Catepillar Game

Generated by Doxygen 1.9.4

Chapter 1

Caterpillar

Author

James Sparrow & Denis Ferenc Caterpiller is a game like snake, popular on old nokia phones. The obejctive is to eat as much food (red $\sim$) as possible. Each time the player eats food, the caterpillar grows larger, and it becomes harder to avoid the body of the caterpillar. If the player moves into a space taken up by the caterpillar's body, the game ends.

Chapter 2

File Index

2.1 File List

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

Chapter 3

File Documentation

3.1 keypad.h

```
void initializeMembranePins (void);
int getInput (void);
```

3.2 main.c File Reference

The main file in the caterpillar game. Contains all the functions and code to run on stm32f7 discovery board.

```
#include "stm32f7xx.h"
#include <stdio.h>
#include "stm32f7xx_hal.h"
#include "GLCD_Config.h"
#include "Board_GLCD.h"
#include "Board_Touch.h"
#include "keypad.h"
#include <stdlib.h>
#include <time.h>
#include "stm32f7xx_hal_gpio.h"
```

Macros

- #define **wait_delay** HAL_Delay
- #define **Height** 15

  Game Area Dimensions.
- #define **Width** 10

  Game Area Dimensions.

Functions

- void SystemClock_Config (void)

  System Clock Configuration.
- void initValues ()

  Init the core values to the default state.
- void resetPinD0 ()
- void resetPinD1 ()
- void resetPinD2 ()
- void resetPinD3 ()
- void resetPinD4 ()
- void resetPinD5 ()
- void resetPinD6 ()
- void resetPins ()
- void initialisePinD0 ()
- void initialisePinD1 ()
- void initialisePinD2 ()
- void initialisePinD3 ()
- void initialisePinD4 ()
- void initialisePinD5 ()
- void initialisePinD6 ()
- void initalizePins ()
- void displayOne ()
- void displayTwo ()
- void displayThree ()
- void displayFour ()
- void displayFive ()
- void displaySix ()
- void displaySeven ()
- void displayEight ()
- void displayNine ()
- void displayZero ()
- void dispNum (int number)

  Displays a number on the 7seg that is passed into the function.
- void levelCalc ()

  Algorithm to calculate level.
- void displayLevel ()

  Retrieves level and displays it on 7seg.
- void clearScreen ()

  Clears the GCLD Screen.
- void scoreCalc ()

  Calculate the score, to be displayed in scoreStr.
- void gameOver ()
- void setGameOver ()
- char **screenNumberTest** (int i)
- void drawGame ()
- void initPlayer ()
- void updatePlayer ()
- void initGameArea ()
- void borders ()
- void generateFood ()
- int **randomDirectionDebug** ()
- void changeDirection ()
- void movement ()

Main function for handling the movement of the catepillar.
- void eat ()
- void checkFood ()
- char **test** (int keypadInput)
- int main (void)

Variables

- GLCD_FONT **GLCD_Font_6x8**
- GLCD_FONT **GLCD_Font_16x24**
- int x = 3
- int y = 4
- int head = 4
- int tail = 1
- int pixelBy = 24
- int direction = 2
- int gameArea [Height][Width]

  2d array of numbers for the game area
- int food = 0
- int numberOfFood = 7
- int score = 0
- int gameIsRunning = 0
- int hasMoved = 0
- int hasEaten = 0
- int membraneNum = 0
- int level = 0
- int gameSpeed = 50
- int **score100**
- int **score10**
- int **score1**
- int **scoreTemp**
- char scoreStr [ ] = "000"

3.2.1 Detailed Description

The main file in the caterpillar game. Contains all the functions and code to run on stm32f7 discovery board.

Author

James Sparrow & Denis Ferenc

Date

14 May 2022

3.2.2 Function Documentation

3.2.2.1 borders()

```
void borders ( )
```

Initializes the borders to the gamearea.

3.2.2.2 changeDirection()

```
void changeDirection ( )
```

Used to change the direction of the catepillar. Has checks to make sure that the catepillar is not going in an illegal direction (i.e. the catepillar cannot start to travel south if it is currently travelling north.)

3.2.2.3 checkFood()

```
void checkFood ( )
```

Extra function used to make sure that the amount of food on the playing field at all times matches the value of the 'food' variable, incrementing if it drops down.

3.2.2.4 clearScreen()

```
void clearScreen ( )
```

Clears the GCLD Screen.

This function will clear the GLCD Screen. It is used in between screen changes, and is run every frame prior to drawing a new frame.

