

LASSO - THE GAME

Course project for CS-101
2020-2021 Abnormal Semester

Asif Shaikh

20d070017

Electrical Engineering Dual Degree

Submission to:
Computer Science and Engineering Department
Indian Institute of Technology, Bombay
February 26, 2021

Contents

	Page
1 General Information	2
1.1 Link to video presentation	2
1.2 General information	2
1.3 Additional features implemented over the basecode	2
2 Instructions	4
2.1 Navigation	4
2.2 Controls	5
2.3 Description of actions	5
2.4 Instructions for running the program on Windows OS	6
3 Classic Game Mode - Description	7
4 Arcade Game Mode - Description	8
5 About the code	10
5.1 Changes to the base code	10
5.2 Additions	10

1 General Information

1.1 Link to video presentation

https://drive.google.com/file/d/13F-rdWH7QI9mTRruB_zZwcLlR5W1ZKAF/view?usp=sharing

Dont mind the birds chirping!

1.2 General information

My name is Asif Shaikh, Roll Number-20d070017, Electrical Engineering Department.

† This is a detailed report of the project that I have developed over the past month.

† The game is loosely modelled on the arcade game- Fruit Ninja.

† The project is made to resemble a general game in terms of navigation, controls and features.

† Enough emphasis has been given to handling corner cases for all the added enhancements, making the game more user friendly and making controls more intuitive. Hence the game, to my knowledge does not show any error or break down for ill-defined or illegal control inputs or for exceptional or corner cases, as these cases are handled by the program.

† It is my request to do the evaluation process on a WINDOWS OS and not on Linux System. The game has been developed entirely on a **WINDOWS** Operating System. A trial run on Linux failed for unknown reasons (the formatting of display was not as expected, some simplecpp library functions did not seem to work, the compiler gave some warnings and the game was very slow, some problems might have been due to the machine I used).

† The click interactions are better with a mouse than with a touch pad, though it is almost the same.

1.3 Additional features implemented over the basecode

† Made two separate game modes- **Classic** and **Arcade** with different features.

† Made multiple screens and pages; buttons on each page to navigate from one page to other.

- † Made the coins go in a parabola.
- † Made multiples coin appear at a time; at random times.
- † Made coins to appear at random places.
- † On catching coins, the change in score and the power ups provided are visible near the lasso for some duration.
- † Made multiple types of Coins with different abilities.
- † Made bombs which should not be collected; score and or 'life' is subtracted if a bomb is caught.
- † Made a Super coin which awards more points than a regular coin.
- † Made a Time Coin which awards bonus time in Arcade mode if caught.
- † Created a Coin-Magnet for Arcade mode which converts the lasso into a magnet which attracts coins; magnet expires after some time.
- † Created a 'Frenzy' Coin. Many coins start falling from the top of the screen for a while.
- † Imposed a time limit for one of the modes(Arcade mode).
- † Limited number of "lives" remaining for player in one of the modes (Classic mode); lives get deducted if a bomb is not caught in Classic mode.
- † Made a realistic playername input box for the user to enter there name via keyboard before beginning of each game.
- † Made provisions for storing the data from every game(username, mode and score) and managing this data.
- † Made provisions for deleting the user progress (i.e. all the data in data file) and creating a backup.
- † Made a score board and a leader board which appears at the end of each game displaying some game statistics based on data from previous games.
- † Made a information page and instruction pages with brief text explaining key controls, rules, and key-components of each mode.
- † Removed some bugs from the original game.

2 Instructions

2.1 Navigation

The navigation for the game is quite intuitive and is entirely done by *click interactions*. The various screens and pages are listed below:

- † The first page encountered on opening the game is the **homepage**. Here the user can close the prompt by clicking on the **EXIT** and can access the General Information Page by clicking on **?Info?**. By clicking on the *play button*, the user is directed to the *Pre Game Page*. There is a very attractive button at the bottom which says **delete all data**, which any curious person would be tempted to press, the consequence being that all the data from previous games is deleted and the leaderboard for upcoming games would be empty. But fear not if you wish to undo things as a backup is stored in the *DataBackup.txt* document. Just copy paste it into the *Data.txt* document manually.
 - The General Information Screen contains some general Information about the game and a short description of the modes. On clicking the *Exit* button on the upper right corner of the screen, the user is redirected to the home page.
- † In the *Pre Game Page*, the user can choose between one of the two modes - **Classic** and **Arcade** to play from by clicking on the name of the two modes. Below the name of each mode there is a link to the instruction sheets for each of the two modes. Here, the user can also change the player-name(which is *player1* by default) to anything of their choice. **The player-name cannot contain spaces, should not be blank and should be less than 9 characters long..** On entering the name, the user must press enter(the user wont be able to click anywhere else before clicking enter). On clicking the *Exit* button on the upper right corner of the screen the user is redirected back to the homepage.
 - The instruction pages contain information about the type of coins in each mode, rules, and controls. In each of the two pages there is an *Exit* button on the upper right corner of the screen clicking which, redirects the user to the *Pre Game Page*.
 - The instructions for arcade mode are two pages long hence the user must click on the next and back button on these pages to toggle between the two.

