McMaster University

SmartServe

Software & Mechatronics Capstone

Low Level System Design

Authors:
Christopher McDonald
Harit Patel
Janak Patel
Jared Rayner
Nisarg Patel
Sam Hamel
Sharon Platkin

Professor:
Dr. Alan Wassyng

Teaching Assistants:
Bennett Mackenzie
Nicholas Annable
Stephen Wynn-Williams
Viktor Smirnov

Last compiled on January 13, 2018

Contents

List of Figures

Date	**Revision**	**Comments**	**Author(s)**
Dec 29, 2017	1.0	Structure made for document including headings	Christopher McDonald
Jan 12, 2018	1.2	Added Data Storage and User Interface modules	Sharon Platkin

Figure 1: Revision History

1 Introduction

1.1 Project Overview

SmartServe is an autonomous table tennis training system for table tennis players with various skill levels. SmartServe aids in diagnosing and improving a player's performance over time. The system trains table tennis players by shooting table tennis balls towards the player and detects successful returns from the player. The system can further adapt to the player's weaknesses and help them overcome it through further training. Importantly, SmartServe alleviates the problems of finding and working with a coach for players, as well as coaches trying to train multiple players simultaneously. The system will be deemed a success if the table tennis players and coaches can enjoy and see some value added by using SmartServe.

The project started at the beginning of the Fall 2017 academic term and will conclude at the end of the Winter 2018 term. In addition, the core project team consists of final year Software and Mechatronics Engineering students who are enrolled in the MECHTRON 4TB6/SFWRENG 4G06 capstone project course.

1.2 Document Overview

1.3 Naming Conventions and Terminology

The following terms and definitions will be used throughout this document:

- **ACID**: a database transaction which is atomic, consistent, isolated and durable
- **CV**: computer vision
- **FPS**: frames per second
- **FSM**: finite state machine, shows transitions between states
- **GUI**: graphical user interface
- **IPO**: input process output
- **Pitch**: rotation along the y-axis; this rotation angle primarily dictates the range of the ball from the net to the edge of the table on the user side
- **Roll**: rotation along the x-axis
- **Shooting Mechanism**: refers to the part of the system that shoots the table tennis balls towards the user side (player) Please refer to Figure 2 for visual illustration
- **System**: encompasses both the hardware and software parts of SmartServe

- **System Side**: the side of the table where the electromechanical system is placed; it is the opposite side of the User Side Please refer to Figure 2 for visual illustration
- **TCP:** transmission control protocol
- **Team**: all team members of the core capstone project, as noted in the list of Authors
- **UI**: user interface
- **User Side**: the side of the table where the user (player) is standing
- **Yaw**: rotation along the z-axis; this rotation angle primarily dictates the panning functionality of the shooting mechanism from the right side to the left side of the table