3.2.2.5 displayEight()

```
void displayEight ( )
```

The code used to initialise all the pins necessary to show on the 7seg the number 8.

3.2.2.6 displayFive()

```
void displayFive ( )
```

The code used to initialise all the pins necessary to show on the 7seg the number 5.

3.2.2.7 displayFour()

```
void displayFour ( )
```

The code used to initialise all the pins necessary to show on the 7seg the number 4.

3.2.2.8 displayLevel()

```
void displayLevel ( )
```

Retrieves level and displays it on 7seg.

This function takes in the level number and calls a function to display specific 7seg segments to represent that number.

3.2.2.9 displayNine()

```
void displayNine ( )
```

The code used to initialise all the pins necessary to show on the 7seg the number 9.

3.2.2.10 displayOne()

```
void displayOne ( )
```

The code used to initialise all the pins necessary to show on the 7seg the number 1.

3.2.2.11 displaySeven()

```
void displaySeven ( )
```

The code used to initialise all the pins necessary to show on the 7seg the number 7.

3.2.2.12 displaySix()

```
void displaySix ( )
```

The code used to initialise all the pins necessary to show on the 7seg the number 6.

3.2.2.13 displayThree()

```
void displayThree ( )
```

The code used to initialise all the pins necessary to show on the 7seg the number 3.

3.2.2.14 displayTwo()

```
void displayTwo ( )
```

The code used to initialise all the pins necessary to show on the 7seg the number 2.

3.2.2.15 displayZero()

```
void displayZero ( )
```

The code used to initialise all the pins necessary to show on the 7seg the number 0.

3.2.2.16 dispNum()

```
void dispNum (
             int number )
```

Displays a number on the 7seg that is passed into the function.

Displays a number on the 7seg that is passed into the function.

3.2.2.17 drawGame()

```
void drawGame ( )
```

Main function for drawing game, based upon data in gameArea

3.2.2.18 eat()

```
void eat ( )
```

Function to increase score when the catepillar has eaten.

3.2.2.19 gameOver()

```
void gameOver ( )
```

Display the game over screen. This function will display the game over screen, which gives the user a "game over" message, as well as providing the final score. The screen is cleaned and scoreCalc() is run before displaying the screen.

3.2.2.20 generateFood()

```
void generateFood ( )
```

Generate new food if there is not enough on the screen.

3.2.2.21 initalizePins()

```
void initalizePins ( )
```

The code used to initialise all the pins, turning all the 7seg parts on at once.

3.2.2.22 initGameArea()

```
void initGameArea ( )
```

Initialize the game area with a 0 in every space. Also adds the border to the playing field.

3.2.2.23 initialisePinD0()

```
void initialisePinD0 ( )
```

The code used to initialise pin D0, turning on the section a of the 7seg.

3.2.2.24 initialisePinD1()

```
void initialisePinD1 ( )
```

The code used to initialise pin D1, turning on the section b of the 7seg.

3.2.2.25 initialisePinD2()

```
void initialisePinD2 ( )
```

The code used to initialise pin D2, turning on the section c of the 7seg.

3.2.2.26 initialisePinD3()

```
void initialisePinD3 ( )
```

The code used to initialise pin D3, turning on the section d of the 7seg.

3.2.2.27 initialisePinD4()

```
void initialisePinD4 ( )
```

The code used to initialise pin D4, turning on the section e of the 7seg.

3.2.2.28 initialisePinD5()

```
void initialisePinD5 ( )
```

The code used to initialise pin D5, turning on the section f of the 7seg.

3.2.2.29 initialisePinD6()

```
void initialisePinD6 ( )
```

The code used to initialise pin D6, turning on the section g of the 7seg.

3.2.2.30 initPlayer()

```
void initPlayer ( )
```

Initialise the catepillar.

3.2.2.31 initValues()

```
void initValues ( )
```

Init the core values to the default state.

This function will reset the values to their initial state. It is run before the game is set up, to make sure a new game can be generated from a clean starting point.

3.2.2.32 levelCalc()

```
void levelCalc ( )
```

Algorithm to calculate level.

================================== /// ========== 7seg code ============ /// ========== ========== ============ /// ========== ========== ============ ///

This function will take in the current score and calculate the level based on that score.

3.2.2.33 main()

```
int main (
            void )
```

Main function, runs when program starts. Initialises everything and starts the game's superloop.

3.2.2.34 movement()

```
void movement ( )
```

Main function for handling the movement of the catepillar.

