VISHWAKARMA GOVERNMENT ENGINEERING COLLEGE, CHANDKHEDA

A Project Report On

Magnar Game

Advanced Java

Semester-6th
(Information Technology Department)

Submitted By
Parmar Fenil (170170116027)

Patel Aman (170170116028)

Reddiar Harsh (170170116047)

Guided By
Prof. Chetna Chand

Academic Year

2019-2020

Objective

This project is the culmination of basics of java and advanced java that provides us an opportunity to build a dynamic game like this. This project is an opportunity to showcase our technical skills. This project is built so that people can have a source for killing time during this hard time of pandemic. This is entirely for entertainment purposes and unlike other games it is not a multiplayer game and a single individual can play even if he doesn't have an internet facility.

Functionalities

- Users can play the game 24×7 without any further requirements.
- Game is much simple to play with minimal controls
- Once installed, the internet is not required to play the game.
- Game has attractive background music with enrich graphics.
- Instructions are available at any time about how to play the game.
- User’s high scores are stored in the records.
- Different levels are rendered dynamically once a user passes a certain level.
- Once the user reaches maximum level then he can design his own obstacles by clicking then dragging and dropping to create the desired size of obstacle.

Tools and Technologies Used

- MySQL for database connection
- Netbeans to develop project

Scope of project

With minimum controls and absence of internet allows this game to be widely played among different age groups. People all over the world can be entertained using it. A 9-10 year old child or an old aged person who doesn’t have much knowledge about technology can also play as they have very basic controls.

Front end and back end

- Front End: HTML5, Java graphics, Java swing, Java window, Java awt
- Back End: Java,JDBC, My SQL

Classes

1. Animations.java
2. Block.java
3. BufferedImageLoader.java
4. Camera.java
5. checkpoint.java
6. Cloud.java
7. Cutscene.java
8. Dbcon.java
9. DeathBoundary.java
10. Game.java
11. GameObject.java
12. Handler.java
13. Highscore.java
14. Input.java
15. Lava.java
16. Levelend.java
17. Lifeup.java
18. Lifeup2.java
19. Menu.java
20. Music.java
21. objectID.java
22. Player.java
23. SpriteSheet.java
24. State.java
25. Texture.java
26. Trap.java
27. Window.java

Methods

1. public Animations(int speed,BufferedImage... args)
2. public void RunAnimation()
3. public void drawAnimation(Graphics g,int x,int y)
4. public void drawAnimation(Graphics g,int x,int y,int ScaleX,int ScaleY)
5. private void nextFrame()
6. public class Block extends GameObject
7. public Rectangle getBounds()
8. public void render(Graphics g)
9. public Rectangle getBounds()
10. public void tick(LinkedList<GameObject> obj)
11. public BufferedImage loadimage(String path)
12. public Camera(float x, float y)
13. public void addscore()
14. public void viewscore(Graphics g)
15. public boolean CheckLife()
16. public void LoadImageLevel(BufferedImage image)
17. public static void NextLevel()

18. private void init()
19. public void run()
20. public static void setLife(int i)
21. public static void setState(State st)
22. public synchronized void start()
23. public void addObject(GameObject object
24. public void clearLevel()
25. public static int geti()
26. public void makelevel()
27. public void keyPressed(KeyEvent e)
28. public void keyReleased(KeyEvent e)
29. public void removeObject(GameObject object)
30. public Rectangle getBounds()
31. public static int getmscreen()
32. public static void setmscreen(int i)
33. public void playMusic(String mus)
34. private void collision(LinkedList<GameObject> object)
35. public SpriteSheet(BufferedImage image)
36. public BufferedImage grabImage(int col,int row,int width,int height)
37. private void getTextures(int i)
38. public boolean trapact()

Getter and setter methods are also implemented for data members of the various classes.

