ACOGHorseRacing Plugin - Requirements Specification

Dave Haenze

February 6, 2014

Contents

1   Introduction

This section introduces the requirements for ACOG's Horse Racing plugin. It provides the purpose and scope of the plugin as well as explaining all references and acronyms. Finally, it outlines the rest of the requirements specification.

1.1   Purpose

The requirements specification document describes the functions and requirements for the Horse Racing plugin for ACOG's Minecraft server. The plugin aims to address the users' want for a way to run and manage horse races. The resulting software shall provide the server's players with the requested functionality, such as creation of race tracks, joining and running races, and managing monetary rewards. Much of that functionality is already available in vanilla Minecraft, however this plugin aims to automate it and reduce the possibility of users cheating.

1.2   Scope

The AHR plugin is designed to provide new functionality to AMG members wanting to organize horse races. It will provide specified users with the ability to create a race track. Creating a race track will require specifying the start line or starting gates, the finish line and the sides of the track. Two types of tracks can be created, tracks beginning with a start line, and tracks with starting gates. Once a track is created, users can join a race on the track. If no races are running on a specified track, a user can create a race to run on said track. During creation, they can specify a number of racers, which is between two and the number of gates (if it is a gate race) or width of start/finish lines divided by two (as each horse takes up two blocks). Once created, other users can join by command, and are teleported to the start line/a starting gate. The race starts in two ways: the race creator starting it with a command, or automatically once each racer has voted to start. Upon the beginning of a race, the gates open (or users are allowed to leave the start line). As each user crosses the finish line, their place and time is posted. Any user that doesn't finish the race within a specified amount of time after that will be disqualified (DNF, did not finish). Upon the completion of the race, users are teleported outside of the race track.

Statistics will be compiled in a database, with four tables. These tables shall be Players, Horses, Race Tracks and Races. Players will contain columns for UID, player name, races won, races lost, winnings, average place. Horses will contain HID, horse name, speed, jump height, jockeys, races won, races lost, winnings, average place. Race Tracks will contain RTID, race track name, creator, location (X,Y,Z coordinates), type, max

number of racers, races. Finally, Races will contain RID, player names, horse names, times, places, bets.

Users will be allowed to gamble on races. This is done by command, allowing a user to select a race to gamble on, then which player/horse to bet on and how much. Based on the databases, odds will be generated. While betting, users will be able to bet an amount in several different currencies (most likely four), such as diamonds, gold ingots, iron ingots and emeralds. Upon completion of the race, users will automatically be given a payout in the currency they origially bet on (that is, a user that bets diamonds will receive diamonds, a user that bets gold ingots will receive gold.

1.3   Definitions, Acronyms and Abbreviations

- **ACOG** – Aberystwyth Community of Gamers, Aberystwyth University's gaming society

- **AMG** – Aberystwyth Minecraft Group, a subgroup of ACOG's, catering to the large community of Minecraft players

- **AHR plugin** – The ACOGHorseRacing plugin

- **Vanilla Minecraft** – unmodified Minecraft, either single player or multiplayer

- **Op or Ops** – short for "operator(s)", users with some administrator privileges on a server

- **UID** – User Identifier

- **HID** – Horse Identifier

- **RTID** – Race Track Identifier

- **RID** – Race Identifier

1.4   Overview

This document provides a high-level description of the ACOGHorseRacing plugin. It identifies the involved users and helps explain their roles. The document then describes general software and hardware constraints as well as any assumptions and dependencies concerning the system. The majority of this document focuses on the specific requirements list. A master list of specific requirements is given first, followed by each requirement explained in detail in the next section.

2   General Description

This section is used to provide a high-level description of the plugin, as well as identify users involved and help explain their roles.

2.1   System Functions

2.2   User Characteristics

There are five main groups of users that will use the plugin. Note however that, unlike in most cases, these groups of users are not necessarily going to be exclusive. Users can be a part of more than one group (though not always simulatenously).

The first category of users are ops, or operators, that administrate the server. They will have the ability to create race tracks. They will need to know how and where to create the tracks. They will be the only users to have the ability to create race-tracks in the plugin, to keep malicious users from creating a race-track in, for example, someone's home.

