

Jeux_De_Seroent

Generated by Doxygen 1.9.4

1 JeuxSerpentConsoleLinux	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 Board Class Reference	7
4.1.1 Detailed Description	7
4.1.2 Constructor & Destructor Documentation	8
4.1.2.1 Board() [1/2]	8
4.1.2.2 Board() [2/2]	8
4.1.3 Member Function Documentation	8
4.1.3.1 getHeightBoard()	8
4.1.3.2 getPosBoard()	9
4.1.3.3 getWidthBoard()	9
4.1.3.4 setElement()	9
4.1.3.5 setPixelColor()	9
4.1.3.6 setPosBoard()	10
4.2 color Struct Reference	10
4.2.1 Detailed Description	10
4.3 Coord Struct Reference	11
4.3.1 Detailed Description	11
4.4 Food Class Reference	11
4.4.1 Detailed Description	11
4.4.2 Constructor & Destructor Documentation	12
4.4.2.1 Food()	12
4.4.3 Member Function Documentation	13
4.4.3.1 aleaNumber()	13
4.5 Snake Class Reference	13
4.5.1 Detailed Description	15
4.5.2 Constructor & Destructor Documentation	15
4.5.2.1 Snake()	15
4.5.3 Member Function Documentation	15
4.5.3.1 changeDirection()	15
4.5.3.2 changePosofSnake()	15
4.5.3.3 checkPositionInterdit()	16
4.5.3.4 getHeadSnakePos()	16
4.5.3.5 getLengthOfSnake()	16
4.5.3.6 getPartSnakePos()	16
4.5.3.7 nextMoveDown()	17

4.5.3.8 nextMoveLeft()	17
4.5.3.9 nextMoveRight()	17
4.5.3.10 nextMoveUp()	18
4.5.3.11 nextMoving()	18
4.5.3.12 setSnakeCoord()	18
5 File Documentation	19
5.1 board.h	19
5.2 controlOutput.h	20
5.3 color.h	21
5.4 food.h	21
5.5 snake_game.h	22
5.6 coord.h	23
Index	25

Chapter 1

JeuxSerpentConsoleLinux

Juste petit temps d'amusement avec la console LINUX

Chapter 2

Class Index

2.1 Class List

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

Board	La classe Board gere la plateforme du jeux	7
color	Structre de la couleur	10
Coord	Coord est la structure gérant le positionnement	11
Food	La classe de la nourriture du serpent	11
Snake	La classe su serpent	13

Chapter 3

File Index

3.1 File List

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

board/board.h	19
board/controlOutput.h	20
color/color.h	21
Game_Snake/food.h	21
Game_Snake/snake_game.h	22
pos/coord.h	23

Chapter 4

Class Documentation

4.1 Board Class Reference

La classe `Board` gere la plateforme du jeux.

```
#include <board.h>
```

Public Member Functions

- `Board` (int w, int h, `Coord` *pos)
`Board` est le premier constructeur de la classe.
- `Board` (int w=MAX_BOARD_WIDTH, int h=MAX_BOARD_HEIGHT)
Le second constructeur du plateau, ici par default le plateau est mis a la position (0, 0)
- void `Show` (void)
Affichage du plateau.
- `Coord` `getPosBoard` ()
recupere la position du plateau
- int `getWidthBoard` ()
recupere la largeur du plateau
- int `getHeightBoard` ()
Recupère la hauteur du plateau.
- void `setPosBoard` (`Coord` *coord1)
Placer le plateau à une position.
- void `setPixelColor` (`Coord` *coord1, `color` *clr1)
Insérer une information de couleur du Pixel à une position du plateau.
- void `setElement` (`Coord` *coord1, char elmnt)
Insérer les caractéristique de l'élément à la position donnée.

4.1.1 Detailed Description

La classe `Board` gere la plateforme du jeux.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 Board() [1/2]

```
Board::Board (
    int w,
    int h,
    Coord * pos )
```

[Board](#) est le premier constructeur de la classe.

Parameters

<i>w</i>	la largeur du plateau
<i>h</i>	la hauteur du plateau
<i>pos</i>	la position du plateau en COORD

4.1.2.2 Board() [2/2]

```
Board::Board (
    int w = MAX_BOARD_WIDTH,
    int h = MAX_BOARD_HEIGHT )
```

Le second constructeur du plateau, ici par défaut le plateau est mis a la position (0, 0)

Parameters

<i>w</i>	les parametres de la largeur du plateau
<i>h</i>	les parametre de la hauteur du plateau

4.1.3 Member Function Documentation

4.1.3.1 getHeightBoard()

```
int Board::getHeightBoard ( ) [inline]
```

Recupère la hauteur du plateau.

