

# The Merry Turnip

## preAlpha

Generated by Doxygen 1.6.1

Wed Sep 9 18:20:14 2009



# Contents

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>The Merry Turnip</b>                          | <b>1</b>  |
| 1.1      | Introduction . . . . .                           | 1         |
| <b>2</b> | <b>Todo List</b>                                 | <b>3</b>  |
| <b>3</b> | <b>Bug List</b>                                  | <b>5</b>  |
| <b>4</b> | <b>Module Index</b>                              | <b>7</b>  |
| 4.1      | Modules . . . . .                                | 7         |
| <b>5</b> | <b>Class Index</b>                               | <b>9</b>  |
| 5.1      | Class Hierarchy . . . . .                        | 9         |
| <b>6</b> | <b>Class Index</b>                               | <b>11</b> |
| 6.1      | Class List . . . . .                             | 11        |
| <b>7</b> | <b>File Index</b>                                | <b>13</b> |
| 7.1      | File List . . . . .                              | 13        |
| <b>8</b> | <b>Module Documentation</b>                      | <b>15</b> |
| 8.1      | XML Scripts . . . . .                            | 15        |
| 8.1.1    | Detailed Description . . . . .                   | 15        |
| <b>9</b> | <b>Class Documentation</b>                       | <b>17</b> |
| 9.1      | actionMesh Class Reference . . . . .             | 17        |
| 9.1.1    | Detailed Description . . . . .                   | 18        |
| 9.1.2    | Constructor & Destructor Documentation . . . . . | 18        |
| 9.1.2.1  | actionMesh . . . . .                             | 18        |
| 9.1.2.2  | ~actionMesh . . . . .                            | 18        |
| 9.1.3    | Member Function Documentation . . . . .          | 19        |
| 9.1.3.1  | getAction . . . . .                              | 19        |

|         |  |    |
|---------|--|----|
| 9.1.3.2 | <a href="#">getColor</a>                                   | 19 |
| 9.1.3.3 | <a href="#">getCombi</a>                                   | 19 |
| 9.1.3.4 | <a href="#">init</a>                                       | 19 |
| 9.1.4   | <a href="#">Member Data Documentation</a>                  | 19 |
| 9.1.4.1 | <a href="#">action</a>                                     | 19 |
| 9.1.4.2 | <a href="#">color</a>                                      | 20 |
| 9.1.4.3 | <a href="#">combi</a>                                      | 20 |
| 9.2     | <a href="#">animateMesh Class Reference</a>                | 21 |
| 9.2.1   | <a href="#">Detailed Description</a>                       | 22 |
| 9.2.2   | <a href="#">Constructor &amp; Destructor Documentation</a> | 22 |
| 9.2.2.1 | <a href="#">animateMesh</a>                                | 22 |
| 9.2.2.2 | <a href="#">~animateMesh</a>                               | 22 |
| 9.2.3   | <a href="#">Member Function Documentation</a>              | 22 |
| 9.2.3.1 | <a href="#">getAnim</a>                                    | 22 |
| 9.2.3.2 | <a href="#">getBegin</a>                                   | 22 |
| 9.2.3.3 | <a href="#">getEnd</a>                                     | 22 |
| 9.2.3.4 | <a href="#">init</a>                                       | 23 |
| 9.2.4   | <a href="#">Member Data Documentation</a>                  | 23 |
| 9.2.4.1 | <a href="#">anim</a>                                       | 23 |
| 9.2.4.2 | <a href="#">begin</a>                                      | 23 |
| 9.2.4.3 | <a href="#">end</a>  | 23 |
| 9.3     | <a href="#">base Class Reference</a>                       | 25 |
| 9.3.1   | <a href="#">Detailed Description</a>                       | 26 |
| 9.3.2   | <a href="#">Constructor &amp; Destructor Documentation</a> | 26 |
| 9.3.2.1 | <a href="#">base</a>                                       | 26 |
| 9.3.2.2 | <a href="#">~base</a>                                      | 27 |
| 9.3.3   | <a href="#">Member Function Documentation</a>              | 27 |
| 9.3.3.1 | <a href="#">close</a>                                      | 27 |
| 9.3.3.2 | <a href="#">getId</a>                                      | 27 |
| 9.3.3.3 | <a href="#">init</a>                                       | 27 |
| 9.3.3.4 | <a href="#">isInit</a>                                     | 28 |
| 9.3.3.5 | <a href="#">log</a>  | 28 |
| 9.3.3.6 | <a href="#">update</a>                                     | 28 |
| 9.3.4   | <a href="#">Member Data Documentation</a>                  | 28 |
| 9.3.4.1 | <a href="#">id</a>   | 28 |
| 9.3.4.2 | <a href="#">initialize</a>                                 | 29 |

|          |  |    |
|----------|--|----|
| 9.3.4.3  | logId                                  | 29 |
| 9.3.4.4  | mycore                                 | 29 |
| 9.3.4.5  | nameClass                              | 29 |
| 9.4      | chap Class Reference                   | 30 |
| 9.4.1    | Detailed Description                   | 31 |
| 9.4.2    | Constructor & Destructor Documentation | 31 |
| 9.4.2.1  | chap                                   | 31 |
| 9.4.2.2  | ~chap                                  | 31 |
| 9.4.3    | Member Function Documentation          | 31 |
| 9.4.3.1  | addItem                                | 31 |
| 9.4.3.2  | close                                  | 32 |
| 9.4.3.3  | getItemFromId                          | 32 |
| 9.4.3.4  | init                                   | 32 |
| 9.4.3.5  | OnEvent                                | 33 |
| 9.4.3.6  | update                                 | 33 |
| 9.4.4    | Member Data Documentation              | 33 |
| 9.4.4.1  | listItem                               | 33 |
| 9.5      | coeur Class Reference                  | 34 |
| 9.5.1    | Detailed Description                   | 38 |
| 9.5.2    | Constructor & Destructor Documentation | 38 |
| 9.5.2.1  | coeur                                  | 38 |
| 9.5.2.2  | ~coeur                                 | 38 |
| 9.5.3    | Member Function Documentation          | 38 |
| 9.5.3.1  | addLight                               | 38 |
| 9.5.3.2  | addMesh                                | 39 |
| 9.5.3.3  | addParticleSystem                      | 39 |
| 9.5.3.4  | anim                                   | 39 |
| 9.5.3.5  | close                                  | 39 |
| 9.5.3.6  | getActionMesh                          | 39 |
| 9.5.3.7  | getActionMesh                          | 39 |
| 9.5.3.8  | getCombinaison                         | 40 |
| 9.5.3.9  | getCurPos                              | 40 |
| 9.5.3.10 | getDim                                 | 40 |
| 9.5.3.11 | getItem                                | 40 |
| 9.5.3.12 | getLogger                              | 40 |
| 9.5.3.13 | getObservationItem                     | 40 |

|          |  |    |
|----------|--|----|
| 9.5.3.14 | <a href="#">getObservationMesh</a>                         | 41 |
| 9.5.3.15 | <a href="#">getSelectedAction</a>                          | 41 |
| 9.5.3.16 | <a href="#">getSelectedNode</a>                            | 41 |
| 9.5.3.17 | <a href="#">getTexture</a>                                 | 41 |
| 9.5.3.18 | <a href="#">getTime</a>                                    | 41 |
| 9.5.3.19 | <a href="#">getVersion</a>                                 | 41 |
| 9.5.3.20 | <a href="#">init</a>                                       | 42 |
| 9.5.3.21 | <a href="#">loadChap</a>                                   | 42 |
| 9.5.3.22 | <a href="#">loadLieu</a>                                   | 42 |
| 9.5.3.23 | <a href="#">loadLocal</a>                                  | 42 |
| 9.5.3.24 | <a href="#">OnEvent</a>                                    | 43 |
| 9.5.3.25 | <a href="#">pickNode</a>                                   | 43 |
| 9.5.3.26 | <a href="#">play</a>                                       | 43 |
| 9.5.3.27 | <a href="#">setCaption</a>                                 | 43 |
| 9.5.3.28 | <a href="#">setSkyBox</a>                                  | 43 |
| 9.5.3.29 | <a href="#">startCredit</a>                                | 43 |
| 9.5.3.30 | <a href="#">update</a>                                     | 44 |
| 9.5.4    | <a href="#">Friends And Related Function Documentation</a> | 44 |
| 9.5.4.1  | <a href="#">console</a>                                    | 44 |
| 9.5.4.2  | <a href="#">pyInt</a>                                      | 44 |
| 9.5.5    | <a href="#">Member Data Documentation</a>                  | 44 |
| 9.5.5.1  | <a href="#">activeIn</a>                                   | 44 |
| 9.5.5.2  | <a href="#">activeScene</a>                                | 44 |
| 9.5.5.3  | <a href="#">bits</a>                                       | 45 |
| 9.5.5.4  | <a href="#">cam</a>  | 45 |
| 9.5.5.5  | <a href="#">co</a>   | 45 |
| 9.5.5.6  | <a href="#">curPos</a>                                     | 45 |
| 9.5.5.7  | <a href="#">currentChap</a>                                | 45 |
| 9.5.5.8  | <a href="#">currentLieu</a>                                | 45 |
| 9.5.5.9  | <a href="#">currentLocal</a>                               | 45 |
| 9.5.5.10 | <a href="#">device</a>                                     | 46 |
| 9.5.5.11 | <a href="#">dim</a>  | 46 |
| 9.5.5.12 | <a href="#">dr</a>   | 46 |
| 9.5.5.13 | <a href="#">driver</a>                                     | 46 |
| 9.5.5.14 | <a href="#">env</a>  | 46 |
| 9.5.5.15 | <a href="#">fullscreen</a>                                 | 46 |

|          |  |    |
|----------|--|----|
| 9.5.5.16 | in                                     | 46 |
| 9.5.5.17 | lang                                   | 47 |
| 9.5.5.18 | metaSelector                           | 47 |
| 9.5.5.19 | mycredit                               | 47 |
| 9.5.5.20 | mylogger                               | 47 |
| 9.5.5.21 | pathChap                               | 47 |
| 9.5.5.22 | pathGame                               | 47 |
| 9.5.5.23 | pathLieu                               | 47 |
| 9.5.5.24 | pathLocal                              | 48 |
| 9.5.5.25 | py                                     | 48 |
| 9.5.5.26 | q                                      | 48 |
| 9.5.5.27 | selectedAction                         | 48 |
| 9.5.5.28 | smgr                                   | 48 |
| 9.5.5.29 | stencilbuffer                          | 48 |
| 9.5.5.30 | vsync                                  | 48 |
| 9.6      | console Class Reference                | 50 |
| 9.6.1    | Detailed Description                   | 51 |
| 9.6.2    | Constructor & Destructor Documentation | 51 |
| 9.6.2.1  | console                                | 51 |
| 9.6.2.2  | ~console                               | 52 |
| 9.6.3    | Member Function Documentation          | 52 |
| 9.6.3.1  | close                                  | 52 |
| 9.6.3.2  | exit                                   | 52 |
| 9.6.3.3  | fps                                    | 52 |
| 9.6.3.4  | help                                   | 52 |
| 9.6.3.5  | init                                   | 52 |
| 9.6.3.6  | OnEvent                                | 53 |
| 9.6.3.7  | position                               | 53 |
| 9.6.3.8  | quit                                   | 53 |
| 9.6.3.9  | rotation                               | 53 |
| 9.6.3.10 | target                                 | 53 |
| 9.6.3.11 | update                                 | 53 |
| 9.6.3.12 | write                                  | 54 |
| 9.6.4    | Member Data Documentation              | 54 |
| 9.6.4.1  | box                                    | 54 |
| 9.6.4.2  | cons                                   | 54 |

|         |   |    |
|---------|---|----|
| 9.6.4.3 | device  | 54 |
| 9.6.4.4 | env   | 54 |
| 9.6.4.5 | FPS   | 55 |
| 9.6.4.6 | history   | 55 |
| 9.6.4.7 | histPos   | 55 |
| 9.6.4.8 | text  | 55 |
| 9.6.4.9 | tx  | 55 |
| 9.7     | credit Class Reference                                  | 56 |
| 9.7.1   | Detailed Description                                    | 56 |
| 9.7.2   | Constructor & Destructor Documentation                  | 56 |
| 9.7.2.1 | credit  | 56 |
| 9.7.2.2 | ~credit   | 57 |
| 9.7.3   | Member Function Documentation                           | 57 |
| 9.7.3.1 | addLogo   | 57 |
| 9.7.3.2 | addText   | 57 |
| 9.7.3.3 | close   | 57 |
| 9.7.3.4 | init  | 57 |
| 9.7.3.5 | OnEvent   | 57 |
| 9.7.3.6 | start   | 58 |
| 9.7.3.7 | stop  | 58 |
| 9.7.3.8 | update  | 58 |
| 9.7.4   | Member Data Documentation                               | 58 |
| 9.7.4.1 | activate  | 58 |
| 9.7.4.2 | env   | 58 |
| 9.7.4.3 | logos   | 58 |
| 9.7.4.4 | mainFont  | 58 |
| 9.7.4.5 | newY  | 59 |
| 9.7.4.6 | sound   | 59 |
| 9.7.4.7 | texts   | 59 |
| 9.7.4.8 | y   | 59 |
| 9.8     | cusString_python< cusString > Struct Template Reference | 60 |
| 9.8.1   | Detailed Description                                    | 60 |
| 9.8.2   | Constructor & Destructor Documentation                  | 60 |
| 9.8.2.1 | cusString_python  | 60 |
| 9.8.3   | Member Function Documentation                           | 60 |
| 9.8.3.1 | construct   | 60 |



|          |  |    |
|----------|--|----|
| 9.8.3.2  | convert                                | 60 |
| 9.8.3.3  | convertible                            | 60 |
| 9.9      | interface Class Reference              | 62 |
| 9.9.1    | Detailed Description                   | 63 |
| 9.9.2    | Constructor & Destructor Documentation | 63 |
| 9.9.2.1  | interface                              | 63 |
| 9.9.2.2  | ~interface                             | 63 |
| 9.9.3    | Member Function Documentation          | 63 |
| 9.9.3.1  | addInventaire                          | 63 |
| 9.9.3.2  | addItemInventaire                      | 64 |
| 9.9.3.3  | afficherInventaire                     | 64 |
| 9.9.3.4  | close                                  | 64 |
| 9.9.3.5  | getFreePlace                           | 64 |
| 9.9.3.6  | getInventaire                          | 64 |
| 9.9.3.7  | init                                   | 64 |
| 9.9.3.8  | OnEvent                                | 64 |
| 9.9.3.9  | setImage                               | 65 |
| 9.9.3.10 | setImagePosition                       | 65 |
| 9.9.3.11 | speak                                  | 65 |
| 9.9.3.12 | update                                 | 65 |
| 9.9.4    | Member Data Documentation              | 66 |
| 9.9.4.1  | classic                                | 66 |
| 9.9.4.2  | curAction                              | 66 |
| 9.9.4.3  | curDefault                             | 66 |
| 9.9.4.4  | curLook                                | 66 |
| 9.9.4.5  | curTake                                | 66 |
| 9.9.4.6  | emCur                                  | 66 |
| 9.9.4.7  | env                                    | 66 |
| 9.9.4.8  | handle                                 | 66 |
| 9.9.4.9  | image                                  | 67 |
| 9.9.4.10 | inv                                    | 67 |
| 9.9.4.11 | mainFont                               | 67 |
| 9.9.4.12 | pafCur                                 | 67 |
| 9.9.4.13 | psCur                                  | 67 |
| 9.9.4.14 | pselectedInv                           | 67 |
| 9.9.4.15 | pselectedItem                          | 67 |

|           |  |    |
|-----------|--|----|
| 9.9.4.16  | selectedInv                            | 67 |
| 9.9.4.17  | selectedItem                           | 68 |
| 9.9.4.18  | soundFind                              | 68 |
| 9.9.4.19  | time                                   | 68 |
| 9.9.4.20  | txt                                    | 68 |
| 9.9.4.21  | v                                      | 68 |
| 9.10      | inventaire Class Reference             | 69 |
| 9.10.1    | Detailed Description                   | 69 |
| 9.10.2    | Constructor & Destructor Documentation | 70 |
| 9.10.2.1  | inventaire                             | 70 |
| 9.10.2.2  | ~inventaire                            | 70 |
| 9.10.3    | Member Function Documentation          | 70 |
| 9.10.3.1  | addItem                                | 70 |
| 9.10.3.2  | afficher                               | 70 |
| 9.10.3.3  | close                                  | 70 |
| 9.10.3.4  | getFreePlace                           | 70 |
| 9.10.3.5  | getImage                               | 70 |
| 9.10.3.6  | getItem                                | 71 |
| 9.10.3.7  | getItem                                | 71 |
| 9.10.3.8  | getNumItem                             | 71 |
| 9.10.3.9  | init                                   | 71 |
| 9.10.3.10 | OnEvent                                | 71 |
| 9.10.3.11 | removeItem                             | 71 |
| 9.10.3.12 | update                                 | 71 |
| 9.10.4    | Member Data Documentation              | 72 |
| 9.10.4.1  | env                                    | 72 |
| 9.10.4.2  | image                                  | 72 |
| 9.10.4.3  | images                                 | 72 |
| 9.10.4.4  | it                                     | 72 |
| 9.10.4.5  | maxItem                                | 72 |
| 9.10.4.6  | position                               | 72 |
| 9.10.4.7  | volume                                 | 72 |
| 9.10.4.8  | vtot                                   | 72 |
| 9.11      | item Class Reference                   | 74 |
| 9.11.1    | Detailed Description                   | 74 |
| 9.11.2    | Constructor & Destructor Documentation | 74 |

|           |  |    |
|-----------|--|----|
| 9.11.2.1  | item                                   | 74 |
| 9.11.2.2  | ~item                                  | 74 |
| 9.11.3    | Member Function Documentation          | 74 |
| 9.11.3.1  | getCombi                               | 74 |
| 9.11.3.2  | getCombiResult                         | 75 |
| 9.11.3.3  | getId                                  | 75 |
| 9.11.3.4  | getTexture                             | 75 |
| 9.11.3.5  | init                                   | 75 |
| 9.11.3.6  | repr                                   | 75 |
| 9.11.4    | Member Data Documentation              | 75 |
| 9.11.4.1  | combi                                  | 75 |
| 9.11.4.2  | combiResult                            | 75 |
| 9.11.4.3  | id                                     | 76 |
| 9.11.4.4  | texture                                | 76 |
| 9.12      | lieu Class Reference                   | 77 |
| 9.12.1    | Detailed Description                   | 77 |
| 9.12.2    | Constructor & Destructor Documentation | 77 |
| 9.12.2.1  | lieu                                   | 77 |
| 9.12.2.2  | ~lieu                                  | 78 |
| 9.12.3    | Member Function Documentation          | 78 |
| 9.12.3.1  | addAnimateMesh                         | 78 |
| 9.12.3.2  | addObservationMesh                     | 78 |
| 9.12.3.3  | addPrendreMesh                         | 78 |
| 9.12.3.4  | addStaticMesh                          | 78 |
| 9.12.3.5  | close                                  | 78 |
| 9.12.3.6  | getActionMesh                          | 79 |
| 9.12.3.7  | getActionMesh                          | 79 |
| 9.12.3.8  | getStaticMeshFromId                    | 79 |
| 9.12.3.9  | init                                   | 79 |
| 9.12.3.10 | OnEvent                                | 79 |
| 9.12.3.11 | update                                 | 79 |
| 9.12.4    | Member Data Documentation              | 80 |
| 9.12.4.1  | listAction                             | 80 |
| 9.12.4.2  | listStatic                             | 80 |
| 9.13      | local Class Reference                  | 81 |
| 9.13.1    | Detailed Description                   | 81 |

