Callan Winfield

Technical Documentation

AIE - Assessment 1 Retro Game

This document is an overview of the Mouse hunter game.

This game does not make use of the aie framework and had been done in the windows console.

As such it makes use of Ascii art in place of 2D sprites.

This document contains the following technical breakdowns:

1 - Breakdown of classes used in this project
2 - UML diagram
3 - Collaboration Diagram
4 - State Diagram

Class Diagram
This class diagram demonstrates the hierarchical relationship of the classes created for this game. Take note of the sprite class, anything that draws to the screen must be aware of it, this is a stark comparison with the Leaderboard class which only the game class is aware of.

State Diagram
this state diagram demonstrates xyz (hint game states) … (2 to 3 lines of text)
refer to image blah blah blah

Sequence Diagram
this dequence diagram demonstrates xyz (hint what we did on the bort the other day) … (2 to 3 lines of text)
refer to image blah blah blah

Feedback Overview and Changes
The following feedback and changes are recomended from users playing the game.

1. Richard L — Mice where to hard, I couldn't see the cheese/ tell them apart from the mice. I recomend using a 2D engine instead of ascii art.
2. Rob S — Game was fairly well balanced, mice where hard to hit. I recomend taking a look at the mouse AI
3. Suzanne N — I found it hard to know how many cheese and mice where left. I recomend adding a count for these to the HUD

File Input Output
The following files are

1. human_readable_leaderBoard.txt(write only) - A human readable version of the leader board is generated on game end
2. leaderBoard(Read/Write) - sqlite3 database

SQLight - LeaderDB
The scores are saved the the sqlight3 database.
A single table is created to contain the score. The table is formatted acording to the following statement.

The create command is submitted using the Sqlite3 prepare statment.
The prepare stament will return a "misuse" code if the table exists which the function then returns as a "false" value.

```
"CREATE TABLE leaderboard("  \

      "ID             INTEGER PRIMARY      KEY," \
      "NAME           TEXT                 NOT NULL," \
      "SCORE          INT                  NOT NULL);";
```

Feedback Overview and Changes

The following feedback and changes are recomended from users playing the game.

1. Richard L — Mice where to hard, I couldn't see the cheese/ tell them apart from the mice. I recomend using a 2D engine instead of ascii art.
2. Rob S — Game was fairly well balanced, mice where hard to hit. I recomend taking a look at the mouse AI
3. Suzanne N — I found it hard to know how many cheese and mice where left. I recomend adding a count for these to the HUD

File Input Output

The following files are

1. human_readable_leaderBoard.txt(write only) - A human readable version of the leader board is generated on game end
2. leaderBoard(Read/Write) - sqlite3 database

SQLight - LeaderDB

The scores are saved the the sqlight3 database.
A single table is created to contain the score. The table is formatted acording to the following statement.

The create command is submitted using the Sqlite3 prepare statment.
The prepare stament will return a "misuse" code if the table exists which the function then returns as a "false" value.

```
"CREATE TABLE leaderboard("  \

      "ID             INTEGER PRIMARY      KEY," \
      "NAME           TEXT                 NOT NULL," \
      "SCORE          INT                  NOT NULL," \
```