

As, this was my first ever assignment of Problem Solving and Programming, I was unsure about how to complete the assignment. I was highly confused in understanding the scenario and solving the coding part. I was not very confident in doing this assignment. In order to completely understand what I was going to do, I took help of my tutor, Mr. Kumar Lamicchane. My friends also assisted me in making my concept clear about the assignment. With their help, I was able to plan the correct method to solve the given assignment. We (Me and my friends) made a group and started discussing about the assignment. It was a wonderful experience studying and learning in a group. Everyone was so dedicated in the assignment and each of us gave our own creative ideas to try to solve the given problem. Thus, discussing with friends helped to increase my confidence level and I was much more familiar with the assignment than before. After that, I started doing my assignment and I was using appropriate techniques that I gained from group discussion in solving the problem.

Working with other friends in a group was a great feeling and it turned out to be fruitful in completing the assignment. There were many other pros that I found by working with my friends to get solution to the problem. Some of them are:

* It helped me to uplift my confidence level in completing the assignment.
* It made me much more clear about the concept of assignment and the subject matter itself.
* It helped everyone in the circle to properly utilize the resources given.
* It assisted me and other friends to get access to the new perspective as there was sharing of ideas and knowledge among us.
* The other good side I found was the development of communication skills.
* Everyone was able to do the assignment itself. Besides completion of assignment, I personally found that working in a group made us close.

Furthermore, while working in a group, I found some disappointment as well. I found some members of my circle being lazy and fully depending upon other’s ideas and opinion. Some of other disadvantages that I found are:

* There was clashing of ideas and opinion of each other and thus, difficulty in finding the right option.
* Blaming each other when their idea doesn’t work.
* There was sometimes unequal involvement of members which delayed the discussion as we have to repeat the same thing the next day for the missed member.
* There was ego among some members, and conflict arised many times.
* Active members were a bit angry with the passive members of the group and thus, misunderstanding created many time.

**3. Analysis and Design:**

Analysis and Design is the main part of the system. In this part, problems are collected so that we can solve it and improve it for a better outcome. In this part, problems are deeply identified and solved for efficient work of the system. Analysis is used to investigate the problem and other requirement whereas Design emphasizes on solution of the identified problem and try to meet the requirements.

**3.1. Algorithm**

Algorithm is a set of method that is used to solve recurrent problems. It is used to obtain a required output of the input given in a certain time. The algorithms of different levels are given below:

**3.1.1. Algorithm for Basic Level**:

Step 1: Start

Step 2: Run the program

Step 3: Press the arrow keys.

Step 4: Press left arrow key and check if ball moves 60px to the left and if ball moves, there is  
 sand block to the left and if it doesn’t then, there is no sand block.

Step 5: Press right arrow key and check if ball moves 60px to the right and if ball moves, there is  
 sand block to the right and if it doesn’t then, there is no sand block.

Step 6: Press up arrow key and check if ball moves 60px to the up and if ball moves, there is  
 sand block upward and if it doesn’t then, there is no sand block.

Step 7: Press down arrow key and check if ball moves 60px downward and if ball moves, there is  
 sand block downward and if it doesn’t then, there is no sand block.

Step 8: When ball reaches the goal, stop scenario.

Step 9: Stop

**3.1.2. Algorithm for Intermediate Level:**

Step 1: Start

Step 2: Run the program.

Step 3: Press the arrow keys

Step 4: Press left arrow key and check if ball moves 60px to the left and if ball moves, there is  
 sand block to the left and if it doesn’t then, there is no sand block.

Step 5: Press right arrow key and check if ball moves 60px to the right and if ball moves, there is  
 sand block to the right and if it doesn’t then, there is no sand block

Step 6: Check if there is sand blocks below the ball, then ball falls automatically producing   
 sound and if not then ball moves in the direction of key pressed.

Step 7: When ball reaches the goal, it displays a congratulations message with sound.

Step 8: Stop.

**3.1.3. Algorithm for Advance Level:**

Step 1: Start

Step 2: Along with the ball, obstacle and cherry is added.

Step 3: Obstacle keeps on moving.

Step 4: Run the program.

Step 5: Press the arrow keys

Step 6: Press left arrow key and check if ball moves 60px to the left and if ball moves, there is  
 sand block to the left and if it doesn’t then, there is no sand block.

Step 7: Press right arrow key and check if ball moves 60px to the right and if ball moves, there is  
 sand block to the right and if it doesn’t then, there is no sand block

Step 8: Check if there is sand blocks below the ball, then ball falls automatically producing   
 sound and if not then ball moves in the direction of key pressed.