|           |  |    |
|-----------|--|----|
| 9.13.2    | Constructor & Destructor Documentation | 82 |
| 9.13.2.1  | local                                  | 82 |
| 9.13.2.2  | ~local                                 | 82 |
| 9.13.3    | Member Function Documentation          | 82 |
| 9.13.3.1  | addCombinaison                         | 82 |
| 9.13.3.2  | addObservationItem                     | 82 |
| 9.13.3.3  | addObservationMesh                     | 82 |
| 9.13.3.4  | close                                  | 82 |
| 9.13.3.5  | getCaption                             | 83 |
| 9.13.3.6  | getCombinaison                         | 83 |
| 9.13.3.7  | getObservationItem                     | 83 |
| 9.13.3.8  | getObservationMesh                     | 83 |
| 9.13.3.9  | init                                   | 83 |
| 9.13.3.10 | OnEvent                                | 83 |
| 9.13.3.11 | update                                 | 83 |
| 9.13.4    | Member Data Documentation              | 84 |
| 9.13.4.1  | caption                                | 84 |
| 9.13.4.2  | combinaison                            | 84 |
| 9.13.4.3  | observationItem                        | 84 |
| 9.13.4.4  | observationMesh                        | 84 |
| 9.14      | localItem Class Reference              | 85 |
| 9.14.1    | Detailed Description                   | 85 |
| 9.14.2    | Constructor & Destructor Documentation | 85 |
| 9.14.2.1  | localItem                              | 85 |
| 9.14.2.2  | ~localItem                             | 85 |
| 9.14.3    | Member Function Documentation          | 85 |
| 9.14.3.1  | getId                                  | 85 |
| 9.14.3.2  | getPath                                | 85 |
| 9.14.3.3  | getText                                | 86 |
| 9.14.3.4  | init                                   | 86 |
| 9.14.4    | Member Data Documentation              | 86 |
| 9.14.4.1  | id                                     | 86 |
| 9.14.4.2  | path                                   | 86 |
| 9.14.4.3  | text                                   | 86 |
| 9.15      | logger Class Reference                 | 87 |
| 9.15.1    | Detailed Description                   | 87 |

|          |  |    |
|----------|--|----|
| 9.15.2   | Constructor & Destructor Documentation | 87 |
| 9.15.2.1 | logger                                 | 87 |
| 9.15.2.2 | ~logger                                | 87 |
| 9.15.3   | Member Function Documentation          | 88 |
| 9.15.3.1 | close                                  | 88 |
| 9.15.3.2 | closeFile                              | 88 |
| 9.15.3.3 | createFile                             | 88 |
| 9.15.3.4 | init                                   | 88 |
| 9.15.3.5 | OnEvent                                | 88 |
| 9.15.3.6 | write                                  | 88 |
| 9.15.4   | Member Data Documentation              | 88 |
| 9.15.4.1 | files                                  | 88 |
| 9.16     | observationMesh Class Reference        | 90 |
| 9.16.1   | Detailed Description                   | 90 |
| 9.16.2   | Constructor & Destructor Documentation | 90 |
| 9.16.2.1 | observationMesh                        | 90 |
| 9.16.2.2 | ~observationMesh                       | 91 |
| 9.16.3   | Member Function Documentation          | 91 |
| 9.16.3.1 | init                                   | 91 |
| 9.17     | persoMesh Class Reference              | 92 |
| 9.17.1   | Detailed Description                   | 92 |
| 9.17.2   | Constructor & Destructor Documentation | 93 |
| 9.17.2.1 | persoMesh                              | 93 |
| 9.17.2.2 | ~persoMesh                             | 93 |
| 9.17.3   | Member Function Documentation          | 93 |
| 9.17.3.1 | getAnim                                | 93 |
| 9.17.3.2 | getBegin                               | 93 |
| 9.17.3.3 | getEnd                                 | 93 |
| 9.17.3.4 | init                                   | 93 |
| 9.17.4   | Member Data Documentation              | 93 |
| 9.17.4.1 | anim                                   | 93 |
| 9.17.4.2 | begin                                  | 93 |
| 9.17.4.3 | end                                    | 93 |
| 9.18     | prendreMesh Class Reference            | 94 |
| 9.18.1   | Detailed Description                   | 94 |
| 9.18.2   | Constructor & Destructor Documentation | 95 |

|          |  |     |
|----------|--|-----|
| 9.18.2.1 | <a href="#">prendreMesh</a>                                | 95  |
| 9.18.2.2 | <a href="#">~prendreMesh</a>                               | 95  |
| 9.18.3   | <a href="#">Member Function Documentation</a>              | 95  |
| 9.18.3.1 | <a href="#">getItem</a>                                    | 95  |
| 9.18.3.2 | <a href="#">init</a>                                       | 95  |
| 9.18.4   | <a href="#">Member Data Documentation</a>                  | 96  |
| 9.18.4.1 | <a href="#">it</a>   | 96  |
| 9.19     | <a href="#">pyInt Class Reference</a>                      | 97  |
| 9.19.1   | <a href="#">Detailed Description</a>                       | 98  |
| 9.19.2   | <a href="#">Constructor &amp; Destructor Documentation</a> | 98  |
| 9.19.2.1 | <a href="#">pyInt</a>                                      | 98  |
| 9.19.2.2 | <a href="#">~pyInt</a>                                     | 98  |
| 9.19.3   | <a href="#">Member Function Documentation</a>              | 98  |
| 9.19.3.1 | <a href="#">addPyEnv</a>                                   | 98  |
| 9.19.3.2 | <a href="#">close</a>                                      | 98  |
| 9.19.3.3 | <a href="#">eval</a>                                       | 99  |
| 9.19.3.4 | <a href="#">exec</a>                                       | 99  |
| 9.19.3.5 | <a href="#">init</a>                                       | 99  |
| 9.19.3.6 | <a href="#">OnEvent</a>                                    | 99  |
| 9.19.3.7 | <a href="#">update</a>                                     | 100 |
| 9.19.4   | <a href="#">Member Data Documentation</a>                  | 100 |
| 9.19.4.1 | <a href="#">dict</a>                                       | 100 |
| 9.19.4.2 | <a href="#">main_module</a>                                | 100 |
| 9.20     | <a href="#">startmenu Class Reference</a>                  | 101 |
| 9.20.1   | <a href="#">Detailed Description</a>                       | 102 |
| 9.20.2   | <a href="#">Constructor &amp; Destructor Documentation</a> | 102 |
| 9.20.2.1 | <a href="#">startmenu</a>                                  | 102 |
| 9.20.2.2 | <a href="#">~startmenu</a>                                 | 102 |
| 9.20.3   | <a href="#">Member Function Documentation</a>              | 102 |
| 9.20.3.1 | <a href="#">close</a>                                      | 102 |
| 9.20.3.2 | <a href="#">getCurPos</a>                                  | 103 |
| 9.20.3.3 | <a href="#">getFileSystem</a>                              | 103 |
| 9.20.3.4 | <a href="#">getTime</a>                                    | 103 |
| 9.20.3.5 | <a href="#">getVersion</a>                                 | 103 |
| 9.20.3.6 | <a href="#">init</a>                                       | 103 |
| 9.20.3.7 | <a href="#">loadDesc</a>                                   | 103 |

|           |                           |     |
|-----------|---------------------------|-----|
| 9.20.3.8  | log                       | 103 |
| 9.20.3.9  | OnEvent                   | 103 |
| 9.20.3.10 | progress                  | 104 |
| 9.20.3.11 | saveConfig                | 104 |
| 9.20.3.12 | setCaption                | 104 |
| 9.20.3.13 | startCoeur                | 104 |
| 9.20.3.14 | update                    | 104 |
| 9.20.4    | Member Data Documentation | 104 |
| 9.20.4.1  | bits                      | 104 |
| 9.20.4.2  | c                         | 104 |
| 9.20.4.3  | curPos                    | 105 |
| 9.20.4.4  | device                    | 105 |
| 9.20.4.5  | dim                       | 105 |
| 9.20.4.6  | dr                        | 105 |
| 9.20.4.7  | driver                    | 105 |
| 9.20.4.8  | env                       | 105 |
| 9.20.4.9  | fullscreen                | 105 |
| 9.20.4.10 | game                      | 105 |
| 9.20.4.11 | game_img                  | 106 |
| 9.20.4.12 | game_name                 | 106 |
| 9.20.4.13 | game_path                 | 106 |
| 9.20.4.14 | game_text                 | 106 |
| 9.20.4.15 | lang                      | 106 |
| 9.20.4.16 | langMenu                  | 106 |
| 9.20.4.17 | menus                     | 106 |
| 9.20.4.18 | menusIn                   | 106 |
| 9.20.4.19 | menusOut                  | 106 |
| 9.20.4.20 | options                   | 107 |
| 9.20.4.21 | options_dv                | 107 |
| 9.20.4.22 | options_fs                | 107 |
| 9.20.4.23 | options_lang              | 107 |
| 9.20.4.24 | options_res               | 107 |
| 9.20.4.25 | path                      | 107 |
| 9.20.4.26 | pathConfig                | 107 |
| 9.20.4.27 | pathMain                  | 107 |
| 9.20.4.28 | pathStart                 | 107 |

|   |            |
|---|------------|
| 9.20.4.29 progressing                         | 108        |
| 9.20.4.30 q                                   | 108        |
| 9.20.4.31 smgr                                | 108        |
| 9.20.4.32 start                               | 108        |
| 9.20.4.33 starting                            | 108        |
| 9.20.4.34 starting_bar                        | 108        |
| 9.20.4.35 starting_text                       | 108        |
| 9.20.4.36 starting_txt                        | 108        |
| 9.20.4.37 stencilbuffer                       | 108        |
| 9.20.4.38 vsync                               | 109        |
| 9.20.4.39 welcome                             | 109        |
| 9.21 staticMesh Class Reference               | 110        |
| 9.21.1 Detailed Description                   | 110        |
| 9.21.2 Constructor & Destructor Documentation | 110        |
| 9.21.2.1 staticMesh                           | 110        |
| 9.21.2.2 ~staticMesh                          | 110        |
| 9.21.3 Member Function Documentation          | 111        |
| 9.21.3.1 close                                | 111        |
| 9.21.3.2 getId                                | 111        |
| 9.21.3.3 getNode                              | 111        |
| 9.21.3.4 init                                 | 111        |
| 9.21.3.5 setVisible                           | 111        |
| 9.21.3.6 update                               | 111        |
| 9.21.4 Member Data Documentation              | 111        |
| 9.21.4.1 id                                   | 111        |
| 9.21.4.2 node                                 | 112        |
| <b>10 File Documentation</b>                  | <b>113</b> |
| 10.1 doc/example/chap.xml File Reference      | 113        |
| 10.1.1 Detailed Description                   | 113        |
| 10.1.2 Description                            | 113        |
| 10.1.3 Xml                                    | 113        |
| 10.1.4 Example                                | 114        |
| 10.2 include/actionMesh.h File Reference      | 115        |
| 10.2.1 Detailed Description                   | 115        |
| 10.2.2 Enumeration Type Documentation         | 115        |
| 10.2.2.1 ACTION_TYPE                          | 115        |



|           |  |     |
|-----------|--|-----|
| 10.3      | include/base.h File Reference . . . . .        | 117 |
| 10.3.1    | Detailed Description . . . . .                 | 117 |
| 10.4      | include/chap.h File Reference . . . . .        | 118 |
| 10.4.1    | Detailed Description . . . . .                 | 118 |
| 10.5      | include/coeur.h File Reference . . . . .       | 119 |
| 10.5.1    | Detailed Description . . . . .                 | 119 |
| 10.6      | include/console.h File Reference . . . . .     | 120 |
| 10.6.1    | Detailed Description . . . . .                 | 120 |
| 10.7      | include/credit.h File Reference . . . . .      | 121 |
| 10.8      | include/game.h File Reference . . . . .        | 122 |
| 10.8.1    | Detailed Description . . . . .                 | 122 |
| 10.9      | include/interface.h File Reference . . . . .   | 123 |
| 10.10     | include/inventaire.h File Reference . . . . .  | 124 |
| 10.10.1   | Enumeration Type Documentation . . . . .       | 124 |
| 10.10.1.1 | INVENTAIRE_POS . . . . .                       | 124 |
| 10.11     | include/item.h File Reference . . . . .        | 125 |
| 10.12     | include/lieu.h File Reference . . . . .        | 126 |
| 10.13     | include/local.h File Reference . . . . .       | 127 |
| 10.14     | include/logger.h File Reference . . . . .      | 128 |
| 10.15     | include/pyInt.h File Reference . . . . .       | 129 |
| 10.15.1   | Detailed Description . . . . .                 | 129 |
| 10.16     | include/startmenu.h File Reference . . . . .   | 130 |
| 10.17     | include/staticMesh.h File Reference . . . . .  | 131 |
| 10.18     | source/actionMesh.cpp File Reference . . . . . | 132 |
| 10.18.1   | Detailed Description . . . . .                 | 132 |
| 10.19     | source/base.cpp File Reference . . . . .       | 133 |
| 10.19.1   | Detailed Description . . . . .                 | 133 |
| 10.20     | source/chap.cpp File Reference . . . . .       | 134 |
| 10.20.1   | Detailed Description . . . . .                 | 134 |
| 10.21     | source/coeur.cpp File Reference . . . . .      | 135 |
| 10.21.1   | Detailed Description . . . . .                 | 135 |
| 10.22     | source/console.cpp File Reference . . . . .    | 136 |
| 10.23     | source/credit.cpp File Reference . . . . .     | 137 |
| 10.24     | source/interface.cpp File Reference . . . . .  | 138 |
| 10.25     | source/inventaire.cpp File Reference . . . . . | 139 |
| 10.26     | source/item.cpp File Reference . . . . .       | 140 |

|   |     |
|---|-----|
| 10.27source/lieu.cpp File Reference . . . . .       | 141 |
| 10.28source/local.cpp File Reference . . . . .      | 142 |
| 10.29source/logger.cpp File Reference . . . . .     | 143 |
| 10.30source/main.cpp File Reference . . . . .       | 144 |
| 10.30.1 Function Documentation . . . . .            | 144 |
| 10.30.1.1 main . . . . .                            | 144 |
| 10.31source/modPyIrr.cpp File Reference . . . . .   | 145 |
| 10.31.1 Detailed Description . . . . .              | 145 |
| 10.31.2 Function Documentation . . . . .            | 145 |
| 10.31.2.1 BOOST_PYTHON_MODULE . . . . .             | 145 |
| 10.31.2.2 reprDim . . . . .                         | 145 |
| 10.31.2.3 reprPos . . . . .                         | 145 |
| 10.32source/modPyMerry.cpp File Reference . . . . . | 146 |
| 10.32.1 Detailed Description . . . . .              | 146 |
| 10.32.2 Function Documentation . . . . .            | 146 |
| 10.32.2.1 BOOST_PYTHON_MODULE . . . . .             | 146 |
| 10.33source/pyInt.cpp File Reference . . . . .      | 147 |
| 10.33.1 Detailed Description . . . . .              | 147 |
| 10.33.2 Function Documentation . . . . .            | 147 |
| 10.33.2.1 initpyIrr . . . . .                       | 147 |
| 10.33.2.2 initpyMerry . . . . .                     | 147 |
| 10.34source/startmenu.cpp File Reference . . . . .  | 148 |
| 10.35source/staticMesh.cpp File Reference . . . . . | 149 |

# Chapter 1

## The Merry Turnip



### 1.1 Introduction

Welcome to the The Merry Turnip code documentation. Here you'll find any information about the merry turnip code.

First of all, it is coded in c++. I used only extreme programming when I coded it.

#### Todo

Description

Compatible operating system :

- Linux (Debian sure, others should too)
- Windows (I have tested XP and Vista)
- Mac is not compatible yet

Library that I use :

- STL
- irrlicht 1.5 : 3d motor
- irrklang 1.1.3 : sound motor

Software i use to develop :

- Debian Unstable(with experimental)
- Gedit 2.22.3
- GCC 4.3.3
- GNU make 3.81

- Gnome-terminal 2.24.1.1
- Blender 2.46
- The Gimp 2.4.7
- Doxygen 1.5.6

**See also:**

<http://merryturnip.tuxfamily.org/>  
<http://www.ambiera.com/irrklang/>  
<http://irrlicht.sourceforge.net/>  
<http://www.irrlicht.fr/>

## **Chapter 2**

### **Todo List**

**Class** [chap](#) add a name to chapter and maybe other stuff like description

**Member** [chap::init](#)(int, [coeur](#) \*, [IXMLReader](#) \*) replace the log by a warning function

**File** [coeur.cpp](#) Remplacer les vector path par des listes indexé de la STL

Creer repertoire log

Use the event system to launch personal event(like changing [chap](#), animation, script,...)

Add in the xml a line to say the path of the texture folder and ad it to irrlicht filesytem

**File** [modPyMerry.cpp](#) Ajouter des repr à tous les objets pour une intégration plus belle avec python cf  
class\_<item>

**Class** [persoMesh](#) Implement and make working this class

**page** [The Merry Turnip](#) Description

## **Chapter 3**

### **Bug List**

**File** [coeur.cpp](#) Not respond when launched fullscreen



# Chapter 4

## Module Index

### 4.1 Modules

Here is a list of all modules:

|                       |    |
|-----------------------|----|
| XML Scripts . . . . . | 15 |
|-----------------------|----|



# Chapter 5

## Class Index

### 5.1 Class Hierarchy

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

|   |     |
|---|-----|
| base . . . . .                          | 25  |
| chap . . . . .                          | 30  |
| coeur . . . . .                         | 34  |
| console . . . . .                       | 50  |
| credit . . . . .                        | 56  |
| interface . . . . .                     | 62  |
| inventaire . . . . .                    | 69  |
| lieu . . . . .                          | 77  |
| local . . . . .                         | 81  |
| logger . . . . .                        | 87  |
| pyInt . . . . .                         | 97  |
| cusString_python< cusString > . . . . . | 60  |
| item . . . . .                          | 74  |
| localItem . . . . .                     | 85  |
| startmenu . . . . .                     | 101 |
| staticMesh . . . . .                    | 110 |
| actionMesh . . . . .                    | 17  |
| observationMesh . . . . .               | 90  |
| animateMesh . . . . .                   | 21  |
| persoMesh . . . . .                     | 92  |
| prendreMesh . . . . .                   | 94  |



# Chapter 6

## Class Index

### 6.1 Class List

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

|   |     |
|---|-----|
| <a href="#">actionMesh</a> (Class containing a mesh who can launched an action when it is clicked, derive from <a href="#">staticMesh</a> ) . . . . . | 17  |
| <a href="#">animateMesh</a> (Class containing a mesh who can launched an animation, derive from <a href="#">observationMesh</a> ) . . . . .           | 21  |
| <a href="#">base</a> (Class from whom most of the other class derive from, derive from <a href="#">IEventReceiver</a> ) . . . .                       | 25  |
| <a href="#">chap</a> (Class containing all the information about a chapter, derive from <a href="#">base</a> ) . . . . .                              | 30  |
| <a href="#">coeur</a> (Main class containing all important call to the irrlicht and a lot of utilities, derive from <a href="#">base</a> )            | 34  |
| <a href="#">console</a> (This class handle all the suff of the <a href="#">console</a> ) . . . . .  | 50  |
| <a href="#">credit</a> . . . . .  | 56  |
| <a href="#">cusString_python&lt; cusString &gt;</a> . . . . .   | 60  |
| <a href="#">interface</a> . . . . .   | 62  |
| <a href="#">inventaire</a> . . . . .  | 69  |
| <a href="#">item</a> . . . . .  | 74  |
| <a href="#">lieu</a> . . . . .  | 77  |
| <a href="#">local</a> . . . . .   | 81  |
| <a href="#">localItem</a> . . . . .   | 85  |
| <a href="#">logger</a> . . . . .  | 87  |
| <a href="#">observationMesh</a> (Class containing a mesh who can be looked at, derive from <a href="#">actionMesh</a> ) . . .                         | 90  |
| <a href="#">persoMesh</a> (Class containing a mesh representing a person, what mean you can speak to it) . . .  | 92  |
| <a href="#">prendreMesh</a> (Class containing a mesh who can be taken, derive from <a href="#">observationMesh</a> ) . . . .                          | 94  |
| <a href="#">pyInt</a> (Class containing all the stuff to use python) . . . . .  | 97  |
| <a href="#">startmenu</a> . . . . .   | 101 |
| <a href="#">staticMesh</a> . . . . .  | 110 |