- † The two modes have a single *Exit* button on the upper right corner of the screen which can be used to quit and end the game midway. When the player quits/ the game ends, the user is redirected to the score page.
- † The **Score page** contains the score board which displays the score of the player, and their personal high score and the leader board which displays the top 3 players in terms of their scores. On clicking the *Exit* button on the upper right corner of the screen the user is redirected back to the homepage.

2.2 Controls

The controls for both the game modes are same. Keyboard buttons are used for all the controls.

- † Press the ↑ arrow key to throw the lasso.
- † Press the ↓ arrow key to yank the lasso.
- † Press the → arrow key to rotate the lasso anticlockwise.
- † Press the ← arrow key to rotate the lasso clockwise.
- † Press the **space bar** key to loop the lasso.

2.3 Description of actions

- † Throwing action - throws the lasso loop in the direction in which the lasso is oriented.
- † Yank action - Pulls back the lasso loop.
- † Loop action - Loop the lasso to catch the nearby coins! Doesnot work when the lasso is in yanked/rest position.
- † Rotating action - Rotate the lasso to change the direction of initial throw. Works only when the lasso is in yanked position

2.4 Instructions for running the program on Windows OS

I am providing the Instructions for running the program on Windows OS, for ease and for the sake of completeness. This is the method that I used:

- † Make sure that you have the simplecpp and codeblocks installed on your machine. If not use the following link- <https://www.cse.iitb.ac.in/~ranade/simplecpp/>.
- † Now, we need to add the simplecpp compiler to our environment variables. Search and open *Edit the system environment variables* on windows.
- † Click on *Environment Variables*.
- † In the System Variables click on *Path* and then click on *Edit...* . **Don't delete anything.**
- † Click on *New* in the left pannel and then click on *Browse....* Now in the drive in which you have installed simplecpp, select *Program Files (x86) > Simple Codeblocks > MinGW > bin*. Click on *OK*.
- † Click on *Move up* to move the path variable just below the system 32 ones. Click on OK on each window. The path variable is now set.
- † Open command prompt and change directory to the one containing the program files for the game. Enter the following command: `g++ -std=c++11 -Wall -o lasso lasso.cpp MovingObject.cpp coin.cpp BigWords.cpp screens.cpp HomeScreen.cpp preGameScreen.cpp InfoScreen.cpp classicInstr.cpp arcadeInstr.cpp postGame.cpp Classic.cpp Arcade.cpp main.cpp -lsprite -lbgi -lgdi32 -lcomdlg32 -luuid -oleaut32 -ole32`
- † Now type *lasso* and press enter.

3 Classic Game Mode - Description

Classic mode is one of the two modes in Lasso - The Game. This mode is based on the game mode of the same name in the game *Fruit Ninja*.

- † The player has to catch gold coins which appear on the screen to gain points. Each gold coin awards the player with 1 point. Occasionally bombs appear on the screen which the player has to avoid. The game ends when the player has caught five bombs. Additionally catching the bomb deducts player score. There is no time limit in this mode. On catching a coin with the loop, the total change in score is visible near the lasso for some time. If the score does not change, no message is visible.
- † As the game progresses the occasionally appearing bombs become more and more frequent. Thus the difficulty level of this mode increases with time. The player can view the number of bombs out of 5 which appear on the upper left corner of the screen. The player score is visible on the upper right side.
- † On the two sides there are blue panels with slightly different colours. The coins appear from below the screens between these two panels which is the main playing area. The player can quit the game by clicking on the *Exit* button on the upper right corner of the screen before the game ends.
- † After the game ends or the player quits, the page with the scoreboard and the leader board comes up. Here the user can see their current score and their personal best score. If the player breaks their record, then a message saying the same also comes up on the screen.
- † The Classic mode does not have any power up coins, but is more challenging than the Arcade mode. It is simpler to understand. It is similar to the Classic mode in Fruit ninja in which the player has to avoid cutting 3 bombs and cut maximum number of fruits.

