CSCE 4255 /5255 (Game Math and Physics)

Program 5 (Final Project)

Due at the start of class on Thursday, December 2, 2021 (the last class day)

Your final project, worth twice as much as a regular programming assignment, is to create a Rube Goldberg Machine using Box2D. The Rube Goldberg Machine, named for the cartoonist Reuben Garrett Lucius Goldberg (1883 –1970), is a machine that performs a simple task in a complicated, arcane way. The following example[1] was originally published in *Collier's* in 1931.

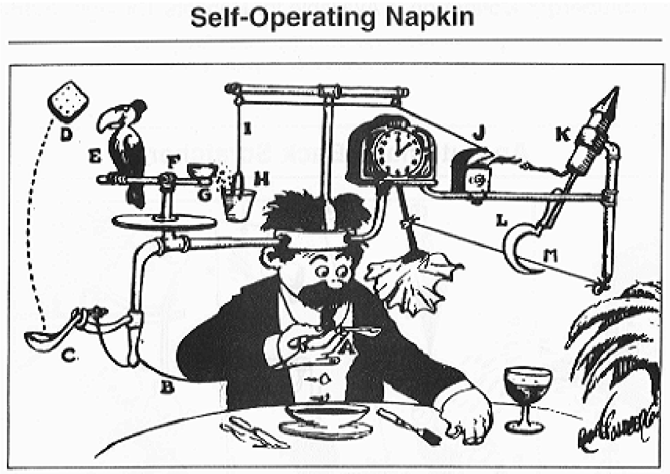

Figure 1: A Rube Goldberg machine.

Your `csegitlab` repository now has a folder called `Program-5` containing a very rudimentary Rube Goldberg Machine, the aim of which is to push the red button at the lower left of the window. There is a timer at top-right of the window that keeps track of elapsed time.

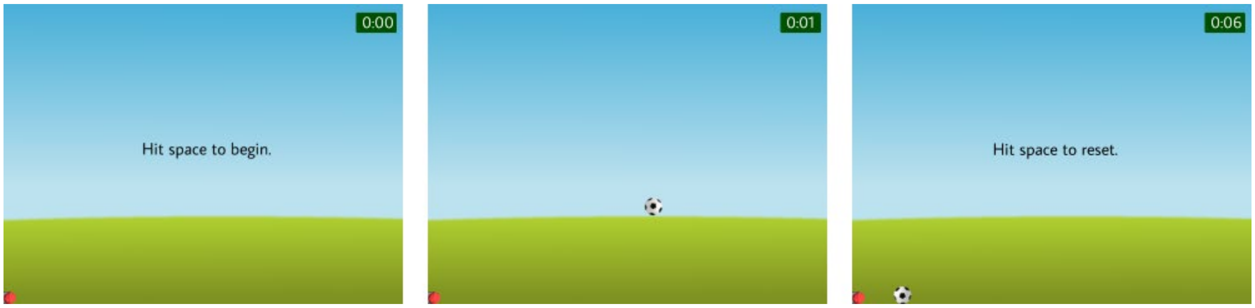

Figure 2 The rudimentary initial Rube Goldberg machine before it is started (left), while it is running (center), and after the ball has hit the red button (right). Note the red button at the bottom right of the screen and the timer at top-left.

You are to modify the code, images, and sounds to make the most awesome Rube Goldberg Machine that you can. Videos from past classes will be posted on Canvas to give some idea of what to do – and what not to do!

[1] Public Domain Image, https://commons.wikimedia.org/w/index.php?curid=9886955.

The Rules

1. Your Rube Goldberg machine must start when the player hits the space bar.
2. You must keep the timer code `CGame::DrawClock()`[2] in your Rube Goldberg machine, although you may change the timer's appearance provided it shows the elapsed time correctly.
3. Your Rube Goldberg machine must finish when something collides with the red button. You may change the button sprite and/or the sound that it makes in a collision.
4. After your Rube Goldberg machine finishes, hitting the space bar must reset it to initial conditions and it must run correctly again after the player hits the space bar a second time.

Grading

Points will be awarded for the following:

1. **Duration**: The amount of time it to run. Aim for something between 30 seconds and 2 minutes.
2. **Complexity**: Are there lots of things happening?
3. **Originality**: Including coolness, humor, style, etc.

Points will be deducted for lameness, including but not limited to the following:

1. **Boredom**: Time in which nothing happens (for example, waiting for balls to roll slowly).
2. **Appropriation**: Using too many things from the class code.
3. **Repetition**: Using too many copies of the same thing.

Final Presentation

Copy your `Release` mode executable and your `Media` folder into the folder `My Rube Goldberg Executable` in your `Program-5` folder before the start of class on the last class day. You will then take turns presenting your Rube Goldberg machine from that folder in class on the last class day (Thursday, December 2, 2021). Failure to attend[3] will result in a score of zero for this program.

[2] Make sure that you do not interfere with the `CCommon` variables `m_eGameState`, `m_fStartTime`, and `m_fTotalTime`, which the timer relies upon.

[3] The usual exceptions excepted.