# Chapter 7

## File Index

### 7.1 File List

Here is a list of all files with brief descriptions:

|   |     |
|---|-----|
| doc/example/chap.xml . . . . .  | 113 |
| include/actionMesh.h (Definition of <a href="#">actionMesh</a> class and its derivate class ) . . . . . | 115 |
| include/base.h (Definition of <a href="#">base</a> class ) . . . . .                                    | 117 |
| include/chap.h (Definition of class <a href="#">chap</a> ) . . . . .                                    | 118 |
| include/coeur.h (Definition of class <a href="#">chap</a> ) . . . . .                                   | 119 |
| include/console.h (Definition of class <a href="#">console</a> ) . . . . .                              | 120 |
| include/credit.h . . . . .  | 121 |
| include/game.h (Main header file ) . . . . .  | 122 |
| include/interface.h . . . . .   | 123 |
| include/inventaire.h . . . . .  | 124 |
| include/item.h . . . . .  | 125 |
| include/lieu.h . . . . .  | 126 |
| include/local.h . . . . .   | 127 |
| include/logger.h . . . . .  | 128 |
| include/pyInt.h (Definition of class <a href="#">pyInt</a> ) . . . . .                                  | 129 |
| include/startmenu.h . . . . .   | 130 |
| include/staticMesh.h . . . . .  | 131 |
| source/actionMesh.cpp (Implement <a href="#">actionMesh</a> class and its derivate class ) . . . . .    | 132 |
| source/base.cpp (Implement <a href="#">base</a> class ) . . . . .                                       | 133 |
| source/chap.cpp (Implement <a href="#">chap</a> class ) . . . . .                                       | 134 |
| source/coeur.cpp (Implement <a href="#">coeur</a> class ) . . . . .                                     | 135 |
| source/console.cpp . . . . .  | 136 |
| source/credit.cpp . . . . .   | 137 |
| source/interface.cpp . . . . .  | 138 |
| source/inventaire.cpp . . . . .   | 139 |
| source/item.cpp . . . . .   | 140 |
| source/lieu.cpp . . . . .   | 141 |
| source/local.cpp . . . . .  | 142 |
| source/logger.cpp . . . . .   | 143 |
| source/main.cpp . . . . .   | 144 |
| source/modPyIrr.cpp (Create the python module who wrap the irrlicht needed class ) . . . . .            | 145 |
| source/modPyMerry.cpp (Create the python module who wrap the game ) . . . . .                           | 146 |
| source/pyInt.cpp (Implement <a href="#">pyInt</a> class ) . . . . .                                     | 147 |

|  |     |
|--|-----|
| source/ <a href="#">startmenu.cpp</a> . . . . .  | 148 |
| source/ <a href="#">staticMesh.cpp</a> . . . . . | 149 |



# Chapter 8

## Module Documentation

### 8.1 XML Scripts

#### Files

- file [chap.xml](#)

#### 8.1.1 Detailed Description

All the game can be configure by xml, this module contain a file by xml you can create to configure the game.

Note that all path in xml are relative to the main.xml.

main.xml, interface.xml, credit.xml must be unical whereas language.xml, [chap.xml](#) and lieu.xml can be numerous.



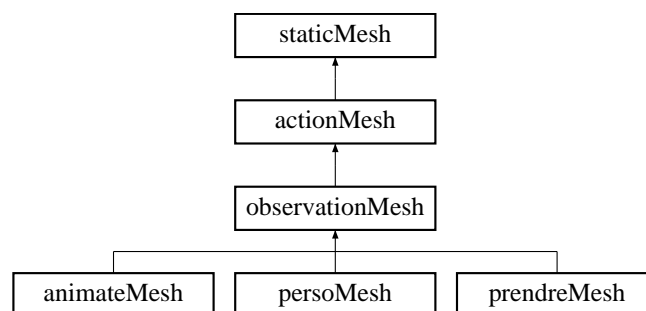
## Chapter 9

# Class Documentation

### 9.1 actionMesh Class Reference

Class containing a mesh who can launched an action when it is clicked, derive from [staticMesh](#).

`#include <actionMesh.h>`Inheritance diagram for `actionMesh::`



#### Public Member Functions

- `actionMesh (ACTION_TYPE=ACTION_AUCUNE)`  
*Constructor.*
- `~actionMesh ()`  
*Destructor.*
- `virtual int init (int, IAnimatedMeshSceneNode *, int)`  
*Init the mesh.*
- `ACTION_TYPE getAction ()`  
*Return the action type.*
- `int getCombi ()`  
*Return `actionMesh::combi`.*
- `SColor getColor ()`

*Return Color.*

## Protected Attributes

- SColor [color](#)

*Color of the ray who will point to the mesh.*

## Private Attributes

- ACTION\_TYPE [action](#)

*Contain the action type.*

- int [combi](#)

*Contain the id of the [item](#) needed to combine.*

### 9.1.1 Detailed Description

Class containing a mesh who can launched an action when it is clicked, derive from [staticMesh](#). This class is most of the time use by child class and not directly. If `combi=0`, the action will be launched just by clicking instead it will wait to be combined with [item](#) of id `actionMesh::combi`. (A bit different for observation Mesh)

Definition at line 50 of file `actionMesh.h`.

### 9.1.2 Constructor & Destructor Documentation

#### 9.1.2.1 `actionMesh::actionMesh (ACTION_TYPE Action = ACTION_AUCUNE)`

Constructor. Construct an [actionMesh](#) and set `actionMesh::action` to Action

##### Parameters:

*Action* The action type

Definition at line 37 of file `actionMesh.cpp`.

References [color](#).

#### 9.1.2.2 `actionMesh::~~actionMesh ()`

Destructor.

Definition at line 42 of file `actionMesh.cpp`.

## 9.1.3 Member Function Documentation

### 9.1.3.1 ACTION\_TYPE actionMesh::getAction ()

Return the action type.

Definition at line 61 of file actionMesh.cpp.

References action.

Referenced by BOOST\_PYTHON\_MODULE(), interface::OnEvent(), and interface::update().

### 9.1.3.2 SColor actionMesh::getColor ()

Return Color.

Definition at line 71 of file actionMesh.cpp.

References color.

Referenced by interface::update().

### 9.1.3.3 int actionMesh::getCombi ()

Return [actionMesh::combi](#).

Definition at line 66 of file actionMesh.cpp.

References combi.

Referenced by BOOST\_PYTHON\_MODULE(), and interface::OnEvent().

### 9.1.3.4 int actionMesh::init (int *Id*, IAnimatedMeshSceneNode \* *Node*, int *Combi*) [virtual]

Init the mesh. Call the [staticMesh::init](#), and set [actionMesh::combi](#) to comb

#### Parameters:

*Id* Id of mesh

*Node* The pointer to the irrlicht mesh corresponding

*Combi* Item to combine with

#### Returns:

[staticMesh::init](#) result

Reimplemented in [observationMesh](#).

Definition at line 52 of file actionMesh.cpp.

References combi, staticMesh::getId(), and staticMesh::getNode().

## 9.1.4 Member Data Documentation

### 9.1.4.1 ACTION\_TYPE actionMesh::action [private]

Contain the action type.

Definition at line 72 of file `actionMesh.h`.

Referenced by `getAction()`.

#### 9.1.4.2 `SColor actionMesh::color` **[protected]**

Color of the ray who will point to the mesh.

Definition at line 77 of file `actionMesh.h`.

Referenced by `actionMesh()`, `animateMesh::animateMesh()`, `getColor()`, `observationMesh::observationMesh()`, and `prendreMesh::prendreMesh()`.

#### 9.1.4.3 `int actionMesh::combi` **[private]**

Contain the id of the [item](#) needed to combine.

Definition at line 74 of file `actionMesh.h`.

Referenced by `getCombi()`, and `init()`.

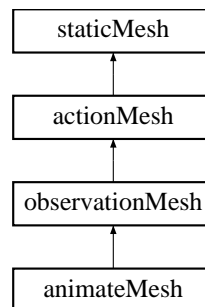
The documentation for this class was generated from the following files:

- `include/actionMesh.h`
- `source/actionMesh.cpp`

## 9.2 animateMesh Class Reference

Class containing a mesh who can launched an animation, derive from [observationMesh](#).

`#include <actionMesh.h>`Inheritance diagram for `animateMesh::`



### Public Member Functions

- [animateMesh](#) ([ACTION\\_TYPE](#)=ACTION\_ANIMATE)  
*Constructor.*
- [~animateMesh](#) ()  
*Destructor.*
- virtual int [init](#) (int, IAnimatedMeshSceneNode \*, int, int, int, int)  
*Init the mesh.*
- int [getAnim](#) ()  
*Return [animateMesh::anim](#).*
- int [getBegin](#) ()  
*Return [animateMesh::begin](#).*
- int [getEnd](#) ()  
*Return [animateMesh::end](#).*

### Private Attributes

- int [anim](#)  
*The mesh Id to anim.*
- int [begin](#)  
*The numeber of the frame where begin the animation.*
- int [end](#)  
*The numeber of the frame where end the animation.*

### 9.2.1 Detailed Description

Class containing a mesh who can launched an animation, derive from [observationMesh](#). When the mesh is clicked the animation of mesh of id [animateMesh::anim](#) is lauchned from [animateMesh::begin](#) to [animateMesh::end](#). And since it's derive from an [observationMesh](#) it can be looked at too.

Definition at line 123 of file `actionMesh.h`.

### 9.2.2 Constructor & Destructor Documentation

#### 9.2.2.1 `animateMesh::animateMesh (ACTION_TYPE act = ACTION_ANIMATE)`

Constructor. Construct a [prendreMesh](#) and call [observationMesh::observationMesh\(\)](#) with parameter Action

##### Parameters:

*act* The action type

Definition at line 155 of file `actionMesh.cpp`.

References `actionMesh::color`.

#### 9.2.2.2 `animateMesh::~~animateMesh ()`

Destructor.

Definition at line 160 of file `actionMesh.cpp`.

### 9.2.3 Member Function Documentation

#### 9.2.3.1 `int animateMesh::getAnim ()`

Return [animateMesh::anim](#).

Definition at line 181 of file `actionMesh.cpp`.

References `anim`.

Referenced by `BOOST_PYTHON_MODULE()`, and `interface::OnEvent()`.

#### 9.2.3.2 `int animateMesh::getBegin ()`

Return [animateMesh::begin](#).

Definition at line 186 of file `actionMesh.cpp`.

References `begin`.

Referenced by `BOOST_PYTHON_MODULE()`, and `interface::OnEvent()`.

#### 9.2.3.3 `int animateMesh::getEnd ()`

Return [animateMesh::end](#).

Definition at line 191 of file `actionMesh.cpp`.



References end.

Referenced by BOOST\_PYTHON\_MODULE(), and interface::OnEvent().

**9.2.3.4** `int animateMesh::init (int Id, IAnimatedMeshSceneNode * Node, int Combi, int Anim, int Begin, int End) [virtual]`

Init the mesh. Call the [actionMesh::init](#), set [prendreMesh::it](#) to ite

#### Parameters:

*Id* Id of mesh

*Node* The pointer to the irrlicht mesh corresponding

*Combi* Item to combine with

*Anim* Id of the mesh to animate

*Begin* Number of the Frame where begin the animation

*End* Number of the Frame where end the animation

#### Returns:

[observationMesh::init](#) result

Definition at line 173 of file actionMesh.cpp.

References [anim](#), [begin](#), and [end](#).

Referenced by [lieu::addAnimateMesh\(\)](#).

## 9.2.4 Member Data Documentation

**9.2.4.1** `int animateMesh::anim [private]`

The mesh Id to anim.

Definition at line 145 of file actionMesh.h.

Referenced by [getAnim\(\)](#), and [init\(\)](#).

**9.2.4.2** `int animateMesh::begin [private]`

The numeber of the frame where begin the animation.

Definition at line 147 of file actionMesh.h.

Referenced by [getBegin\(\)](#), and [init\(\)](#).

**9.2.4.3** `int animateMesh::end [private]`

The numeber of the frame where end the animation.

Definition at line 149 of file actionMesh.h.

Referenced by [getEnd\(\)](#), and [init\(\)](#).

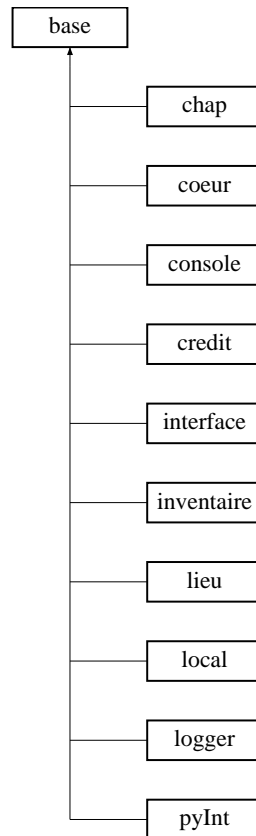
The documentation for this class was generated from the following files:

- [include/actionMesh.h](#)
- [source/actionMesh.cpp](#)

## 9.3 base Class Reference

Class from whom most of the other class derive from, derive from IEventReceiver.

#include <base.h>Inheritance diagram for base::



### Public Member Functions

- `base` (stringc=stringc("base"))  
*Constructor.*
- `~base` ()  
*Destructor.*
- virtual int `init` (int, `coeur` \*)  
*Init the `base`.*
- virtual int `update` ()  
*Updating(should be call on each loop).*
- virtual int `close` ()  
*Close the `base`.*
- int `getId` ()

Return the `base::id`.

- int `log` (stringc)  
Log some text.
- bool `isInit` ()  
return `base::initialize`

## Protected Attributes

- `coeur * mycore`  
Pointer to the core.

## Private Attributes

- int `id`  
Id of the class.
- stringc `nameClass`  
Name of the class.
- int `logId`  
Id of the log.
- bool `initialize`  
Contain True if `base` initialized.

### 9.3.1 Detailed Description

Class from whom most of the other class derive from, derive from `IEventReceiver`. Contain some basic feature needed by a lot of class.

This class handle more precisely the id, the name and the log function.

Derive from `IEventReceiver`, in order to be able to "surcharger" `OnEvent()` to handle events.

Definition at line 27 of file `base.h`.

### 9.3.2 Constructor & Destructor Documentation

#### 9.3.2.1 `base::base (stringc name = stringc ("base"))`

Constructor. Construct the `base` initiate `base::nameClass` to name and `base::initialize` to false

#### Parameters:

**name** Name of the class(needed for logging)

Definition at line 35 of file base.cpp.

References initialize, and nameClass.

#### 9.3.2.2 base::~~base ()

Destructor.

Definition at line 41 of file base.cpp.

### 9.3.3 Member Function Documentation

#### 9.3.3.1 int base::close () [virtual]

Close the [base](#). Close the log and put [base::initialize](#) to false

**Returns:**

[logger::closeFile\(\)](#)

Reimplemented in [chap](#), [coeur](#), [console](#), [credit](#), [interface](#), [inventaire](#), [lieu](#), [local](#), [logger](#), and [pyInt](#).

Definition at line 73 of file base.cpp.

References [logger::closeFile\(\)](#), [coeur::getLogger\(\)](#), [initialize](#), [log\(\)](#), [logId](#), [mycore](#), and [nameClass](#).

#### 9.3.3.2 int base::getId ()

Return the [base::id](#).

Definition at line 80 of file base.cpp.

References [id](#).

Referenced by [inventaire::addItem\(\)](#), [inventaire::getFreePlace\(\)](#), and [inventaire::removeItem\(\)](#).

#### 9.3.3.3 int base::init (int Id, coeur \* c) [virtual]

Init the [base](#). Put [base::mycore](#) to c, [base::id](#) to Id, init the [logger](#) and put initialized to true

**Parameters:**

*Id* Id of the class

*c* Pointer to the core

**Returns:**

The id of the log ([base::logId](#))

Definition at line 50 of file base.cpp.

References [logger::createFile\(\)](#), [coeur::getLogger\(\)](#), [initialize](#), [log\(\)](#), [logId](#), [mycore](#), and [nameClass](#).

#### 9.3.3.4 bool base::isInit ()

return [base::initialize](#)

Definition at line 93 of file base.cpp.

References [initialize](#).

Referenced by [coeur::loadChap\(\)](#), [coeur::loadLieu\(\)](#), and [coeur::loadLocal\(\)](#).

#### 9.3.3.5 int base::log (stringc *text*)

Log some text. Send text to the [logger](#) to log it in the proper file

##### Parameters:

*text* text to log(pure text, time and \n will be add letter)

##### Returns:

[logger::write\(\)](#)

Definition at line 88 of file base.cpp.

References [coeur::getLogger\(\)](#), [logId](#), [mycore](#), and [logger::write\(\)](#).

Referenced by [lieu::addAnimateMesh\(\)](#), [local::addCombinaison\(\)](#), [interface::addInventaire\(\)](#), [chap::addItem\(\)](#), [interface::addItemInventaire\(\)](#), [credit::addLogo\(\)](#), [coeur::addMesh\(\)](#), [local::addObservationItem\(\)](#), [local::addObservationMesh\(\)](#), [lieu::addObservationMesh\(\)](#), [lieu::addPrendreMesh\(\)](#), [lieu::addStaticMesh\(\)](#), [credit::addText\(\)](#), [local::close\(\)](#), [lieu::close\(\)](#), [interface::close\(\)](#), [credit::close\(\)](#), [coeur::close\(\)](#), [chap::close\(\)](#), [close\(\)](#), [logger::closeFile\(\)](#), [logger::createFile\(\)](#), [pyInt::eval\(\)](#), [pyInt::exec\(\)](#), [console::exit\(\)](#), [console::fps\(\)](#), [coeur::getTexture\(\)](#), [console::help\(\)](#), [local::init\(\)](#), [lieu::init\(\)](#), [interface::init\(\)](#), [credit::init\(\)](#), [console::init\(\)](#), [coeur::init\(\)](#), [chap::init\(\)](#), [init\(\)](#), [coeur::loadChap\(\)](#), [coeur::loadLieu\(\)](#), [coeur::loadLocal\(\)](#), [interface::OnEvent\(\)](#), [console::OnEvent\(\)](#), [coeur::play\(\)](#), [console::position\(\)](#), [console::quit\(\)](#), [console::rotation\(\)](#), [coeur::setCaption\(\)](#), [credit::start\(\)](#), [credit::stop\(\)](#), [console::target\(\)](#), and [coeur::update\(\)](#).

#### 9.3.3.6 int base::update () [virtual]

Updating(should be call on each loop). Does nothing

##### Returns:

Always 0

Reimplemented in [chap](#), [coeur](#), [console](#), [credit](#), [interface](#), [inventaire](#), [lieu](#), [local](#), and [pyInt](#).

Definition at line 66 of file base.cpp.

### 9.3.4 Member Data Documentation

#### 9.3.4.1 int base::id [private]

Id of the class.

Definition at line 60 of file base.h.

Referenced by [getId\(\)](#).

#### 9.3.4.2 `bool base::initialize` `[private]`

Contain True if [base](#) initialized.

Definition at line 69 of file `base.h`.

Referenced by `base()`, `close()`, `init()`, and `isInit()`.

#### 9.3.4.3 `int base::logId` `[private]`

Id of the log.

Definition at line 66 of file `base.h`.

Referenced by `close()`, `init()`, and `log()`.

#### 9.3.4.4 `coeur* base::mycore` `[protected]`

Pointer to the core.

Definition at line 56 of file `base.h`.

