

My Project

Generated by Doxygen 1.8.13

Contents

1	Namespace Index	1
1.1	Namespace List	1
2	Hierarchical Index	3
2.1	Class Hierarchy	3
3	Class Index	5
3.1	Class List	5
4	File Index	7
4.1	File List	7
5	Namespace Documentation	9
5.1	Ui Namespace Reference	9
5.1.1	Detailed Description	9
6	Class Documentation	11
6.1	Debug Class Reference	11
6.1.1	Detailed Description	11
6.1.2	Constructor & Destructor Documentation	11
6.1.2.1	Debug()	11
6.1.3	Member Function Documentation	12
6.1.3.1	speedUp()	12
6.2	Intelligence Class Reference	12
6.2.1	Detailed Description	12
6.2.2	Constructor & Destructor Documentation	12

6.2.2.1	Intelligence()	13
6.2.3	Member Function Documentation	13
6.2.3.1	increaseIntelligence()	13
6.2.4	Member Data Documentation	13
6.2.4.1	intelligenceBalance	13
6.3	MainWindow Class Reference	13
6.3.1	Detailed Description	14
6.3.2	Constructor & Destructor Documentation	14
6.3.2.1	MainWindow()	14
6.3.2.2	~MainWindow()	14
6.4	Pencil Class Reference	15
6.4.1	Detailed Description	15
6.4.2	Constructor & Destructor Documentation	15
6.4.2.1	Pencil()	16
6.4.3	Member Function Documentation	16
6.4.3.1	activateIntelligence()	16
6.4.3.2	apm2000()	16
6.4.3.3	buyApm()	16
6.4.3.4	decreasePrice()	16
6.4.3.5	getInventory()	16
6.4.3.6	increasePrice()	17
6.4.3.7	newRate()	17
6.4.3.8	producePencil()	17
6.4.3.9	round()	17
6.4.3.10	sell()	17
6.4.3.11	upgradeApm()	17
6.4.3.12	upgradeMarketing()	17
6.4.4	Member Data Documentation	18
6.4.4.1	apmFractional	18
6.4.4.2	apmUpgradePrice	18

6.4.4.3	marketing	18
6.4.4.4	marketingUpgradePrice	18
6.4.4.5	numberOfApm	18
6.4.4.6	numberOfPencil	18
6.4.4.7	pencilsForUpgrade	18
6.4.4.8	priceofApm	19
6.4.4.9	priceofPencil	19
6.4.4.10	rateofApm	19
6.4.4.11	rateofPencil	19
6.4.4.12	totalNumberOfPencil	19
6.5	Production Class Reference	19
6.5.1	Detailed Description	20
6.5.2	Constructor & Destructor Documentation	20
6.5.2.1	Production()	20
6.5.3	Member Function Documentation	20
6.5.3.1	buyGraphite()	20
6.5.3.2	buyWood()	21
6.5.3.3	calculatePrices()	21
6.5.3.4	decreaseGraphite()	21
6.5.3.5	decreaseWood()	21
6.5.3.6	disable_buyGraphite()	21
6.5.3.7	disable_buyWood()	21
6.5.3.8	getGraphite()	21
6.5.3.9	getWood()	22
6.5.4	Member Data Documentation	22
6.5.4.1	numberOfGraphite	22
6.5.4.2	numberOfWood	22
6.5.4.3	priceofGraphite	22
6.5.4.4	priceofWood	22
6.6	Wallet Class Reference	22
6.6.1	Detailed Description	23
6.6.2	Constructor & Destructor Documentation	23
6.6.2.1	Wallet()	23
6.6.3	Member Function Documentation	23
6.6.3.1	getBalance()	23
6.6.3.2	setBalance()	23
6.6.4	Member Data Documentation	23
6.6.4.1	balance	23

7 File Documentation	25
7.1 /home/fatine/Documents/se-03-team-06/src/debug.h File Reference	25
7.2 /home/fatine/Documents/se-03-team-06/src/intelligence.h File Reference	25
7.2.1 Detailed Description	25
7.3 /home/fatine/Documents/se-03-team-06/src/mainwindow.h File Reference	25
7.3.1 Detailed Description	26
7.4 /home/fatine/Documents/se-03-team-06/src/pencil.h File Reference	26
7.4.1 Detailed Description	26
7.5 /home/fatine/Documents/se-03-team-06/src/production.h File Reference	26
7.5.1 Detailed Description	27
7.6 /home/fatine/Documents/se-03-team-06/src/wallet.h File Reference	27
7.6.1 Detailed Description	27
Index	29

