

Tetris_STM_code

Generated by Doxygen 1.10.0

1 Tetris Game	1
1.1 Introduction	1
2 Class Index	3
2.1 Class List	3
3 File Index	5
3.1 File List	5
4 Class Documentation	7
4.1 TETRIMINO Struct Reference	7
4.1.1 Detailed Description	7
5 File Documentation	9
5.1 C:/Users/Pierre/Documents/Git_Repos/Tetris-STM/Tetris_STM_code/tetrisGame/tetrisGame.c File Reference	9
5.1.1 Detailed Description	11
5.1.2 Function Documentation	11
5.1.2.1 addPieceIntoPlayZone()	11
5.1.2.2 falling()	11
5.1.2.3 getLinesCompleted()	12
5.1.2.4 HAL_GPIO_EXTI_Callback()	12
5.1.2.5 HAL_TIM_PeriodElapsedCallback()	12
5.1.2.6 isClippingInPlayZone()	12
5.1.2.7 moveDownPlayZone()	13
5.1.2.8 movePiece()	13
5.1.2.9 randomPiece()	13
5.1.2.10 removeLine()	13
5.1.2.11 removePieceFromPlayZone()	14
5.1.2.12 rotatePiece()	14
5.1.2.13 tetrisGame()	14
5.1.3 Variable Documentation	14
5.1.3.1 pieceBar	14
5.1.3.2 pieceJ	15
5.1.3.3 pieceL	15
5.1.3.4 pieceS	15
5.1.3.5 pieceSquare	15
5.1.3.6 pieceT	16
5.1.3.7 pieceZigzag	16
5.1.3.8 playZone	16
5.2 tetrisGame.h	16
5.3 C:/Users/Pierre/Documents/Git_Repos/Tetris-STM/Tetris_STM_code/tetrisGame/tetrisLeveling.c File Reference	17
5.3.1 Detailed Description	17

5.3.2 Function Documentation	17
5.3.2.1 getLevelNumber()	17
5.3.2.2 newLevelCalculate()	17
5.4 tetrisLeveling.h	18
5.5 C:/Users/Pierre/Documents/Git_Repos/Tetris-STM/Tetris_STM_code/tetrisGame/tetrisScoring.c File Reference	18
5.5.1 Detailed Description	19
5.5.2 Function Documentation	19
5.5.2.1 addScoreLine()	19
5.5.2.2 calculateScore()	19
5.5.2.3 getScore()	19
5.6 tetrisScoring.h	19
5.7 C:/Users/Pierre/Documents/Git_Repos/Tetris-STM/Tetris_STM_code/tetrisGame/userOled.c File Reference	20
5.7.1 Detailed Description	21
5.7.2 Function Documentation	21
5.7.2.1 drawBorderDecor()	21
5.7.2.2 drawPlayZone()	21
5.7.2.3 drawTile()	21
5.7.2.4 ssd1306_SetCursorVertical()	22
5.7.2.5 ssd1306_WriteCharVertical()	22
5.7.2.6 ssd1306_WriteStringVertical()	22
5.8 userOled.h	22
Index	25

Chapter 1

Tetris Game

1.1 Introduction

Ce jeu de Tetris a entièrement été réalisé par Adrien Kerfriden, Lou Vacher et Pierre Pontet.

Il a été conçu pour fonctionner sur une carte STM32 L031K6.

Il suffit de brancher 5 boutons, sur les broches PA1, PA3, PA4, PA6 et PA7.

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

TETRIMINO	
Structure to locate the current tetrimino in the game	7

Chapter 3

File Index

3.1 File List

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

C:/Users/Pierre/Documents/Git_Repos/Tetris-STM/Tetris_STM_code/tetrisGame/ tetrisGame.c	
: Main program of the Tetris Game	9
C:/Users/Pierre/Documents/Git_Repos/Tetris-STM/Tetris_STM_code/tetrisGame/ tetrisGame.h	16
C:/Users/Pierre/Documents/Git_Repos/Tetris-STM/Tetris_STM_code/tetrisGame/ tetrisLeveling.c	
: Program for Tetris Leveling system	17
C:/Users/Pierre/Documents/Git_Repos/Tetris-STM/Tetris_STM_code/tetrisGame/ tetrisLeveling.h	18
C:/Users/Pierre/Documents/Git_Repos/Tetris-STM/Tetris_STM_code/tetrisGame/ tetrisScoring.c	
: Program for Tetris Scoring system	18
C:/Users/Pierre/Documents/Git_Repos/Tetris-STM/Tetris_STM_code/tetrisGame/ tetrisScoring.h	19
C:/Users/Pierre/Documents/Git_Repos/Tetris-STM/Tetris_STM_code/tetrisGame/ userOled.c	
: Others functions for Oled drawing	20
C:/Users/Pierre/Documents/Git_Repos/Tetris-STM/Tetris_STM_code/tetrisGame/ userOled.h	22

Chapter 4

Class Documentation

4.1 TETRIMINO Struct Reference

Structure to locate the current tetrimino in the game.