Referenced by `lieu::addAnimateMesh()`, `interface::addInventaire()`, `inventaire::addItem()`, `credit::addLogo()`, `lieu::addObservationMesh()`, `lieu::addPrendreMesh()`, `lieu::addStaticMesh()`, `credit::addText()`, `console::close()`, `close()`, `console::help()`, `lieu::init()`, `inventaire::init()`, `interface::init()`, `console::init()`, `init()`, `log()`, `interface::OnEvent()`, `console::OnEvent()`, `console::position()`, `console::quit()`, `console::rotation()`, `interface::setImage()`, `interface::speak()`, `credit::start()`, `console::target()`, `interface::update()`, and `logger::write()`.

#### 9.3.4.5 `stringc base::nameClass` `[private]`

Name of the class.

Definition at line 63 of file `base.h`.

Referenced by `base()`, `close()`, and `init()`.

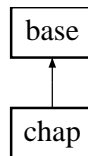
The documentation for this class was generated from the following files:

- [include/base.h](#)
- [source/base.cpp](#)

## 9.4 chap Class Reference

Class containing all the information about a chapter, derive from [base](#).

`#include <chap.h>`Inheritance diagram for chap::



### Public Member Functions

- [chap](#) ()  
*Constructor.*
- [~chap](#) ()  
*Destructor.*
- int [init](#) (int, [coeur](#) \*, IXMLReader \*)  
*Init the chapter.*
- int [update](#) ()  
*Updating(should be call on each loop).*
- int [close](#) ()  
*Close the [base](#).*
- [item](#) [addItem](#) (int [id](#), stringc tex, int it, int cr)  
*add an [Item](#) to the [item](#) list ([chap::listItem](#))*
- [item](#) [getItemFromId](#) (int [id](#))  
*Get an [item](#) from the list ([chap::listItem](#)) by its id.*
- bool [OnEvent](#) (const SEvent &)  
*Called when an event occured.*

### Private Attributes

- std::vector< [item](#) > \* [listItem](#)  
*Contain all the [item](#) needed by the chapter.*



### 9.4.1 Detailed Description

Class containing all the information about a chapter, derive from [base](#). This class contain mainly the [item](#) list ([chap::listItem](#)) of the chapter.

When you call the init ([chap::init\(\)](#)) function it will read an xml containing a list of item and store them in the class.

If you want to know more about chapter and how the xml must be formatted, see [chap.xml](#)

#### Todo

add a name to chapter and maybe other stuff like description

Definition at line 27 of file chap.h.

### 9.4.2 Constructor & Destructor Documentation

#### 9.4.2.1 chap::chap ()

Constructor. Construct the [chap](#) class, call [base::base\(\)](#) and create [chap::listItem](#).

Definition at line 77 of file chap.cpp.

References [listItem](#).

#### 9.4.2.2 chap::~~chap ()

Destructor. Destruct the [chap](#) class, delete [chap::listItem](#).

Definition at line 83 of file chap.cpp.

References [listItem](#).

### 9.4.3 Member Function Documentation

#### 9.4.3.1 item chap::addItem (int *Id*, stringc *tex*, int *Combi*, int *CombiResult*)

add an Item to the [item](#) list ([chap::listItem](#)) Create the [item](#), init it ([item::init\(\)](#)) and add it to the [chap::listItem](#)

##### Parameters:

*Id* Id of the [item](#)

*tex* Pointer to the irrlicht texture corresponding to the [item](#)

*Combi* Id of the [item](#) which this one can be combine

*CombiResult* Id of the [item](#) resulting from the combinaison

##### Returns:

Pointer to the [item](#) created if succesful, 0 instead

Definition at line 165 of file chap.cpp.

References [item::init\(\)](#), [listItem](#), and [base::log\(\)](#).

Referenced by [BOOST\\_PYTHON\\_MODULE\(\)](#), and [init\(\)](#).

#### 9.4.3.2 int chap::close () [virtual]

Close the [base](#). Remove all the [item](#) from [chap::listItem](#) and call [base::close\(\)](#)

##### Returns:

[base::close\(\)](#)

Reimplemented from [base](#).

Definition at line 148 of file chap.cpp.

References [listItem](#), and [base::log\(\)](#).

Referenced by [coeur::close\(\)](#), and [coeur::loadChap\(\)](#).

#### 9.4.3.3 item chap::getItemFromId (int Id)

Get an [item](#) from the list ([chap::listItem](#)) by its id. Check for all element of [chap::listItem](#) if the id correspond to Id

##### Parameters:

*Id* Id of the [item](#) to search

##### Returns:

The [item](#) if succesful, 0 instead

Definition at line 178 of file chap.cpp.

References [listItem](#).

Referenced by [BOOST\\_PYTHON\\_MODULE\(\)](#), and [coeur::getItem\(\)](#).

#### 9.4.3.4 int chap::init (int Id, coeur \* c, IXMLReader \* xmlChap)

Init the chapter. Call [base::init\(\)](#) and read xmlChap

##### Parameters:

*Id* Id of the [chap](#)

*c* Pointer to the core

*xmlChap* Pointer to the IXMLReader you want to load

##### Returns:

The first [lieu](#) of the Chapter if succesful, -1 instead

##### Todo

replace the log by a warning function

Definition at line 93 of file chap.cpp.

References [addItem\(\)](#), [listItem](#), and [base::log\(\)](#).

Referenced by [coeur::loadChap\(\)](#).

#### 9.4.3.5 bool chap::OnEvent (const SEvent & *event*)

Called when an event occurred. This method is called when an event has to be treated, but this class do not wait event so this function does nothing

##### Parameters:

*event* The event to parse

##### Returns:

Always false

Definition at line 190 of file chap.cpp.

#### 9.4.3.6 int chap::update () [virtual]

Updating(should be call on each loop). Does nothing, apart from slowing the game

##### Returns:

always 0

Reimplemented from [base](#).

Definition at line 141 of file chap.cpp.

### 9.4.4 Member Data Documentation

#### 9.4.4.1 std::vector<item>\* chap::listItem [private]

Contain all the [item](#) needed by the chapter.

Definition at line 56 of file chap.h.

Referenced by [addItem\(\)](#), [chap\(\)](#), [close\(\)](#), [getItemFromId\(\)](#), [init\(\)](#), and [~chap\(\)](#).

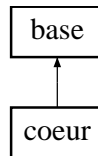
The documentation for this class was generated from the following files:

- [include/chap.h](#)
- [source/chap.cpp](#)

## 9.5 coeur Class Reference

Main class containing all important call to the irrlicht and a lot of utilities, derive from [base](#).

`#include <coeur.h>`Inheritance diagram for `coeur::`



### Public Member Functions

- [coeur](#) ()  
*Constructor.*
- [~coeur](#) ()  
*Destructor.*
- int [init](#) (startmenu \*)  
*Init the [coeur](#).*
- int [update](#) ()  
*Updating (must be call on each loop).*
- int [close](#) ()  
*Close the [coeur](#).*
- int [setCaption](#) (stringw)  
*Set the title of the window.*
- IAnimatedMeshSceneNode \* [addMesh](#) (stringc)  
*Add a mesh to irrlicht.*
- ILightSceneNode \* [addLight](#) (int)  
*Add a light to irrlicht.*
- ITexture \* [getTexture](#) (stringc)  
*Load a texture.*
- dimension2d< s32 > [getDim](#) ()  
*Return the dimension of the window.*
- [actionMesh](#) \* [getSelectedAction](#) ()  
*Return wich [actionMesh](#) the user is looking at.*
- [item](#) [getItem](#) (int)  
*Return an Item from his Id.*

- `actionMesh * getActionMesh (int)`  
*Return an ActionMesh from his id.*
- `actionMesh * getActionMesh (ISceneNode *)`  
*Return an ActionMesh from his node.*
- `localItem * getObservationMesh (int)`  
*Return the [localItem](#) of an [observationMesh](#) from his id.*
- `localItem * getObservationItem (int)`  
*Return the [localItem](#) of an [observationItem](#) from his id.*
- `localItem * getCombinaison ()`  
*Return the [localItem](#) corresponding to a failing combinaison.*
- `position2d< s32 > getCurPos ()`  
*Return the postion of the cursor.*
- `u32 getTime ()`  
*Return the current time.*
- `stringc getVersion ()`  
*Return the version of the soft.*
- `int play (stringc)`  
*Play a sound.*
- `int anim (int, int, int)`  
*Launch the animation of an [actionMesh](#).*
- `int setSkyBox (stringc, stringc, stringc, stringc, stringc, stringc)`  
*Set the skybox.*
- `virtual bool OnEvent (const SEvent &)`  
*Called when an event occured.*
- `logger * getLogger ()`  
*Return the pointer to the [logger](#).*
- `int startCredit ()`  
*Launch the Credit.*
- `int loadChap (int)`  
*Load a new chapter.*
- `int loadLieu (int)`  
*Load a new [lieu](#).*
- `int loadLocal (int)`  
*Load a new [local](#).*
- `IParticleSystemSceneNode * addParticleSystem ()`

## Public Attributes

- ISceneManager \* [smgr](#)  
*The Irrlicht SceneManager.*
- ICameraSceneNode \* [cam](#)  
*The Camera.*

## Private Member Functions

- ISceneNode \* [getSelectedNode](#) ()  
*Get Selected Node.*
- void [pickNode](#) (ISceneNode \*, line3df, f32 &, ISceneNode \*&)  
*Pick a node.*

## Private Attributes

- IrrlichtDevice \* [device](#)  
*The Irrlicht Device.*
- IVideoDriver \* [driver](#)  
*The Irrlicht Driver.*
- IGUEnvironment \* [env](#)  
*The Irrlicht Envirronement.*
- E\_DRIVER\_TYPE [dr](#)  
*The driver chosen.*
- dimension2d< s32 > [dim](#)  
*The dimension chosen.*
- int [bits](#)  
*The bits o the window.*
- bool [fullscreen](#)  
*Does the render take all the screen.*
- bool [stencilbuffer](#)  
*Does it draw the stencil shadow.*
- bool [vsync](#)  
*Does the certical synchronisation is activated.*
- int [lang](#)  
*Language of the game loaded.*

- [interface](#) \* [in](#)  
*Pointer to the [interface](#) class.*
- [console](#) \* [co](#)  
*Pointer to the [console](#) class.*
- [credit](#) \* [mycredit](#)  
*Pointer to the [credit](#) class.*
- [logger](#) \* [mylogger](#)
- [bool](#) [q](#)  
*Set to true if the [coeur](#) must be quit.*
- [position2d](#) < [s32](#) > [curPos](#)  
*Position of the cursor.*
- [std::vector](#) < [stringc](#) \* > \* [pathChap](#)  
*List containing the path of all the chapters.*
- [std::vector](#) < [stringc](#) \* > \* [pathLieu](#)  
*List containing the path of all the [lieu](#).*
- [std::vector](#) < [stringc](#) \* > \* [pathLocal](#)  
*List containing the path of all the [local](#).*
- [chap](#) \* [currentChap](#)  
*Pointer to the [chap](#) class.*
- [lieu](#) \* [currentLieu](#)  
*Pointer to the [lieu](#) class.*
- [local](#) \* [currentLocal](#)  
*Pointer to the [local](#) class.*
- [pyInt](#) \* [py](#)
- [actionMesh](#) \* [selectedAction](#)  
*Pointer to the [actionMesh](#) selected.*
- [IMetaTriangleSelector](#) \* [metaSelector](#)  
*Pointer to the [metaSelector](#).*
- [stringc](#) [pathGame](#)  
*Folder containg all the data of the game.*
- [bool](#) [activeScene](#)  
*True if the [coeur](#) must draw the scene.*
- [bool](#) [activeIn](#)  
*True if the [coeur](#) must draw the [interface](#).*

## Friends

- class [console](#)

*The [console](#) has all right on this class.*

- class [pyInt](#)

*The python [interface](#) has all right on this class.*

### 9.5.1 Detailed Description

Main class containing all important call to the irrlicht and a lot of utilities, derive from [base](#). This class contain all the reference to the irrlicht and irrklang engine It contains most of the stuff needed by other class. It's the only class who have to be create, all th others are handle by this one, and the xml config file. This is class is configured by two xml files : the config.xml for all basic stuff and the main.xml which is the main config filoe of the game

If you want to know more about the xml configuration see config.xml and main.xml

Definition at line 27 of file coeur.h.

### 9.5.2 Constructor & Destructor Documentation

#### 9.5.2.1 `coeur::coeur ()`

Constructor. Construct the [coeur](#) class, call [base::base\(\)](#) and create all the class and initialize some value.

Definition at line 105 of file coeur.cpp.

References [bits](#), [co](#), [currentChap](#), [currentLieu](#), [currentLocal](#), [dim](#), [dr](#), [fullscreen](#), [in](#), [lang](#), [mycredit](#), [mylogger](#), [pathChap](#), [pathLieu](#), [pathLocal](#), [py](#), [stencilbuffer](#), and [vsync](#).

#### 9.5.2.2 `coeur::~~coeur ()`

Destructor. Destruct the [coeur](#) class, delete all what is needed.

Definition at line 133 of file coeur.cpp.

References [co](#), [currentChap](#), [currentLieu](#), [currentLocal](#), [in](#), [mycredit](#), [mylogger](#), [pathChap](#), [pathLieu](#), [pathLocal](#), and [py](#).

### 9.5.3 Member Function Documentation

#### 9.5.3.1 `ILightSceneNode * coeur::addLight (int radius)`

Add a light to irrlicht.

Definition at line 473 of file coeur.cpp.

References [smgr](#).

Referenced by [lieu::init\(\)](#).



### 9.5.3.2 IAnimatedMeshSceneNode \* coeur::addMesh (stringc *pathm*)

Add a mesh to irrlicht.

Definition at line 449 of file coeur.cpp.

References base::log(), metaSelector, and smgr.

Referenced by lieu::addAnimateMesh(), lieu::addObservationMesh(), lieu::addPrendreMesh(), and lieu::addStaticMesh().

### 9.5.3.3 IParticleSystemSceneNode \* coeur::addParticleSystem ()

Definition at line 670 of file coeur.cpp.

References smgr.

Referenced by interface::init().

### 9.5.3.4 int coeur::anim (int *id*, int *begin*, int *end*)

Launch the animation of an [actionMesh](#).

Definition at line 643 of file coeur.cpp.

References getActionMesh(), staticMesh::getNode(), metaSelector, and smgr.

Referenced by init(), and interface::OnEvent().

### 9.5.3.5 int coeur::close () [virtual]

Close the [coeur](#).

Reimplemented from [base](#).

Definition at line 398 of file coeur.cpp.

References logger::close(), pyInt::close(), credit::close(), interface::close(), chap::close(), lieu::close(), local::close(), console::close(), co, currentChap, currentLieu, currentLocal, device, in, base::log(), mycredit, mylogger, pathChap, pathLieu, pathLocal, and py.

Referenced by startmenu::startCoeur().

### 9.5.3.6 actionMesh \* coeur::getActionMesh (ISceneNode \* *node*)

Return an ActionMesh from his node.

Definition at line 613 of file coeur.cpp.

References currentLieu, and lieu::getActionMesh().

### 9.5.3.7 actionMesh \* coeur::getActionMesh (int *id*)

Return an ActionMesh from his id. /todo Maybe returning a pointer is not a good idea

Definition at line 608 of file coeur.cpp.

References currentLieu, and lieu::getActionMesh().

Referenced by `anim()`, `BOOST_PYTHON_MODULE()`, and `update()`.

#### 9.5.3.8 `localItem * coeur::getCombinaison ()`

Return the `localItem` corresponding to a failing combinaison.

Definition at line 628 of file `coeur.cpp`.

References `currentLocal`, and `local::getCombinaison()`.

Referenced by `BOOST_PYTHON_MODULE()`, and `interface::OnEvent()`.

#### 9.5.3.9 `position2d< s32 > coeur::getCurPos ()`

Return the position of the cursor.

Definition at line 496 of file `coeur.cpp`.

References `curPos`.

Referenced by `interface::update()`.

#### 9.5.3.10 `dimension2d< s32 > coeur::getDim ()`

Return the dimension of the window.

Definition at line 486 of file `coeur.cpp`.

References `dim`.

Referenced by `interface::addInventaire()`, `credit::addLogo()`, `credit::addText()`, `BOOST_PYTHON_MODULE()`, `interface::init()`, `credit::init()`, and `interface::update()`.

#### 9.5.3.11 `item coeur::getItem (int id)`

Return an Item from his Id.

Definition at line 602 of file `coeur.cpp`.

References `currentChap`, and `chap::getItemFromId()`.

Referenced by `BOOST_PYTHON_MODULE()`, and `interface::OnEvent()`.

#### 9.5.3.12 `logger * coeur::getLogger ()`

Return the pointer to the `logger`.

Definition at line 633 of file `coeur.cpp`.

References `mylogger`.

Referenced by `base::close()`, `base::init()`, and `base::log()`.

#### 9.5.3.13 `localItem * coeur::getObservationItem (int id)`

Return the `localItem` of an observationItem from his id.

Definition at line 623 of file `coeur.cpp`.

References `currentLocal`, and `local::getObservationItem()`.

Referenced by `BOOST_PYTHON_MODULE()`, and `interface::OnEvent()`.

#### 9.5.3.14 `localItem * coeur::getObservationMesh (int id)`

Return the `localItem` of an `observationMesh` from his id.

Definition at line 618 of file `coeur.cpp`.

References `currentLocal`, and `local::getObservationMesh()`.

Referenced by `BOOST_PYTHON_MODULE()`, and `interface::OnEvent()`.

#### 9.5.3.15 `actionMesh * coeur::getSelectedAction ()`

Return wich `actionMesh` the user is looking at.

Definition at line 491 of file `coeur.cpp`.

References `selectedAction`.

Referenced by `BOOST_PYTHON_MODULE()`, `interface::OnEvent()`, and `interface::update()`.

#### 9.5.3.16 `ISceneNode * coeur::getSelectedNode () [private]`

Get Selected Node.

Definition at line 329 of file `coeur.cpp`.

References `curPos`, `pickNode()`, and `smgr`.

Referenced by `update()`.

#### 9.5.3.17 `ITexture * coeur::getTexture (stringc path)`

Load a texture.

Definition at line 480 of file `coeur.cpp`.

References `driver`, and `base::log()`.

Referenced by `inventaire::addItem()`, `credit::addLogo()`, `lieu::init()`, `inventaire::init()`, `interface::init()`, `interface::setImage()`, and `setSkyBox()`.

#### 9.5.3.18 `u32 coeur::getTime ()`

Return the current time.

Definition at line 501 of file `coeur.cpp`.

References `device`.

Referenced by `BOOST_PYTHON_MODULE()`, `interface::speak()`, and `logger::write()`.

#### 9.5.3.19 `stringc coeur::getVersion ()`

Return the version of the soft.

Definition at line 638 of file coeur.cpp.

Referenced by BOOST\_PYTHON\_MODULE(), and console::help().

#### 9.5.3.20 int coeur::init (startmenu \* cb)

Init the [coeur](#). Call [base::init\(\)](#) and read xmlChap

##### Parameters:

[startmenu](#) pointer to the [startmenu](#) to update the progress bar

##### Returns:

0 if all goes well

Definition at line 154 of file coeur.cpp.

References [activeIn](#), [activeScene](#), [anim\(\)](#), [bits](#), [cam](#), [co](#), [currentLocal](#), [device](#), [dim](#), [dr](#), [driver](#), [env](#), [fullscreen](#), [local::getCaption\(\)](#), [startmenu::getFileSystem\(\)](#), [in](#), [console::init\(\)](#), [pyInt::init\(\)](#), [credit::init\(\)](#), [interface::init\(\)](#), [logger::init\(\)](#), [lang](#), [loadChap\(\)](#), [loadLocal\(\)](#), [base::log\(\)](#), [metaSelector](#), [mycredit](#), [mylogger](#), [pathChap](#), [pathGame](#), [pathLieu](#), [pathLocal](#), [startmenu::progress\(\)](#), [py](#), [q](#), [selectedAction](#), [setCaption\(\)](#), [smgr](#), [stencilbuffer](#), and [vsync](#).

Referenced by [startmenu::startCoeur\(\)](#).

#### 9.5.3.21 int coeur::loadChap (int i)

Load a new chapter.