Main function for handling the movement of the catepillar. Checks what direction the catepillar is travelling in, then attempts to move in that direction. If the next slot on the grid contains a food, then the catepillar will eat it. If the next slot contains a border, the catepillar will teleport to the other end of the screen. If the slot contains a part of the catepillars body, the game will end. Finally, if the slot contains nothing (i.e. is empty) then the catepillar will move normally into that slot.

3.2.2.35 resetPinD0()

```
void resetPinD0 ( )
```

================================== /// ========== 7seg init ============ /// ========== ========== ============ /// ========== ========== ============ /// The code used to reset pin D0.

3.2.2.36 resetPinD1()

```
void resetPinD1 ( )
```

The code used to reset pin D1.

3.2.2.37 resetPinD2()

```
void resetPinD2 ( )
```

The code used to reset pin D2.

3.2.2.38 resetPinD3()

```
void resetPinD3 ( )
```

The code used to reset pin D3.

3.2.2.39 resetPinD4()

```
void resetPinD4 ( )
```

The code used to reset pin D4.

3.2.2.40 resetPinD5()

```
void resetPinD5 ( )
```

The code used to reset pin D5.

3.2.2.41 resetPinD6()

```
void resetPinD6 ( )
```

The code used to reset pin D6.

3.2.2.42 resetPins()

```
void resetPins ( )
```

Reset all the pins. This should be run before trying to update the 7seg.

3.2.2.43 scoreCalc()

```
void scoreCalc ( )
```

Calculate the score, to be displayed in scoreStr.

This function will convert the int score value into a string, so it can be easily displayed on the game screen.

3.2.2.44 setGameOver()

```
void setGameOver ( )
```

Used to set the gameIsRunning variable to 0.

3.2.2.45 SystemClock_Config()

```
void SystemClock_Config (
             void  )
```

System Clock Configuration.

This is the system clock configuration from the GLCD lab code on blackboard. It is required to get the GLCD working on the discovery board.

3.2.2.46 updatePlayer()

```
void updatePlayer ( )
```

Update the catepillar. This code mostly deals with the tail, or removing the last part of the catepillar and increasing the pointing variable so it matches the new end of the catepillar.

3.2.3 Variable Documentation

3.2.3.1 direction

```
int direction = 2
```

direction var (can be between 1 and 4)

3.2.3.2 food

```
int food = 0
```

Number of food that is on the playing field currently.

3.2.3.3 gameArea

```
int gameArea[Height][Width]
```

2d array of numbers for the game area

This is the playing field for the game. It is a grid of integers, with each integer representing a different concept in the game. They are as follows: -2 = food (∼) -1 = border (+) 0 = empty space ( ) Any other number above 0 = parts of the catepillar. The head of the catepillar Will be represented with a @ character, and the body with # characters.

3.2.3.4 gameIsRunning

```
int gameIsRunning = 0
```

A boolean used to check if the game is running. If this is set to false (as it is by default), then the game will go to the main menu first.

3.2.3.5 gameSpeed

```
int gameSpeed = 50
```

The game speed (or more accurately, the delay between screen refreshes) in milliseconds.

3.2.3.6 hasEaten

```
int hasEaten = 0
```

Boolean for if the player has eaten.

3.2.3.7 hasMoved

```
int hasMoved = 0
```

Boolean for if the player has moved.

3.2.3.8 head

```
int head = 4
```

The current value of the head of the caterpillar.

3.2.3.9 level

```
int level = 0
```

The number of the level the player is currently on. This is displayed on the 7seg, and increases as the player reaches new score milestones. As the level increases, so does the game speed.

3.2.3.10 membraneNum

```
int membraneNum = 0
```

The number pressed down on the membrane keypad.

3.2.3.11 numberOfFood

```
int numberOfFood = 7
```

Number of food to generate on the playing field. Acts as the max number that can appear at any one time.

3.2.3.12 pixelBy

```
int pixelBy = 24
```

Int used for the amount of spacing per character (Since font is 16x24, this is set to 24)

3.2.3.13 score

```
int score = 0
```

Count of the score.

3.2.3.14 scoreStr

```
char scoreStr[] = "000"
```

The string (array of chars) used to display the score.

3.2.3.15 tail

```
int tail = 1
```

The current value of the tail of the caterpillar.

3.2.3.16 x

```
int x = 3
```

player's x coord position

3.2.3.17 y

```
int y = 4
```

player's y coord position

3.3 main.h

Index