
University of Dhaka

Department of Computer Science and Engineering

*Project Report:*
*Fundamentals of Programming Lab(CSE-1211)*

*Project Name:*
*Goofy Pinkman : On The Run*

*Team Members:*

1. *Md jawad Ul Islam Abir(FH-40)*
2. *Arif Billah(FH-32)*
3. *Tonmoy Banik(JN-54)*

# 1. Introduction

Goofy Pinkman : On The Run, An endless runner game, linear in design with no end, without pauses or breaks for rest and no stages or changing levels. It has one continuous level. The game difficulty starts slow and easy. There's this character named Pinkman who appears to run faster and faster as the game speeds up. Obstacles approach faster and faster, requiring ever faster reactions. The player succumbs to intensity and it's game over.

# 2. Objectives

The main objective of our game is to make use of our C/C++ skill and use SDL to make an adventure game for everyone. We decided to make an infinite 2D runner that consists of two different kinds of maps. And also we implement our ideas in a way that makes future extensions and updates very easy and fast.

# 3. Project Features

1. **Maps:** There are two maps in the game named Braavos and Westeros.
    - **Westeros:**

        In the world of Westeros, Goofy Pinkman can only run and jump to avoid obstacles.

- **Braavos:**

  But in Braavos, Pinkman gets the ability to fly like superman and face some different kinds of obstacles than westeros.

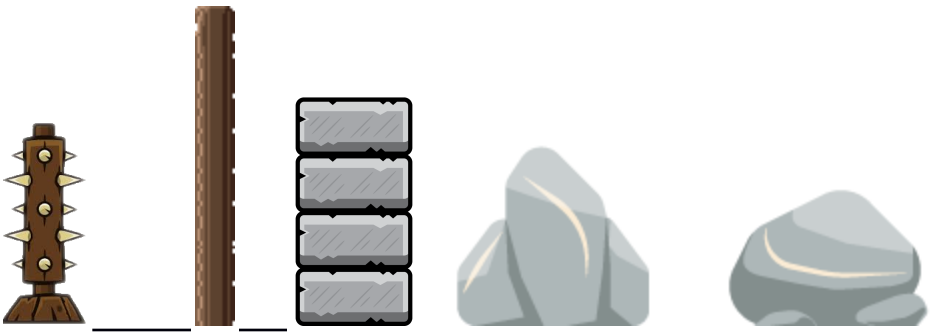

2. **Running**: In Westeros our hero is always in a hurry. He is running to see the end of time

3. **Jumping**: In Westeros, when our hero encounters any kind of obstacle or lake, the player can make him jump by entering “spacebar”, “enter” and “j" from the keyboard.

4. **Flying:** In Braavos, Our hero can fly. Player can control him by pressing the right key to move right, left key to move left, up key to move up and down key to move down.

5. **Sounds**: We added background music, in menu music, collision sound, jump sound, game over music, sound when the player collects bonus points etc.

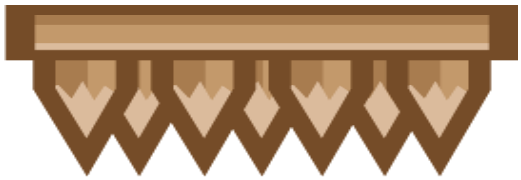
6. **Different types of obstacles:** Any kind of touch with these obstacles will give you instant death.
    - **Westeros:**



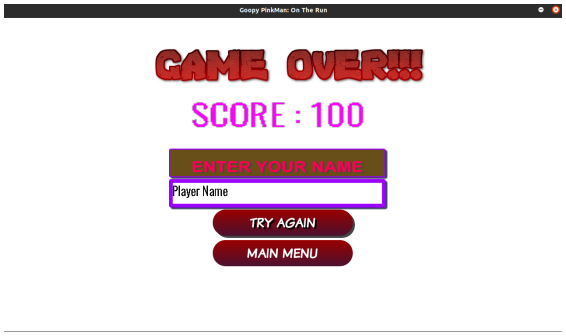
    - **Braavos:**

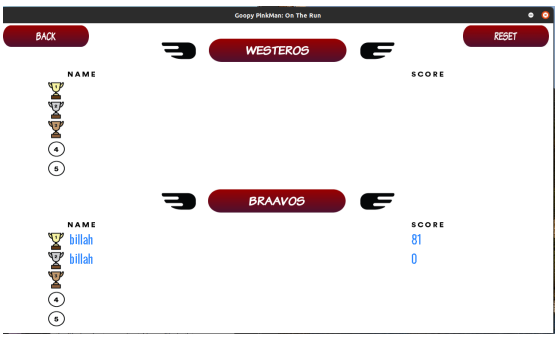
7. **Bonus Points && life gain:** throughout the game, the player has numerous opportunities to collect bonus points. Different colored points increase the player's score at different rates. And there are also life players who can gain this and increase life but total lives will always be not more than three at any time.

8. **Falling and rotating objects:** There are some special obstacles like falling obstacles which tries to fall on the hero and make him dead, rotating obstacles which spin and try to cut the hero into pisces.Touches with those obstacles will instantly kill him.