Public Attributes

- int **coordY**
- int **coordX**
- int **piece** [4][4]

4.1.1 Detailed Description

Structure to locate the current tetrimino in the game.

The documentation for this struct was generated from the following file:

- C:/Users/Pierre/Documents/Git_Repos/Tetris-STM/Tetris_STM_code/tetrisGame/[tetrisGame.c](#)

Chapter 5

File Documentation

5.1 C:/Users/Pierre/Documents/Git_Repos/Tetris-STM/Tetris_STM_↵ code/tetrisGame/tetrisGame.c File Reference

: Main program of the Tetris Game

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <tim.h>
#include "tetrisScoring.h"
#include "userOled.h"
```

Classes

- struct [TETRIMINO](#)

Structure to locate the current tetrimino in the game.

Macros

- #define **PIECE_TOUCHED** 2
- #define **PIECE_FALLED** 1
- #define **FALL_COOLDOWN** 0

Enumerations

- enum [game_state](#) {
 PLAY_STATE = 0 , **SCORE_STATE** = 1 , **SPAWN_STATE** = 2 , **GAMEOVER_STATE** = 3 ,
 START_STATE = 4 }

Game States.

- enum [inputs](#) {
 gauche = -1 , **droite** = 1 , **fall** = 2 , **rotate** = 3 ,
 menu = 4 }

Inputs du jeu, permet de simplifier le code.

Functions

- void [addPieceIntoPlayZone](#) (TETRIMINO *currentTetrimino)
Permet de placer la pièce dans le jeu.
- void [removePieceFromPlayZone](#) (TETRIMINO *currentTetrimino)
Permet de supprimer la pièce du jeu.
- int [isClippingInPlayZone](#) (TETRIMINO *currentTetrimino)
Permet de détecter si la pièce rentrerait en collision avec les blocs déjà présents.
- int [falling](#) (TETRIMINO *currentTetrimino)
Permet de gérer la chute de la pièce.
- void **printPlayZone** (void)
Affiche la play zone sur le terminal.
- int [movePiece](#) (TETRIMINO *currentTetrimino, int direction)
Permet de déplacer une pièce vers la gauche ou la droite.
- int [rotatePiece](#) (TETRIMINO *currentTetrimino)
Permet de pivoter de 90 ° une pièce, seulement si elle ne va pas clip.
- int [getLinesCompleted](#) ()
Calculate amount of lines completed, and shifts down the playZone accordingly.
- void [removeLine](#) (int line)
Remove the selected line (fill it with 0)
- void **resetPlayZone** ()
Reset the PlayZone.
- void [moveDownPlayZone](#) (int startLine)
Move all the play zone above "startLine" down by one line.
- void [randomPiece](#) (TETRIMINO *currentTetrimino)
Select a random piece among the tetriminos.
- void **tetrisInit** ()
Permet d'initialiser le jeu.
- int [tetrisGame](#) ()
Permet de lancer le jeu.
- void **reduceFallDelay** ()
Reduce the delay of fallTime.
- void [HAL_GPIO_EXTI_Callback](#) (uint16_t GPIO_Pin)
Gestion des interruptions.
- void [HAL_TIM_PeriodElapsedCallback](#) (TIM_HandleTypeDef *htim)
Fin du timer pour lire les boutons.

Variables

- int [pieceZigzag](#) [4][4]
Zigzag piece in 4 by 4 grid.
- int [pieceS](#) [4][4]
S piece in 4 by 4 grid.
- int [pieceBar](#) [4][4]
Bar piece in 4 by 4 grid.
- int [pieceSquare](#) [4][4]
Square piece in 4 by 4 grid.
- int [pieceL](#) [4][4]
L piece in 4 by 4 grid.
- int [pieceJ](#) [4][4]

- J piece in 4 by 4 grid.*
 - int `pieceT` [4][4]
- T piece in 4 by 4 grid.*
 - int `playZone` [23][16]
- PlayZone.*
 - uint32_t `nextFallTime` = 0
 - int `fallDelay` = 1000
 - int `rotateFallDelay` = 500
 - int `moveFallDelay` = 500
 - int `userInput` = 0

5.1.1 Detailed Description

: Main program of the Tetris Game

Date

08 février 2024

5.1.2 Function Documentation

5.1.2.1 addPieceIntoPlayZone()

