Creating a CheatEngine like PinTool and get a high score in the Moon-Buggy game.

Description:

You are given the Moon-buggy game. Your goal is to use Intel Pin to make the moon buggy keep going even after crashes.

In short, watch the video. It shows two moon buggy programs. The first one is the original game. You will see it crashes if you do not properly jump to avoid the holes on the ground. The second one is a modified version. It shows even after it crashes, the buggy keeps going. Even though the buggy looks damaged, it still functions (e.g., keep going and firing).
* Live demo will be presented during the class.

You need to figure out the details of how to make the second program's buggy keep going. You are supposed to figure this out by yourself by analyzing the source code. For some hints, you need to look at "crash detection" mechanism. In other words, you need to focus, how the program detects the buggy ran into the holes and crashes the buggy.

Extra Challenge:

The score is proportional to the time you survive. Longer you survive, higher the score. Can you make the game give you a super high score within a short amount of time?
Hints: You need to find out how the score is calculated.

What to submit?

1. Your Pin tool code
2. Report that includes
   (1) high-level descriptions of how it works,
   (2) instructions and memory locations (i.e., variables) you identified by analyzing the program,
   (3) strategies to implement your Pin tool, and
   (4) explanations of your code (pin tool's code) -- per basic block.

How to compile and run the moon-buggy program:

The source code of the moon buggy game is attached. Download and extract it.
Install required packages.
* Required Packages: autoconf, automake, texinfo, libncurses5-dev, libncursesw5-dev

To compile and run, do the following commands.
./autogen.sh
./configure
make