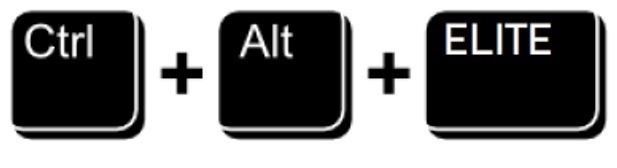
Target Acquisition Game

**

User’s Manual

Document

**Benjamin Lichtman, Grace Conner, Julian Gomez, Haley Donaldson, Connor Sullivan**

**11-30-18**

**USER'S MANUAL**

**TABLE OF CONTENTS**

Page #

[1.0 GENERAL INFORMATION 3](#_Toc531804710)

[1.1 System Overview 3](#_Toc531804711)

[1.2 Authorized Use Permission 3](#_Toc531804712)

[1.3 Points of Contact 3](#_Toc531804713)

[1.3.1 Information 3](#_Toc531804714)

[1.3.2 Coordination 3](#_Toc531804715)

[2.0 SYSTEM SUMMARY 4](#_Toc531804716)

[2.1 System Configuration 4](#_Toc531804717)

[2.2 Data Flows 5](#_Toc531804718)

[2.3 User Access Levels 5](#_Toc531804719)

[3.0 QUICK START GUIDE 6](#_Toc531804720)

[3.1 System Entry (Access admin page or user page) 6](#_Toc531804721)

[3.2 Player menu 6](#_Toc531804722)

[3.2.1 Play New Game 6](#_Toc531804723)

[3.2.2 High Scores 6](#_Toc531804724)

[3.2.3 Change Theme 6](#_Toc531804725)

[3.2.4 Quit 6](#_Toc531804726)

[3.3 Administrative Login (Gaining Access to the System) 6](#_Toc531804727)

[3.4 Administrative menu 6](#_Toc531804728)

[3.4.1 Change Speed 6](#_Toc531804729)

[3.4.2 Change Velocity Direction 7](#_Toc531804730)

[3.4.3 Change Size 7](#_Toc531804731)

[3.4.4 Change Direction 7](#_Toc531804732)

[3.4.5 Change Distance 7](#_Toc531804733)

[3.4.6 Export a csv 8](#_Toc531804734)

[3.4.7 Delete the current database 8](#_Toc531804735)

[3.4.8 Restore a database backup 8](#_Toc531804736)

[3.4.9 Visually seeing the content of the database 8](#_Toc531804737)

[3.5 Exiting the System 8](#_Toc531804738)

[3.6 Special Instructions for Error Correction 8](#_Toc531804739)

[3.7 Caveats and Exceptions 9](#_Toc531804740)

[4.0 FUTURE ENHANCEMENTS 10](#_Toc531804741)

[4.1 Future Functionality 10](#_Toc531804742)

[4.2 Similar Systems – Additional Functionality 10](#_Toc531804743)

[4.3 Maintenance Capabilities 10](#_Toc531804744)

# GENERAL INFORMATION

## 1.1 System Overview

* A target acquisition game with an administrative interface
* Allows administrators to modify gameplay variables e.g. size, speed, direction, etc. and users to play games, view scores and change themes.
* Web based game and administrative tools
* A game that includes music (Current track: *HumanMusic* by Benjamin Lichtman)
* Deployment
  + LAMPP/XAMPP Server
  + MySQL Database
  + GODOT export
  + Modification of IP specific files
  + Allow access on port 80
  + Debugging

## 1.2 Authorized Use Permission

Any unauthorized use of this system is prohibited, unless explicit consent is given by Dr. Joseph Chase.

## 1.3 Points of Contact

### 1.3.1 Information

* First Semester Design
  + Andrew McGuiness, Andrew Albanese, Ryan Kelley, Michael Hall
* Second Semester Design (Administrative functions + game modifications)
  + Benjamin Lichtman
    - [Balichtman@radford.edu](mailto:Balichtman@radford.edu)
  + Connor Sullivan
    - Csullivan22@radford.edu
  + Julian Gomez, Grace Conner, Haley Donaldson
* Client
  + Dr. Joe Chase
    - [jchase@radford.edu](mailto:jchase@radford.edu)

### 1.3.2 Coordination

To successfully coordinate the integrity of the system, several steps must be taken to ensure the reliability and functionality of the system.

