**Street Runner: Java 2D Racing Game**

Software Project I Submitted

By

**Name**  **Id**

Rahman, Md. Ashekur **[**12-21301-2**]**

Ishaque, Usama Ibna **[**12-21419-2**]**

Under the supervision of

**A.G.M. Zaman**

Assistant Professor, Department of Computer Science

Faculty of Science and Information Technology

American International University - Bangladesh

Spring Semester 14-15

**Declaration**

This is to certify that this project is our original work. No part of this has been submitted elsewhere partially or fully for the award of any other degree. Any material reproduced in this project has been properly acknowledged.

|  |  |
| --- | --- |
| 1. Name : Rahman, Md. Ashekur  ID NO. : 12-21301-2  Department : CSSE | 2. Name : Ishaque, Usama Ibna  ID NO. : 12-21419-2  Department : CSSE |

**Approval**

Project  titled  “**Street Runner: Java 2D Racing Game**”  has  been  submitted  to  the  following  respected  members  of  the  Board  of Examiners of the Faculty of Science and Information Technology  in partial fulfillment of the requirements for the degree  of  Bachelor  of  Science  in  Computer Science and  engineering  by the following students and has been accepted satisfactory.

|  |  |
| --- | --- |
| **A.G.M. Zaman**  Assistant Professor & Supervisor Department of Computer Science  American International University- Bangladesh | **Dr. Dip Nandi** Assistant Professor and Head (Undergraduate Program)  Department of Computer Science  American International University-Bangladesh |
| **Prof. Dr. Tafazzal Hossain**  Dean  Faculty of Computer Science and IT  American International University-Bangladesh | **Dr. Carmen Z. Lamagna**  Vice Chancellor  American International University-Bangladesh |

**Acknowledgement**

We are very grateful to Mr. A.G.M. Zaman, our project supervisor for his continuous support, assistance, instruction and encouragement, without which our work would have never been completed. We would like to thank Mr. A.G.M. Zaman for his help for showing us how to design and develop a java gaming application. Without the proper guidance of him this project would not have been successful.

We convey our thanks to our honorable Vice Chancellor, Dr. Carmen Z. Lamagna for encouragement.

**Table of Content**

**Chapter-1 Introduction**

* 1. **Overview**
  2. **Introduction**
  3. **Background Study**
  4. **Objectives**

**Chapter-2 Project Scope**

* 1. Requirements
     1. U**sed technologies**
     2. O**perating** E**nvironment**
  2. Roles and Responsibility

**Chapter-3 Game Design and Specification**

* 1. Class Diagram
  2. Game Scenario
  3. Project Features

**Chapter-4 Conclusion and Future Work**

**References**

**Appendix**

Distribution

**Chapter 1- Introduction**

1. **Overview**
   1. **Introduction**

Since the early 1990s, people have been using their mobiles to play games. Major gaming companies never paid much attention to pc running game development because it was not possible to develop powerful running games for computers. When game development companies started developing their operating system and the games became popular and easily accessible. In short, gaming is exploding! The explosive growth of the gaming industry, in addition to the profusion of powerful games, has fuelled the demand of gamers. It is known that games can be used to teach traditional concepts from computer science in our regular computer science classes because games motivate students, which we believe increases enrolment, motivation and retention and thus helps us to educate more. In general, games can be used to teach almost every area of computer science. For example, computer architecture is important for understanding how game consoles work. “Street Runner” is a 2D racing game. “Street Runner” belongs to endless runner genre of games. This is 2D and Java based game. The runner will be running at a certain pace at the beginning but as the time will pass, it will fasten its motion. The trick is the gamer will face some challenges while running faster. Background will be changing periodically in the night mode and day mode. The gamer will be required to hold the pace and keep pace with the speed and be careful whether the runner collides with obstacles and die. The game ends where the runner dies. These kinds of games can be used to improve the gamers' concentration, furthermore, teach a variety of important job skills for computer scientists in both academia and industry, including technical skills such as computational thinking, software engineering and programming skills, creativity, design skills, problem solving Skills and teamwork skills. The project is on an animated game and this sector is a complex subject in game programming. Java games are expected to run on multiple operating systems with different hardware specification. Threads give the most accurate timing solutions