Definition at line 552 of file coeur.cpp.

References [chap::close\(\)](#), [currentChap](#), [device](#), [chap::init\(\)](#), [base::isInit\(\)](#), [loadLieu\(\)](#), [base::log\(\)](#), and [pathChap](#).

Referenced by BOOST\_PYTHON\_MODULE(), and [init\(\)](#).

#### 9.5.3.22 int coeur::loadLieu (int i)

Load a new [lieu](#).

Definition at line 567 of file coeur.cpp.

References [lieu::close\(\)](#), [currentLieu](#), [device](#), [lieu::init\(\)](#), [base::isInit\(\)](#), [base::log\(\)](#), and [pathLieu](#).

Referenced by [loadChap\(\)](#).

#### 9.5.3.23 int coeur::loadLocal (int i)

Load a new [local](#).

Definition at line 580 of file coeur.cpp.

References [local::close\(\)](#), [currentLocal](#), [device](#), [local::init\(\)](#), [base::isInit\(\)](#), [base::log\(\)](#), and [pathLocal](#).

Referenced by [init\(\)](#).

**9.5.3.24 bool coeur::OnEvent (const SEvent & *event*) [virtual]**

Called when an event occurred.

Definition at line 516 of file coeur.cpp.

References cam, co, in, mycredit, interface::OnEvent(), credit::OnEvent(), console::OnEvent(), q, and startCredit().

**9.5.3.25 void coeur::pickNode (ISceneNode \* *root*, line3df *ray*, f32 & *outbestdistance*, ISceneNode \*& *outbestnode*) [private]**

Pick a node.

Definition at line 341 of file coeur.cpp.

References smgr.

Referenced by getSelectedNode().

**9.5.3.26 int coeur::play (stringc *path*)**

Play a sound.

Definition at line 507 of file coeur.cpp.

References base::log(), and pathGame.

Referenced by BOOST\_PYTHON\_MODULE(), interface::OnEvent(), interface::speak(), and credit::start().

**9.5.3.27 int coeur::setCaption (stringw *caption*)**

Set the title of the window.

Definition at line 442 of file coeur.cpp.

References device, and base::log().

Referenced by BOOST\_PYTHON\_MODULE(), and init().

**9.5.3.28 int coeur::setSkyBox (stringc *top*, stringc *bottom*, stringc *left*, stringc *right*, stringc *front*, stringc *back*)**

Set the skybox.

Definition at line 663 of file coeur.cpp.

References getTexture(), and smgr.

Referenced by lieu::init().

**9.5.3.29 int coeur::startCredit ()**

Launch the Credit.

Definition at line 593 of file coeur.cpp.

References `activeIn`, `activeScene`, `cam`, `mycredit`, and `credit::start()`.

Referenced by `BOOST_PYTHON_MODULE()`, and `OnEvent()`.

#### 9.5.3.30 `int coeur::update ()` **[virtual]**

Updating (must be call on each loop). Desc

##### Returns:

Reimplemented from [base](#).

Definition at line 369 of file `coeur.cpp`.

References `activeIn`, `activeScene`, `co`, `curPos`, `device`, `driver`, `env`, `getActionMesh()`, `getSelectedNode()`, `in`, `base::log()`, `mycredit`, `q`, `selectedAction`, `smgr`, `credit::update()`, `console::update()`, and `interface::update()`.

Referenced by `startmenu::startCoeur()`.

### 9.5.4 Friends And Related Function Documentation

#### 9.5.4.1 `friend class console` **[friend]**

The [console](#) has all right on this class.

Definition at line 118 of file `coeur.h`.

#### 9.5.4.2 `friend class pyInt` **[friend]**

The python [interface](#) has all right on this class.

Definition at line 121 of file `coeur.h`.

### 9.5.5 Member Data Documentation

#### 9.5.5.1 `bool coeur::activeIn` **[private]**

True if the [coeur](#) must draw the [interface](#).

Definition at line 219 of file `coeur.h`.

Referenced by `init()`, `startCredit()`, and `update()`.

#### 9.5.5.2 `bool coeur::activeScene` **[private]**

True if the [coeur](#) must draw the scene.

Definition at line 216 of file `coeur.h`.

Referenced by `init()`, `startCredit()`, and `update()`.

### 9.5.5.3 int coeur::bits [private]

The bits o the window.

Definition at line 155 of file coeur.h.

Referenced by coeur(), and init().

### 9.5.5.4 ICameraSceneNode\* coeur::cam

The Camera.

Definition at line 146 of file coeur.h.

Referenced by init(), OnEvent(), console::position(), console::rotation(), startCredit(), console::target(), and interface::update().

### 9.5.5.5 console\* coeur::co [private]

Pointer to the [console](#) class.

Definition at line 173 of file coeur.h.

Referenced by close(), coeur(), init(), OnEvent(), update(), and ~coeur().

### 9.5.5.6 position2d<s32> coeur::curPos [private]

Position of the cursor.

Definition at line 184 of file coeur.h.

Referenced by getCurPos(), getSelectedNode(), and update().

### 9.5.5.7 chap\* coeur::currentChap [private]

Pointer to the [chap](#) class.

Definition at line 196 of file coeur.h.

Referenced by close(), coeur(), getItem(), console::init(), loadChap(), and ~coeur().

### 9.5.5.8 lieu\* coeur::currentLieu [private]

Pointer to the [lieu](#) class.

Definition at line 199 of file coeur.h.

Referenced by close(), coeur(), getActionMesh(), loadLieu(), and ~coeur().

### 9.5.5.9 local\* coeur::currentLocal [private]

Pointer to the [local](#) class.

Definition at line 202 of file coeur.h.

Referenced by close(), coeur(), getCombinaison(), getObservationItem(), getObservationMesh(), console::init(), init(), loadLocal(), and ~coeur().

**9.5.5.10 IrrlichtDevice\* coeur::device [private]**

The Irrlicht Device.

Definition at line 134 of file coeur.h.

Referenced by close(), getTime(), init(), loadChap(), loadLieu(), loadLocal(), setCaption(), and update().

**9.5.5.11 dimension2d<s32> coeur::dim [private]**

The dimension chosen.

Definition at line 152 of file coeur.h.

Referenced by coeur(), getDim(), and init().

**9.5.5.12 E\_DRIVER\_TYPE coeur::dr [private]**

The driver chosen.

Definition at line 149 of file coeur.h.

Referenced by coeur(), and init().

**9.5.5.13 IVideoDriver\* coeur::driver [private]**

The Irrlicht Driver.

Definition at line 137 of file coeur.h.

Referenced by getTexture(), init(), and update().

**9.5.5.14 IGUIEnvironment\* coeur::env [private]**

The Irrlicht Environnement.

Definition at line 143 of file coeur.h.

Referenced by init(), and update().

**9.5.5.15 bool coeur::fullscreen [private]**

Does the render take all the screen.

Definition at line 158 of file coeur.h.

Referenced by coeur(), and init().

**9.5.5.16 interface\* coeur::in [private]**

Pointer to the [interface](#) class.

Definition at line 170 of file coeur.h.

Referenced by close(), coeur(), console::init(), init(), OnEvent(), update(), and ~coeur().



**9.5.5.17 int coeur::lang [private]**

Language of the game loaded.

Definition at line 167 of file coeur.h.

Referenced by coeur(), and init().

**9.5.5.18 IMetaTriangleSelector\* coeur::metaSelector [private]**

Pointer to the metaSelector.

Definition at line 210 of file coeur.h.

Referenced by addMesh(), anim(), and init().

**9.5.5.19 credit\* coeur::mycredit [private]**

Pointer to the [credit](#) class.

Definition at line 176 of file coeur.h.

Referenced by close(), coeur(), console::init(), init(), OnEvent(), startCredit(), update(), and ~coeur().

**9.5.5.20 logger\* coeur::mylogger [private]**

Definition at line 178 of file coeur.h.

Referenced by close(), coeur(), getLogger(), init(), and ~coeur().

**9.5.5.21 std::vector<stringc\*>\* coeur::pathChap [private]**

List containing the path of all the chapters.

Definition at line 187 of file coeur.h.

Referenced by close(), coeur(), init(), loadChap(), and ~coeur().

**9.5.5.22 stringc coeur::pathGame [private]**

Folder containing all the data of the game.

Definition at line 213 of file coeur.h.

Referenced by init(), and play().

**9.5.5.23 std::vector<stringc\*>\* coeur::pathLieu [private]**

List containing the path of all the [lieu](#).

Definition at line 190 of file coeur.h.

Referenced by close(), coeur(), init(), loadLieu(), and ~coeur().

**9.5.5.24 std::vector<stringc\*> coeur::pathLocal [private]**

List containing the path of all the [local](#).

Definition at line 193 of file coeur.h.

Referenced by close(), coeur(), init(), loadLocal(), and ~coeur().

**9.5.5.25 pyInt\* coeur::py [private]**

Definition at line 204 of file coeur.h.

Referenced by console::close(), close(), coeur(), console::init(), init(), console::OnEvent(), and ~coeur().

**9.5.5.26 bool coeur::q [private]**

Set to true if the [coeur](#) must be quit.

Definition at line 181 of file coeur.h.

Referenced by init(), OnEvent(), console::quit(), and update().

**9.5.5.27 actionMesh\* coeur::selectedAction [private]**

Pointer to the [actionMesh](#) selected.

Definition at line 207 of file coeur.h.

Referenced by getSelectedAction(), init(), and update().

**9.5.5.28 ISceneManager\* coeur::smgr**

The Irrlicht SceneManager.

Definition at line 140 of file coeur.h.

Referenced by addLight(), addMesh(), addParticleSystem(), anim(), getSelectedNode(), init(), pickNode(), setSkyBox(), interface::update(), and update().

**9.5.5.29 bool coeur::stencilbuffer [private]**

Does it draw the stencil shadow.

Definition at line 161 of file coeur.h.

Referenced by coeur(), and init().

**9.5.5.30 bool coeur::vsync [private]**

Does the certical synchronisation is activated.

Definition at line 164 of file coeur.h.

Referenced by coeur(), and init().

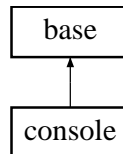
The documentation for this class was generated from the following files:

- [include/coeur.h](#)
- [source/coeur.cpp](#)

## 9.6 console Class Reference

This class handle all the suff of the [console](#).

`#include <console.h>`Inheritance diagram for console::



### Public Member Functions

- [console](#) ()  
*Constructor.*
- [~console](#) ()  
*Destructor.*
- int [init](#) (coeur \*, IGUIEnvironment \*)  
*Init the [console](#).*
- int [update](#) ()  
*Updating (must be call on each loop).*
- int [close](#) ()  
*Close the [console](#).*
- bool [OnEvent](#) (const SEvent &)  
*Call when an event occured.*
- void [write](#) (std::string const &)  
*Write some text on the [console](#).*
- void [position](#) ()  
*Display the position of the camera.*
- void [rotation](#) ()  
*Display the rotation of the camera.*
- void [target](#) ()  
*Display the position of the target of the camera.*
- void [exit](#) ()
- void [quit](#) ()  
*Quit the game.*
- void [help](#) ()

*Display an helping text.*

- void `fps` ()

*Display the actual fps.*

## Private Attributes

- IrrlichtDevice \* `device`

*The Irrlicht Device.*

- IGUEnvironment \* `env`

*The Irrlicht Envirronement.*

- stringw \* `text`

*Contain the text display by the `console`.*

- IGUIImage \* `cons`

*Just use to contain all the `item` of the `console`.*

- IGUEditBox \* `tx`

*Box where are displayed the text of the `console`.*

- IGUEditBox \* `box`

*Box where you type the command to launch to the `console`.*

- IGUStaticText \* `FPS`

*Box who displayed the fps.*

- std::vector< stringw > \* `history`

*Contain the history of the command enter.*

- unsigned int `histPos`

*Contain the position in the history.*

### 9.6.1 Detailed Description

This class handle all the suff of the `console`.

Definition at line 27 of file console.h.

### 9.6.2 Constructor & Destructor Documentation

#### 9.6.2.1 `console::console` ()

Constructor. /todo Hide the fps when hide the console(or not)

Definition at line 9 of file console.cpp.

References `history`, `histPos`, and `text`.

### 9.6.2.2 console::~~console ()

Destructor.

Definition at line 17 of file console.cpp.

## 9.6.3 Member Function Documentation

### 9.6.3.1 int console::close () [virtual]

Close the [console](#).

Reimplemented from [base](#).

Definition at line 84 of file console.cpp.

References [pyInt::exec\(\)](#), [base::mycore](#), and [coeur::py](#).

Referenced by [coeur::close\(\)](#).

### 9.6.3.2 void console::exit ()

Close the [console](#) /todo Change the name of this function because it exit as much as it lake visible

Definition at line 155 of file console.cpp.

References [cons](#), and [base::log\(\)](#).

Referenced by [BOOST\\_PYTHON\\_MODULE\(\)](#), and [OnEvent\(\)](#).

### 9.6.3.3 void console::fps ()

Display the actual fps.

Definition at line 194 of file console.cpp.

References [env](#), [base::log\(\)](#), and [write\(\)](#).

Referenced by [BOOST\\_PYTHON\\_MODULE\(\)](#).

### 9.6.3.4 void console::help ()

Display an helping text.

Definition at line 176 of file console.cpp.

References [coeur::getVersion\(\)](#), [base::log\(\)](#), [base::mycore](#), and [write\(\)](#).

Referenced by [BOOST\\_PYTHON\\_MODULE\(\)](#).

### 9.6.3.5 int console::init (coeur \* c, IGUIEnvironment \* e)

Init the [console](#).

Definition at line 22 of file console.cpp.

References [pyInt::addPyEnv\(\)](#), [box](#), [cons](#), [coeur::currentChap](#), [coeur::currentLocal](#), [env](#), [pyInt::exec\(\)](#), [FPS](#), [coeur::in](#), [base::log\(\)](#), [base::mycore](#), [coeur::mycredit](#), [coeur::py](#), and [tx](#).

Referenced by `coeur::init()`.

#### 9.6.3.6 `bool console::OnEvent (const SEvent & event)`

Call when an event occurred.

Definition at line 92 of file `console.cpp`.

References `box`, `cons`, `pyInt::eval()`, `exit()`, `history`, `histPos`, `base::log()`, `base::mycore`, `coeur::py`, and `write()`.

Referenced by `coeur::OnEvent()`.

#### 9.6.3.7 `void console::position ()`

Display the position of the camera.

Definition at line 137 of file `console.cpp`.

References `coeur::cam`, `base::log()`, `base::mycore`, and `write()`.

Referenced by `BOOST_PYTHON_MODULE()`.

#### 9.6.3.8 `void console::quit ()`

Quit the game.

Definition at line 169 of file `console.cpp`.

References `base::log()`, `base::mycore`, `coeur::q`, and `write()`.

Referenced by `BOOST_PYTHON_MODULE()`.

#### 9.6.3.9 `void console::rotation ()`

Display the rotation of the camera.

Definition at line 143 of file `console.cpp`.

References `coeur::cam`, `base::log()`, `base::mycore`, and `write()`.

Referenced by `BOOST_PYTHON_MODULE()`.

#### 9.6.3.10 `void console::target ()`

Display the position of the target of the camera.

Definition at line 149 of file `console.cpp`.

References `coeur::cam`, `base::log()`, `base::mycore`, and `write()`.

Referenced by `BOOST_PYTHON_MODULE()`.

#### 9.6.3.11 `int console::update () [virtual]`

Updating (must be call on each loop).

Reimplemented from [base](#).

Definition at line 72 of file console.cpp.

References box, cons, env, and FPS.

Referenced by coeur::update().

#### 9.6.3.12 void console::write (std::string const & *message*)

Write some text on the [console](#). This method log a text to the [console](#), used by python to replace stdout

##### Parameters:

*message* Text to log

##### Returns:

Nothing

Definition at line 130 of file console.cpp.

References text, and tx.

Referenced by BOOST\_PYTHON\_MODULE(), fps(), help(), OnEvent(), position(), quit(), rotation(), and target().

### 9.6.4 Member Data Documentation

#### 9.6.4.1 IGUIMain\* console::box [private]

Box where you type the command to launch to the [console](#).

Definition at line 90 of file console.h.

Referenced by init(), OnEvent(), and update().

#### 9.6.4.2 IGUIImage\* console::cons [private]

Just use to contain all the [item](#) of the [console](#).

Definition at line 84 of file console.h.

Referenced by exit(), init(), OnEvent(), and update().

#### 9.6.4.3 IrrlichtDevice\* console::device [private]

The Irrlicht Device.

Definition at line 75 of file console.h.

#### 9.6.4.4 IGUIEnvironment\* console::env [private]

The Irrlicht Environment.

Definition at line 78 of file console.h.

Referenced by fps(), init(), and update().



**9.6.4.5 IGUIStaticText\* console::FPS [private]**

Box who displayed the fps.

Definition at line 93 of file console.h.

Referenced by init(), and update().

**9.6.4.6 std::vector<stringw>\* console::history [private]**

Contain the history of the command enter.

Definition at line 96 of file console.h.

Referenced by console(), and OnEvent().

**9.6.4.7 unsigned int console::histPos [private]**

Contain the position in the history.

Definition at line 99 of file console.h.

Referenced by console(), and OnEvent().

**9.6.4.8 stringw\* console::text [private]**

Contain the text display by the [console](#).

Definition at line 81 of file console.h.

Referenced by console(), and write().

**9.6.4.9 IGUIEditBox\* console::tx [private]**

Box where are displayed the text of the [console](#).

Definition at line 87 of file console.h.

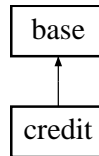
Referenced by init(), and write().

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

- include/[console.h](#)
- source/[console.cpp](#)

## 9.7 credit Class Reference

`#include <credit.h>`Inheritance diagram for `credit::`



### Public Member Functions

- `credit ()`
- `~credit ()`
- `int init (coeur *, IGUIEnvironment *, IXMLReader *)`
- `int update ()`  
*Updating(should be call on each loop).*
- `int close ()`  
*Close the `base`.*
- `int start ()`
- `int stop ()`
- `int addLogo (stringc)`
- `int addText (stringw)`
- `bool OnEvent (const SEvent &)`

### Private Attributes

- `IGUIEnvironment * env`
- `IGUIFont * mainFont`
- `bool activate`
- `int newY`
- `int y`
- `stringc sound`
- `std::vector< IGUIImage * > * logos`
- `std::vector< IGUIStaticText * > * texts`

#### 9.7.1 Detailed Description

Definition at line 2 of file `credit.h`.

#### 9.7.2 Constructor & Destructor Documentation

##### 9.7.2.1 `credit::credit ()`

Definition at line 3 of file `credit.cpp`.

References `logos`, and `texts`.

### 9.7.2.2 credit::~~credit ()

Definition at line 9 of file credit.cpp.

References logos, and texts.

## 9.7.3 Member Function Documentation

### 9.7.3.1 int credit::addLogo (stringc *path*)

Definition at line 127 of file credit.cpp.

References activate, env, coeur::getDim(), coeur::getTexture(), base::log(), logos, base::mycore, and newY.

Referenced by BOOST\_PYTHON\_MODULE(), and init().

### 9.7.3.2 int credit::addText (stringw *text*)

Definition at line 143 of file credit.cpp.

References activate, env, coeur::getDim(), base::log(), base::mycore, newY, and texts.

Referenced by BOOST\_PYTHON\_MODULE(), and init().

### 9.7.3.3 int credit::close () [virtual]

Close the [base](#). Close the log and put [base::initialize](#) to false

#### Returns:

[logger::closeFile\(\)](#)

Reimplemented from [base](#).

Definition at line 78 of file credit.cpp.

References base::log(), logos, stop(), and texts.

Referenced by coeur::close().

### 9.7.3.4 int credit::init (coeur \* *c*, IGUIEnvironment \* *e*, IXMLReader \* *xmlCredit*)

Definition at line 15 of file credit.cpp.