```
void addPieceIntoPlayZone (
    TETRIMINO * currentTetrimino )
```

Permet de placer la pièce dans le jeu.

Parameters

<i>*currentTetrimino</i>	Current tetrimino used
--------------------------	------------------------

5.1.2.2 falling()

```
int falling (
    TETRIMINO * currentTetrimino )
```

Permet de gérer la chute de la pièce.

Parameters

<i>*currentTetrimino</i>	Current tetrimino used
--------------------------	------------------------

Return values

<i>state</i>	PIECE_TOUCHED, or PIECE_FALLED
--------------	--------------------------------

5.1.2.3 getLinesCompleted()

```
int getLinesCompleted ( )
```

Calculate amount of lines completed, and shifts down the playZone accordingly.

Return values

<i>linesCount</i>	number of Lines completed
-------------------	---------------------------

5.1.2.4 HAL_GPIO_EXTI_Callback()

```
void HAL_GPIO_EXTI_Callback (
    uint16_t GPIO_Pin )
```

Gestion des interruptions.

Parameters

<i>GPIO_Pin</i>	Pin sur lequel a lieu l'interruption
-----------------	--------------------------------------

5.1.2.5 HAL_TIM_PeriodElapsedCallback()

```
void HAL_TIM_PeriodElapsedCallback (
    TIM_HandleTypeDef * htim )
```

Fin du timer pour lire les boutons.

Parameters

<i>*htim</i>	timer
--------------	-------

5.1.2.6 isClippingInPlayZone()

```
int isClippingInPlayZone (
    TETRIMINO * currentTetrimino )
```

Permet de détecter si la pièce rentrerait en collision avec les blocs déjà présents.

Parameters

<i>*currentTetrimino</i>	Current tetrimino used
--------------------------	------------------------

Return values

<i>state</i>	1 if would clip, 0 if not
--------------	---------------------------

5.1.2.7 moveDownPlayZone()

```
void moveDownPlayZone (
    int startLine )
```

Move all the play zone above "startLine" down by one line.

Parameters

<i>startLine</i>	line which got cleared
------------------	------------------------

5.1.2.8 movePiece()

```
int movePiece (
    TETRIMINO * currentTetrimino,
    int direction )
```

Permet de déplacer une pièce vers la gauche ou la droite.

Parameters

<i>*currentTetrimino</i>	Current tetrimino used
<i>direction</i>	userInputs

Return values

<i>state</i>	1 si déplacement OK, 0 si non
--------------	-------------------------------

5.1.2.9 randomPiece()

```
void randomPiece (
    TETRIMINO * currentTetrimino )
```

Select a random piece among the tetriminos.

Parameters

<i>currentTetrimino</i>	list of tetriminos
-------------------------	--------------------

5.1.2.10 removeLine()

```
void removeLine (
    int line )
```

Remove the selected line (fill it with 0)

Parameters

<i>line</i>	line selected
-------------	---------------

5.1.2.11 removePieceFromPlayZone()

```
void removePieceFromPlayZone (
    TETRIMINO * currentTetrimino )
```

Permet de supprimer la pièce du jeu.

Parameters

<i>*currentTetrimino</i>	Current tetrimino used
--------------------------	------------------------

5.1.2.12 rotatePiece()

```
int rotatePiece (
    TETRIMINO * currentTetrimino )
```

Permet de pivoter de 90° une pièce, seulement si elle ne va pas clip.

Parameters

<i>*currentTetrimino</i>	Current tetrimino used
--------------------------	------------------------

Return values

<i>state</i>	1 if rotated and 0 if clipping
--------------	--------------------------------

5.1.2.13 tetrisGame()

```
int tetrisGame ( )
```

Permet de lancer le jeu.

Return values

<i>0</i>	si erreur
----------	-----------

5.1.3 Variable Documentation**5.1.3.1 pieceBar**