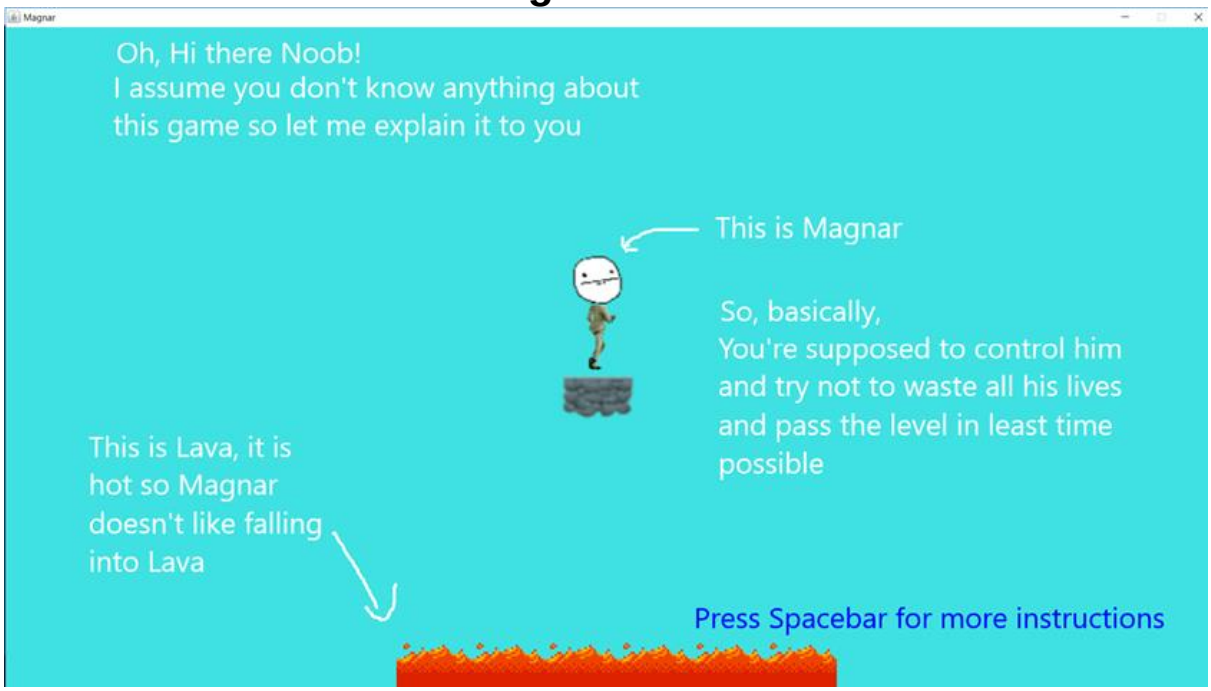
Objects

- Magnar (User)
- Lava
- Obstacles
- Clouds
- Death Boundary
- Checkpoint

User Interface of Magnar Game

Screen 1 : Main Menu

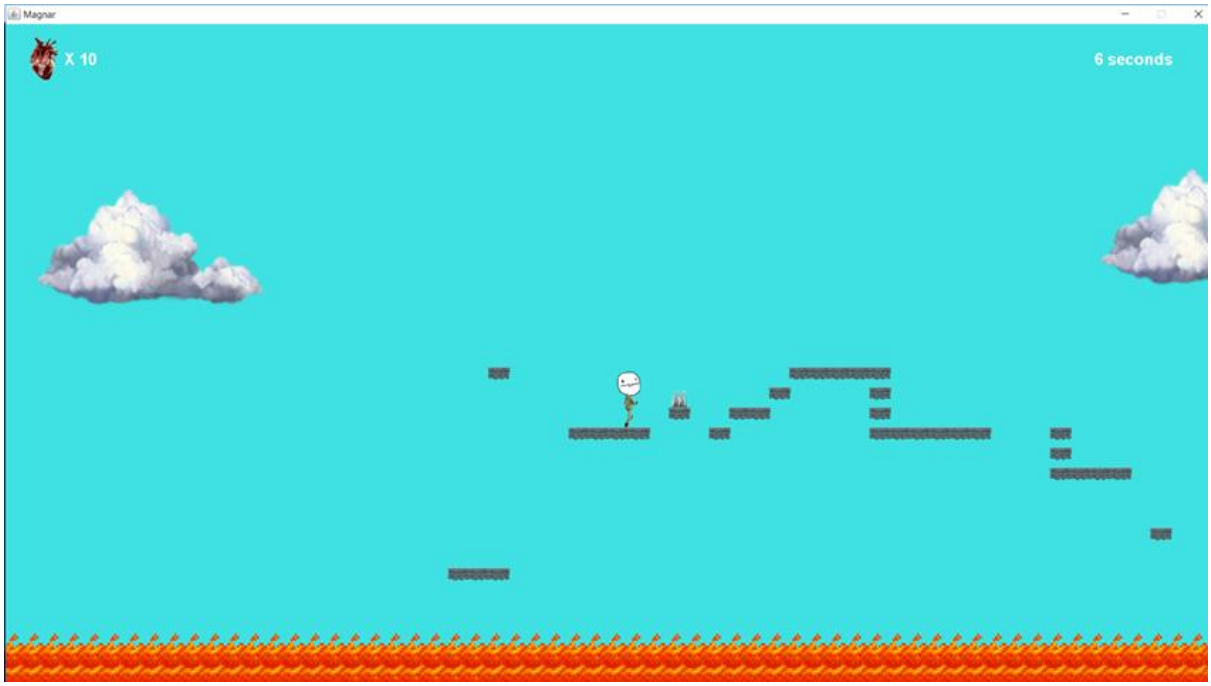
Screen 2 : Instructions Page-1



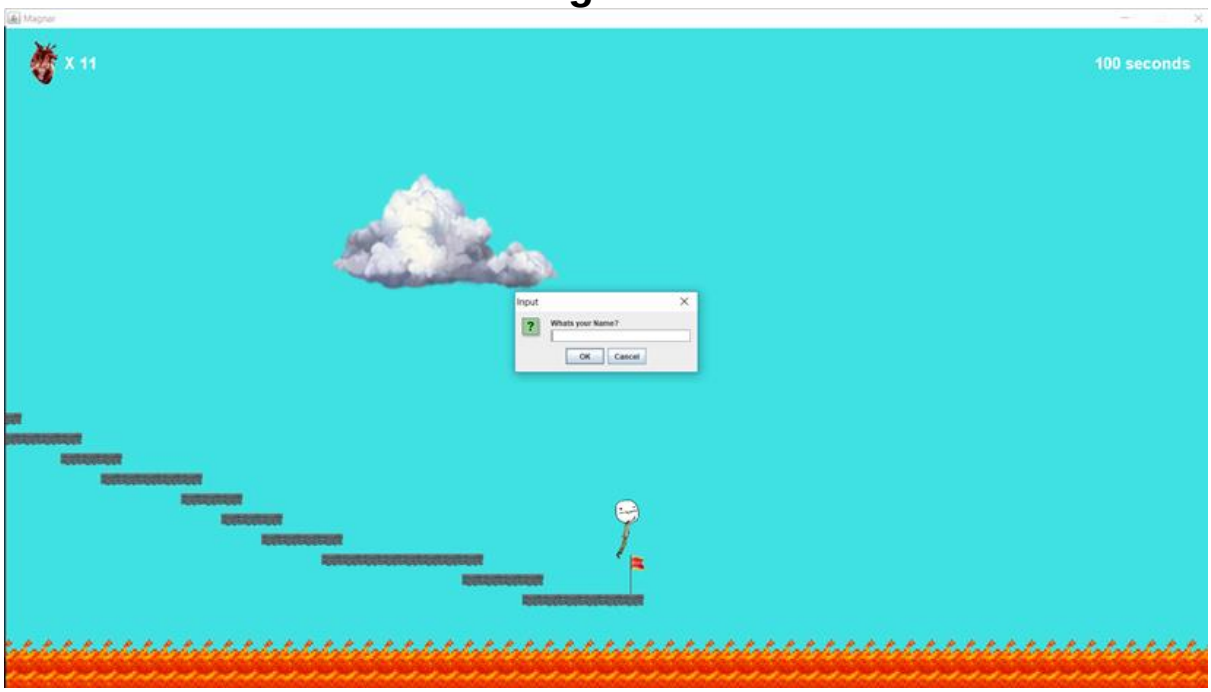
Screen 3: Instructions Page-2

Screen 4: View High Score

Screen 5: The Game



Screen 6: Enter Name for Highscore



Conclusion

Our game provided a good experience for learning java and also was fun while playing it. We have combined different modules and various abstract classes By using the functions we can make this game playable, java graphics library has made the game visually pleasing as compared to c++. Also by connecting this game with Database we can show the high scores to the user. It makes the game more competitive and also will learn how to deal with databases.