4 Arcade Game Mode - Description

Arcade mode is the second mode in Lasso - The game. This mode is also loosely based on the Arcade mode of the game -*Fruit Ninja*.

† The game has a time limit of 2 minutes at the beginning of the game. The player has to maximize his score by the end of the game. The game has many different type of coins:

- The **Gold coin** is the coin which regularly appears on the screen and awards the player 1 point on being caught.
- The **Bomb** also regularly appears on the screen. The player will loose a point if the bomb is caught.
- The **Super coin** is large and orange in colour. It awards the player with 5 points on being caught.
- The green coloured **Time Coin** awards the player with 15 bonus seconds of run time. This coin is rarer than the above mentioned coins
- The **Coin Magnet** is a grey coloured power up coin. On catching this coin the lasso gets magnetized for 10s. This makes all the gold coins to be attracted towards the lasso. On catching this coin the back ground colour changes and a countdown timer for the coin magnet to run out is displayed on the top of the screen. At the end of this timer the lasso loses its magnetization, the coins start moving normally, and the background colour returns to normal. This coin, too is rarer than the first 3 coins.
- The **Frenzy coin**, is the last type of coin. Frenzy means wildness or uncontrolled behaviour, because this coin literally causes a shower of gold coins to fall from the top of the screen. For a while coins keep falling. The shower coin along with the coin magnet is the most fun part. This coin too, is rare.

† On catching any of the above coins a message displaying the change in score and the power up provided by the coin is visible near the lasso for a while.

† There are two orange coloured panels on each side of the screen. On the left hand side panel, the player can see all the different coins and their abilities listed below them. On the upper left corner, there is a timer which displays the amount of time remaining before the game ends. The user can quit the game before the timer runs out by clicking on the *Exit* button visible on the upper right corner in the right hand side panel.

- † After the game ends or the player quits, the page with the scoreboard and the leader board comes up. Here the user can see their current score and their personal best score. If the player break their record, then a message saying the same also comes up on the screen
- † The arcade mode has many additional features as compared to the Classic mode. There are a lot more things, to know here but most of the features are made to be intuitive, or as the player might expect the to be. The Arcade mode in the *Fruit Ninja* game has similar power ups and time limit.

5 About the code

5.1 Changes to the base code

The Coin class has been heavily modified in order to include all types of coins. The lasso class functions for catching the coins have also been modified accordingly. Additionally some functions were modified so that the lasso cannot be looped while it is in yanked position, and the lasso must not count the coins which it has already caught when it is looped every time. These were some bugs in the older game. Additional functions for magnetizing and demagnetizing the coins have also been included in the lasso class. The main program has been changed to only navigate between the screens which are member functions of the *screens* class.

5.2 Additions

- † The screens class is the class made for managing various screens. The screens, which are member functions of this class have been implemented in different files. Screens.cpp contains some simple functions for these screens and a function for managing the data file.
- † A data file has been created for storing the scores of the player. The code has been written in such a way that this file does not need any manual management. This has added a memory element to our game and made the game more realistic.
- † For screens the main idea was that auto deletion of the screen objects once these screens run out of scope. The pre game screen is able to take user input for their name. This interface was tried to be made to look as realistic as possible. I also wanted to implement an additional settings menu, but time was not available.
- † Random behaviour has also been integrated into the two game modes. The coins appear at random times, at random places, hurled approximately towards the centre of the screen. The coins later get reset to a random position. The coins are unpaused by a certain probability factor.
- † To synchronise the game with actual time, a buffer time is used for the wait function instead of the step time to account for the time consumed in each loop, so that the game clock runs in sync with a real clock.

- † A coin pointer vector is used in each of the two modes to generate coins on the stack memory. The only drawback (or is it?) of using it is that they are to be deleted at the end of each game which takes some time, as they don't disappear like other objects which run out of scope. If instead, a regular vector of coins were used the program takes more time to close.
- † While adding the data file, enough care was taken so that the user is not needed to directly interact with the data file, at any time. The only time this is needed now is when the user has accidentally deleted the data and wishes to recover it from the backup file.
- † The player name is saved in the data file before the start of each game and the score is stored after each game. If the user forcefully ends the game in between by closing the Canvas the score page might not work properly in the next game. To prevent this, a guard number is stored before the score. Its working is mentioned in the program file, in the comments.

End of document