* 1. **Background Study**

The word "game" is familiar among people from the young until the old ones. It is recognized that the current existence of game has become one of the necessities in life. Game can be used as a mean of recreation or entertainment. Even for some people game becomes their profession. There are similar types of games like Street Runner. Popular android games like Flappy Bird [3], Temple Run [4], Subway Surfer [5] all fall into the same category. These games are endless. Players can play forever until they collide with something in the game.

This game is developed in Java programming language. As java is platform independent this game will suit in almost every OS. Java offers many built it method to the developers which are being used in this game like event handler which helps to make the runner jump, for repainting and ranging from moving *paintComponent*() method is being used [1]. This game has been developed by using some built in classes like *JFrame*(creates frame), *JPanel*(creates panel) , *JButton*(creates button) etc [1]. As this is a running game the gamer would expect running object and in order to fulfill this expectation as well as make the game look incredible multiple threads has been used to make things look moving at different ways [2]. There are two separate threads running in background. Another working system in the game is Event handler to detect mouse clicks. When the user clicks mouse for once, the runner will jump. The game has a well-organized framework so that the code is readable and it is easier to find you one looks for.

* 1. **Objectives**

The objective was to develop a simple 2D Racing game using JAVA, make the game user-friendly and fun to play. Also make the graphics pleasant for the user. Completing the documentation regarding the project is also an important part of the objectives.

**Chapter-2 Project Scope**

* 1. **Requirements**
     1. **Used Technologies**

For Graphics designing and animation part of the game Adobe Flash and Photoshop was used. Java was selected as the programming language to develop the game, so JDK and JRE were used in the project. “eclipse Luna” was used for the coding part.

1. JRE version 8u51(Java Runtime Environment)
2. Eclipse IDE for Java EE Developers
3. Adobe Flash Professional CS 6.0
4. Adobe Photoshop CS 6.0
   * 1. **Operating Environment**

The game will run on any device (Computer/Mobile) which supports JDK and have the ability to run any java application.

* 1. **Roles and Responsibilities**

The software developers are responsible for all documentation to be developed and also for all work to be done. The work was distributed among the developers and integrated under the supervision of Project Supervisor and Team Leader.