# SYSTEM SUMMARY

## 2.1 System Configuration

* To complete the successful implementation of this system. The following are required:

1. Web Server (preferably apache via XAMPP/LAMPP)
2. Database (preferably MySQL via XAMPP/LAMPP)
   1. Compromised of two tables ‘test\_data’ and ‘game\_preferences’. The ‘test\_data’ table has the auto-incrementing column ‘ID’ which is the primary key, there are also columns ‘input’,’age’,’skill’,’score’,’theme’ and ‘Targets’. In addition to this, there are 35 columns for target misses and 35 columns for target time represented by the names ‘T#\_MISS’ and ‘T#\_Time’ respectively, where the ‘#’ symbol is a number 1-35.
3. Web Code (htdocs) {Modify IP specific lines in AdminPage.php, Login.php, TargetHunter.php

* The GODOT game files have been included in the ‘htdocs’ folder in the ‘370\_Project\_Game-master’ folder. If modifications are made to the game, it is critical that the game be re-exported using the provided custom export. These files may then be added to the ‘target\_hunter’ folder and the updates will be reflected.
* All references to ‘provided’, ‘login page’, ‘administrative page’, ‘user page’, and corresponding functionality is currently represented by the following information:

User URL: <http://137.45.241.80/>

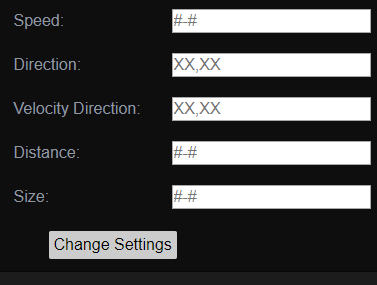
Administrator URL: <http://137.45.241.80/target_hunter/login.php>

Administrator Username: user

Administrator Password: pass

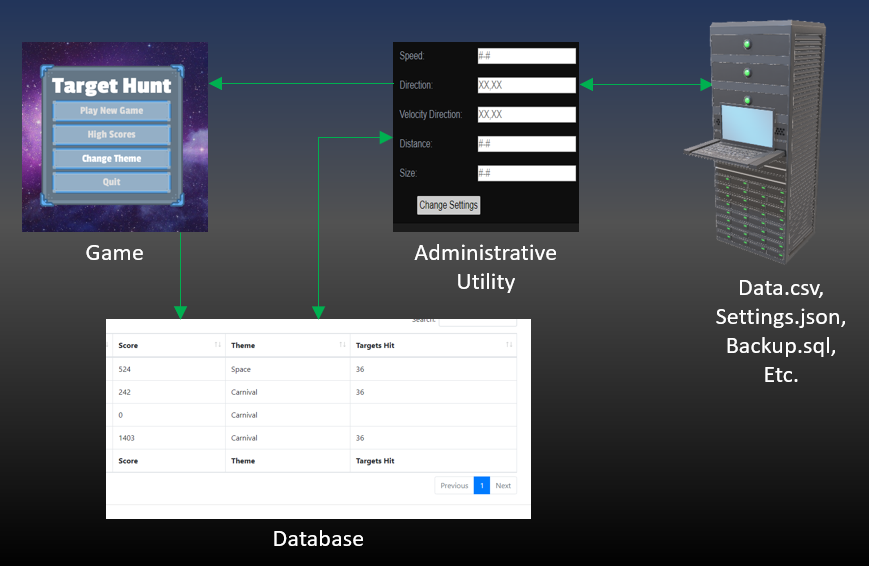
To login to the administrative interface we created for the Target Acquisition game, you must have access to a computer that will allow use of a Chrome browser. You also must have the credentials required to access the administrative account. Once logged in, the administrator will have the ability to modify the game functions, export data to a .csv file, delete data and recover data that has been deleted and backed up.

This is an image of the functions we created that the administrator will have the ability to modify.



## 2.2 Data Flows

To provide a visual representation of the system, a simplified diagram is below:



## 2.3 User Access Levels

There are two primary forms of access to the system, a user level and an administrative level.

The purpose of the administrative interface is to allow researchers to collect data that will be generated each time the target acquisition game is played. A user is simply the player of one of these games. A third form of access does exist through the web server that is hosting the system. A user may access this system to change the server settings, modify the game, or view database backups.

# QUICK START GUIDE

## 3.1 System Entry (Access admin page or user page)

A user may access the system through the provided User URL and an administrator may access the administrative interface via the provided Administrator URL. An administrative log on with the provided credentials will be required upon accessing the initial administrative log on page.

## 3.2 Player menu

The player menu includes options to start a new game, change the current theme, view high scores, and exit the game.

### 3.2.1 Play New Game

The ‘Play New Game’ button will change screen from the player menu to the game screen. The game screen includes a score counter and an exit button. One target will be spawned on screen at a time. Score is counted based on how close to the center of the target the user is when clicking the target. The server will keep track of hits and misses to determine the score.

### 3.2.2 High Scores

The ‘High Scores’ button creates a new window and displays (in descending order) the list of all high scores saved on the server. Users also have the option of viewing group scores which represent the scores within one difficulty preference. The user can select the exit button to return to the main menu.