Step 9: When the ball touches the obstacle, game stops producing sound with the “GAME  
 OVER” message displayed.

Step 10: When the ball touches the cherry, the cherry disappears looking like the ball eating the  
 cherry

Step 11: When ball reaches the goal, it displays a congratulations message with sound.

Step 12: Stop.

**3.2. Design:**

**4. Coding:**

**Maze**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Write a description of class Maze here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class Maze extends World

{

/\*\*

\* Constructor for objects of class Maze.

\*

\*/

public Maze()

{

// Create a new world with 600x400 cells with a cell size of 1x1 pixels.

super(1000, 850, 1);

for (int loop=40;loop<960;loop=loop+60)

{

addObject(new sandroad(), loop,50);

addObject(new sandroad(), loop,230);

addObject(new sandroad(), loop,410);

addObject(new sandroad(), loop,590);

addObject(new sandroad(), loop,770);

for (int loop\_2=0;loop\_2<=1;loop\_2=loop\_2+1)

{

addObject(new sandroad(), 100,110+(loop\_2\*60));

addObject(new sandroad(), 340,110+(loop\_2\*60));

addObject(new sandroad(), 580,110+(loop\_2\*60));

addObject(new sandroad(), 160,290+(loop\_2\*60));

addObject(new sandroad(), 400,290+(loop\_2\*60));

addObject(new sandroad(), 700,290+(loop\_2\*60));

addObject(new sandroad(), 100,470+(loop\_2\*60));

addObject(new sandroad(), 340,470+(loop\_2\*60));

addObject(new sandroad(), 760,470+(loop\_2\*60));

addObject(new sandroad(), 160,650+(loop\_2\*60));

addObject(new sandroad(), 400,650+(loop\_2\*60));

}

}

addObject(new Goal(), 40,770);

addObject(new GoldenBall(),940 ,50);

prepare();

}

/\*\*

\* Prepare the world for the start of the program.

\* That is: create the initial objects and add them to the world.

\*/

private void prepare()

{

Obstacle obstacle = new Obstacle();

addObject(obstacle,345,58);

Obstacle obstacle2 = new Obstacle();

addObject(obstacle2,466,237);

Obstacle obstacle3 = new Obstacle();

addObject(obstacle3,108,415);

Obstacle obstacle4 = new Obstacle();

addObject(obstacle4,344,587);

obstacle4.setLocation(342,767);

Obstacle obstacle5 = new Obstacle();

addObject(obstacle5,226,596);

obstacle.setLocation(45,51);

Cherry cherry = new Cherry();

addObject(cherry,642,54);

Cherry cherry2 = new Cherry();

addObject(cherry2,527,234);

Cherry cherry3 = new Cherry();

addObject(cherry3,341,418);

Cherry cherry4 = new Cherry();

addObject(cherry4,711,591);

Cherry cherry5 = new Cherry();

addObject(cherry5,772,772);

Cherry cherry6 = new Cherry();

addObject(cherry6,157,768);

}

}

**Goldenball**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Write a description of class GoldenBall here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class GoldenBall extends Actor

{

/\*\*

\* Act - do whatever the GoldenBall wants to do. This method is called whenever

\* the 'Act' or 'Run' button gets pressed in the environment.

\*/

public void act()

{

key();

win();

lose();

eat();

//This method is called to make Goldenball fall down

if (canMove(0, 60)==true){

Greenfoot.delay(1);

setLocation(getX(), getY()+60) ;

Greenfoot.playSound("falling.wav");//This produces sound when falling

}

}

public boolean canMove(int x, int y)

{

Actor sand;

sand=getOneObjectAtOffset(x,y,sandroad.class);

//the section below checks if there is a block you can move to

// if there is it sets sand to a vlaue otherwise it says null

// The errors are in this section

boolean flag=false;

if (sand !=null)

{

flag=true;

}

return flag;

}

//This method is called for moving the ball

public void key()

{

//Note 1: Down the page increase the y value and going to the right increases the x value

//Note 2: Each block is 60 pixels wide and high

int leftChange=-60;//This moves the ball to left ;

int rightChange=60;//This moves the ball to right;

int upChange=-60;//This moves the ball to up ;

if (Greenfoot.isKeyDown("left"))

{

if (canMove(leftChange, 0)==true){

setLocation(getX()+leftChange, getY()) ;

Greenfoot.playSound("move.wav");//This is used to play sound while it moves.