```
int pieceBar[4][4]
```

Initial value:

```
= { {0,1,0,0},
    {0,1,0,0},
    {0,1,0,0},
    {0,1,0,0} }
```

Bar piece in 4 by 4 grid.

5.1.3.2 pieceJ

```
int pieceJ[4][4]
```

Initial value:

```
= { {0,0,0,0},
    {0,0,1,0},
    {0,0,1,0},
    {0,1,1,0} }
```

J piece in 4 by 4 grid.

5.1.3.3 pieceL

```
int pieceL[4][4]
```

Initial value:

```
= { {0,0,0,0},
    {0,1,0,0},
    {0,1,0,0},
    {0,1,1,0} }
```

L piece in 4 by 4 grid.

5.1.3.4 pieceS

```
int pieceS[4][4]
```

Initial value:

```
= { {0,0,0,0},
    {0,0,1,1},
    {0,1,1,0},
    {0,0,0,0} }
```

S piece in 4 by 4 grid.

5.1.3.5 pieceSquare

```
int pieceSquare[4][4]
```

Initial value:

```
= { {0,0,0,0},
    {0,1,1,0},
    {0,1,1,0},
    {0,0,0,0} }
```

Square piece in 4 by 4 grid.

5.1.3.6 pieceT

```
int pieceT[4][4]
```

Initial value:

```
= { {0,0,0,0},
    {1,1,1,0},
    {0,1,0,0},
    {0,0,0,0} }
```

T piece in 4 by 4 grid.

5.1.3.7 pieceZigzag

```
int pieceZigzag[4][4]
```

Initial value:

```
= { {0,0,0,0},
    {1,1,0,0},
    {0,1,1,0},
    {0,0,0,0} }
```

Zigzag piece in 4 by 4 grid.

5.1.3.8 playZone

```
int playZone[23][16]
```

Initial value:

```
=
{{1,1,1,0,0,0,0,0,0,0,0,0,0,1,1,1},
 {1,1,1,0,0,0,0,0,0,0,0,0,0,1,1,1},
 {1,1,1,0,0,0,0,0,0,0,0,0,0,1,1,1},
 {1,1,1,0,0,0,0,0,0,0,0,0,0,1,1,1},
 {1,1,1,0,0,0,0,0,0,0,0,0,0,1,1,1},
 {1,1,1,0,0,0,0,0,0,0,0,0,0,1,1,1},
 {1,1,1,0,0,0,0,0,0,0,0,0,0,1,1,1},
 {1,1,1,0,0,0,0,0,0,0,0,0,0,1,1,1},
 {1,1,1,0,0,0,0,0,0,0,0,0,0,1,1,1},
 {1,1,1,0,0,0,0,0,0,0,0,0,0,1,1,1},
 {1,1,1,0,0,0,0,0,0,0,0,0,0,1,1,1},
 {1,1,1,0,0,0,0,0,0,0,0,0,0,1,1,1},
 {1,1,1,0,0,0,0,0,0,0,0,0,0,1,1,1},
 {1,1,1,0,0,0,0,0,0,0,0,0,0,1,1,1},
 {1,1,1,0,0,0,0,0,0,0,0,0,0,1,1,1},
 {1,1,1,0,0,0,0,0,0,0,0,0,0,1,1,1},
 {1,1,1,0,0,0,0,0,0,0,0,0,0,1,1,1},
 {1,1,1,0,0,0,0,0,0,0,0,0,0,1,1,1},
 {1,1,1,0,0,0,0,0,0,0,0,0,0,1,1,1},
 {1,1,1,0,0,0,0,0,0,0,0,0,0,1,1,1},
 {1,1,1,0,0,0,0,0,0,0,0,0,0,1,1,1},
 {1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1},
 {1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1},
 {1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1} }
```

PlayZone.

5.2 tetrisGame.h

```
00001 /*
00002  * tetrisGame.h
00003  *
00004  * Created on: Feb 7, 2024
00005  * Author: PONTET
00006  */
00007
00008 #ifndef TETRISGAME_H_
00009 #define TETRISGAME_H_
00010
00011 int tetrisGame();
00012 void reduceFallDelay();
00013
00014 #endif /* TETRISGAME_H_ */
```

5.3 C:/Users/Pierre/Documents/Git_Repos/Tetris-STM/Tetris_STM_code/tetrisGame/tetrisLeveling.c File Reference

: Program for Tetris Leveling system