Chapter 1

Namespace Index

1.1 Namespace List

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

Ui	Getting MainWondow class from Ui namespace	9
--------------------	--	-------------------

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Intelligence	12
Pencil	15
Debug	11
MainWindow	13
QMainWindow	
MainWindow	13
Wallet	22
Production	19
Pencil	15

Chapter 3

Class Index

3.1 Class List

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

Debug	Class to implement Debug functionality	11
Intelligence	Class to implement Intelligence currency	12
MainWindow	Class to implement the MainWindow	13
Pencil	Class to implement pencil game	15
Production	Class to implement production of the pencils	19
Wallet	Class to implement the wallet of the player	22

Chapter 4

File Index

4.1 File List

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

/home/fatine/Documents/se-03-team-06/src/ debug.h	
This header file contains functionalities to debug the game	25
/home/fatine/Documents/se-03-team-06/src/ intelligence.h	
Intelligence currency implentation and functionalities	25
/home/fatine/Documents/se-03-team-06/src/ mainwindow.h	
Main window functionality	25
/home/fatine/Documents/se-03-team-06/src/ pencil.h	
This header file contains the functionalities for buying and selling pencil and the Auto Pencil Machine (APM)	26
/home/fatine/Documents/se-03-team-06/src/ production.h	
This header file contains functionalities to buy and sell wood and graphite	26
/home/fatine/Documents/se-03-team-06/src/ wallet.h	
This header file contains required definitions for Wallet . It only stores Balance which is the money the user has in the bank	27

Chapter 5

Namespace Documentation

5.1 Ui Namespace Reference

Getting MainWindow class from [Ui](#) namespace.

5.1.1 Detailed Description

Getting MainWindow class from [Ui](#) namespace.

Chapter 6

Class Documentation

6.1 Debug Class Reference

Class to implement [Debug](#) functionality.

```
#include <debug.h>
```

Inheritance diagram for Debug:

Collaboration diagram for Debug:

Public Member Functions

- [Debug](#) ()
- void [speedUp](#) ()

Additional Inherited Members

6.1.1 Detailed Description

Class to implement [Debug](#) functionality.

We define the constructor, destructor and necessary methods.

6.1.2 Constructor & Destructor Documentation

6.1.2.1 Debug()

```
Debug::Debug ( )
```

Constructor.

6.1.3 Member Function Documentation

6.1.3.1 speedUp()

```
void Debug::speedUp ( )
```

Function to speed up the specified values.

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

- [/home/fatine/Documents/se-03-team-06/src/debug.h](#)
- [/home/fatine/Documents/se-03-team-06/src/debug.cpp](#)

6.2 Intelligence Class Reference

Class to implement [Intelligence](#) currency.

```
#include <intelligence.h>
```

Inheritance diagram for Intelligence:

Public Member Functions

- [Intelligence](#) ()
- void [increaseIntelligence](#) ()

Public Attributes

- int [intelligenceBalance](#)
- bool [intelligenceIsActive](#)

6.2.1 Detailed Description

Class to implement [Intelligence](#) currency.

We define the constructor and necessary methods.

6.2.2 Constructor & Destructor Documentation

6.2.2.1 Intelligence()

```
Intelligence::Intelligence ( )
```

Constructor.

6.2.3 Member Function Documentation

6.2.3.1 increaseIntelligence()

```
void Intelligence::increaseIntelligence ( )
```

Insert your comment here.

6.2.4 Member Data Documentation

6.2.4.1 intelligenceBalance

```
int Intelligence::intelligenceBalance
```

Insert your comment here.