**Chapter-3 Game Design and Specification**

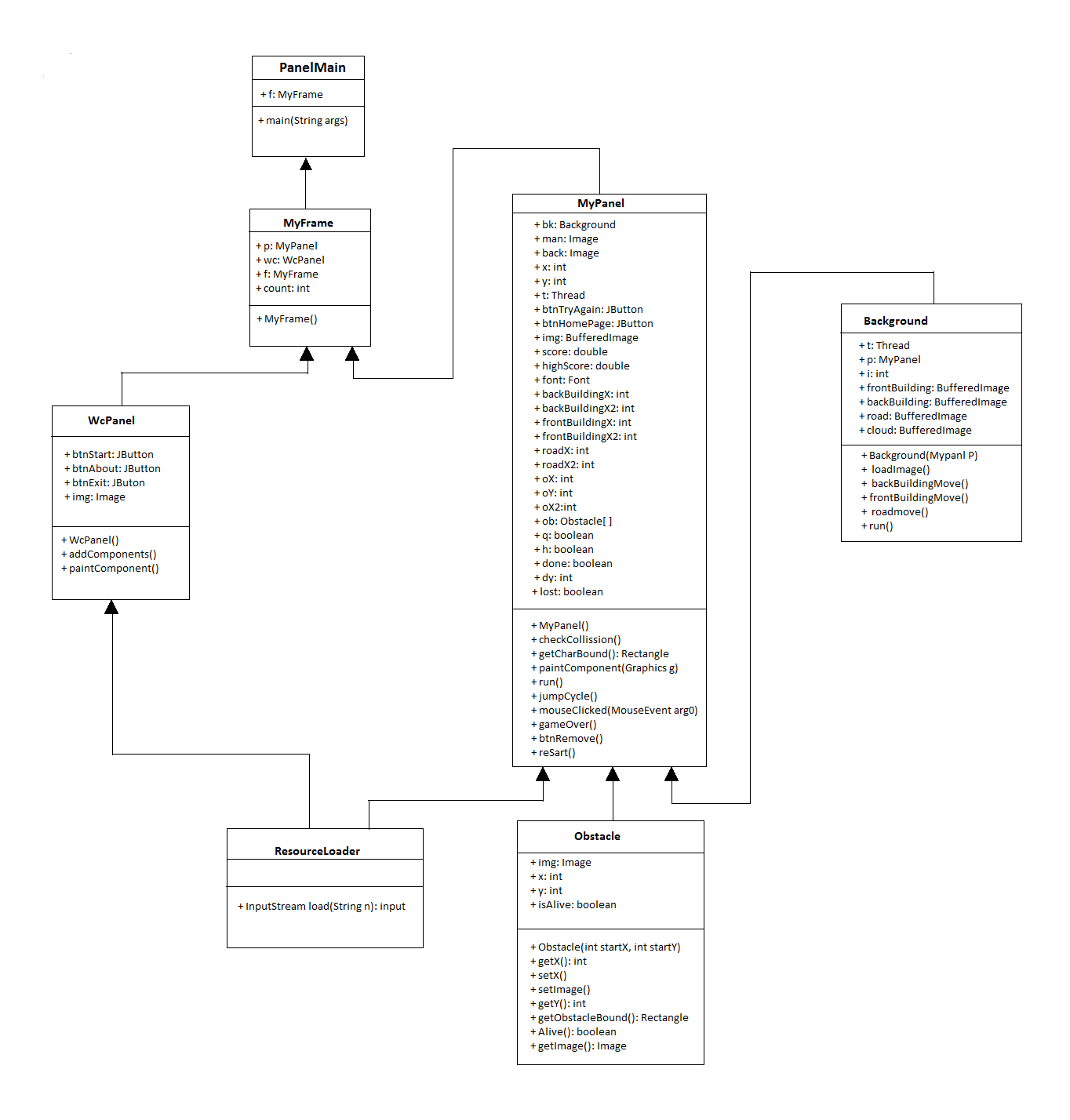
* 1. **Class Diagram**

Fig. 1: Simplified Class Diagram of Street Runner

Fig. 1 shows the simplified class diagram of the project. *PanelMain* has the main function. *PanelMain* creates an object of *MyFrame* which extends *JFrame* and creates the initial Game Frame. At the beginning, *MyFrame* adds the contents of *WcPanel* which extends JPanel to the frame. When user clicks the Start button, *MyFrame* adds the contents of MyPanel which also extends *JPanel*. *MyPanel* loads all the images and paints it on the screen using *paintComponent( )* function. Thread is also used for updating the screen continuously. *MyPanel* implements *MouseListener* to detect mouse click which will make the charater jump to avoid the obstacles. Background class has contents of the background of the game. It loads the images and *MyPanel* paints them. Some variables are used to represent the x,y coordinate of the character, obsctacles, buildings and other things in the game. Threads in *MyPanel* and Background changes their values to move the contents and the game and create the animation. Obstacle class has all the information about the obstacls of the game. *MyPanel* loads the images of the obstacles from the Obstacle class and paints them in the screen. The main character of the game and obstacles has their own Rectangle which helps the program for collision detection. When Rectangles representing the character and any obstacle intersects with each other; they collide and the game stops by stopping the Threads and showing *“Try Again?”* and *“Home”* button. If *“Try Again?”* button is clicked then the states of all the variables resets and Threads start. *“Home”* button loads *WcPanel* removing *MyPanel* from the *MyFrame*.

