1. **INSTRUCTIONS**

The main character (Solider) is controlled with WAD, it has the ability to slam to the ground by pressing the DOWN arrow and jump with SPACE. This is to increase accuracy in the movement of the solider.

With left click, the solider will shoot. The bullets spawn at the beginning of each level and need to be collected in order to shoot.

The game is speed run style, so the goal is to finish it as fast as possible. To advance to each level, all enemies present in the world must be killed and the Solider to go to the green portal.

*To get to the portal in level 2, you must collect the blue potion to fly up to the portal.*

1. **EXTRA FEATURE**

I created my own stopwatch mainly using System.nanoTime() to embrace the speed-run nature of the game. Its purpose is to track how long each user takes to finish the game. The stopwatch only starts when the user has submitted their name and pressed play to accurately track the time. The stopwatch can be started, paused and ended. It can be reset to a new set of values, used when loading a save file.

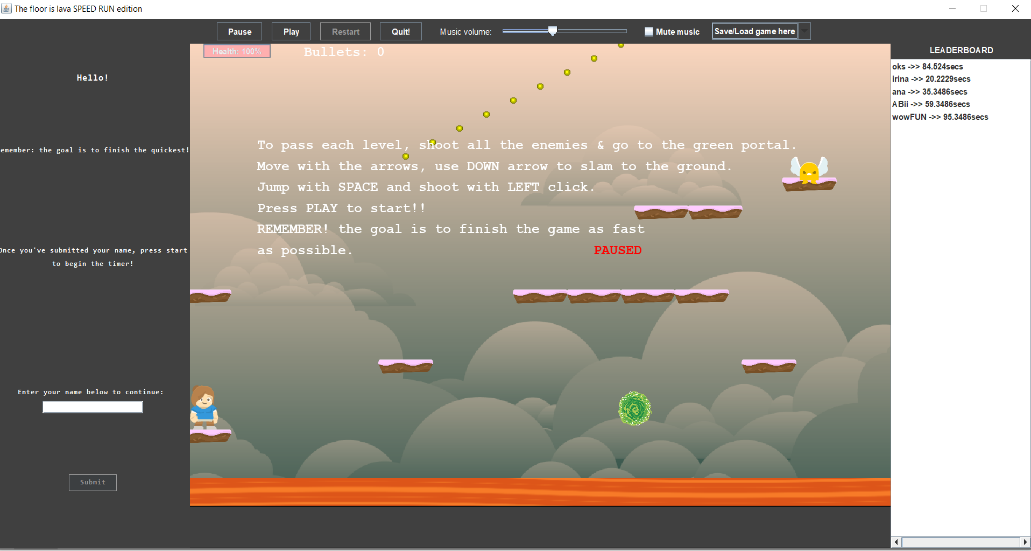
1. **SAVE/LOAD**

The save feature in my game stores: the level the user is on, the health and bullet count of the solider, the solider x and y coordinates in the world, the name the user gave at the start of the game and the current status of the stopwatch.