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

- [/home/fatine/Documents/se-03-team-06/src/intelligence.h](#)
- [/home/fatine/Documents/se-03-team-06/src/intelligence.cpp](#)

6.3 MainWindow Class Reference

Class to implement the [MainWindow](#).

```
#include <mainwindow.h>
```

Inheritance diagram for MainWindow:

Collaboration diagram for MainWindow:

Public Member Functions

- [MainWindow](#) (QWidget *parent=nullptr)
Mainwindow constructor.
- [~MainWindow](#) ()

Additional Inherited Members

6.3.1 Detailed Description

Class to implement the [MainWindow](#).

class [MainWindow](#) extending from QMainWindow uses Q_OBJECT macro defined in QMainWindow header

6.3.2 Constructor & Destructor Documentation

6.3.2.1 MainWindow()

```
MainWindow::MainWindow (
    QWidget * parent = nullptr ) [explicit]
```

Mainwindow constructor.

Constructor. Label is set to display as soon as the [MainWindow](#) class is constructed.

Timer initialization.

This timer is used for calculating prices of graphite and wood every 5 sec.

Action to be taken when save is clicked

Action to be taken when load is clicked

Not displaying the debugging tool option

Not displaying the debugging tool

Action to be taken when debug is clicked

Action to be taken when speed up is clicked

6.3.2.2 ~MainWindow()

```
MainWindow::~MainWindow ( )
```

Destructor.

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

- [/home/fatine/Documents/se-03-team-06/src/mainwindow.h](#)
- [/home/fatine/Documents/se-03-team-06/src/mainwindow.cpp](#)

6.4 Pencil Class Reference

Class to implement pencil game.

```
#include <pencil.h>
```

Inheritance diagram for Pencil:

Collaboration diagram for Pencil:

Public Member Functions

- [Pencil](#) ()
- int [getInventory](#) ()
- void [producePencil](#) ()
- void [increasePrice](#) ()
- void [decreasePrice](#) ()
- void [sell](#) ()
- void [newRate](#) ()
- void [buyApm](#) ()
- void [apm2000](#) ()
- void [upgradeApm](#) ()
- bool [activateIntelligence](#) ()
- double [round](#) (double var)
- void [upgradeMarketing](#) ()

Public Attributes

- double [priceofPencil](#)
- double [rateofPencil](#)
- int [numberofPencil](#)
- double [totalnumberofPencil](#)
- int [pencilsForUpgrade](#)
- double [numberofApm](#)
- double [priceofApm](#)
- double [rateofApm](#)
- double [apmFractional](#)
- int [apmUpgradePrice](#)
- int [marketing](#)
- double [marketingUpgradePrice](#)

6.4.1 Detailed Description

Class to implement pencil game.

We define the constructor, variables and necessary methods.

6.4.2 Constructor & Destructor Documentation

6.4.2.1 Pencil()

```
Pencil::Pencil ( )
```

Constructor.

6.4.3 Member Function Documentation

6.4.3.1 activateIntelligence()

```
bool Pencil::activateIntelligence ( )
```

Insert your comment here.

6.4.3.2 apm2000()

```
void Pencil::apm2000 ( )
```

APM 2000.

Function to produce pencils for the APM

6.4.3.3 buyApm()

```
void Pencil::buyApm ( )
```

Buy APM.

Function to buy APM and increase its price by 10% after every purchase

6.4.3.4 decreasePrice()

```
void Pencil::decreasePrice ( )
```

Decrease price of pencil.

6.4.3.5 getInventory()

```
int Pencil::getInventory ( )
```

Returns pencils in inventory.

6.4.3.6 increasePrice()

```
void Pencil::increasePrice ( )
```

Increase price of pencil.

6.4.3.7 newRate()

```
void Pencil::newRate ( )
```

New rate of pencil sells.

Function to calculate rate of a pencil every 0.20 seconds

6.4.3.8 producePencil()

```
void Pencil::producePencil ( )
```

Produce pencils.

6.4.3.9 round()

```
double Pencil::round (
    double var )
```

Function to round number to 2 decimal places.

Function to round doubles to two decimal places

6.4.3.10 sell()

```
void Pencil::sell ( )
```

Sells pencil.

Sell pencils (according to rate) every second Selling pencils in bulk

6.4.3.11 upgradeApm()

```
void Pencil::upgradeApm ( )
```

Upgrade of APM.

6.4.3.12 upgradeMarketing()

```
void Pencil::upgradeMarketing ( )
```

Function to upgrade market.