References activate, addLogo(), addText(), env, coeur::getDim(), base::log(), logos, newY, sound, texts, and y.

Referenced by coeur::init().

### 9.7.3.5 bool credit::OnEvent (const SEvent & *event*)

Definition at line 156 of file credit.cpp.

Referenced by coeur::OnEvent().

#### 9.7.3.6 `int credit::start ()`

Definition at line 100 of file `credit.cpp`.

References `activate`, `base::log()`, `base::mycore`, `coeur::play()`, `sound`, and `y`.

Referenced by `BOOST_PYTHON_MODULE()`, and `coeur::startCredit()`.

#### 9.7.3.7 `int credit::stop ()`

Definition at line 109 of file `credit.cpp`.

References `activate`, `base::log()`, `logos`, `texts`, and `y`.

Referenced by `BOOST_PYTHON_MODULE()`, `close()`, and `update()`.

#### 9.7.3.8 `int credit::update () [virtual]`

Updating(should be call on each loop). Does nothing

##### Returns:

Always 0

Reimplemented from [base](#).

Definition at line 55 of file `credit.cpp`.

References `activate`, `logos`, `newY`, `stop()`, `texts`, and `y`.

Referenced by `coeur::update()`.

### 9.7.4 Member Data Documentation

#### 9.7.4.1 `bool credit::activate [private]`

Definition at line 24 of file `credit.h`.

Referenced by `addLogo()`, `addText()`, `init()`, `start()`, `stop()`, and `update()`.

#### 9.7.4.2 `IGUIEnvironment* credit::env [private]`

Definition at line 20 of file `credit.h`.

Referenced by `addLogo()`, `addText()`, and `init()`.

#### 9.7.4.3 `std::vector<IGUIImage*>* credit::logos [private]`

Definition at line 31 of file `credit.h`.

Referenced by `addLogo()`, `close()`, `credit()`, `init()`, `stop()`, `update()`, and `~credit()`.

#### 9.7.4.4 `IGUIFont* credit::mainFont [private]`

Definition at line 22 of file `credit.h`.

**9.7.4.5 int credit::newY [private]**

Definition at line 26 of file credit.h.

Referenced by addLogo(), addText(), init(), and update().

**9.7.4.6 stringc credit::sound [private]**

Definition at line 29 of file credit.h.

Referenced by init(), and start().

**9.7.4.7 std::vector<IGUIStaticText\*>\* credit::texts [private]**

Definition at line 32 of file credit.h.

Referenced by addText(), close(), credit(), init(), stop(), update(), and ~credit().

**9.7.4.8 int credit::y [private]**

Definition at line 27 of file credit.h.

Referenced by init(), start(), stop(), and update().

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

- [include/credit.h](#)
- [source/credit.cpp](#)

## 9.8 cusString\_python< cusString > Struct Template Reference

### Public Member Functions

- [cusString\\_python\(\)](#)

### Static Public Member Functions

- static PyObject \* [convert](#) (cusString const &s)
- static void \* [convertible](#) (PyObject \*obj\_ptr)
- static void [construct](#) (PyObject \*obj\_ptr, boost::python::converter::rvalue\_from\_python\_stage1\_data \*data)

### 9.8.1 Detailed Description

`template<class cusString> struct cusString_python< cusString >`

Definition at line 26 of file modPyIrr.cpp.

### 9.8.2 Constructor & Destructor Documentation

**9.8.2.1** `template<class cusString > cusString_python< cusString >::cusString_python()`  
[inline]

Definition at line 28 of file modPyIrr.cpp.

References `cusString_python< cusString >::construct()`, and `cusString_python< cusString >::convertible()`.

### 9.8.3 Member Function Documentation

**9.8.3.1** `template<class cusString > static void cusString_python< cusString >::construct`  
(PyObject \* *obj\_ptr*, boost::python::converter::rvalue\_from\_python\_stage1\_data \* *data*)  
[inline, static]

Definition at line 44 of file modPyIrr.cpp.

Referenced by `cusString_python< cusString >::cusString_python()`.

**9.8.3.2** `template<class cusString > static PyObject* cusString_python< cusString >::convert`  
(cusString const & *s*) [inline, static]

Definition at line 33 of file modPyIrr.cpp.

**9.8.3.3** `template<class cusString > static void* cusString_python< cusString >::convertible`  
(PyObject \* *obj\_ptr*) [inline, static]

Definition at line 38 of file modPyIrr.cpp.

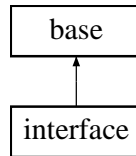
Referenced by `cusString_python< cusString >::cusString_python()`.

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

- [source/modPyIrr.cpp](#)

## 9.9 interface Class Reference

#include <interface.h> Inheritance diagram for interface::



### Public Member Functions

- [interface](#) ()
- [~interface](#) ()
- int [init](#) ([coeur](#) \*, [IGUIEnvironment](#) \*, [IXMLReader](#) \*, bool=false)
- int [update](#) ()
  - Updating(should be call on each loop).*
- int [close](#) ()
  - Close the [base](#).*
- int [setImage](#) (stringc)
- int [afficherInventaire](#) (bool)
- int [setImagePosition](#) (position2d< s32 >)
- int [addItemInventaire](#) (item)
- int [addInventaire](#) (int, stringc, int, [INVENTAIRE\\_POS](#), stringc)
- [inventaire](#) \* [getInventaire](#) (int)
- bool [OnEvent](#) (const [SEvent](#) &)

### Private Member Functions

- int [speak](#) ([localItem](#) \*)
- int [getFreePlace](#) ()

### Private Attributes

- [IGUIEnvironment](#) \* [env](#)
- stringc [curDefault](#)
- stringc [curTake](#)
- stringc [curLook](#)
- stringc [curAction](#)
- stringc [soundFind](#)
- [IGUIFont](#) \* [mainFont](#)
- [IGUIImage](#) \* [image](#)
- [IGUIStaticText](#) \* [txt](#)
- int [selectedInv](#)
- int [selectedItem](#)
- int [pselectedInv](#)



- int [pselectedItem](#)
- item [handle](#)
- std::vector< [inventaire](#) \* > \* [inv](#)
- u32 [time](#)
- bool [v](#)
- bool [classic](#)
- IParticleSystemSceneNode \* [psCur](#)  
*The particle system for the cursor.*
- IParticleEmitter \* [emCur](#)  
*The particle system for the cursor.*
- IParticleAttractionAffector \* [pafCur](#)  
*The particle affector for the cursor.*

### 9.9.1 Detailed Description

Definition at line 2 of file interface.h.

### 9.9.2 Constructor & Destructor Documentation

#### 9.9.2.1 interface::interface ()

/todo refaire [inventaire](#) pour mettre un nombre infini d'item /todo Afficher [inventaire](#) en fonction taille(redimensionner) /todo Check sound really launched /todo Load the texture at the beginning in order not to slow down the game

Definition at line 8 of file interface.cpp.

References [inv](#).

#### 9.9.2.2 interface::~~interface ()

Definition at line 13 of file interface.cpp.

References [inv](#).

### 9.9.3 Member Function Documentation

#### 9.9.3.1 int interface::addInventaire (int *i*, stringc *path*, int *v*, INVENTAIRE\_POS *p*, stringc *pathvoid*)

Definition at line 213 of file interface.cpp.

References [env](#), [coeur::getDim\(\)](#), [inventaire::init\(\)](#), [inv](#), [INVENTAIRE\\_POS\\_BAS](#), [INVENTAIRE\\_POS\\_DROITE](#), [INVENTAIRE\\_POS\\_GAUCHE](#), [INVENTAIRE\\_POS\\_HAUT](#), [base::log\(\)](#), and [base::mycore](#).

Referenced by [init\(\)](#).

### 9.9.3.2 `int interface::addItemInventaire (item m)`

Definition at line 244 of file `interface.cpp`.

References `item::getId()`, `inv`, `base::log()`, `selectedInv`, and `selectedItem`.

Referenced by `BOOST_PYTHON_MODULE()`, and `OnEvent()`.

### 9.9.3.3 `int interface::afficherInventaire (bool a)`

Definition at line 203 of file `interface.cpp`.

References `inv`, and `v`.

Referenced by `BOOST_PYTHON_MODULE()`, `init()`, and `update()`.

### 9.9.3.4 `int interface::close () [virtual]`

Close the [base](#). Close the log and put [base::initialize](#) to false

#### Returns:

[logger::closeFile\(\)](#)

Reimplemented from [base](#).

Definition at line 184 of file `interface.cpp`.

References `inv`, and `base::log()`.

Referenced by `coeur::close()`.

### 9.9.3.5 `int interface::getFreePlace () [private]`

Definition at line 356 of file `interface.cpp`.

References `inv`.

Referenced by `OnEvent()`.

### 9.9.3.6 `inventaire* interface::getInventaire (int)`

### 9.9.3.7 `int interface::init (coeur * c, IGUIEnvironment * e, IXMLReader * xmlInterface, bool cl = false)`

Definition at line 18 of file `interface.cpp`.

References `addInventaire()`, `coeur::addParticleSystem()`, `afficherInventaire()`, `classic`, `curAction`, `curDefault`, `curLook`, `curTake`, `emCur`, `env`, `coeur::getDim()`, `coeur::getTexture()`, `handle`, `image`, `inv`, `base::log()`, `mainFont`, `base::mycore`, `pafCur`, `psCur`, `pselectedInv`, `pselectedItem`, `selectedInv`, `selectedItem`, `soundFind`, and `txt`.

Referenced by `coeur::init()`.

### 9.9.3.8 `bool interface::OnEvent (const SEvent & event)`

Definition at line 251 of file `interface.cpp`.

References ACTION\_ANIMATE, ACTION\_OBSERVER, ACTION\_PRENDRE, addItemInventaire(), coeur::anim(), actionMesh::getAction(), animateMesh::getAnim(), animateMesh::getBegin(), item::getCombi(), actionMesh::getCombi(), coeur::getCombinaison(), item::getCombiResult(), animateMesh::getEnd(), getFreePlace(), staticMesh::getId(), item::getId(), coeur::getItem(), staticMesh::getNode(), coeur::getObservationItem(), coeur::getObservationMesh(), coeur::getSelectedAction(), handle, inv, base::log(), base::mycore, coeur::play(), pselectedInv, pselectedItem, selectedInv, selectedItem, soundFind, and speak().

Referenced by coeur::OnEvent().

### 9.9.3.9 int interface::setImage (stringc *tex*)

Definition at line 197 of file interface.cpp.

References coeur::getTexture(), image, and base::mycore.

Referenced by BOOST\_PYTHON\_MODULE(), and update().

### 9.9.3.10 int interface::setImagePosition (position2d< s32 > *pos*)

Definition at line 238 of file interface.cpp.

References image.

Referenced by BOOST\_PYTHON\_MODULE(), and update().

### 9.9.3.11 int interface::speak (localItem \* *loc*) [private]

Definition at line 366 of file interface.cpp.

References localItem::getPath(), localItem::getText(), coeur::getTime(), base::mycore, coeur::play(), time, and txt.

Referenced by OnEvent().

### 9.9.3.12 int interface::update () [virtual]

Updating(should be call on each loop). Does nothing

#### Returns:

Always 0

Reimplemented from [base](#).

Definition at line 108 of file interface.cpp.

References ACTION\_ANIMATE, ACTION\_OBSERVER, ACTION\_PRENDRE, afficherInventaire(), coeur::cam, classic, curAction, curDefault, curLook, curTake, emCur, env, actionMesh::getAction(), actionMesh::getColor(), coeur::getCurPos(), coeur::getDim(), item::getId(), staticMesh::getNode(), coeur::getSelectedAction(), item::getTexture(), handle, image, inv, base::mycore, pafCur, psCur, selectedInv, selectedItem, setImage(), setImagePosition(), coeur::smgr, txt, and v.

Referenced by coeur::update().

## 9.9.4 Member Data Documentation

### 9.9.4.1 `bool interface::classic` `[private]`

Definition at line 62 of file interface.h.

Referenced by `init()`, and `update()`.

### 9.9.4.2 `stringc interface::curAction` `[private]`

Definition at line 38 of file interface.h.

Referenced by `init()`, and `update()`.

### 9.9.4.3 `stringc interface::curDefault` `[private]`

Definition at line 35 of file interface.h.

Referenced by `init()`, and `update()`.

### 9.9.4.4 `stringc interface::curLook` `[private]`

Definition at line 37 of file interface.h.

Referenced by `init()`, and `update()`.

### 9.9.4.5 `stringc interface::curTake` `[private]`

Definition at line 36 of file interface.h.

Referenced by `init()`, and `update()`.

### 9.9.4.6 `IParticleEmitter* interface::emCur` `[private]`

The particle system for the cursor.

Definition at line 68 of file interface.h.

Referenced by `init()`, and `update()`.

### 9.9.4.7 `IGUIEnvironment* interface::env` `[private]`

Definition at line 33 of file interface.h.

Referenced by `addInventaire()`, `init()`, and `update()`.

### 9.9.4.8 `item interface::handle` `[private]`

Definition at line 54 of file interface.h.

Referenced by `init()`, `OnEvent()`, and `update()`.

**9.9.4.9 IGUIImage\* interface::image [private]**

Definition at line 44 of file interface.h.

Referenced by init(), setImage(), setImagePosition(), and update().

**9.9.4.10 std::vector<inventaire\*>\* interface::inv [private]**

Definition at line 56 of file interface.h.

Referenced by addInventaire(), addItemInventaire(), afficherInventaire(), close(), getFreePlace(), init(), interface(), OnEvent(), update(), and ~interface().

**9.9.4.11 IGUIFont\* interface::mainFont [private]**

Definition at line 42 of file interface.h.

Referenced by init().

**9.9.4.12 IParticleAttractionAffector\* interface::pafCur [private]**

The particle affector for the cursor.

Definition at line 71 of file interface.h.

Referenced by init(), and update().

**9.9.4.13 IParticleSystemSceneNode\* interface::psCur [private]**

The particle system for the cursor.

Definition at line 65 of file interface.h.

Referenced by init(), and update().

**9.9.4.14 int interface::pselectedInv [private]**

Definition at line 51 of file interface.h.

Referenced by init(), and OnEvent().

**9.9.4.15 int interface::pselectedItem [private]**

Definition at line 52 of file interface.h.

Referenced by init(), and OnEvent().

**9.9.4.16 int interface::selectedInv [private]**

Definition at line 48 of file interface.h.

Referenced by addItemInventaire(), init(), OnEvent(), and update().

**9.9.4.17 int interface::selectedItem [private]**

Definition at line 49 of file interface.h.

Referenced by addItemInventaire(), init(), OnEvent(), and update().

**9.9.4.18 stringc interface::soundFind [private]**

Definition at line 40 of file interface.h.

Referenced by init(), and OnEvent().

**9.9.4.19 u32 interface::time [private]**

Definition at line 58 of file interface.h.

Referenced by speak().

**9.9.4.20 IGUIStaticText\* interface::txt [private]**

Definition at line 46 of file interface.h.

Referenced by init(), speak(), and update().

**9.9.4.21 bool interface::v [private]**

Definition at line 60 of file interface.h.

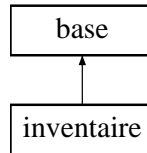
Referenced by afficherInventaire(), and update().

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

- [include/interface.h](#)
- [source/interface.cpp](#)

## 9.10 inventaire Class Reference

`#include <inventaire.h>`Inheritance diagram for `inventaire`::



### Public Member Functions

- `inventaire ()`
- `~inventaire ()`
- `int init (int, coeur \*, IGUIEnvironment \*, stringc, rect< s32 >, int, INVENTAIRE\_POS, stringc)`
- `int update ()`  
*Updating(should be call on each loop).*
- `int close ()`  
*Close the [base](#).*
- `int afficher (bool)`
- `IGUIImage \* getImage ()`
- `int addItem (int n, item)`
- `int removeItem (int)`
- `int getNumItem (const position2d< s32 >)`
- `item getItem (int)`
- `item getItem (const position2d< s32 >)`
- `int getFreePlace ()`
- `bool OnEvent (const SEvent &)`

### Private Attributes

- [IGUIEnvironment \\*](#) `env`
- `int vtot`
- `int volume`
- `item \* it`
- `IGUIImage \*\* images`
- `int maxItem`
- `INVENTAIRE\_POS position`
- `IGUIImage \* image`

#### 9.10.1 Detailed Description

Definition at line 10 of file `inventaire.h`.

## 9.10.2 Constructor & Destructor Documentation

### 9.10.2.1 `inventaire::inventaire ()`

Definition at line 8 of file `inventaire.cpp`.

### 9.10.2.2 `inventaire::~~inventaire ()`

Definition at line 13 of file `inventaire.cpp`.

## 9.10.3 Member Function Documentation

### 9.10.3.1 `int inventaire::addItem (int n, item m)`

Definition at line 107 of file `inventaire.cpp`.

References `base::getId()`, `item::getTexture()`, `coeur::getTexture()`, `images`, `it`, `maxItem`, and `base::mycore`.

### 9.10.3.2 `int inventaire::afficher (bool a)`

Definition at line 96 of file `inventaire.cpp`.

References `image`.

Referenced by `init()`.

### 9.10.3.3 `int inventaire::close () [virtual]`

Close the `base`. Close the log and put `base::initialize` to false

#### Returns:

`logger::closeFile()`

Reimplemented from `base`.

Definition at line 90 of file `inventaire.cpp`.

### 9.10.3.4 `int inventaire::getFreePlace ()`

Definition at line 145 of file `inventaire.cpp`.

References `base::getId()`, `it`, and `maxItem`.

### 9.10.3.5 `IGUIImage * inventaire::getImage ()`

Definition at line 102 of file `inventaire.cpp`.

References `image`.



**9.10.3.6 item inventaire::getItem (const position2d< s32 > pos)**

Definition at line 140 of file inventaire.cpp.

References `getNumItem()`, and `it`.

**9.10.3.7 item inventaire::getItem (int n)**

Definition at line 133 of file inventaire.cpp.

References `it`, and `maxItem`.

**9.10.3.8 int inventaire::getNumItem (const position2d< s32 > pos)**

Definition at line 125 of file inventaire.cpp.

References `images`, and `maxItem`.

Referenced by `getItem()`.

**9.10.3.9 int inventaire::init (int i, coeur \* c, IGUIEnvironment \* e, stringc path, rect< s32 > pos, int v, INVENTAIRE\_POS p, stringc pathvoid)**

Definition at line 18 of file inventaire.cpp.

References `afficher()`, `env`, `coeur::getTexture()`, `image`, `images`, `INVENTAIRE_POS_BAS`, `INVENTAIRE_POS_DROITE`, `INVENTAIRE_POS_GAUCHE`, `INVENTAIRE_POS_HAUT`, `it`, `maxItem`, `base::mycore`, `position`, `volume`, and `vtot`.

Referenced by `interface::addInventaire()`.

**9.10.3.10 bool inventaire::OnEvent (const SEvent & event)**

Definition at line 153 of file inventaire.cpp.

**9.10.3.11 int inventaire::removeItem (int n)**

Definition at line 116 of file inventaire.cpp.

References `base::getId()`, `images`, `it`, and `maxItem`.

**9.10.3.12 int inventaire::update () [virtual]**

Updating(should be call on each loop). Does nothing

**Returns:**

Always 0

Reimplemented from [base](#).

Definition at line 85 of file inventaire.cpp.

## 9.10.4 Member Data Documentation

### 9.10.4.1 IGUIEnvironment\* inventaire::env [private]

Definition at line 35 of file inventaire.h.

Referenced by `init()`.

### 9.10.4.2 IGUIImage\* inventaire::image [private]

Definition at line 46 of file inventaire.h.

Referenced by `afficher()`, `getImage()`, and `init()`.

### 9.10.4.3 IGUIImage\*\* inventaire::images [private]

Definition at line 41 of file inventaire.h.

Referenced by `addItem()`, `getNumItem()`, `init()`, and `removeItem()`.

### 9.10.4.4 item\* inventaire::it [private]

Definition at line 40 of file inventaire.h.