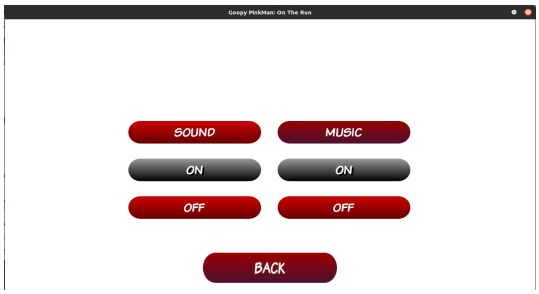


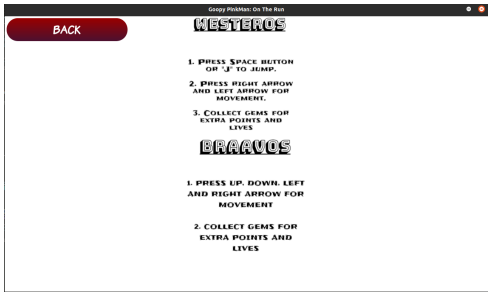
9. **Scoreboard & Score:** After finishing three lives game will ask to enter the player name for the scoreboard. We added a full scoreboard of two maps. For every map, the top five scorer name will be on the leaderboard with their saved name.



10. **Complete Menu:** we created a complete menu, settings, help and scoreboard.


11. **Lives:** The player has three lives. Lives refer to a finite number of tries before the game ends with a game over. Also there are life gems during the game, players can collect life to gain extra lives.

12. **A little mercy to the hero:** When our hero dies, he has some time to revive himself. In that time, if he counters an obstacle he doesn't die.

## 4. Project Modules

- ★ header.h ( Render object class and class function. initialization, close and other necessary function and variables)
- ★ loadmedia.h( loading all kinds of media like image, sounds, fonts etc)
- ★ gui.h (menu and other necessary screen)

Inside **header.h** there is a class to render object

**Texture:** All attributes and methods are public

| | |
|---|---|
| Attributes | mTexture: SDL_Texture<br>mWidth: int<br>mHeight |
| Methods | • LoadFromFile(path:string):bool [to load images]<br>• loadFromRenderedText( std::string textureText, SDL_Color textColor ): bool [to render a text]<br>• void render( int x, int y, SDL_Rect* clip = NULL, double angle = 0.0, SDL_Point* center = NULL, SDL_RendererFlip flip = SDL_FLIP_NONE ); [to render an object]<br>• setColor( Uint8 red, Uint8 green, Uint8 blue ): void [set color modulation] |

| | |
|---|---|
| | ● setBlendMode( SDL_BlendMode blending ): void [Adjust color Blending]<br>● setAlpha( Uint8 alpha ): void [set alpha modulation]<br>● free(): void [Delocates texture]<br>● getWidth():int [return object width]<br>● getHeight():int [return object height] |

## 5. Team Member Responsibilities

1. Md Jawad Ul Islam Abir: Project ideas, Menu and other screen implementation, Map1 water obstacle added, jump implementation, resources collection, debugging and testing.
2. Arif Billah: Map1 ideas, Map2 obstacle and character collision detection, character rendering and moving, settings, sounds, fonts, scorecard saved and restoration (bassically file handling), bonus and life addition, testing and debugging.
3. Tonmoy Banik: Map2 ideas, Map2 rigid obstacles and movable obstacle implementation, Map2 scrolling and score checking and all kinds of resources collection.

## 6. Platform, Library & Tools

Platform: Linux

Language: C, C++

Library:

- build-essential
- libsdl2-dev
- libsdl2-image-dev
- libsdl2-ttf-dev
- libsdl2-mixer-dev

Tools: Sublime Text, Codeblocs, Gimp

## 7. Limitations

1. **Lack of variety of obstacles:** We wanted to put several types of obstacles (Both static and dynamic), but we weren't able to bring the variety in the obstacles as much as we wanted.

2. **The absence of the hero's super powers:** We wanted to give some super powers to our game character like shooting, running slowly for a few seconds to overcome the obstacles more easily etc. But failed to do so as the libraries couldn't provide much facilities.

3. **Collision detections are not perfect:** We have implemented the "2D Two Rectangle Rectangle'' collision method to detect collision between character and obstacle which has vulnerabilities.

## 8. Conclusions

Most importantly, we had the basic idea of how games are actually created. We learned about the cross-platform development library SDL(Simple DirectMedia Layer) in C++. We learned how to work in a project with teammates, for example, how to distribute work among the group members. Though at the very beginning, it was not easy at all for us, as everything was kind of completely new to us. We had hardly any idea about SDL and creating games. But gradually, everything started making sense as we were approaching the procedure.

## 9. Future plan

We want to make this game more interesting and super addictive.We would like to add some stories and some more characters, mainly villains. So, our hero can fight with them. We write our code in a way that has made future extensions and updates very easy and fast. We may add some more maps and give our hero some super powers. We also hope to link between maps and make different heroes for different maps. And we want to implement the “Per Pixel Collision method” which will give more accuracy to detect collisions.

And also we would like to implement some more settings like for character speed control, frame rates control and saving slots.

# *Repositories*

*GitHub Repository: https://github.com/arfbllh/Goofy-Pinkman-on-the-run*

*Youtube Video:*

# *References*


1. https://wiki.libsdl.org/
2. https://opengameart.org/
3. https://www.deviantart.com/
4. https://lazyfoo.net/