```
#include "tetrisLeveling.h"
#include "tetrisGame.h"
```

Functions

- `uint8_t getLevelNumber ()`
Return levelNumber.
- `void newLevelCalculate (uint8_t _numberOfLineCompleted)`
Verify and change levelNumber.

Variables

- `uint8_t levelNumber = 0`

5.3.1 Detailed Description

: Program for Tetris Leveling system

Date

08 février 2024

5.3.2 Function Documentation

5.3.2.1 getLevelNumber()

```
uint8_t getLevelNumber ( )
```

Return levelNumber.

Return values

<i>levelNumber</i>	
--------------------	--

5.3.2.2 newLevelCalculate()

```
void newLevelCalculate (
    uint8_t _numberOfLineCompleted )
```

Verify and change levelNumber.

Parameters

<code>_numberOfLineCompleted</code>	each level requires a numberOfLine completed to be incremented
-------------------------------------	--

5.4 tetrisLeveling.h

```

00001 /*
00002  * tetrisLeveling.h
00003  *
00004  * Created on: Feb 7, 2024
00005  * Author: PONTET
00006  */
00007
00008 #ifndef TETRISLEVELING_H_
00009 #define TETRISLEVELING_H_
00010
00011 #include "main.h"
00012
00013 uint8_t getLevelNumber();
00014 void newLevelCalculate(uint8_t);
00015
00016 #endif /* TETRISLEVELING_H_ */

```

5.5 C:/Users/Pierre/Documents/Git_Repos/Tetris-STM/Tetris_STM_↵ code/tetrisGame/tetrisScoring.c File Reference

: Program for Tetris Scoring system

```

#include "tetrisScoring.h"
#include "ssd1306_fonts.h"
#include "ssd1306.h"
#include <stdio.h>
#include "string.h"
#include "tetrisLeveling.h"
#include "userOled.h"

```

Functions

- `uint32_t` [getScore](#) ()
Getter of score.
- `void` [calculateScore](#) (int _numberOfLineCompleted)
Calculate the Score.
- `void` [addScorePiece](#) ()
Add points to the score when a Piece fall to its place.
- `void` [addScoreLine](#) (uint8_t _numberOfLineCompleted)
Add points to the score, using numberOfLineCompleted and levelNumber as factors.
- `void` [printScore](#) ()
Print the score on the screen.

Variables

- `uint32_t` [tetrisScore](#) = 0
- `uint8_t` [piecePosee](#) = 100

5.5.1 Detailed Description

: Program for Tetris Scoring system

Date

08 février 2024

5.5.2 Function Documentation

5.5.2.1 addScoreLine()

```
void addScoreLine (
    uint8_t _numberOfLineCompleted )
```

Add points to the score, using numberOfLineCompleted and levelNumber as factors.

Parameters

<code>_numberOfLineCompleted</code>	numberOfLineCompleted in one fall by a piece
-------------------------------------	--

5.5.2.2 calculateScore()

```
void calculateScore (
    int _numberOfLineCompleted )
```

Calculate the Score.

Parameters

<code>_numberOfLineCompleted</code>	
-------------------------------------	--

5.5.2.3 getScore()

```
uint32_t getScore ( )
```

Getter of score.

Return values

<i>Return</i>	the value of score
---------------	--------------------

5.6 tetrisScoring.h

```
00001 /*
00002  * tetrisScoring.h
```

```

00003  *
00004  *   Created on: Feb 1, 2024
00005  *       Author: PONTET
00006  */
00007
00008 #ifndef TETRISSCORING_H_
00009 #define TETRISSCORING_H_
00010
00011 #include "main.h"
00012
00013
00014 uint32_t getScore();
00015 void calculateScore(int);
00016 void addScorePiece();
00017 void addScoreLine(uint8_t);
00018
00019 void printScore();
00020
00021 #endif /* TETRISSCORING_H_ */

```

5.7 C:/Users/Pierre/Documents/Git_Repos/Tetris-STM/Tetris_STM_↵ code/tetrisGame/userOled.c File Reference

: Others functions for Oled drawing