Referenced by `addItem()`, `getFreePlace()`, `getItem()`, `init()`, and `removeItem()`.

### 9.10.4.5 int inventaire::maxItem [private]

Definition at line 42 of file inventaire.h.

Referenced by `addItem()`, `getFreePlace()`, `getItem()`, `getNumItem()`, `init()`, and `removeItem()`.

### 9.10.4.6 INVENTAIRE\_POS inventaire::position [private]

Definition at line 44 of file inventaire.h.

Referenced by `init()`.

### 9.10.4.7 int inventaire::volume [private]

Definition at line 38 of file inventaire.h.

Referenced by `init()`.

### 9.10.4.8 int inventaire::vtot [private]

Definition at line 37 of file inventaire.h.

Referenced by `init()`.

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

- [include/inventaire.h](#)

- [source/inventaire.cpp](#)

## 9.11 item Class Reference

```
#include <item.h>
```

### Public Member Functions

- [item](#) ()
- [~item](#) ()
- [init](#) (int, stringc, int=-1, int=-1)
- [getId](#) ()
- [stringc getTexture](#) ()
- [int getCombi](#) ()
- [int getCombiResult](#) ()
- [stringc repr](#) ()

### Private Attributes

- [int id](#)
- [stringc texture](#)
- [int combi](#)
- [int combiResult](#)

#### 9.11.1 Detailed Description

Definition at line 2 of file item.h.

#### 9.11.2 Constructor & Destructor Documentation

##### 9.11.2.1 item::item ()

Definition at line 3 of file item.cpp.

##### 9.11.2.2 item::~~item ()

Definition at line 8 of file item.cpp.

#### 9.11.3 Member Function Documentation

##### 9.11.3.1 int item::getCombi ()

Definition at line 32 of file item.cpp.

References [combi](#).

Referenced by [BOOST\\_PYTHON\\_MODULE\(\)](#), and [interface::OnEvent\(\)](#).

### 9.11.3.2 int item::getCombiResult ()

Definition at line 37 of file item.cpp.

References combiResult.

Referenced by BOOST\_PYTHON\_MODULE(), and interface::OnEvent().

### 9.11.3.3 int item::getId ()

Definition at line 22 of file item.cpp.

References id.

Referenced by interface::addItemInventaire(), BOOST\_PYTHON\_MODULE(), interface::OnEvent(), and interface::update().

### 9.11.3.4 stringc item::getTexture ()

Definition at line 27 of file item.cpp.

References texture.

Referenced by inventaire::addItem(), BOOST\_PYTHON\_MODULE(), and interface::update().

### 9.11.3.5 int item::init (int *i*, stringc *tex*, int *c* = -1, int *cr* = -1)

Definition at line 13 of file item.cpp.

References combi, combiResult, and texture.

Referenced by chap::addItem(), and BOOST\_PYTHON\_MODULE().

### 9.11.3.6 stringc item::repr ()

Definition at line 43 of file item.cpp.

References combi, combiResult, and texture.

Referenced by BOOST\_PYTHON\_MODULE().

## 9.11.4 Member Data Documentation

### 9.11.4.1 int item::combi [private]

Definition at line 19 of file item.h.

Referenced by getCombi(), init(), and repr().

### 9.11.4.2 int item::combiResult [private]

Definition at line 20 of file item.h.

Referenced by getCombiResult(), init(), and repr().

**9.11.4.3 int item::id [private]**

Definition at line 17 of file item.h.

Referenced by getId().

**9.11.4.4 stringc item::texture [private]**

Definition at line 18 of file item.h.

Referenced by getTexture(), init(), and repr().

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

- include/[item.h](#)
- source/[item.cpp](#)

## 9.12 lieu Class Reference

`#include <lieu.h>`Inheritance diagram for lieu::



### Public Member Functions

- [lieu \(\)](#)
- [~lieu \(\)](#)
- [int init](#) (int, [coeur \\*](#), [IXMLReader \\*](#))
- [int update](#) ()  
*Updating(should be call on each loop).*
- [int close](#) ()  
*Close the [base](#).*
- [staticMesh \\*](#) [addStaticMesh](#) (stringc, int)
- [observationMesh \\*](#) [addObservationMesh](#) (stringc, int, int)
- [prendreMesh \\*](#) [addPrendreMesh](#) (stringc, int, int, int)
- [animateMesh \\*](#) [addAnimateMesh](#) (stringc, int, int, int, int, int)
- [staticMesh \\*](#) [getStaticMeshFromId](#) (int id)
- [actionMesh \\*](#) [getActionMesh](#) (int id)
- [actionMesh \\*](#) [getActionMesh](#) (ISceneNode \*)
- [bool OnEvent](#) (const [SEvent](#) &event)

### Private Attributes

- `std::vector< staticMesh * > * listStatic`
- `std::vector< actionMesh * > * listAction`

#### 9.12.1 Detailed Description

Definition at line 2 of file lieu.h.

#### 9.12.2 Constructor & Destructor Documentation

##### 9.12.2.1 lieu::lieu ()

Definition at line 3 of file lieu.cpp.

References [listAction](#), and [listStatic](#).

### 9.12.2.2 lieu::~~lieu ()

Definition at line 9 of file lieu.cpp.

References listAction, and listStatic.

## 9.12.3 Member Function Documentation

### 9.12.3.1 animateMesh \* lieu::addAnimateMesh (stringc *path*, int *i*, int *c*, int *ite*, int *b*, int *e*)

Definition at line 209 of file lieu.cpp.

References coeur::addMesh(), staticMesh::getNode(), animateMesh::init(), listAction, base::log(), and base::mycore.

Referenced by init().

### 9.12.3.2 observationMesh \* lieu::addObservationMesh (stringc *path*, int *i*, int *c*)

Definition at line 189 of file lieu.cpp.

References coeur::addMesh(), staticMesh::getNode(), observationMesh::init(), listAction, base::log(), and base::mycore.

Referenced by init().

### 9.12.3.3 prendreMesh \* lieu::addPrendreMesh (stringc *path*, int *i*, int *c*, int *ite*)

Definition at line 199 of file lieu.cpp.

References coeur::addMesh(), staticMesh::getNode(), prendreMesh::init(), listAction, base::log(), and base::mycore.

Referenced by init().

### 9.12.3.4 staticMesh \* lieu::addStaticMesh (stringc *path*, int *i*)

Definition at line 179 of file lieu.cpp.

References coeur::addMesh(), staticMesh::getNode(), staticMesh::init(), listStatic, base::log(), and base::mycore.

Referenced by init().

### 9.12.3.5 int lieu::close () [virtual]

Close the [base](#). Close the log and put [base::initialize](#) to false

#### Returns:

[logger::closeFile\(\)](#)

Reimplemented from [base](#).

Definition at line 159 of file lieu.cpp.



References listAction, listStatic, and base::log().

Referenced by coeur::close(), and coeur::loadLieu().

#### 9.12.3.6 actionMesh \* lieu::getActionMesh (ISceneNode \* node)

Definition at line 239 of file lieu.cpp.

References listAction.

#### 9.12.3.7 actionMesh \* lieu::getActionMesh (int id)

Definition at line 229 of file lieu.cpp.

References listAction.

Referenced by coeur::getActionMesh().

#### 9.12.3.8 staticMesh \* lieu::getStaticMeshFromId (int id)

Definition at line 219 of file lieu.cpp.

References listStatic.

#### 9.12.3.9 int lieu::init (int i, coeur \* c, IXMLReader \* xmlLieu)

Definition at line 15 of file lieu.cpp.

References addAnimateMesh(), coeur::addLight(), addObservationMesh(), addPrendreMesh(), addStaticMesh(), staticMesh::getNode(), coeur::getTexture(), listAction, listStatic, base::log(), base::mycore, and coeur::setSkyBox().

Referenced by coeur::loadLieu().

#### 9.12.3.10 bool lieu::OnEvent (const SEvent & event)

Definition at line 249 of file lieu.cpp.

#### 9.12.3.11 int lieu::update () [virtual]

Updating(should be call on each loop). Does nothing

##### Returns:

Always 0

Reimplemented from [base](#).

Definition at line 154 of file lieu.cpp.

## 9.12.4 Member Data Documentation

### 9.12.4.1 `std::vector<actionMesh*>* lieu::listAction` **[private]**

Definition at line 24 of file lieu.h.

Referenced by `addAnimateMesh()`, `addObservationMesh()`, `addPrendreMesh()`, `close()`, `getActionMesh()`, `init()`, `lieu()`, and `~lieu()`.

### 9.12.4.2 `std::vector<staticMesh*>* lieu::listStatic` **[private]**

Definition at line 23 of file lieu.h.

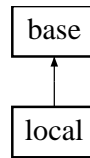
Referenced by `addStaticMesh()`, `close()`, `getStaticMeshFromId()`, `init()`, `lieu()`, and `~lieu()`.

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

- [include/lieu.h](#)
- [source/lieu.cpp](#)

## 9.13 local Class Reference

`#include <local.h>`Inheritance diagram for local::



### Public Member Functions

- `local ()`
- `~local ()`
- `int init (int, coeur *, IXMLReader *)`
- `int update ()`  
*Updating(should be call on each loop).*
- `int close ()`  
*Close the base.*
- `localItem * getObservationMesh (int)`
- `localItem * getObservationItem (int)`
- `localItem * getCombinaison ()`
- `stringw getCaption ()`
- `bool OnEvent (const SEvent &)`

### Private Member Functions

- `int addObservationMesh (int, stringw, stringc)`
- `int addObservationItem (int, stringw, stringc)`
- `int addCombinaison (int, stringw, stringc)`

### Private Attributes

- `std::vector< localItem * > * observationMesh`
- `std::vector< localItem * > * observationItem`
- `std::vector< localItem * > * combinaison`
- `stringw caption`

#### 9.13.1 Detailed Description

Definition at line 20 of file local.h.

## 9.13.2 Constructor & Destructor Documentation

### 9.13.2.1 `local::local ()`

Definition at line 40 of file `local.cpp`.

References `combinaison`, and `observationItem`.

### 9.13.2.2 `local::~~local ()`

Definition at line 47 of file `local.cpp`.

References `combinaison`, and `observationItem`.

## 9.13.3 Member Function Documentation

### 9.13.3.1 `int local::addCombinaison (int id, stringw text, stringc path) [private]`

Definition at line 145 of file `local.cpp`.

References `combinaison`, `localItem::init()`, and `base::log()`.

Referenced by `init()`.

### 9.13.3.2 `int local::addObservationItem (int id, stringw text, stringc path) [private]`

Definition at line 136 of file `local.cpp`.

References `localItem::init()`, `base::log()`, and `observationItem`.

Referenced by `init()`.

### 9.13.3.3 `int local::addObservationMesh (int id, stringw text, stringc path) [private]`

Definition at line 127 of file `local.cpp`.

References `localItem::init()`, and `base::log()`.

Referenced by `init()`.

### 9.13.3.4 `int local::close () [virtual]`

Close the `base`. Close the log and put `base::initialize` to false

#### Returns:

`logger::closeFile()`

Reimplemented from `base`.

Definition at line 100 of file `local.cpp`.

References `combinaison`, `base::log()`, and `observationItem`.

Referenced by `coeur::close()`, and `coeur::loadLocal()`.

#### 9.13.3.5 stringw local::getCaption ()

Definition at line 159 of file local.cpp.

References caption.

Referenced by BOOST\_PYTHON\_MODULE(), and coeur::init().

#### 9.13.3.6 localItem \* local::getCombinaison ()

Definition at line 154 of file local.cpp.

References combinaison.

Referenced by BOOST\_PYTHON\_MODULE(), and coeur::getCombinaison().

#### 9.13.3.7 localItem \* local::getObservationItem (int id)

Definition at line 179 of file local.cpp.

References observationItem.

Referenced by BOOST\_PYTHON\_MODULE(), and coeur::getObservationItem().

#### 9.13.3.8 localItem \* local::getObservationMesh (int id)

Definition at line 169 of file local.cpp.

Referenced by BOOST\_PYTHON\_MODULE(), and coeur::getObservationMesh().

#### 9.13.3.9 int local::init (int i, coeur \* c, IXMLReader \* xmlLocal)

Definition at line 54 of file local.cpp.

References addCombinaison(), addObservationItem(), addObservationMesh(), caption, combinaison, base::log(), and observationItem.

Referenced by coeur::loadLocal().

#### 9.13.3.10 bool local::OnEvent (const SEvent & event)

Definition at line 164 of file local.cpp.

#### 9.13.3.11 int local::update () [virtual]

Updating(should be call on each loop). Does nothing

##### Returns:

Always 0

Reimplemented from [base](#).

Definition at line 95 of file local.cpp.

## 9.13.4 Member Data Documentation

### 9.13.4.1 `stringw local::caption` `[private]`

Definition at line 46 of file local.h.

Referenced by `getCaption()`, and `init()`.

### 9.13.4.2 `std::vector<localItem*>* local::combinaison` `[private]`

Definition at line 44 of file local.h.

Referenced by `addCombinaison()`, `close()`, `getCombinaison()`, `init()`, `local()`, and `~local()`.

### 9.13.4.3 `std::vector<localItem*>* local::observationItem` `[private]`

Definition at line 43 of file local.h.

Referenced by `addObservationItem()`, `close()`, `getObservationItem()`, `init()`, `local()`, and `~local()`.

### 9.13.4.4 `std::vector<localItem*>* local::observationMesh` `[private]`

Definition at line 42 of file local.h.

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

- [include/local.h](#)
- [source/local.cpp](#)

## 9.14 localItem Class Reference

```
#include <local.h>
```

### Public Member Functions

- [localItem](#) ()
- [~localItem](#) ()
- [init](#) (int, stringw, stringc)
- [getId](#) ()
- [getText](#) ()
- [getPath](#) ()

### Private Attributes

- [id](#)
- [text](#)
- [path](#)

#### 9.14.1 Detailed Description

Definition at line 2 of file local.h.

#### 9.14.2 Constructor & Destructor Documentation

##### 9.14.2.1 localItem::localItem ()

Definition at line 3 of file local.cpp.

##### 9.14.2.2 localItem::~~localItem ()

Definition at line 8 of file local.cpp.

#### 9.14.3 Member Function Documentation

##### 9.14.3.1 int localItem::getId ()

Definition at line 21 of file local.cpp.

References [id](#).

Referenced by [BOOST\\_PYTHON\\_MODULE\(\)](#).

##### 9.14.3.2 stringc localItem::getPath ()

Definition at line 33 of file local.cpp.

References [path](#).

Referenced by [BOOST\\_PYTHON\\_MODULE\(\)](#), and [interface::speak\(\)](#).

### 9.14.3.3 `stringw localItem::getText ()`

Definition at line 27 of file local.cpp.

References text.

Referenced by `BOOST_PYTHON_MODULE()`, and `interface::speak()`.

### 9.14.3.4 `int localItem::init (int i, stringw t, stringc p)`

Definition at line 13 of file local.cpp.

References path, and text.

Referenced by `local::addCombinaison()`, `local::addObservationItem()`, `local::addObservationMesh()`, and `BOOST_PYTHON_MODULE()`.

## 9.14.4 Member Data Documentation

### 9.14.4.1 `int localItem::id` **[private]**

Definition at line 14 of file local.h.

Referenced by `getId()`.

### 9.14.4.2 `stringc* localItem::path` **[private]**

Definition at line 16 of file local.h.

Referenced by `getPath()`, and `init()`.

### 9.14.4.3 `stringw* localItem::text` **[private]**

Definition at line 15 of file local.h.

Referenced by `getText()`, and `init()`.

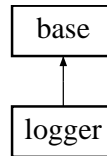
The documentation for this class was generated from the following files:

- [include/local.h](#)
- [source/local.cpp](#)



## 9.15 logger Class Reference

`#include <logger.h>`Inheritance diagram for `logger::`



### Public Member Functions

- `logger ()`
- `~logger ()`
- `int init (coeur *)`
- `int close ()`  
*Close the `base`.*
- `int createFile (stringc name, int id)`
- `int write (int id, stringc=stringc(""))`
- `int closeFile (int)`
- `bool OnEvent (const SEvent &)`

### Private Attributes

- `std::vector< std::ofstream * > * files`

### 9.15.1 Detailed Description

Definition at line 2 of file `logger.h`.

### 9.15.2 Constructor & Destructor Documentation

#### 9.15.2.1 `logger::logger ()`

Definition at line 18 of file `logger.cpp`.

References `files`.

#### 9.15.2.2 `logger::~~logger ()`

Definition at line 23 of file `logger.cpp`.

References `files`.

### 9.15.3 Member Function Documentation

#### 9.15.3.1 `int logger::close ()` `[virtual]`

Close the `base`. Close the log and put `base::initialize` to false

**Returns:**

`logger::closeFile()`

Reimplemented from `base`.

Definition at line 34 of file `logger.cpp`.

Referenced by `coeur::close()`.

#### 9.15.3.2 `int logger::closeFile (int idLog)`

Definition at line 69 of file `logger.cpp`.

References `files`, `base::log()`, and `write()`.

Referenced by `base::close()`.

#### 9.15.3.3 `int logger::createFile (stringc name, int id)`

Definition at line 40 of file `logger.cpp`.

References `files`, `base::log()`, and `write()`.

Referenced by `base::init()`.

#### 9.15.3.4 `int logger::init (coeur * c)`

Definition at line 28 of file `logger.cpp`.

Referenced by `coeur::init()`.

#### 9.15.3.5 `bool logger::OnEvent (const SEvent & event)`

Definition at line 80 of file `logger.cpp`.

#### 9.15.3.6 `int logger::write (int id, stringc text = stringc (""))`

Definition at line 61 of file `logger.cpp`.

References `files`, `coeur::getTime()`, and `base::mycore`.

Referenced by `closeFile()`, `createFile()`, and `base::log()`.

### 9.15.4 Member Data Documentation

#### 9.15.4.1 `std::vector<std::ofstream*> logger::files` `[private]`

Definition at line 17 of file `logger.h`.

Referenced by `closeFile()`, `createFile()`, `logger()`, `write()`, and `~logger()`.

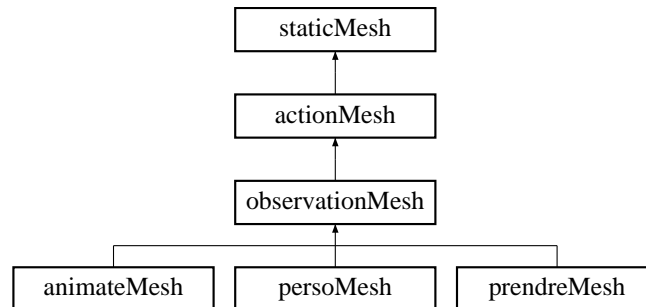
The documentation for this class was generated from the following files:

- [include/logger.h](#)
- [source/logger.cpp](#)

## 9.16 observationMesh Class Reference

Class containing a mesh who can be looked at, derive from [actionMesh](#).

`#include <actionMesh.h>`Inheritance diagram for `observationMesh::`



### Public Member Functions

- [observationMesh](#) ([ACTION\\_TYPE](#)=[ACTION\\_OBSERVER](#))

*Constructor.*

- [~observationMesh](#) ()

*Destructor.*

- virtual int [init](#) (int, IAnimatedMeshSceneNode \*, int)

*Init the mesh.*

### 9.16.1 Detailed Description

Class containing a mesh who can be looked at, derive from [actionMesh](#). It's really like an [actionMesh](#), the only difference is that it can be looked at. The action of looking at should be launched when pressing mouse right button

Definition at line 83 of file `actionMesh.h`.

### 9.16.2 Constructor & Destructor Documentation

#### 9.16.2.1 `observationMesh::observationMesh (ACTION_TYPE Action = ACTION_OBSERVER)`

Constructor. Construct an [observationMesh](#) and set [actionMesh::actionMesh\(\)](#) with parameter Action

#### Parameters:

*Action* The action type

Definition at line 86 of file `actionMesh.cpp`.

References `actionMesh::color`.