The second group of users are race creators. Any player on the server will have the ability to create a race, provided the creator of the race-track has not requested that only certain users be allowed to race on their track. They will create a race by specifying the arena, the number of racers and the number of laps.

The third group of users are racers. These are people that join a race made by a race creator. They will have access to the /join command, in which they specify the race they wish to join. Additionally, they will be able to vote to start a race if there are fewer racers present than the max number of racers.

The fourth group of users are spectators. They will be able to gamble and place bets on different races. Provided they are within a certain distance of the track, they can place bets on individual racers, either the jockeys or the horses themselves.

Finally, the fifth group of users are general users on the server. They will be able to view the database by command, giving them access to a leaderboard (top 10 horses/jockeys), viewing individual horses/jockeys stats, and viewing overall stats for a specific track.

3 General Constraints

3.1 Software limitations

- JRE v7 is used to run this plugin.

- Servers running the plugin will be required to use Bukkit 1.7.2-R02, as this is the build that the plugin will be tested with. Older versions will not necessarily work, and neither will new versions. Any Bukkit builds pre-1.6 will definitely not function, as horses did not even exist in Minecraft then.

- Servers will have to have access to a MySQL database if stats are to be enabled.

3.2 Hardware constraints

- A server wishing to run this plugin will require the same specs as the test server.

4 Assumptions and Dependencies

- The plugin assumes users have sufficient skill to use simple commands in Minecraft.

- Saving and displaying statistics is dependent on MySQL

5   Requirements Master List

This section contains the listing of all requirements for the ACOGHorseRacing plugin. The list contains unique requirement numbers and names with a short description of each requirement. The following section describes these requirements in full detail.

- **REQ 1**: Race Track Creation – The plugin will allow operators to create race tracks, specifying type of race track, starting line, finish line, and spawns/max number of racers.

- **REQ 2**: Race Creation – The plugin will allow users to create races, specifying the track, number of racers and number of laps (default 1).

- **REQ 3**: Joining a Race – The plugin will allow users to join a race, up to the maximum number of spawns available for said race.

- **REQ 4**: Starting a Race – The plugin shall begin a race automatically a certain amount of time after the max number of racers has been reached, or once a majority of racers present have voted to start.

- **REQ 5**: Running a race – The plugin will announce the start of a race and keep track of the each racers position. It will warn users if they are leaving the race track, and disqualify any attempted cheaters.

- **REQ 6**: Finishing a race – The plugin shall report each users place and time upon successfull completion of a race, and disqualify users who do not finish within a configurable amount of time after the first person finishes.

- **REQ 7**: Statistics generation – The plugin can be configured to have stats saving on or off. If on, it will automatically save the results of each race to a database, and add their respective results to each players/race tracks/horses existing row in the respective table.

- **REQ 8**: Statistics viewing – The plugin shall allow users to view statistics, such as a leaderboard, available for horses and jockeys seperately, both overall and per race track. The plugin will also facilitate viewing a specific horse's, player's, race's and/or race track's individual statistics.

- **REQ 9**: Gambling – The plugin can be configured to allow users to place bets on a race. If enabled, users can place bets on a race. These bets are placed in various currencies (with configurable eligibility and exchange rate between them). If statistics are enabled, odds can be generated automatically as well.

6   Requirements Detail

6.1   REQ 1: Race Track Creation

6.1.1   Description

The plugin will allow operators (or other users that are give the permission) to create race tracks. This process is invoked whenever a users with sufficient privileges issues the `/ahr createtrack <name> [track type] [lap boolean]` command. The available choices of track type are `line` and `gate` for having a starting line or starting gates, respectively. Depending on the type chosen, the user will be given different instructions. If a starting line is chosen, the user is prompted to select a line of blocks - this will be the starting line. This is done by selecting the first then last block of the line. After selecting the starting line, the user is prompted to create "spawns". This is done by selecting a 2x3 area, with the one of the short sides being directly adjacent to the starting line. In the case of a race track with gates being created, the user is prompted to select starting gates, by clicking the first gate in the line and then the last. Finally, the user is prompted to select the finish line - this is done by selecting the first and last block of the line, just like with creating a starting line. Lastly, the user is given the choice to select a "delimiter", a block that separates the race track from the rest of the world.