```

#include "userOled.h"
#include "tetrisScoring.h"

```

Functions

- void **drawBorder** ()
Draw the border of the game.
- void **drawTile** (uint8_t x, uint8_t y, SSD1306_COLOR color)
Draw a normal tile.
- void **drawPlayZone** (int stack[23][16])
Draw the entire play zone.
- void **drawGameOver** ()
Plays the gameOver Animation.
- void **drawTetrisStartGame** ()
Plays the StartMenu Animation.
- void **clearPlayZone** ()
Clear the PlayZone.
- char **ssd1306_WriteCharVertical** (char ch, FontDef Font, SSD1306_COLOR color)
Ecrit un caractère verticalement.
- char **ssd1306_WriteStringVertical** (char *str, FontDef Font, SSD1306_COLOR color)
Ecrit une chaîne de caractères verticalement.
- void **ssd1306_SetCursorVertical** (uint8_t x, uint8_t y)
Déplace le curseur à sa position Verticale.
- void **drawBorderDecor** (SSD1306_COLOR color)
Draw the decor.
- void **drawTetriminos** ()
Affiche les tetriminos en bas du StartMenu.
- void **Clignotement_Ecran** ()
Fait "clignoter" l'écran en inversant ses couleurs pendant un court instant.
- void **Clignotement_Click** ()
Fait "clignoter" le mot "Click" à l'écran de démarrage, non bloquant.

5.7.1 Detailed Description

: Others functions for Oled drawing

Date

08 février 2024

5.7.2 Function Documentation

5.7.2.1 drawBorderDecor()

```
void drawBorderDecor (
    SSD1306_COLOR color )
```

Draw the decor.

Parameters

<i>color</i>	couleur sélectionnée
--------------	----------------------

5.7.2.2 drawPlayZone()

```
void drawPlayZone (
    int stack[23][16] )
```

Draw the entire play zone.

Parameters

<i>stack</i> [][]	the play zone of the game
-------------------	---------------------------

5.7.2.3 drawTile()

```
void drawTile (
    uint8_t x,
    uint8_t y,
    SSD1306_COLOR color )
```

Draw a normal tile.

Parameters

<i>x,y</i>	position de la Tile à changer
<i>color</i>	couleur de la Tile

5.7.2.4 `ssd1306_SetCursorVertical()`

```
void ssd1306_SetCursorVertical (
    uint8_t x,
    uint8_t y )
```

Déplace le curseur à sa position Verticale.

Parameters

<i>x,y</i>	: position du curseur à changer
------------	---------------------------------

5.7.2.5 `ssd1306_WriteCharVertical()`

```
char ssd1306_WriteCharVertical (
    char ch,
    FontDef Font,
    SSD1306_COLOR color )
```

Ecrit un caractère verticalement.

Parameters

<i>ch</i>	caractère à afficher
<i>Font</i>	police sélectionnée
<i>color</i>	couleur sélectionnée

5.7.2.6 `ssd1306_WriteStringVertical()`

```
char ssd1306_WriteStringVertical (
    char * str,
    FontDef Font,
    SSD1306_COLOR color )
```

Ecrit une chaîne de caractères verticalement.

Parameters

<i>str</i>	chaîne de caractères à afficher
<i>Font</i>	police sélectionnée
<i>color</i>	couleur sélectionnée

5.8 `userOled.h`

```
00001 /*
00002  * userOled.h
00003  *
00004  * Created on: Feb 7, 2024
00005  * Author: VACHER
```