Returns

INT la hauteur du plateau

4.1.3.2 getPosBoard()

```
Coord Board::getPosBoard ( ) [inline]
```

recupere la position du plateau

Returns

Coord la position du plateau

4.1.3.3 getWidthBoard()

```
int Board::getWidthBoard ( ) [inline]
```

recupere la largeur du plateau

Returns

INT la largeur

4.1.3.4 setElement()

```
void Board::setElement (
    Coord * coord1,
    char elmnt )
```

Insérer les caractéristique de l'élément à la position donnée.

Parameters

<i>coord1</i>	Les coordonnées de position
<i>elmnt</i>	La valeur de l'élément

4.1.3.5 setPixelColor()

```
void Board::setPixelColor (
    Coord * coord1,
    color * clr1 )
```

Insérer une information de couleur du Pixel à une position du plateau.

Parameters

<i>coord1</i>	La position sur laquelle on veut insérer
<i>clr1</i>	La couleur à insérer

4.1.3.6 setPosBoard()

```
void Board::setPosBoard (
    Coord * coord1 ) [inline]
```

Placer le plateau à une position.

Parameters

<i>coord1</i>	La nouvelle position du plateau
---------------	---------------------------------

The documentation for this class was generated from the following files:

- board/board.h
- board/board.cpp

4.2 color Struct Reference

Structre de la couleur.

```
#include <color.h>
```

Public Attributes

- int **r**
- int **v**
- int **b**

4.2.1 Detailed Description

Structre de la couleur.

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

- color/color.h

4.3 Coord Struct Reference

[Coord](#) est la structure gérant le positionnement.

```
#include <coord.h>
```

Public Attributes

- `int x`
- `int y`

4.3.1 Detailed Description

[Coord](#) est la structure gérant le positionnement.

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

- `pos/coord.h`

4.4 Food Class Reference

La classe de la nourriture du serpent.

```
#include <food.h>
```

Public Member Functions

- [Food](#) (int x=0, int y=0, [Board](#) *frame=NULL)
Le constructeur de la classe.
- void **changeFoodPos** ([Coord](#) *newCoord)
Changer la position de la nourriture.
- [Coord](#) **aleaNumber** (int borneY, int borneX, [Snake](#) *snakeEat=NULL)
Générer aléatoirement la position.
- void **setFoodToBoard** ()
Insérer la nourriture sur le plateau.
- void **clearFoodToBoard** ()
Supprimer du plateau.
- [Coord](#) **getPosFood** ()
Récupère la position de la nourriture.

4.4.1 Detailed Description

La classe de la nourriture du serpent.

4.4.2 Constructor & Destructor Documentation

4.4.2.1 Food()

```
Food::Food (
    int x = 0,
    int y = 0,
    Board * frame = NULL )
```

Le constructeur de la classe.

Parameters

<i>x</i>	Les paramètres en x
<i>y</i>	Paramètre en y
<i>frame</i>	Le tableau le contenant

4.4.3 Member Function Documentation

4.4.3.1 aleaNumber()

```
Coord Food::aleaNumber (
    int borneY,
    int borneX,
    Snake * snakeEat = NULL )
```

Générer aléatoirement la position.

Parameters

<i>borneY</i>	Limite en Y
<i>borneX</i>	Limite en X
<i>snakeEat</i>	Le serpent ayant manger

Returns

COORD la position nouvelle

The documentation for this class was generated from the following files:

- Game_Snake/food.h
- Game_Snake/food.cpp

4.5 Snake Class Reference

La classe su serpent.

```
#include <snake_game.h>
```

Public Member Functions

- **Snake** (int x=0, int y=0, **Board** *frame=NULL)
Le constructeur du serpent.
- void **setSnakeCoord** (**Coord** *coord1)
Insérer le serpent à une position.
- **Coord** **getHeadSnakePos** ()
Chercher la t^{ête} du serpent.
- int **getLengthOfSnake** ()
Récupérer la longueur du serpent.
- **Coord** **getPartSnakePos** (int pos)
La position d'une partie du serpent.
- void **changePosofSnake** (**Coord** *newCoordHead)
Changer la position d'élément du serpent.
- void **putSnakeToBoard** ()
Insérer le serpent dans le tableau.
- void **clearSnakeToBoard** ()
Supprimer Le serpent du tableau.
- void **makeSnakeHead** ()
Insérer la t^{ête} du serpent.
- void **moving** ()
Déplacement du serpent.
- **Coord** **nextMoving** ()
vérification de la position future
- void **moveUp** ()
Déplacement d'en haut.
- **Coord** **nextMoveUp** ()
La position futur d'en haut.
- void **moveRight** ()
Déplacement à droite.
- **Coord** **nextMoveRight** ()
Position future de droite.
- void **moveDown** ()
Deplacement en bas.
- **Coord** **nextMoveDown** ()
Déplacement future en bas.
- void **moveLeft** ()
Déplacement de gauche.
- **Coord** **nextMoveLeft** ()
Déplacement future de gauche.
- void **grandi** ()
Grandir le serpent.
- void **changeDirection** (char d)
Changer des direction.
- void **changeVelocity** ()
Changer la vitesse
- void **checkPositionInterdit** (int *game)
Vérifier les position interdites.

