

Tap wars

Generated by Doxygen 1.9.4

1 Tap Wars	1
2 Module Index	3
2.1 Modules	3
3 File Index	5
3.1 File List	5
4 Module Documentation	7
4.1 MAIN	7
4.1.1 Detailed Description	7
4.1.2 Function Documentation	7
4.1.2.1 SystemClock_Config()	7
4.2 Screens	8
4.2.1 Detailed Description	8
4.2.2 Function Documentation	8
4.2.2.1 Game()	8
4.2.2.2 MainMenu()	8
4.2.2.3 Periphral()	9
4.2.2.4 Winner()	9
4.3 Main_Loop	10
4.3.1 Detailed Description	10
4.3.2 Function Documentation	10
4.3.2.1 main()	10
5 File Documentation	11
5.1 TapWars.c File Reference	11
5.1.1 Detailed Description	12
Index	13

Chapter 1

Tap Wars

Authors

Andrew Sturdy, Cameron Shipman

Tap wars is a player vs. player tapping game where the person who reaches the requires score first wins. To start the game simply choose what goal you want to be aiming for then you can mash away. Each button press increases your score by 2 and reduces your opponents by 1. At the end of the game you can see how many points each player finished with.

We chose to increase your points by 2 and lower your opponents by 1 each button press to make the end scores give a more accurate comparison in tapping speeds. The longer the game the more similar you both are in tapping speed.

The application sometimes experiences a visual bug where negative numbers are not displayed accuratly, this only occures when one player goes significantly into the negatives. If both players are playing at a similar pace the bug does not occure. Another bug is that the winning screen can be skipped if players press the button as someone wins causing them to go straight to the main menu.

The original idea was to record the taps per second of each player and display them at the end of the game, this didnt happen because we could not get the RTC (Real Time Clock) to work.

Chapter 2

Module Index

2.1 Modules

Here is a list of all modules:

MAIN	7
Screens	8
Main_Loop	10

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

TapWars.c	This file provides all program functions	11
---------------------------	--	----

Chapter 4

Module Documentation

4.1 MAIN

main file

Modules

- [Screens](#)
- [Main_Loop](#)

Functions

- void [SystemClock_Config](#) (void)

4.1.1 Detailed Description

main file

4.1.2 Function Documentation

4.1.2.1 SystemClock_Config()

```
void SystemClock_Config (  
    void )
```

System Clock Configuration

4.2 Screens

Functions

- void [Peripheral](#) (void)
This function turns on and of GPIO pins needed to control the buzzer and LED.
- int [Winner](#) (int winner, int points[], TOUCH_STATE tsc_state)
This function draws the Win screen and controls the button on the screen.
- int [Game](#) (int goal, TOUCH_STATE tsc_state)
This function draws the game screen and controls the gaining of points.
- int [MainMenu](#) (void)
This function draws the main menu and controls all the buttons on the main menu.

4.2.1 Detailed Description

4.2.2 Function Documentation

4.2.2.1 Game()

```
int Game (
    int goal,
    TOUCH_STATE tsc_state )
```

This function draws the game screen and controls the gaining of points.

Note

This function only works with external GPIO pins D1 & D2.

Parameters

<i>goal</i>	: The amount of points required to win, tsc_state : the state of the screen.
-------------	--

Return values

<i>NA</i>	(Returns 0 to exit function).
-----------	-------------------------------

4.2.2.2 MainMenu()

```
int MainMenu (
    void )
```

This function draws the main menu and controls all the buttons on the main menu.

Parameters

<i>None</i>	
-------------	--

Return values

<i>NA</i>	(Returns 0 to exit function).
-----------	-------------------------------

4.2.2.3 Periphral()

```
void Periphral (
    void )
```

This function turns on and of GPIO pins needed to control the buzzer and LED.

Note

This function is hard coded to turn on D3 & D4.

Parameters

<i>None</i>	
-------------	--

Return values

<i>None</i>	
-------------	--

4.2.2.4 Winner()

```
int Winner (
    int winner,
    int points[],
    TOUCH_STATE tsc_state )
```

This function draws the Win screen and controls the button on the screen.

Parameters

<i>winner</i>	: Who won, <i>points[]</i> : the array holding each players points, <i>tsc_state</i> : the state of the screen.
---------------	---

Return values

<i>NA</i>	(Returns 0 to exit function).
-----------	-------------------------------

4.3 Main_Loop

Functions

- int `main` (void)

(Main Loop) This function initialises GPIO pins and calls the mainmenu screen.

4.3.1 Detailed Description

4.3.2 Function Documentation

4.3.2.1 `main()`

```
int main (  
        void )
```

(Main Loop) This function initialises GPIO pins and calls the mainmenu screen.

Note

This function initialises D0,D1,D2,D3 & D4

Parameters

<i>None</i>	
-------------	--

Return values

<i>None</i>	
-------------	--

Chapter 5

File Documentation

5.1 TapWars.c File Reference

This file provides all program functions.

```
#include "stdio.h"
#include "stm32f7xx_hal.h"
#include "stm32f7xx_hal_gpio.h"
#include "GLCD_Config.h"
#include "Board_GLCD.h"
#include "Board_Touch.h"
```

Macros

- #define **wait_delay** HAL_Delay

Functions

- void [SystemClock_Config](#) (void)
- void [Peripheral](#) (void)
This function turns on and of GPIO pins needed to control the buzzer and LED.
- int [Winner](#) (int winner, int points[], TOUCH_STATE tsc_state)
This function draws the Win screen and controls the button on the screen.
- int [Game](#) (int goal, TOUCH_STATE tsc_state)
This function draws the game screen and controls the gaining of points.
- int [MainMenu](#) (void)
This function draws the main menu and controls all the buttons on the main menu.
- int [main](#) (void)
(Main Loop) This function initialises GPIO pins and calls the mainmenu screen.

Variables

- GLCD_FONT **GLCD_Font_6x8**
- GLCD_FONT **GLCD_Font_16x24**

5.1.1 Detailed Description

This file provides all program functions.

Author

Andrew Sturdy, Cameron Shipman

Version

V1.0.0

Date

13-May-2022

Index

- Game
 - Screens, [8](#)
- MAIN, [7](#)
 - SystemClock_Config, [7](#)
- main
 - Main_Loop, [10](#)
- Main_Loop, [10](#)
 - main, [10](#)
- MainMenu
 - Screens, [8](#)
- Periphral
 - Screens, [9](#)
- Screens, [8](#)
 - Game, [8](#)
 - MainMenu, [8](#)
 - Periphral, [9](#)
 - Winner, [9](#)
- SystemClock_Config
 - MAIN, [7](#)
- TapWars.c, [11](#)
- Winner
 - Screens, [9](#)