}

}

if (Greenfoot.isKeyDown("right"))

{

if (canMove(rightChange, 0)==true){

setLocation(getX()+rightChange, getY()) ;

Greenfoot.playSound("move.wav");

}

}

if (Greenfoot.isKeyDown("up"))

{

if (canMove(0, upChange)==true){

setLocation(getX(), getY()+upChange) ;

Greenfoot.playSound("move.wav");

}

}

}

//This method is called when GoldenBall reaches to the goal.

public void win()

{

Actor win;

win=getOneObjectAtOffset(0,0,Goal.class);

if (win !=null)

{

Greenfoot.stop();//This is used to stop greenfoot when the ball reaches the goal

World myWorld= getWorld();

Congratulations congrats= new Congratulations();//This creates new method

myWorld.addObject(congrats, myWorld.getWidth()/3,myWorld.getHeight()/3);//This is used to set the height and width of the message

Greenfoot.playSound("clap.wav");//This is used to play sound when you reach the goal.

}

}

//This method is called when ball touches the obstacle

public void lose(){

if(isTouching(Obstacle.class)==true){

World myWorld=getWorld();

Gameover g = new Gameover();

myWorld.addObject(g,500, 400);

Greenfoot.stop();//This is used to stop the greenfoot operation when Goldenball touches the Obstacle

Greenfoot.playSound("out.wav");//This is used to play sound when you touch the Obstacle

}

}

//This method is called when ball touches cherry

public void eat(){

Actor cherry;//This creates a new actor cherry

cherry = getOneObjectAtOffset(0, 0, Cherry.class);

if(cherry!=null){

World myWorld=getWorld();

myWorld.removeObject(cherry);

Greenfoot.playSound("eat.wav");//This is used to play sound when you eat cherry

}

}

}

**sandroad**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Write a description of class sandroad here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class sandroad extends Actor

{

/\*\*

\* Act - do whatever the sandroad wants to do. This method is called whenever

\* the 'Act' or 'Run' button gets pressed in the environment.

\*/

public void act()

{

// Add your action code here.

}

}

**Goal**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Write a description of class Goal here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class Goal extends Actor

{

/\*\*

\* Act - do whatever the Goal wants to do. This method is called whenever

\* the 'Act' or 'Run' button gets pressed in the environment.

\*/

public void act()

{

// Add your action code here.

}

}

**Congratulations**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Write a description of class Congratulations here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class Congratulations extends Actor

{

/\*\*

\* Act - do whatever the Congratulations wants to do. This method is called whenever

\* the 'Act' or 'Run' button gets pressed in the environment.

\*/

public Congratulations()

{

setImage (new GreenfootImage("CONGRATULATIONS\n VICTORY!",50, Color.BLACK, Color.YELLOW));//This is used to setimage, message and its color

}

}

**Obstacle**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Write a description of class Obstacle here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class Obstacle extends Actor

{

/\*\*

\* Act - do whatever the Obstacle wants to do. This method is called whenever

\* the 'Act' or 'Run' button gets pressed in the environment.

\*/

public void act()

{

// Add your action code here.

move(30);

if(isAtEdge()==true){

turn(180);//This is used to make the obstacle turn when it touches at edge.

}

}

}

**Cherry**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Write a description of class Cherry here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class Cherry extends Actor

{

/\*\*

\* Act - do whatever the Cherry wants to do. This method is called whenever

\* the 'Act' or 'Run' button gets pressed in the environment.

\*/

public void act()

{

// Add your action code here.

}

}

**Gameover**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Write a description of class Gameover here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class Gameover extends Actor

{

/\*\*

\* Act - do whatever the Gameover wants to do. This method is called whenever

\* the 'Act' or 'Run' button gets pressed in the environment.

\*/

public Gameover()

{

// Add your action code here.

setImage (new GreenfootImage("GAMEOVER!",50, Color.BLACK, Color.YELLOW));//This is used to set image and message

}

}

**5. Testing:**Testing is an investigation carried out in the system to make sure that the objectives set are being met. It allows us to know the differences between given input and expected outputs.

**5.1 Walkthroughs:**

Walkthrough is a step by step technique to test the code and find the errors to note it down for improvement in the future.