4.5.1 Detailed Description

La classe du serpent.

4.5.2 Constructor & Destructor Documentation

4.5.2.1 Snake()

```
Snake::Snake (
    int x = 0,
    int y = 0,
    Board * frame = NULL )
```

Le constructeur du serpent.

Parameters

<i>x</i>	Les coordonnées en x du serpent
<i>y</i>	Les coordonnées en y du serpent
<i>frame</i>	Le plateau

4.5.3 Member Function Documentation

4.5.3.1 changeDirection()

```
void Snake::changeDirection (
    char d ) [inline]
```

Changer des direction.

Parameters

<i>d</i>	La direction
----------	--------------

4.5.3.2 changePosofSnake()

```
void Snake::changePosofSnake (
    Coord * newCoordHead )
```

Changer la position d'élément du serpent.

Parameters

<i>newCoordHead</i>	Les nouvelles positions
---------------------	-------------------------

4.5.3.3 checkPositionInterdit()

```
void Snake::checkPositionInterdit (
    int * game )
```

Vérifier les position interdites.

Parameters

<i>game</i>	Voir comment arreter le jeux
-------------	------------------------------

4.5.3.4 getHeadSnakePos()

```
Coord Snake::getHeadSnakePos ( ) [inline]
```

Chercher la tête du serpent.

Returns

Les coordonnées de la position de la tête du serpent

4.5.3.5 getLengthOfSnake()

```
int Snake::getLengthOfSnake ( ) [inline]
```

Récupérer la longueur du serpent.

Returns

La longueur

4.5.3.6 getPartSnakePos()

```
Coord Snake::getPartSnakePos (
    int pos ) [inline]
```

La position d'une partie du serpent.

Parameters

<i>pos</i>	L'élément
------------	-----------

Returns

La position

4.5.3.7 nextMoveDown()

```
Coord Snake::nextMoveDown ( )
```

Déplacement future en bas.

Returns

Déplacement future en bas

4.5.3.8 nextMoveLeft()

```
Coord Snake::nextMoveLeft ( )
```

Déplacement future de gauche.

Returns

Position future de gauche

4.5.3.9 nextMoveRight()

```
Coord Snake::nextMoveRight ( )
```

Position future de droite.

Returns

Position futur de droite

4.5.3.10 nextMoveUp()

```
Coord Snake::nextMoveUp ( )
```

La position futur d'en haut.

Returns

La position futur d'en haut

4.5.3.11 nextMoving()

```
Coord Snake::nextMoving ( )
```

vérification de la position future

Returns

La position futur

4.5.3.12 setSnakeCoord()

```
void Snake::setSnakeCoord (
    Coord * coord1 ) [inline]
```

Insérer le serpent à une position.

Parameters

<i>coord1</i>	La nouvelle position du serpent
---------------	---------------------------------

The documentation for this class was generated from the following files:

- Game_Snake/snake_game.h
- Game_Snake/snake_game.cpp

Chapter 5

File Documentation

5.1 board.h

```
1 #pragma once
2
3 #ifndef BOARD_H
4 #define BOARD_H
5
6 #include <iostream>
7 #include <cstdlib>
8 #include <ctime>
9 #include <cstdio>
10 #include <vector>
11
12 #include "../color/color.h"
13 #include "../pos/coord.h"
14
15 #define MAX_BOARD_WIDTH 120 // By default
16 #define MAX_BOARD_HEIGHT 160 // By default
17
18 #define VACCUM '\0'
19 #define BORDURE '#'
20 #define HEAD '@'
21
22 class Board
23 {
24 public:
25 /**
26  * CONSTRUCTEURS
27  */
28
29 Board(int w, int h, Coord *pos);
30
31 Board(int w = MAX_BOARD_WIDTH, int h = MAX_BOARD_HEIGHT);
32
33 /**
34  * METHODES PUBLIQUES
35  */
36 void Show(void);
37
38 Coord getPosBoard() {return this->posBoard;}
39
40 int getWidthBoard() {return this->m_width;}
41
42 int getHeightBoard() {return this->m_height;}
43
44 void setPosBoard(Coord *coord1) {CopyCoord(coord1, &this->posBoard);}
45
46 void setPixelColor(Coord *coord1, color *clr1);
47
48 void setElement(Coord *coord1, char elmnt);
49 private:
50 int m_width; // La largeur de la fenetre
51
52 int m_height; // La hauteur de la fenetre
```

```

89
90     Coord posBoard; // La position de la fenetre par rapport a l'ecran
91
92     std::vector< std::vector<color> > pixelBoard;
93
94     std::vector< std::vector<char> > element;
95
96
97     Coord centerBoard(int wScreen, int hScreen);
98
99     void defineBoard(void);
100
101     void initMap(void);
102
103     void initElements(void);
104
105 };
106
107 typedef Board Cell;
108
109 #endif // BOARD_H

```

5.2 controlOutput.h

```

1 #ifndef FILE_CONTROL_OUTPUT_H
2 #define FILE_CONTROL_OUTPUT_H
3
4 #pragma once
5
6 #include <stdio>
7 #include <iostream>
8
9 #define TC_NRM "\x1B[0m"
10
11 #define TC_RED "\x1B[1;31m"
12
13 #define TC_GRN "\x1B[1;32m"
14
15 #define TC_YEL "\x1B[1;33m"
16
17 #define TC_BLU "\x1B[1;34m"
18
19 #define TC_MAG "\x1B[1;34m"
20
21 #define TC_CYN "\x1B[1;36m"
22
23 #define TC_WTH "\x1B[1;37m"
24
25 #define TC_B_NRM "\x1B[0m"
26
27 #define TC_B_RED "\x1B[0;31m"
28
29 #define TC_B_GRN "\x1B[0;32m"
30
31 #define TC_B_YEL "\x1B[0;33m"
32
33 #define TC_B_BLU "\x1B[0;34m"
34
35 #define TC_B_MAG "\x1B[0;34m"
36
37 #define TC_B_CYN "\x1B[0;36m"
38
39 #define TC_B_WTH "\x1B[0;37m"
40
41 #define TC_BG_NRM "\x1B[40m"
42
43 #define TC_BG_RED "\x1B[41m"
44
45 #define TC_BG_GRN "\x1B[42m"
46
47 #define TC_BG_YEL "\x1B[43m"
48
49 #define TC_BG_BLU "\x1B[44m"
50
51 #define TC_BG_MAG "\x1B[45m"
52
53 #define TC_BG_CYN "\x1B[46m"
54
55 #define TC_BG_WTH "\x1B[47m"
56
57 #define tc_clear_screen() std::cout<<"\x1B[2J"
58
59 #define tc_move_cursor(X,Y) std::cout<<"\033["<<Y<<"<<X<<"H"

```



```
87
88 #endif // FILE_CONTROL_OUTPUT_H
```

5.3 color.h

```
1 #ifndef COLOR_CELL_H
2 #define COLOR_CELL_H
3
4 #pragma once
5
6 #include <iostream>
7
9 struct color
10 {
11     /* data */
12     int r;
13     int v;
14     int b;
15 };
17 typedef struct color color;
18
21 void initColor(color *src);
22
26 void ColorCopy(color *clsrc, color *clrdest);
27
33 void createColor(int r, int v, int b, color *clr);
34
38 void permutColor(color *clr1, color *clr2);
39
40 #endif // COLOR_CELL_H
41
```

5.4 food.h

```
1 #ifndef FOOD_H
2 #define FOOD_H
3 #include <iostream>
4 #include <cstdlib>
5 #include <ctime>
6 #include <cstdio>
7 #include <termio.h>
8 #include <unistd.h>
9 #include <fcntl.h>
10
11 #include <vector>
12
13 #include "../board/board.h"
14 #include "../board/controlOutput.h"
15 #include "snake_game.h"
16
18 class Food
19 {
20 public:
21
22     Food(int x = 0, int y = 0, Board *frame = NULL);
23
24     void changeFoodPos(Coord *newCoord);
25
26     Coord aleaNumber(int borneY, int borneX, Snake *snakeEat=NULL);
27
28     void setFoodToBoard();
29
30     void clearFoodToBoard();
31
32     Coord getPosFood() {return this->posFood;}
33
34 private:
35
36     Coord posFood;
37
38     Board *m_frameParent;
39 };
40
41 #endif // FOOD_H
```