6.4.4 Member Data Documentation

6.4.4.1 apmFractional

```
double Pencil::apmFractional
```

APM fractional.

6.4.4.2 apmUpgradePrice

```
int Pencil::apmUpgradePrice
```

Price of APM.

6.4.4.3 marketing

```
int Pencil::marketing
```

Marketing price.

6.4.4.4 marketingUpgradePrice

```
double Pencil::marketingUpgradePrice
```

Marketing price upgrade.

6.4.4.5 numberOfApm

```
double Pencil::numberOfApm
```

Number of APMs.

6.4.4.6 numberOfPencil

```
int Pencil::numberOfPencil
```

Number of pencil produced.

6.4.4.7 pencilsForUpgrade

```
int Pencil::pencilsForUpgrade
```

Pencils for upgrade.

6.4.4.8 priceofApm

```
double Pencil::priceofApm
```

Price of APMS.

6.4.4.9 priceofPencil

```
double Pencil::priceofPencil
```

Price of pencil.

6.4.4.10 rateofApm

```
double Pencil::rateofApm
```

Rate of APMS.

6.4.4.11 rateofPencil

```
double Pencil::rateofPencil
```

Sell rate of pencil.

6.4.4.12 totalnumberofPencil

```
double Pencil::totalnumberofPencil
```

Total number of pencil produced.

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

- /home/fatine/Documents/se-03-team-06/src/[pencil.h](#)
- /home/fatine/Documents/se-03-team-06/src/pencil.cpp

6.5 Production Class Reference

Class to implement production of the pencils.

```
#include <production.h>
```

Inheritance diagram for Production:

Collaboration diagram for Production:

Public Member Functions

- [Production](#) ()
- void [buyWood](#) ()
- void [buyGraphite](#) ()
- bool [disable_buyWood](#) ()
- bool [disable_buyGraphite](#) ()
- void [calculatePrices](#) ()
- void [decreaseWood](#) ()
- void [decreaseGraphite](#) ()
- double [getWood](#) ()
- double [getGraphite](#) ()

Public Attributes

- double [priceofWood](#)
- double [priceofGraphite](#)
- double [numberofWood](#)
- double [numberofGraphite](#)

6.5.1 Detailed Description

Class to implement production of the pencils.

We define the constructor, variables and necessary methods.

6.5.2 Constructor & Destructor Documentation

6.5.2.1 Production()

```
Production::Production ( )
```

Constructor.

6.5.3 Member Function Documentation

6.5.3.1 buyGraphite()

```
void Production::buyGraphite ( )
```

Buy graphite function.

Function to buy Graphite if there is enough balance

6.5.3.2 buyWood()

```
void Production::buyWood ( )
```

Buy wood function.

Function to buy wood if there is enough balance

6.5.3.3 calculatePrices()

```
void Production::calculatePrices ( )
```

Calculates price of wood and graphite function.

Function to calculate price of Wood and Graphite. Price of Wood and Graphite is a random number between 1000 to 2000.

Price of Graphite is a random number between 1500 to 2500.

6.5.3.4 decreaseGraphite()

```
void Production::decreaseGraphite ( )
```

Consume graphite function.

6.5.3.5 decreaseWood()

```
void Production::decreaseWood ( )
```

Consumes wood function.

6.5.3.6 disable_buyGraphite()

```
bool Production::disable_buyGraphite ( )
```

Disable buy graphite function.

6.5.3.7 disable_buyWood()

```
bool Production::disable_buyWood ( )
```

Disable buy wood function.

6.5.3.8 getGraphite()

```
double Production::getGraphite ( )
```

return number of graphite.

6.5.3.9 getWood()

```
double Production::getWood ( )
```

return number of wood.

6.5.4 Member Data Documentation

6.5.4.1 numberOfGraphite

```
double Production::numberOfGraphite
```

Amount of graphite.

6.5.4.2 numberOfWood

```
double Production::numberOfWood
```

Amount of wood.

6.5.4.3 priceofGraphite

```
double Production::priceofGraphite
```

Price of Graphite.

6.5.4.4 priceofWood

```
double Production::priceofWood
```

Price of Wood.