6.1.2   Input

Several types of user inputs are used. First of all, there is the `/ahr createtrack <track type>` command. Secondly, the user selects various parts of the game world to create start and finish lines, as well as the "delimiter" block. Lastly, the user gives the track a name.

6.1.3   Display

At each stage of track creation, the system will output a message to the user, prompting them to complete the next stage of race track creation. This should be a simple, unambigious and easily understandable message told directly to the user.

6.1.4   System Processing

Once the user has finished creating the track, it is added to either a) the MySQL database (if it is enabled) or b) a flat file. The database/file will be updated with the location and name of the track, the number of spawns, the location of the start line/gates and the finish line.

6.1.5   System Output

At each stage of the track creation process, the system will output messages such as "Start line selected!" or "Starting point created!", confirming that the user has completed these stages.

6.1.6   Other

N/A

6.1.7   Constraints

N/A

6.1.8   Data Handling

The database/flat file is updated each time a track is completed. At each stage of the track creation process, the data input by the user is validated to be certain that correct data is entered. Examples include preventing the user from selecting anything but a 2x3 area as a starting point, only allowing a one block wide area be selected as a starting line, etc.

6.2 REQ 2: Race Creation

6.2.1 Description

The plugin will allow users to create races on select tracks, using the `/ahr createrace <track> [no of racers] [no of laps]` command. While creating a race, the user must provide the name of the track to race on. Optionally, they may set the number of racers (default is number of starting points on the track) and/or the number of laps (default is 1). Upon creating a race, the creator is automatically teleported to a starting point/gate. However, being mounted on a horse is required - if the user is not mounted, an error message will be displayed.

6.2.2 Input

`/ahr createrace`, the track name, the number of racers and the number of laps (the last two being optional).

6.2.3 Display

A message will be sent by the server, saying that the race has been created and displaying the data, e.g.,
`Race created on Race Track, with 4 racers and 3 lap(s)!`
    If incorrect options/parameters are provided, the system will output a warning message, such as `No such track exists!` or `Too many racers` or `Incorrect number of laps!`

6.2.4 System Processing

6.2.5 System Output

6.2.6 Other

6.2.7 Constraints

6.2.8 Data Handling

6.3   REQ 3: Joining a Race

6.3.1   Description

6.3.2   Input

6.3.3   Display

6.3.4   System Processing

6.3.5   System Output

6.3.6   Other

6.3.7   Constraints

6.3.8   Data Handling

6.4   REQ 4: Starting a Race

6.4.1   Description

6.4.2   Input

6.4.3   Display

6.4.4   System Processing

6.4.5   System Output

6.4.6   Other

6.4.7   Constraints

6.4.8   Data Handling

6.5 REQ 5: Running a race

6.5.1 Description

6.5.2 Input

6.5.3 Display

6.5.4 System Processing

6.5.5 System Output

6.5.6 Other

6.5.7 Constraints

6.5.8 Data Handling

6.6   REQ 6: Finishing a race

6.6.1   Description

6.6.2   Input

6.6.3   Display

6.6.4   System Processing

6.6.5   System Output

6.6.6   Other

6.6.7   Constraints

6.6.8   Data Handling

6.7   REQ 7: Statistics generation

6.7.1   Description

6.7.2   Input

6.7.3   Display

6.7.4   System Processing

6.7.5   System Output

6.7.6   Other

6.7.7   Constraints

6.7.8   Data Handling

6.8   REQ 8: Statistics viewing

6.8.1   Description

6.8.2   Input

6.8.3   Display

6.8.4   System Processing

6.8.5   System Output

6.8.6   Other

6.8.7   Constraints

6.8.8   Data Handling

6.9   REQ 9: Gambling

6.9.1   Description

6.9.2   Input

6.9.3   Display

6.9.4   System Processing

6.9.5   System Output

6.9.6   Other

6.9.7   Constraints

6.9.8   Data Handling