5.5 snake_game.h

```

1  #ifndef SNAKE_GAME_H
2  #define SNAKE_GAME_H
3
4  #include <iostream>
5  #include <cstdlib>
6  #include <ctime>
7  #include <cstdio>
8  #include <termio.h>
9  #include <unistd.h>
10 #include <fcntl.h>
11
12 #include <vector>
13
14 #include "../board/board.h"
15
16
17 class Snake
18 {
19 public:
20     Snake(int x = 0, int y = 0, Board *frame = NULL);
21
22     void setSnakeCoord(Coord *coordl) {
23         int x = coordl->x;
24         int y = coordl->y;
25
26         CreateCoord(x, y, &this->posSnake);
27     }
28
29     Coord getHeadSnakePos() {return this->the_snake[0].getPosBoard();}
30
31     int getLengthOfSnake() {return this->the_snake.size();}
32
33     Coord getPartSnakePos(int pos) {return this->the_snake[pos].getPosBoard();}
34
35     void changePosofSnake(Coord *newCoordHead);
36
37     void putSnakeToBoard();
38
39     void clearSnakeToBoard();
40
41     void makeSnakeHead();
42
43     void moving();
44
45     Coord nextMoving();
46
47     void moveUp();
48
49     Coord nextMoveUp();
50
51     void moveRight();
52
53     Coord nextMoveRight();
54
55     void moveDown();
56
57     Coord nextMoveDown();
58
59     void moveLeft();
60
61     Coord nextMoveLeft();
62
63     void grandi();
64
65     void changeDirection(char d) {this->m_direction = d;}
66
67     void changeVelocity() {this->m_velocity += 1;}
68
69     void checkPositionInterdit(int *game);
70
71 private:
72     Coord posSnake;
73
74     std::vector< Cell> the_snake;
75
76     Board *m_frameParent;
77
78     char m_direction;
79
80     int m_velocity;
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130

```

```
132     void createHead();
133 };
134
135 #endif //
```

5.6 coord.h

```
1 #ifndef COORD_CELL_H
2 #define COORD_CELL_H
3
4 #pragma once
5
6 #include <iostream>
7
9 struct Coord
10 {
11     int x;
12     int y;
13 };
15 typedef struct Coord Coord;
16
19 void initCoord(Coord *src);
20
24 void CopyCoord(Coord *src, Coord *dest);
25
30 void CreateCoord(int x, int y, Coord *coord1);
31
35 void permutCoord(Coord *coord1, Coord *coord2);
36
41 int compareCoords(Coord *coord1, Coord *coord2);
42
43 #endif // COORD_CELL_H
```


Index

aleaNumber
Food, [13](#)

Board, [7](#)
Board, [8](#)
getHeightBoard, [8](#)
getPosBoard, [8](#)
getWidthBoard, [9](#)
setElement, [9](#)
setPixelColor, [9](#)
setPosBoard, [10](#)
board/board.h, [19](#)
board/controlOutput.h, [20](#)

changeDirection
Snake, [15](#)
changePosofSnake
Snake, [15](#)
checkPositionInterdit
Snake, [16](#)
color, [10](#)
color/color.h, [21](#)
Coord, [11](#)

Food, [11](#)
aleaNumber, [13](#)
Food, [12](#)

Game_Snake/food.h, [21](#)
Game_Snake/snake_game.h, [22](#)
getHeadSnakePos
Snake, [16](#)
getHeightBoard
Board, [8](#)
getLengthOfSnake
Snake, [16](#)
getPartSnakePos
Snake, [16](#)
getPosBoard
Board, [8](#)
getWidthBoard
Board, [9](#)

nextMoveDown
Snake, [17](#)
nextMoveLeft
Snake, [17](#)
nextMoveRight
Snake, [17](#)
nextMoveUp
Snake, [17](#)

nextMoving
Snake, [18](#)

pos/coord.h, [23](#)

setElement
Board, [9](#)
setPixelColor
Board, [9](#)
setPosBoard
Board, [10](#)
setSnakeCoord
Snake, [18](#)
Snake, [13](#)
changeDirection, [15](#)
changePosofSnake, [15](#)
checkPositionInterdit, [16](#)
getHeadSnakePos, [16](#)
getLengthOfSnake, [16](#)
getPartSnakePos, [16](#)
nextMoveDown, [17](#)
nextMoveLeft, [17](#)
nextMoveRight, [17](#)
nextMoveUp, [17](#)
nextMoving, [18](#)
setSnakeCoord, [18](#)
Snake, [15](#)