### 3.2.3 Change Theme

The ‘Change Theme’ button selects a new theme based on the current list of available themes on the server. The user is unable to select which theme will be used, they must click the change theme button until the desired theme is displayed.

### 3.2.4 Quit

The ‘Quit’ button returns a user to the landing page where they may re-enter personal preferences and play again.

## 3.3 Administrative Login (Gaining Access to the System)

The user can login to the system with provided administrator credentials to gain access to the administrative menu.

## 3.4 Administrative menu

At the administrative menu an administrator has full control over the primary interface provided by this system. The administrator can complete the listed change operations to modify further gameplays.

### 3.4.1 Change Speed

To change the speed of the targets the administrator must enter the speed or speed range they want the targets to be into the ‘Speed’ text field.



Format must be (some number) – (a higher number) or be a single number. Each number must be zero or above but less than one million. Upon the administrator clicking the  button, if there is data in the ‘Speed’ text field it uses the python file ChangeSpeed.py passing it the user entered data. This iteratively assigns each target in the targetFile JSON a new speed that falls within the given range. After each target is assigned a new speed, the currentData JSON file is updated such that it represents the most recent version of the targetFile.

### 3.4.2 Change Velocity Direction

To change the direction that the targets move, the administrator must enter one or more directions into the ‘Velocity Direction’ text field . Format must be some cardinal or diagonal direction, followed by any number of other cardinal or diagonal directions separated by comma. Upon the administrator clicking the  button, if there is data in the ‘Velocity Direction’ text field it uses the python file ChangeVDirection.py passing it the user entered data. This iteratively assigns each target in the targetFile JSON a new velocity direction that falls within the given list of directions. After each target is assigned a new velocity direction, the currentData JSON file is updated such that it represents the most recent version of the targets.

### 3.4.3 Change Size

To change the size of the targets the administrator must enter the size or size range, that they want the targets to be, into the ‘Size’ text field . Format must be (some number) – (a higher number) or be a single number. Each number must be above zero but less than five, or the target will not display. Upon the administrator clicking the  button, if there is data in the ‘Size’ text field it uses the python file ChangeSize.py passing it the user entered data. This iteratively assigns each target in the targetFile JSON a new size that falls within the given range. After each target is assigned a new size, the currentData JSON file is updated such that it represents the most recent version of the targets.

### 3.4.4 Change Direction

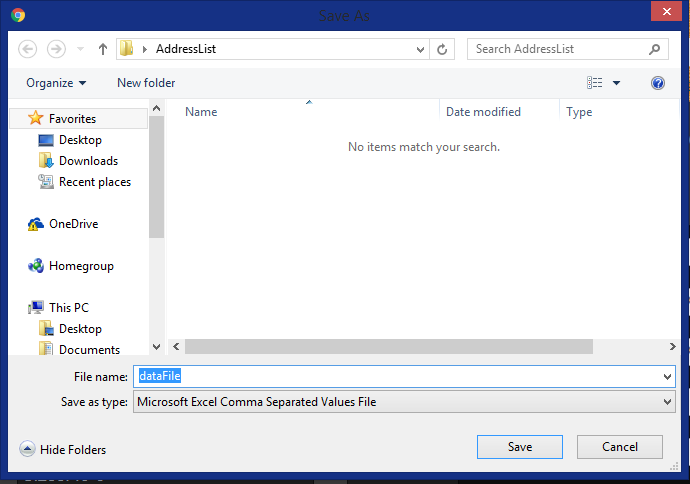
To change the direction the targets spawn from the administrator must enter the directions, that they want the targets to be allowed to move, into the ‘Direction’ text field . Format must be some cardinal or diagonal direction, followed by any number of other cardinal or diagonal directions separated by comma. Upon the administrator clicking the  button, if there is data in the ‘Direction’ text field it uses the python file ChangeDirection.py passing it the user entered data. This iteratively assigns each target in the targetFile JSON a new direction that falls within the given list of directions. After each target is assigned a new direction, the currentData JSON file is updated such that it represents the most recent version of the targets.

### 3.4.5 Change Distance

To change the distance the targets spawn from each other the administrator must enter the distance or distance range, that they want the targets to spawn from the previous target, into the ‘Distance’ text field . Format must be (some number) – (a higher number) or be a single number. Each number must be above zero but less than five, or the target will not display. Upon the administrator clicking the  button, if there is data in the ‘Distance’ text field it uses the python file ChangeDistance.py passing it the user entered data. This iteratively assigns each target in the targetFile JSON a new distance that falls within the given range. After each target is assigned a new distance, the currentData JSON file is updated such that it represents the most recent version of the targets.

### 3.4.6 Export a csv

The button  exports the current database containing all target information into a csv, after clicking a prompt to download the file will be shown.