**For Basic:**

|  |  |  |
| --- | --- | --- |
| **Data** | **Expected Outcome** | **Actually Happened** |
| Left key pressed | The ball should move a step in the left direction. | As we expected, the ball moves a step to the left. |
| Right key pressed | The ball should move a step in the right direction. | As we expected, the ball moves a step to the right direction. |
| Up key pressed | The ball should move a step in the upward direction. | As we expected, the ball moves a step to upward direction. |
| Down key pressed | The ball should move a step in the downward direction. | As we expected, the ball moves a step to downward direction. |

**For Intermediate:**

|  |  |  |
| --- | --- | --- |
| **Data** | **Outcome Expected** | **Actually Happened** |
| Right key pressed | The ball should move a step in the right direction and if there are blocks below the block it should fall downward automatically producing sound. | As we expected, the ball moves a step to right and when there are blocks below the block it fell downward producing sound. |
| Left key pressed | The ball should move a step in the left direction and if there are blocks below the block it should fall downward automatically producing sound | As we expected, the ball moves a step to left and when there are blocks below the block it fell downward producing sound. |

**For Advanced:**

|  |  |  |
| --- | --- | --- |
| **Data** | **Outcome expected** | **Actually Happened** |
| Right key pressed | The ball should move a step in the right direction and if there are blocks below the block it should fall downward automatically producing sound. | As we expected, the ball moves a step to right and when there are blocks below the block it fell downward producing sound. |
| Left key pressed | The ball should move a step in the left direction and if there are blocks below the block it should fall downward automatically producing sound | As we expected, the ball moves a step to left and when there are blocks below the block it fell downward producing sound. |
| Ball touches the obstacle | The game should be stopped and produce a sound. It should also display a GAMEOVER message. | As expected, GAMEOVER message appeared along with the sound when the ball touches the obstacle. |
| Ball touches the cherry | The cherry should disappear when it comes in contact with the ball and should produce a sound. | As expected, the cherry disappears when it touches the ball. |
| Ball reaches the Goal | The game should get stopped and produce a congratulation message along with the sound. | As expected, congratulations message appears with presence of clapping sound. |

1. **Conclusion:**

In conclusion, the assignment was given to us for the completion of 1st Term of Year 1.The given assignment was very fun and worth doing. Firstly, I had some problem in understanding and using the correct code in the right place. Then it was my tutor who helped me a lot in making me understand the assignment in brief. Additionally, my friends were always there for me and we did some research for better understanding. It was difficult at the first to complete the assignment but I continuously did research and tried to learn many things. After the completion of assignment, I found that it was a completely new experience for me and I was able to learn many new interesting things. I am now confident that I can solve some of the basic problems of the greenfoot scenario by doing research and through learning resources.

The scenario was all about taking the golden ball to the goal. I was able to solve the coding problem and followed the Basic, Intermediate and Advance level to make the ball reach to the goal as we were told to. I made the scenario in such a way that the golden ball only moves in the sand block and when there are blocks below the ball it will automatically go downwards and produce sound. Furthermore, I also added some extra features like Obstacle and Cherry to make the game scenario more fun and attractive. Lastly, when ball reaches the goal, the player is appreciated by congratulations message with clapping sound.

If I was provided with some extra time and if I am told to do the scenario again then, I am very confident that I will create a better scenario than I made this time. I would add some extra features which will make it more attractive and exciting to play. I would have added stages of the game and I would add some more obstacles and add such a feature in which difficulty gets increased as the stage increases. I would also add points in the game. This could surely be a driving factor for my game.

* 1. **Limitation:**

As this was my first time doing this assignment, I know I have many limitation and I need to learn more to get somewhere around perfection. I made a very simple and basic game .Due to lack of proper time and resources, I was unable to add other more feature like Level of the game. Also, I was unable to put a score board on the game. And, it is lacking proper sound effects as well. These certainly are the major drawbacks of my game which can be some negative factors.

**7. Referencing:**

1. Available form: <http://static1.grsites.com/archive/sounds/cartoon/cartoon015.wav> - Eat sound  
 effect [online]  
 [Accessed on 17 April, 2018]

2. Available form: out- <http://static1.grsites.com/archive/sounds/cartoon/cartoon010.wav> - Out   
 sound effect [online]  
 [Accessed on 17 April, 2018]

3. Jefferson Moses (12 Jan 2018)- Intermediate Level Greenfoot [online] for adding sounds and   
 make ball fall down.  
 Available at: <https://youtu.be/4Xan9tdqXNc>   
 [Accessed on 14 April, 2018]

4. Available form: fall- <http://static1.grsites.com/archive/sounds/cartoon/cartoon006.wav> -Falling  
 sound effect [online]  
 [Accessed on 17 April, 2018]

5. Available form: move- <http://static1.grsites.com/archive/sounds/cartoon/cartoon002.wav> -   
 Moving sound effect [online]  
 [Accessed on 17 April, 2018]