* 1. **Game Scenario**

Street Runner is an endless running game. Runner is constantly running, day and night, but that is not an easy task. He has to face a lot of obstacles while running and must avoid the obstacles to stay alive. Runner will be gaining points and the longer he stays alive, the highest points he will get. The speed of the runner will be increasing time to time, making the simple task more difficult. The pace of gaining points will also increase with the running speed. If the runner collides with any obstacle, the game ends.

The game has a well-organized framework so that the code is readable and it is easier to find you one looks for. The game flow in the framework goes like this: In the state "Launch Game", first a window is created then a panel is put on for drawing and to the keyboard and the mouse events. In this state, an initialization of variables and objects and loading files such as images, sounds are followed. When the state "Launch Game" is finished we are in “Main menu". Here player can choose between "Start", “About” and "Exit". If "Exit" is selected then the game ends there. If "About" is selected then credits is shown. When "Start" is selected, the actual game is created then. There are two separate threads running in background. Each and every program is running on different threads. This is basically an animated game. Another working system in the game is Event Handler. When the user clicks mouse for once, the runner will jump. Then the runner will get down and run. The score will be increasing continuously. The speed will increase after specific times, making the game more difficult. If the runner and any obstacle collide, the game stops and shows two buttons- “Try Again?” and “Home”. “Try Again?” starts the game again and “Home” takes the player back to the main menu.

* 1. **Project Features**

“Street Runner” is a simple gaming application. Users need to run the program to play the game.

****

Fig. 2: The Start Screen

Fig. 2 shows the Start screen of the game. There are three options in the Start Screen -

1. Start – Starts the game.
2. About – Shows information regarding game development.
3. Exit – Exits the program.

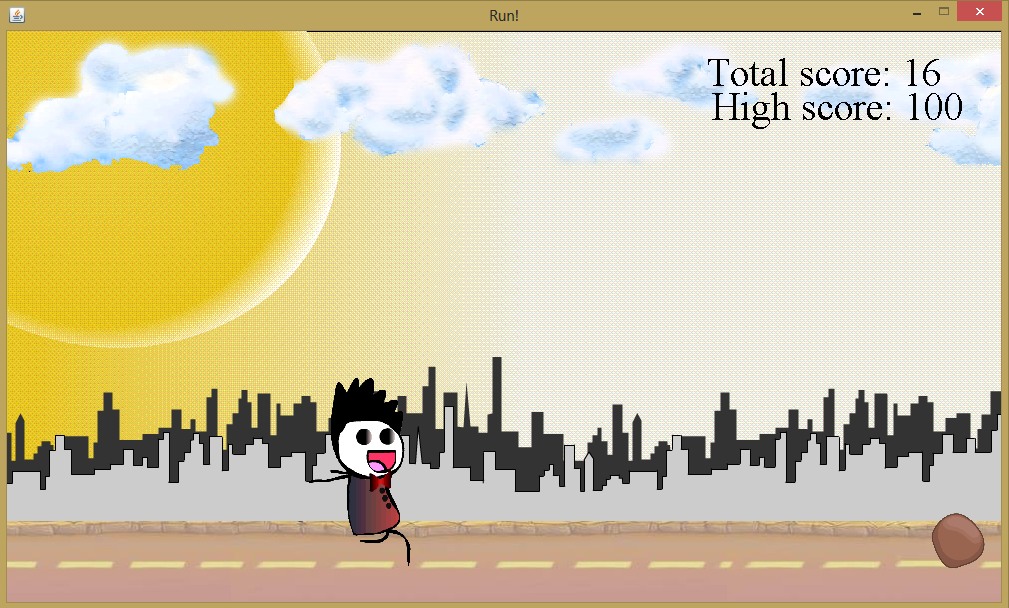
****

Fig 3: Game Screen

Fig. 3 is showing the screen after the game starts. During the game, the runner will be running at a constant speed and will be increasing after specific times. Obstacles will be coming from the right side of the screen. Player has to click on the screen to jump and avoid the obstacles. Player can see his score and high score in the top right corner of the screen. On the background, after day passes, nigh comes and then night passes, day comes.