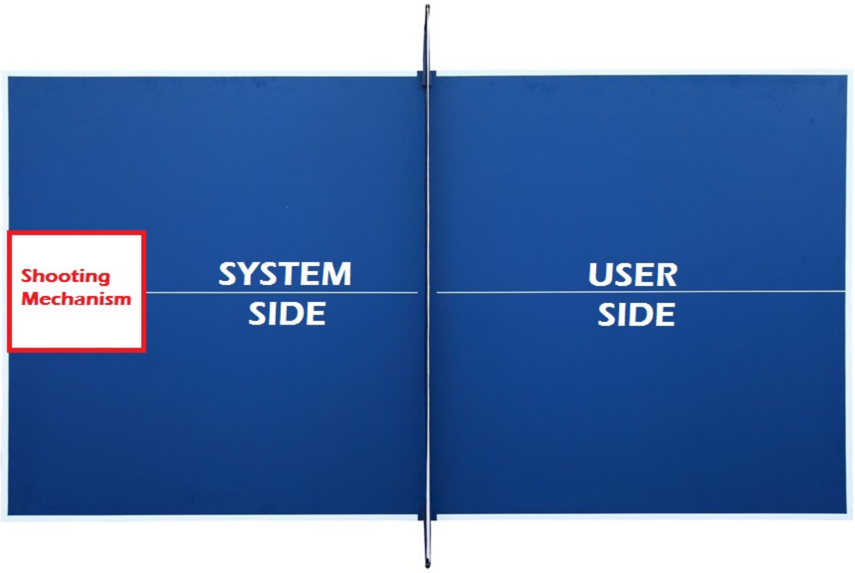

Figure 2: Top View of the Tennis Table

2 Detailed Class Diagram

3 Module Guide

3.1 SmartServe Modules

Module 1

Responsibilities

some text

Secrets
some text
MID
some text
MIS
some text

3.2 Shot Recommendation Modules

Module 1

Responsibilities
some text
Secrets
some text
MID
some text
MIS
some text

3.3 Shooting Model Modules

Module 1

Responsibilities
some text
Secrets
some text
MID
some text
MIS
some text

3.4 Shot Optimizer Modules

Module 1

Responsibilities
some text
Secrets

some text
MID
some text
MIS
some text

3.5 Computer Vision Modules

Module 1

Responsibilities
some text
Secrets
some text
MID
some text
MIS
some text

3.6 Data Storage Modules

Global

Responsibilities
Communicates with the database to create, delete and update rows in various tables.
Secrets
All connection information between the sub-systems, table contents.
MID

- User **user** - String user_name, String password, int user_id
  description: representation of a user
- Shot **shotType** - int zone_id, int omega_id, int shot_id
  description: representation of a shot type
- Zone **user** - int zone_id, double x_loc, double y_loc
  description: representation of a zone on the table, used as a look up table
- Omega **user** - int omega_id, double angle, double velocity
  description: representation of angular velocity of the ball, used as a look up table

- ReturnRate **return_rate** - int user_id, int shot_id, Timestamp timeStamp, boolean returned
  description: return statistics for each user
- **signUp** - String userName, String password
  returns: *none*
  description: adds a user row to the *User* table
- **nextShot** - int zone
  returns: *Shot*
  description: determines which shot type to perform
- **returned** - Timestamp timeStamp, User user, Shot shot
  returns: *none*
  description: updates returnRate table for user for performance statistics

3.7 Shooting Mechanism Modules

Module 1

Responsibilities
some text
Secrets
some text
MID
some text
MIS
some text

3.8 UI (User Interface) Modules

Controller

Responsibilities
Responsible for translating the desired action of the user from the View module to the SmartServe module.
Secrets
Connection between modules and sub-systems.
MID

- **main** - String args[]
  returns: *none*
  description: calls the View module to initialize the UI

- **setTrainingMode** - Mode m
  returns: *none*
  description: sets mode from View module and sends it to SmartServe

- **setShootingParameters** - none
  returns: *ShootingParameters*
  description: sets shooting parameters from View module and sends it to SmartServe

- **setStatistics** - none
  returns: *StatisticsParameters*
  description: sets statistics parameters and sends them to View module

- **signUp** - User user
  returns *none*
  description: gets user information from view module and sends it to smartServe

- **login** - User user
  returns *boolean*
  description: gets user name and password from the View module and authenticates by calling SmartServe

View

Responsibilities
The view module will contain all the actual visual aspects of the system that the user will interact with. This includes text, pictures, buttons, etc.
Secrets
The structure and implementation of the view.
MID

- **start** - none
  returns: *none*
  description: general user interface code, called to initiate UI

- **selectMode** - MouseEvent
  returns: *Mode*
  description: listens for a mouse click on mode types, and returns that mode

- **calibrate** - none
  returns: *ShootingParameters*
  description: collects user input for table size in order to calibrate the system

- **startTrainingBtn** - MouseEvent
  returns: *boolean*
  description: listens for mouse click on start training button

- **stopTrainingBtn** - MouseEvent
  returns: *boolean*
  description: listens for mouse click on stop training button

- **viewStatsBtn** - MouseEvent
  returns: *none*
  description: listens for mouse click on statistics button and displays statistics to the user

- **signUpBtn** - MouseEvent
  returns: *User*
  description: listens for mouse click on sign up button and collects user information

- **loginBtn** - MouseEvent
  returns: *User*
  description: listens for mouse click on sign in button and displays profile if authenticated properly

4 Communication Protocols

4.1 Smart Serve to Shot Recommendation

4.2 Smart Serve to Shooting Mechanism

4.3 Smart Serve to Computer Vision

4.4 Smart Serve to User Interface

4.5 Smart Serve to Shot Optimizer

4.6 Smart Serve to Shooting Model

4.7 Smart Serve to Data Storage

4.8 Shot Recommendation to Data Storage