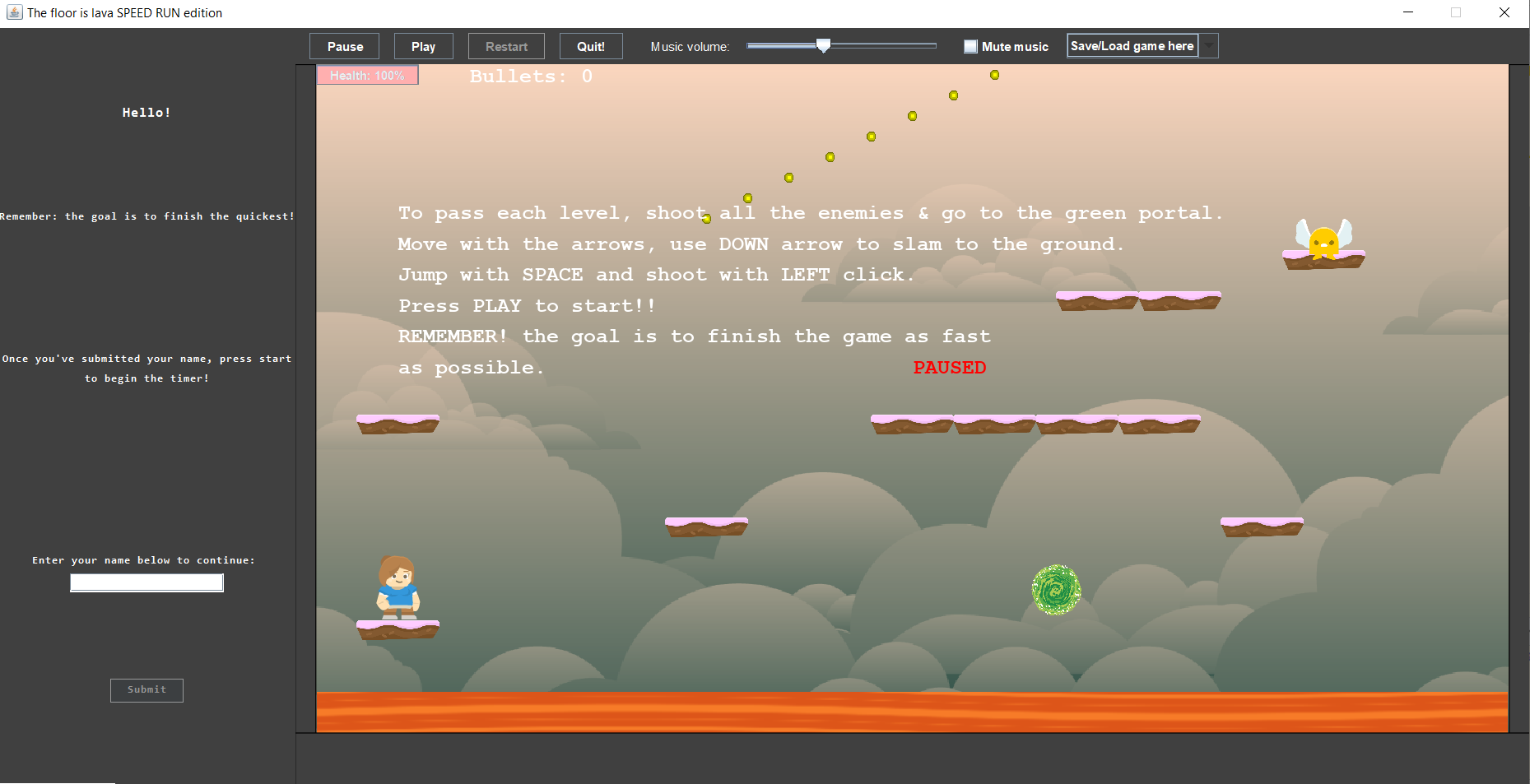
It is critical to store the status of the stopwatch so the user can resume their game seamlessly. The system nanosecond of when the stopwatch was started and the number of nanoseconds it was paused for is stored in the save file.

When loading a save file, the current world and main player(solider) is overwritten to the data from the file, as well as the current running stopwatch data is reset. The start nanosecond and the total seconds it was paused values are set to the ones from the save file. Depending on which level number is started, the correct background song is played.

1. **OTHER FEATURES**

To compliment the stopwatch, a leader board is created that holds each user and the time taken to complete the game in seconds to 1 decimal point. It has a fixed size with a vertical and horizontal scrollbar to preserve the look of the game window, and it is placed to the right of the game. If there are no entries in the leader board, it will not be displayed to maintain the neat appearance of the game.

*Window with entries in the leader board.*



*Window with* ***no*** *entries in the leader board.*

Another feature is sound and the GUI to control it. There is one song for the first 2 levels, at level 3, a more alert song begins to indicate the game is becoming more intense. This is to compliment the falling spike in level 3 that is triggered when it reaches a certain point in the game, it also halves the health of the solider upon collision. Once the 2 enemies are destroyed in the final level, level 4, a happier upbeat song is played as the player makes its way to the portal.

The GUI for the music is on the top of the game window, it contains a scroll bar and a “Mute” check box. The scroll bar goes from muting the music to maximum volume. A more advanced feature is the “Mute” check box, in which I had to make sure the music is still muted even if the user is moving the scroll bar or the song in changed in level 3.

The solider shooting is directional. If the cursor is to the left of the solider, the bullet will shoot at a 45-degree angle to the left of the solider. The same applies if the cursor is to the right of the solider.

The health of the soldier is transferred from one level to the next one. This feature is implemented with the use of text files. If the level is restarted while playing, the solider health is set to the value from the previous level seamlessly.

The game has a FSM implemented as a String attribute of the Game class. The states are: “paused”, “play” and “saved”. This feature is mainly used to display helpful messages on the window, such as, “PAUSED” when the game is paused or “Saved progress” when the game is saved by the user.

Timers are used to make the solider fly for approx. 3 seconds when it collects a potion and to display the “Saved progress” message on the screen for one second when the user saves the game.

1. **JAVADOC CLASSES**

All the classes in my game are documented, but the most detailed ones are game.GameLoader, game.GameSaver & game.StopWatch.