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

- /home/fatine/Documents/se-03-team-06/src/[production.h](#)
- /home/fatine/Documents/se-03-team-06/src/production.cpp

6.6 Wallet Class Reference

Class to implement the wallet of the player.

```
#include <wallet.h>
```

Inheritance diagram for Wallet:

Public Member Functions

- [Wallet](#) ()
- void [setBalance](#) (double newBalance)
- double [getBalance](#) ()

Public Attributes

- double [balance](#)

6.6.1 Detailed Description

Class to implement the wallet of the player.

We define the constructor, variables and necessary methods.

6.6.2 Constructor & Destructor Documentation

6.6.2.1 [Wallet\(\)](#)

```
Wallet::Wallet ( )
```

Constructor.

6.6.3 Member Function Documentation

6.6.3.1 [getBalance\(\)](#)

```
double Wallet::getBalance ( )
```

Returns the balance.

6.6.3.2 [setBalance\(\)](#)

```
void Wallet::setBalance (
    double newBalance )
```

Sets the balance.

6.6.4 Member Data Documentation

6.6.4.1 [balance](#)

```
double Wallet::balance
```

Balance in wallet.

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

- [/home/fatine/Documents/se-03-team-06/src/wallet.h](#)
- [/home/fatine/Documents/se-03-team-06/src/wallet.cpp](#)

Chapter 7

File Documentation

7.1 /home/fatine/Documents/se-03-team-06/src/debug.h File Reference

This header file contains functionalities to debug the game.

This graph shows which files directly or indirectly include this file:

7.2 /home/fatine/Documents/se-03-team-06/src/intelligence.h File Reference

[Intelligence](#) currency implentation and functionalities.

This graph shows which files directly or indirectly include this file:

Classes

- class [Intelligence](#)
Class to implement [Intelligence](#) currency.

7.2.1 Detailed Description

[Intelligence](#) currency implentation and functionalities.

7.3 /home/fatine/Documents/se-03-team-06/src/mainwindow.h File Reference

Main window functionality.

```
#include <QMainWindow>
#include "pencil.h"
#include "production.h"
#include "wallet.h"
#include "intelligence.h"
#include "debug.h"
```

Include dependency graph for mainwindow.h:

Classes

- class [MainWindow](#)

Class to implement the [MainWindow](#).

Namespaces

- [Ui](#)

Getting [MainWondow](#) class from [Ui](#) namespace.

7.3.1 Detailed Description

Main window functionality.

7.4 /home/fatine/Documents/se-03-team-06/src/pencil.h File Reference

this header file contains the functionalities for buying and selling pencil and the Auto [Pencil](#) Machine (APM).

```
#include "production.h"
#include "wallet.h"
#include "intelligence.h"
```

Include dependency graph for pencil.h: This graph shows which files directly or indirectly include this file:

Classes

- class [Pencil](#)

Class to implement pencil game.

7.4.1 Detailed Description

this header file contains the functionalities for buying and selling pencil and the Auto [Pencil](#) Machine (APM).

7.5 /home/fatine/Documents/se-03-team-06/src/production.h File Reference

This header file contains functionalities to buy and sell wood and graphite.

```
#include "wallet.h"
```

Include dependency graph for production.h: This graph shows which files directly or indirectly include this file:

Classes

- class [Production](#)

Class to implement production of the pencils.

7.5.1 Detailed Description

This header file contains functionalities to buy and sell wood and graphite.

7.6 /home/fatine/Documents/se-03-team-06/src/wallet.h File Reference

This header file contains required definitions for [Wallet](#). It only stores Balance which is the money the user has in the bank.

This graph shows which files directly or indirectly include this file:

Classes

- class [Wallet](#)

Class to implement the wallet of the player.

7.6.1 Detailed Description

This header file contains required definitions for [Wallet](#). It only stores Balance which is the money the user has in the bank.