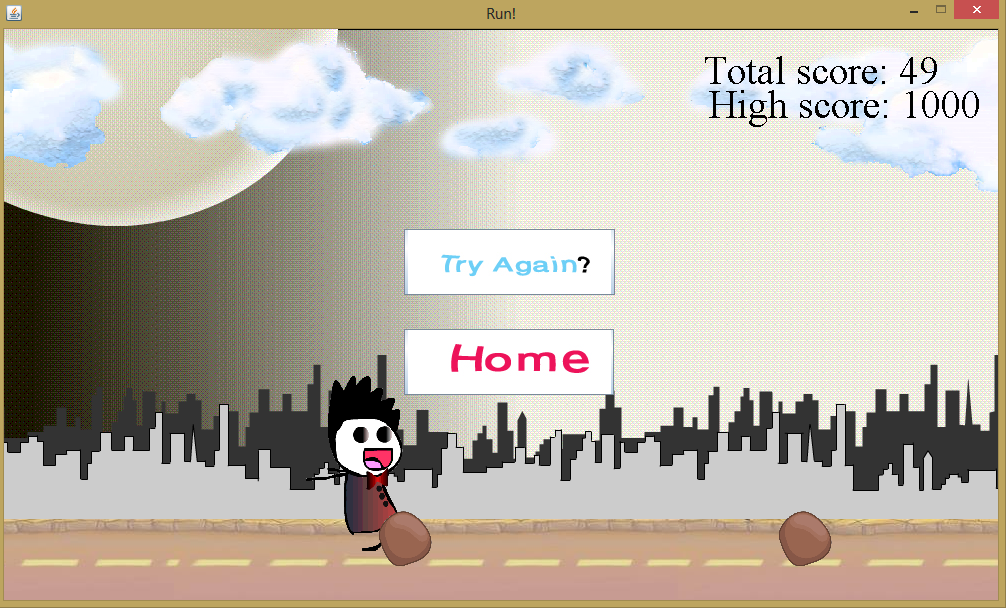


Fig 4: After colliding with the obstacle

What happens if the user collides with an obstacle is shown in Fig. 4. If the player collides with an obstacle, the game ends immediately and a two options is shown-

1. Try Again? – Restarts the game.
2. Home – Takes back to the start screen.

**Chapter-4 Conclusion and Future Work**

Game development requires logic and creativity at the same time. It also requires a good amount programming knowledge. A lot of things can be learned from gaming development. It is known that games can be used to teach traditional concepts from computer science in our regular computer science classes because games motivate students, which we believe increases enrolment, motivation and retention and thus helps us to educate more. In general, games can be used to teach almost every area of computer science. For example, computer architecture is important for understanding how game consoles work. Java is a suitable platform for game development. “Street Runner” is a simple game implement using Java. It’s both pleasant and challenging at the same time. The interface is very user-friendly and it is fun to play. The game is complete but many additional features can still be added in the game like weather changes, bullets to destroy obstacles, different levels, enemies and many more. “Street Runner” is a 2D racing game and these kind of games can be used to improve the gamer’s concentration, furthermore, teach a variety of important job skills for computer scientists in both academia and industry, including technical skills such as computational thinking, software engineering and programming skills, creativity, design skills, problem solving Skills and teamwork skills. Many versions of the game can be made to support other platform and devices. The game will be developed to run on Android devices.

**References**

[1] Andrew Davison, Killer Game Programming in Java, May 2005.

[2] David Brackeen, Bret Barker, Laurence Vanhelsuwé, Developing Games in Java, New Riders Publishing, August 20, 2003.

[3] Flappy Bird. [Online]. Available: <http://flappybird.io/>

[4] Subway Serfer. [Online]. Available: <https://play.google.com/store/apps/details?id=com.kiloo.subwaysurf>

[5] Temple Run. [Online]. Available: <https://play.google.com/store/apps/details?id=com.imangi.templerun>

**Appendix**

**Distribution**

|  |  |  |
| --- | --- | --- |
| **Recipient Name** | **Recipient Organization** | **Distribution Method** |
| A.G.M. Zaman | AIUB | Hard Copy, Soft Copy |