### 3.4.7 Delete the current database

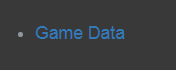
The button  backs up all the data from the current database, then deletes that data. This data is saved as a unique filename based on the date and stored in the backups folder inside the servers htdocs folder.

### 3.4.8 Restore a database backup

To restore a backup, the administrator must enter the correct filename for the backup they wish to restore, it is recommended that they rename the backups to make this step easier. After entering the correct filename into the  text field, the administrator should click the  button. This deletes and backs up the current data within the database, and generates new data based on the backup sql file.

### 3.4.9 Visually seeing the content of the database

To query and interact with the database without using the server directly, we have linked a page created by the previous team. To access this page the administrator should click the “Game Data” hyperlink.



## 3.5 Exiting the System

To exit the system a user or an administrator may simply close their browser window and/or navigate to a different web page. If a user is at the game menu and wishes to exit to the landing page, they are provided a quit button that provides this functionality, on the game menu. If an administrator is logged in to the administrative interface, they may exit the interface via the logout option, which will return them to the administrative login page.

## 3.6 Special Instructions for Error Correction

The system is prone to small documented errors. They are listed below:

* An administrator can provide variable game data updates via the administrative interface that ‘break’ the game. Some of these updates are listed below:
  + Speeds exceeding 500 are relatively unplayable.
  + Sizes more than ~6 will render partially out of bounds and thus will not appear.
  + VDIR should not be N, E, S, W, as this may result in a target that spawns slightly out of view to continue in a direction where it never comes in to view
  + Distance should not exceed ~6 to prevent targets from being rendered out of bounds and thus not appearing.
* The game may sometimes not allow a player to click the ‘Change Theme’ button because of the browser being changed in some form. This can be worked around by selecting the ‘High Scores’ button and then returning to the main menu with the ‘X’ button, where the ‘Change Theme’ button should once again be functional.

## 3.7 Caveats and Exceptions

To maintain the system, the database must be checked for storage capacity issues. The system also may be prone to failure depending on the processing power available on the server, and the amount of server traffic.

# FUTURE ENHANCEMENTS

## 4.1 Future Functionality

To enhance the functionality of the system, the system could adopt several postulated functionalities listed below:

1. Targets shoot back: The player can be represented by something such as a static object on the play screen and targets can have a rate by which they fire at this object, steadily lowering the amount of ‘health’ a player has. At the end of the player’s health, the player’s game can end.
2. Change sounds: An administrator could be given the option of changing the sounds and/or music that are used in the game.
3. View more scores: Users currently have access to an overall high score list and a ‘group score’ list

## 4.2 Similar Systems – Additional Functionality

Duck Hunt is another ‘target clicking’ game that allows two players to play the game, where one player can move the ‘targets’ (which are in this case ducks) while the other tries to click them. It would be an interesting idea to implement the light gun functionality.

Mouseaccuracy.com is a web-based target clicking game much like ours, it lacks the visual flair that ours has but has several features that would be interesting to see implemented. Each target grows then shrinks until it cannot be clicked, there is also an overall time limit. These would be great changes to make and well within the scope of the class.

AimBooster.com is another web-based target clicking game in this case there is a ‘life’ meter which depletes if the user fails to click the target in time, and the game ends when you run out of life. This is a fantastic functionality to add but would require a method of determining when a user can no longer click the target. This could be on a miss, or if the target becomes too small like in the previous example. This game also has a record of where on the targets the user clicked and shows the user at the end of the game on a target board where they clicked, allowing the user to see a visual representation of their accuracy.

The game ‘osu’ takes this target clicking idea to a completely different level, the game uses targets as a mechanism to construct its rhythm game, allowing users to dictate where exactly the targets go, at what point they show up and the type of target. The game has targets that require you to click in a specific order, i.e. clicking 1 then 2 then 3, targets that must be dragged along a path and targets that require you to spin your mouse in a circle. The targets give you a higher score based on a circle that shrinks to the target, giving you a better score the closer the outer circle is to the actual target. Perhaps the most interesting feature of osu is that the targets act as nodes act merely as a means to an end for a rhythm game, meaning the targets must get clicked in association with a song, though this might cause licensing issues for Radford. These features would be a much larger scale project than ours, however; they would indeed be very cool features to add.

## 4.3 Maintenance Capabilities

If Godot continues to export our code without error, there should not be a problem changing the code, however; if that is ever the case we were able to export using Godot version 3 and as such this version should be used if there is ever a problem exporting the game. Any MySQL database should work with the given code, no matter which version. We used XAMPP a tool which includes both Apache and MySQL, so we recommend using that to prevent errors we did not encounter. It is important that all future browsers you allow the game to be run on support WebGL as that is the method by which the Godot engine creates the graphics.