Index

/home/fatine/Documents/se-03-team-06/src/debug.h, [25](#)
/home/fatine/Documents/se-03-team-06/src/intelligence.h, [25](#)
/home/fatine/Documents/se-03-team-06/src/mainwindow.h, [25](#)
/home/fatine/Documents/se-03-team-06/src/pencil.h, [26](#)
/home/fatine/Documents/se-03-team-06/src/production.h, [26](#)
/home/fatine/Documents/se-03-team-06/src/wallet.h, [27](#)
~MainWindow
 MainWindow, [14](#)

activateIntelligence
 Pencil, [16](#)
apm2000
 Pencil, [16](#)
apmFractional
 Pencil, [18](#)
apmUpgradePrice
 Pencil, [18](#)

balance
 Wallet, [23](#)
buyApm
 Pencil, [16](#)
buyGraphite
 Production, [20](#)
buyWood
 Production, [20](#)

calculatePrices
 Production, [21](#)

Debug, [11](#)
 Debug, [11](#)
 speedUp, [12](#)
decreaseGraphite
 Production, [21](#)
decreasePrice
 Pencil, [16](#)
decreaseWood
 Production, [21](#)
disable_buyGraphite
 Production, [21](#)
disable_buyWood
 Production, [21](#)

getBalance
 Wallet, [23](#)
getGraphite
 Production, [21](#)
getInventory
 Pencil, [16](#)
getWood
 Production, [21](#)

increaseIntelligence
 Intelligence, [13](#)
increasePrice
 Pencil, [16](#)
Intelligence, [12](#)
 increaseIntelligence, [13](#)
 Intelligence, [12](#)
 intelligenceBalance, [13](#)
intelligenceBalance
 Intelligence, [13](#)

MainWindow, [13](#)
 ~MainWindow, [14](#)
 MainWindow, [14](#)
marketing
 Pencil, [18](#)
marketingUpgradePrice
 Pencil, [18](#)

newRate
 Pencil, [17](#)
numberOfApm
 Pencil, [18](#)
numberOfGraphite
 Production, [22](#)
numberOfPencil
 Pencil, [18](#)
numberOfWood
 Production, [22](#)

Pencil, [15](#)
 activateIntelligence, [16](#)
 apm2000, [16](#)
 apmFractional, [18](#)
 apmUpgradePrice, [18](#)
 buyApm, [16](#)
 decreasePrice, [16](#)
 getInventory, [16](#)
 increasePrice, [16](#)
 marketing, [18](#)
 marketingUpgradePrice, [18](#)
 newRate, [17](#)
 numberOfApm, [18](#)
 numberOfPencil, [18](#)
 Pencil, [15](#)

- pencilsForUpgrade, [18](#)
- priceofApm, [18](#)
- priceofPencil, [19](#)
- producePencil, [17](#)
- rateofApm, [19](#)
- rateofPencil, [19](#)
- round, [17](#)
- sell, [17](#)
- totalnumberofPencil, [19](#)
- upgradeApm, [17](#)
- upgradeMarketing, [17](#)
- pencilsForUpgrade
 - Pencil, [18](#)
- priceofApm
 - Pencil, [18](#)
- priceofGraphite
 - Production, [22](#)
- priceofPencil
 - Pencil, [19](#)
- priceofWood
 - Production, [22](#)
- producePencil
 - Pencil, [17](#)
- Production, [19](#)
 - buyGraphite, [20](#)
 - buyWood, [20](#)
 - calculatePrices, [21](#)
 - decreaseGraphite, [21](#)
 - decreaseWood, [21](#)
 - disable_buyGraphite, [21](#)
 - disable_buyWood, [21](#)
 - getGraphite, [21](#)
 - getWood, [21](#)
 - numberOfGraphite, [22](#)
 - numberOfWood, [22](#)
 - priceofGraphite, [22](#)
 - priceofWood, [22](#)
 - Production, [20](#)
- rateofApm
 - Pencil, [19](#)
- rateofPencil
 - Pencil, [19](#)
- round
 - Pencil, [17](#)
- sell
 - Pencil, [17](#)
- setBalance
 - Wallet, [23](#)
- speedUp
 - Debug, [12](#)
- totalnumberofPencil
 - Pencil, [19](#)
- Ui, [9](#)
- upgradeApm
 - Pencil, [17](#)
- upgradeMarketing
 - Pencil, [17](#)
- Wallet, [22](#)
 - balance, [23](#)
 - getBalance, [23](#)
 - setBalance, [23](#)
 - Wallet, [23](#)