Project Description: The Lasso Game

CS101 2020-2021 Abnormal Semester

Background

- What is a **lasso**? See link
- You are given the code for a simple game called **lasso**, based on simplecpp graphics
- As you can find out by playing and/or looking at the code, the game is about throwing a lasso to catch coins
- How to compile and run?
    - `s++ -o lasso lasso.cpp MovingObject.cpp coin.cpp main.cpp`
    - Run the program after compiling by typing `./lasso`
    - For windows, please see link
    - If you are adding more code to windows in another file, say `xyz.cpp` , you can include it in the main file by using `#include "xyz.cpp"`
    - Note that in general it is not a good idea to include CPP files in another CPP file; you are doing this only to get around the restriction of single file in codeblocks novice version
- Initial key controls: you can figure out the key controls through a combination of reading the given code, and by actually playing the game

What to do in the project?

- In the project, you have to develop this game further
- A suggested list of possible enhancements are given below.
    - Each suggestion can be a game "level"
    - You can choose a suitable subset of suggestions to implement.
    - You can also be creative and come up with your own game levels and variations, so long as you stick to the rules given below.
- Suggestions for enhancements:
    - Make the coin go in a parabola
    - Make more than one coin appear at a time; at random times
    - Make bombs which should not be collected; score is subtracted if you catch a bomb
    - Impose a time limit for a game level
    - Some number of "lives" remaining for player; lives get deducted if "level" is not completed
    - Some coins which definitely need to be collected; they will deduct points or cost lives otherwise
    - Instead of coins, alphabets come up, which need to be caught to make a word

    - If you are math person: have a linear equation on the side, numbers get thrown instead of coins, numbers satisfying a certain property have to be caught
    - Once in a while, a magnet is thrown; if caught, it has the ability to attract nearby coins; magnet expires after a delay
    - Some types of coins turn your lasso into a magnet which repels coins; magnet expires after delay
    - In the default game, the lasso can be made faster/slow with key-presses; you will notice that having a faster lasso makes the game easier; you can allow lasso speedup only on catching a special “speedup” coin
    - Likewise, on catching a “slowdown” coin, the lasso should slow down
    - Catch magnets to repel bombs, with expiry
    - Catch gift box, open gift box to get one of: time extension, magnet, magnet to repel bombs, twice the points
    - Enter name, maintain high score against name; optionally store this info in a file to restore it on next run
    - Make a help page with brief text explaining key controls

Project Rules

- All effort must be INDIVIDUAL. No discussion with anyone including friends/family, no code sharing is allowed. No help from TAs as well. (TAs will clarify concepts as related to course).
- Make at least 3 enhancements, worth a total of ~~at least 500~~ about 300-350 lines of additional C++ code (line count under “normal” coding/indentation).
- Make a short help document giving key controls (or you could include this as part of the game itself)
- Make a short 4-5 min screen-recording video showing the features of the game; max video size: 15MB (you can use vlc to compress video)
- Submission of code, document, video will be on BodhiTree
- Deadline for submission: Sun 28 Feb 2021, 23:59 IST

Evaluation Plan

- Evaluation will be done largely by TAs
- Evaluation will be based on: (a) program working, (b) how well the code is written, (c) good programming practices such as comments, indentation, appropriate variable/function names.
- Project will be pass/no-pass
- Instructors’ decision will be final
- Pass means: AB → AA
- No-pass means: no change in grade obtained otherwise
- Reminder: only way to obtain AA in this course: get AB in normal evaluation, AND go a successful project