```
00006  */
00007
00008 #ifndef USEROLED_H_
00009 #define USEROLED_H_
00010
00011 #include "main.h"
00012 #include "ssd1306.h"
00013
00014 void drawBorder();
00015 void drawTile(uint8_t x, uint8_t y, SSD1306_COLOR color);
00016 void drawPlayZone(int stack[23][16]);
00017 void drawGameOver();
00018 void ssd1306_SetCursorVertical(uint8_t x, uint8_t y);
00019 char ssd1306_WriteStringVertical(char* str, FontDef Font, SSD1306_COLOR color);
00020 void drawTetrisStartGame();
00021 void clearPlayZone();
00022 void drawBorderDecor(SSD1306_COLOR color);
00023 void drawTetriminos();
00024 void Clignotement_Ecran();
00025 void Clignotement_Click();
00026
00027 #endif /* USEROLED_H_ */
```


Index

addPieceIntoPlayZone
 tetrisGame.c, [11](#)

addScoreLine
 tetrisScoring.c, [19](#)

C:/Users/Pierre/Documents/Git_Repos/Tetris-STM/Tetris_STM_code/tetrisGame/tetrisGame.c,
 [9](#)

C:/Users/Pierre/Documents/Git_Repos/Tetris-STM/Tetris_STM_code/tetrisGame/tetrisGame.h,
 [16](#)

C:/Users/Pierre/Documents/Git_Repos/Tetris-STM/Tetris_STM_code/tetrisGame/tetrisLeveling.c,
 [17](#)

C:/Users/Pierre/Documents/Git_Repos/Tetris-STM/Tetris_STM_code/tetrisGame/tetrisLeveling.h,
 [18](#)

C:/Users/Pierre/Documents/Git_Repos/Tetris-STM/Tetris_STM_code/tetrisGame/tetrisScoring.c,
 [18](#)

C:/Users/Pierre/Documents/Git_Repos/Tetris-STM/Tetris_STM_code/tetrisGame/tetrisScoring.h,
 [19](#)

C:/Users/Pierre/Documents/Git_Repos/Tetris-STM/Tetris_STM_code/tetrisGame/userOled.c,
 [20](#)

C:/Users/Pierre/Documents/Git_Repos/Tetris-STM/Tetris_STM_code/tetrisGame/userOled.h,
 [22](#)

calculateScore
 tetrisScoring.c, [19](#)

drawBorderDecor
 userOled.c, [21](#)

drawPlayZone
 userOled.c, [21](#)

drawTile
 userOled.c, [21](#)

falling
 tetrisGame.c, [11](#)

getLevelNumber
 tetrisLeveling.c, [17](#)

getLinesCompleted
 tetrisGame.c, [11](#)

getScore
 tetrisScoring.c, [19](#)

HAL_GPIO_EXTI_Callback
 tetrisGame.c, [12](#)

HAL_TIM_PeriodElapsedCallback
 tetrisGame.c, [12](#)

isClippingInPlayZone
 tetrisGame.c, [12](#)

moveDownPlayZone
 tetrisGame.c, [13](#)

movePiece
 tetrisGame.c, [13](#)

newLevelCalculate
 tetrisLeveling.c, [17](#)

pieceBar
 tetrisGame.c, [14](#)

pieceJ
 tetrisGame.c, [15](#)

pieceL
 tetrisGame.c, [15](#)

pieceS
 tetrisGame.c, [15](#)

pieceSquare
 tetrisGame.c, [15](#)

pieceT
 tetrisGame.c, [15](#)

pieceZigzag
 tetrisGame.c, [16](#)

playZone
 tetrisGame.c, [16](#)

randomPiece
 tetrisGame.c, [13](#)

removeLine
 tetrisGame.c, [13](#)

removePieceFromPlayZone
 tetrisGame.c, [14](#)

rotatePiece
 tetrisGame.c, [14](#)

ssd1306_SetCursorVertical
 userOled.c, [21](#)

ssd1306_WriteCharVertical
 userOled.c, [22](#)

ssd1306_WriteStringVertical
 userOled.c, [22](#)

TETRIMINO, [7](#)

Tetris Game, [1](#)

tetrisGame
 tetrisGame.c, [14](#)

tetrisGame.c
 addPieceIntoPlayZone, [11](#)
 falling, [11](#)
 getLinesCompleted, [11](#)
 HAL_GPIO_EXTI_Callback, [12](#)
 HAL_TIM_PeriodElapsedCallback, [12](#)
 isClippingInPlayZone, [12](#)
 moveDownPlayZone, [13](#)

- movePiece, [13](#)
- pieceBar, [14](#)
- pieceJ, [15](#)
- pieceL, [15](#)
- pieceS, [15](#)
- pieceSquare, [15](#)
- pieceT, [15](#)
- pieceZigzag, [16](#)
- playZone, [16](#)
- randomPiece, [13](#)
- removeLine, [13](#)
- removePieceFromPlayZone, [14](#)
- rotatePiece, [14](#)
- tetrisGame, [14](#)
- tetrisLeveling.c
 - getLevelNumber, [17](#)
 - newLevelCalculate, [17](#)
- tetrisScoring.c
 - addScoreLine, [19](#)
 - calculateScore, [19](#)
 - getScore, [19](#)
- userOled.c
 - drawBorderDecor, [21](#)
 - drawPlayZone, [21](#)
 - drawTile, [21](#)
 - ssd1306_SetCursorVertical, [21](#)
 - ssd1306_WriteCharVertical, [22](#)
 - ssd1306_WriteStringVertical, [22](#)