### 9.16.2.2 observationMesh::~~observationMesh ()

Destructor.

Definition at line 91 of file actionMesh.cpp.

## 9.16.3 Member Function Documentation

### 9.16.3.1 int observationMesh::init (int *Id*, IAnimatedMeshSceneNode \* *Node*, int *Combi*) [virtual]

Init the mesh. Call the [actionMesh::init](#)

#### Parameters:

*Id* Id of mesh

*Node* The pointer to the irrlicht mesh corresponding

*Combi* Item to combine with

#### Returns:

[staticMesh::init](#) result

Reimplemented from [actionMesh](#).

Definition at line 101 of file actionMesh.cpp.

Referenced by `lieu::addObservationMesh()`.

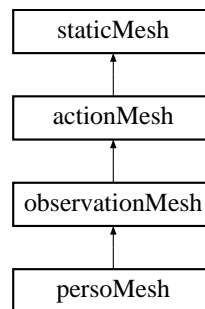
The documentation for this class was generated from the following files:

- include/[actionMesh.h](#)
- source/[actionMesh.cpp](#)

## 9.17 persoMesh Class Reference

Class containing a mesh representing a person, what mean you can speak to it.

`#include <actionMesh.h>`Inheritance diagram for persoMesh::



### Public Member Functions

- [persoMesh](#) ([ACTION\\_TYPE](#)=[ACTION\\_PERSO](#))
- [~persoMesh](#) ()
- virtual int [init](#) (int, IAnimatedMeshSceneNode \*, int, int)
- int [getAnim](#) ()
- int [getBegin](#) ()
- int [getEnd](#) ()

### Private Attributes

- int [anim](#)
- int [begin](#)
- int [end](#)

#### 9.17.1 Detailed Description

Class containing a mesh representing a person, what mean you can speak to it.

##### Todo

Implement and make working this class

Definition at line 154 of file `actionMesh.h`.

## 9.17.2 Constructor & Destructor Documentation

9.17.2.1 `persoMesh::persoMesh (ACTION_TYPE = ACTION_PERSO)`

9.17.2.2 `persoMesh::~~persoMesh ()`

## 9.17.3 Member Function Documentation

9.17.3.1 `int persoMesh::getAnim ()`

9.17.3.2 `int persoMesh::getBegin ()`

9.17.3.3 `int persoMesh::getEnd ()`

9.17.3.4 `virtual int persoMesh::init (int, IAnimatedMeshSceneNode *, int, int) [virtual]`

## 9.17.4 Member Data Documentation

9.17.4.1 `int persoMesh::anim [private]`

Definition at line 166 of file `actionMesh.h`.

9.17.4.2 `int persoMesh::begin [private]`

Definition at line 167 of file `actionMesh.h`.

9.17.4.3 `int persoMesh::end [private]`

Definition at line 168 of file `actionMesh.h`.

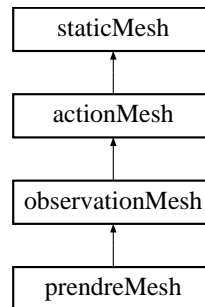
The documentation for this class was generated from the following file:

- [include/actionMesh.h](#)

## 9.18 prendreMesh Class Reference

Class containing a mesh who can be taken, derive from [observationMesh](#).

`#include <actionMesh.h>`Inheritance diagram for `prendreMesh::`



### Public Member Functions

- [prendreMesh](#) ([ACTION\\_TYPE](#)=[ACTION\\_PRENDRE](#))  
*Constructor.*
- [~prendreMesh](#) ()  
*Destructor.*
- virtual int [init](#) (int, IAnimatedMeshSceneNode \*, int, int)  
*Init the mesh.*
- int [getItem](#) ()  
*Return the [item](#) id corresponding to the mesh.*

### Private Attributes

- int [it](#)  
*Item id corresponding to the mesh.*

#### 9.18.1 Detailed Description

Class containing a mesh who can be taken, derive from [observationMesh](#). This mesh is linked to the [item](#) of id [prendreMesh::it](#), when action launched, the user should get the [item](#). And since it's derive from an [observationMesh](#) it can be looked at too.

Definition at line 101 of file `actionMesh.h`.



## 9.18.2 Constructor & Destructor Documentation

### 9.18.2.1 prendreMesh::prendreMesh (ACTION\_TYPE *act* = ACTION\_PRENDRE)

Constructor. Construct a [prendreMesh](#) and call [observationMesh::observationMesh\(\)](#) with parameter *Action*

#### Parameters:

*act* The action type

Definition at line 117 of file `actionMesh.cpp`.

References `actionMesh::color`.

### 9.18.2.2 prendreMesh::~~prendreMesh ()

Destructor.

Definition at line 122 of file `actionMesh.cpp`.

## 9.18.3 Member Function Documentation

### 9.18.3.1 int prendreMesh::getItem ()

Return the [item](#) id corresponding to the mesh.

Definition at line 139 of file `actionMesh.cpp`.

References `it`.

Referenced by `BOOST_PYTHON_MODULE()`.

### 9.18.3.2 int prendreMesh::init (int *Id*, IAnimatedMeshSceneNode \* *Node*, int *Combi*, int *It*) [virtual]

Init the mesh. Call the [actionMesh::init](#), set [prendreMesh::it](#) to `ite`

#### Parameters:

*Id* Id of mesh

*Node* The pointer to the irrlicht mesh corresponding

*Combi* Item to combine with

*It* Item associate to the mesh

#### Returns:

[observationMesh::init](#) result

Definition at line 133 of file `actionMesh.cpp`.

References `it`.

Referenced by `lieu::addPrendreMesh()`.

## 9.18.4 Member Data Documentation

### 9.18.4.1 `int prendreMesh::it` `[private]`

Item id corresponding to the mesh.

Definition at line 117 of file `actionMesh.h`.

Referenced by `getItem()`, and `init()`.

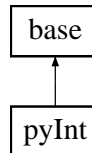
The documentation for this class was generated from the following files:

- [include/actionMesh.h](#)
- [source/actionMesh.cpp](#)

## 9.19 pyInt Class Reference

Class containing all the stuff to use python.

#include <pyInt.h> Inheritance diagram for pyInt::



### Public Member Functions

- `pyInt ()`  
*Constructor.*
- `~pyInt ()`  
*Destructor.*
- `int init (coeur *)`  
*Init the [interface](#).*
- `int update ()`  
*Updating(should be call on each loop).*
- `int close ()`  
*Close the [base](#).*
- `bool OnEvent (const SEvent &)`  
*Called when an event occured.*
- `stringw eval (stringw)`  
*Run a python command.*
- `int exec (stringw)`  
*Run a python script.*
- `int addPyEnv (stringc, object)`  
*Add a variable to python environnement.*

### Private Attributes

- object `dict`  
*Dictionnary for python wrapper.*
- object `main_module`  
*Main module for python wrapper.*

### 9.19.1 Detailed Description

Class containing all the stuff to use python. It will create the python embedded system and make it accesible for other class in order to launch instruction, script and to get the output

Definition at line 27 of file pyInt.h.

### 9.19.2 Constructor & Destructor Documentation

#### 9.19.2.1 `pyInt::pyInt ()`

Constructor. Construct the `pyInt` class, call `base::base()`.

Definition at line 36 of file pyInt.cpp.

#### 9.19.2.2 `pyInt::~~pyInt ()`

Destructor. Destruct the `pyInt` class.

Definition at line 42 of file pyInt.cpp.

### 9.19.3 Member Function Documentation

#### 9.19.3.1 `int pyInt::addPyEnv (stringc s, object o)`

Add a variable to python environnement. This method add a variable to the python environnement

##### Parameters:

- s* Name of the variable
- o* The object to put in the variable

##### Returns:

0

Definition at line 138 of file pyInt.cpp.

References `main_module`.

Referenced by `init()`, and `console::init()`.

#### 9.19.3.2 `int pyInt::close () [virtual]`

Close the `base`. Quit python and call `base::close()`

##### Returns:

`base::close()`

Reimplemented from `base`.

Definition at line 79 of file pyInt.cpp.

Referenced by `coeur::close()`.

### 9.19.3.3 stringw pyInt::eval (stringw *str*)

Run a python command. This method run a python command

**Parameters:**

*str* command to run

**Returns:**

the result of the command

Definition at line 97 of file pyInt.cpp.

References dict, exec(), and base::log().

Referenced by console::OnEvent().

### 9.19.3.4 int pyInt::exec (stringw *str*)

Run a python script. This method run a python script

**Parameters:**

*str* script to run

**Returns:**

0 if succesful, -1 instead

Definition at line 119 of file pyInt.cpp.

References dict, and base::log().

Referenced by console::close(), eval(), init(), and console::init().

### 9.19.3.5 int pyInt::init (coeur \* *c*)

Init the [interface](#). Call [base::init\(\)](#) and initialize python

**Parameters:**

*c* Pointer to the core

**Returns:**

0 if succesful, -1 instead

Definition at line 50 of file pyInt.cpp.

References addPyEnv(), dict, exec(), initpyIrr(), initpyMerry(), and main\_module.

Referenced by coeur::init().

### 9.19.3.6 bool pyInt::OnEvent (const SEvent & *event*)

Called when an event occured. This method is called when an event has to be treated, but this class do not wait event so this function does nothing

**Parameters:**

*event* The event to parse

**Returns:**

Always false

Definition at line 89 of file pyInt.cpp.

**9.19.3.7 int pyInt::update () [virtual]**

Updating(should be call on each loop). Does nothing, apart from slowing the game

**Returns:**

always 0

Reimplemented from [base](#).

Definition at line 72 of file pyInt.cpp.

**9.19.4 Member Data Documentation****9.19.4.1 object pyInt::dict [private]**

Dictionary for python wrapper.

Definition at line 58 of file pyInt.h.

Referenced by eval(), exec(), and init().

**9.19.4.2 object pyInt::main\_module [private]**

Main module for python wrapper.

Definition at line 61 of file pyInt.h.

Referenced by addPyEnv(), and init().

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

- [include/pyInt.h](#)
- [source/pyInt.cpp](#)

## 9.20 startmenu Class Reference

```
#include <startmenu.h>
```

### Public Member Functions

- [startmenu](#) ()
- [~startmenu](#) ()
- int [init](#) (stringc, stringc)
- int [update](#) ()
- int [close](#) ()
- int [progress](#) ()
- position2d< s32 > [getCurPos](#) ()
- u32 [getTime](#) ()
- stringc [getVersion](#) ()
- int [setCaption](#) (stringw)
- virtual bool [OnEvent](#) (const SEvent &)
- IFileSystem \* [getFileSystem](#) ()

### Private Member Functions

- int [log](#) (stringc)
- int [startCoeur](#) ()
- int [loadDesc](#) ()
- int [saveConfig](#) ()

### Private Attributes

- IrrlichtDevice \* [device](#)
- IVideoDriver \* [driver](#)
- ISceneManager \* [smgr](#)
- IGUIEnvironment \* [env](#)
- E\_DRIVER\_TYPE [dr](#)
- dimension2d< s32 > [dim](#)
- int [bits](#)
- bool [fullscreen](#)
- bool [stencilbuffer](#)
- bool [vsync](#)
- int [lang](#)
- stringc [langMenu](#)
- bool [q](#)
- position2d< s32 > [curPos](#)
- [coeur](#) \* [c](#)
- stringc [path](#)
- stringc [pathConfig](#)
- stringc [pathStart](#)
- stringc [pathMain](#)
- bool [start](#)
- std::vector< IGUIButton \* > \* [menus](#)

- `std::vector< ITexture * > * menusOut`
- `std::vector< ITexture * > * menusIn`
- `IGUIImage * welcome`
- `IGUIImage * game`
- `IGUIImage * options`
- `IGUIImage * starting`
- `IGUIEditBox * game_path`
- `IGUIImage * game_img`
- `IGUIStaticText * game_name`
- `IGUIStaticText * game_text`
- `IGUIComboBox * options_dv`
- `IGUIComboBox * options_res`
- `IGUICheckBox * options_fs`
- `IGUIComboBox * options_lang`
- `IGUIStaticText * starting_text`
- `IGUIImage * starting_bar`
- `std::vector< stringw > * starting_txt`
- `unsigned int progressing`

### 9.20.1 Detailed Description

Definition at line 3 of file `startmenu.h`.

### 9.20.2 Constructor & Destructor Documentation

#### 9.20.2.1 `startmenu::startmenu ()`

Definition at line 22 of file `startmenu.cpp`.

References `bits`, `dim`, `dr`, `fullscreen`, `lang`, `menus`, `menusIn`, `menusOut`, `progressing`, `starting_txt`, `stencil-buffer`, and `vsync`.

#### 9.20.2.2 `startmenu::~~startmenu ()`

Definition at line 41 of file `startmenu.cpp`.

References `menus`, `menusIn`, `menusOut`, and `starting_txt`.

### 9.20.3 Member Function Documentation

#### 9.20.3.1 `int startmenu::close ()`

Definition at line 362 of file `startmenu.cpp`.

References `device`, and `log()`.

Referenced by `startCoeur()`, and `update()`.



### 9.20.3.2 position2d< s32 > startmenu::getCurPos ()

Definition at line 374 of file startmenu.cpp.

References curPos.

### 9.20.3.3 IFileSystem \* startmenu::getFileSystem ()

Definition at line 625 of file startmenu.cpp.

References device.

Referenced by coeur::init().

### 9.20.3.4 u32 startmenu::getTime ()

Definition at line 379 of file startmenu.cpp.

References device.

### 9.20.3.5 stringc startmenu::getVersion ()

Definition at line 480 of file startmenu.cpp.

### 9.20.3.6 int startmenu::init (stringc *p*, stringc *la*)

Definition at line 49 of file startmenu.cpp.

References bits, device, dim, dr, driver, env, fullscreen, game, game\_img, game\_name, game\_path, game\_text, lang, langMenu, log(), menus, menusIn, menusOut, options, options\_dv, options\_fs, options\_lang, options\_res, path, pathConfig, pathMain, pathStart, smgr, start, starting, starting\_bar, starting\_text, starting\_txt, stencilbuffer, vsync, and welcome.

Referenced by main().

### 9.20.3.7 int startmenu::loadDesc () [private]

Definition at line 526 of file startmenu.cpp.

References device, driver, game\_img, game\_name, game\_path, game\_text, langMenu, and pathMain.

Referenced by OnEvent().

### 9.20.3.8 int startmenu::log (stringc *text*) [private]

Definition at line 520 of file startmenu.cpp.

Referenced by close(), init(), progress(), setCaption(), startCoeur(), and update().

### 9.20.3.9 bool startmenu::OnEvent (const SEvent & *event*) [virtual]

Definition at line 384 of file startmenu.cpp.

References device, dim, dr, env, fullscreen, game, game\_path, lang, loadDesc(), options, options\_dv, options\_fs, options\_lang, options\_res, path, q, saveConfig(), start, and welcome.

#### 9.20.3.10 int startmenu::progress ()

Definition at line 608 of file startmenu.cpp.

References log(), progressing, starting\_bar, starting\_text, starting\_txt, and update().

Referenced by coeur::init().

#### 9.20.3.11 int startmenu::saveConfig () [private]

Definition at line 566 of file startmenu.cpp.

References bits, device, dim, dr, fullscreen, lang, pathConfig, pathMain, pathStart, stencilbuffer, and vsync.

Referenced by OnEvent().

#### 9.20.3.12 int startmenu::setCaption (stringw *caption*)

Definition at line 485 of file startmenu.cpp.

References device, and log().

#### 9.20.3.13 int startmenu::startCoeur () [private]

Definition at line 492 of file startmenu.cpp.

References c, coeur::close(), close(), game, coeur::init(), log(), options, starting, coeur::update(), and welcome.

Referenced by update().

#### 9.20.3.14 int startmenu::update ()

Definition at line 325 of file startmenu.cpp.

References close(), curPos, device, driver, env, log(), menus, menusIn, menusOut, q, start, and startCoeur().

Referenced by main(), and progress().

### 9.20.4 Member Data Documentation

#### 9.20.4.1 int startmenu::bits [private]

Definition at line 43 of file startmenu.h.

Referenced by init(), saveConfig(), and startmenu().

#### 9.20.4.2 coeur\* startmenu::c [private]

Definition at line 55 of file startmenu.h.

Referenced by startCoeur().

#### 9.20.4.3 position2d<s32> startmenu::curPos [private]

Definition at line 53 of file startmenu.h.

Referenced by getCurPos(), and update().

#### 9.20.4.4 IrrlichtDevice\* startmenu::device [private]

Definition at line 36 of file startmenu.h.

Referenced by close(), getFileSystem(), getTime(), init(), loadDesc(), OnEvent(), saveConfig(), setCaption(), and update().

#### 9.20.4.5 dimension2d<s32> startmenu::dim [private]

Definition at line 42 of file startmenu.h.

Referenced by init(), OnEvent(), saveConfig(), and startmenu().

#### 9.20.4.6 E\_DRIVER\_TYPE startmenu::dr [private]

Definition at line 41 of file startmenu.h.

Referenced by init(), OnEvent(), saveConfig(), and startmenu().

#### 9.20.4.7 IVideoDriver\* startmenu::driver [private]

Definition at line 37 of file startmenu.h.

Referenced by init(), loadDesc(), and update().

#### 9.20.4.8 IGUIEnvironment\* startmenu::env [private]

Definition at line 39 of file startmenu.h.

Referenced by init(), OnEvent(), and update().

#### 9.20.4.9 bool startmenu::fullscreen [private]

Definition at line 44 of file startmenu.h.

Referenced by init(), OnEvent(), saveConfig(), and startmenu().

#### 9.20.4.10 IGUIImage\* startmenu::game [private]

Definition at line 69 of file startmenu.h.

Referenced by init(), OnEvent(), and startCoeur().

**9.20.4.11 IGUIImage\* startmenu::game\_img [private]**

Definition at line 74 of file startmenu.h.

Referenced by init(), and loadDesc().

**9.20.4.12 IGUIStaticText\* startmenu::game\_name [private]**

Definition at line 75 of file startmenu.h.

Referenced by init(), and loadDesc().

**9.20.4.13 IGUIEditBox\* startmenu::game\_path [private]**

Definition at line 73 of file startmenu.h.

Referenced by init(), loadDesc(), and OnEvent().

**9.20.4.14 IGUIStaticText\* startmenu::game\_text [private]**

Definition at line 76 of file startmenu.h.

Referenced by init(), and loadDesc().

**9.20.4.15 int startmenu::lang [private]**

Definition at line 47 of file startmenu.h.

Referenced by init(), OnEvent(), saveConfig(), and startmenu().

**9.20.4.16 stringc startmenu::langMenu [private]**

Definition at line 49 of file startmenu.h.

Referenced by init(), and loadDesc().

**9.20.4.17 std::vector<IGUIButton\*> startmenu::menus [private]**

Definition at line 64 of file startmenu.h.

Referenced by init(), startmenu(), update(), and ~startmenu().

**9.20.4.18 std::vector<ITexture\*> startmenu::menusIn [private]**

Definition at line 66 of file startmenu.h.

Referenced by init(), startmenu(), update(), and ~startmenu().

**9.20.4.19 std::vector<ITexture\*> startmenu::menusOut [private]**

Definition at line 65 of file startmenu.h.

Referenced by init(), startmenu(), update(), and ~startmenu().

**9.20.4.20 IGUIImage\* startmenu::options [private]**

Definition at line 70 of file startmenu.h.

Referenced by init(), OnEvent(), and startCoeur().

**9.20.4.21 IGUIComboBox\* startmenu::options\_dv [private]**

Definition at line 78 of file startmenu.h.

Referenced by init(), and OnEvent().

**9.20.4.22 IGUICheckBox\* startmenu::options\_fs [private]**

Definition at line 80 of file startmenu.h.

Referenced by init(), and OnEvent().

**9.20.4.23 IGUIComboBox\* startmenu::options\_lang [private]**

Definition at line 81 of file startmenu.h.

Referenced by init(), and OnEvent().

**9.20.4.24 IGUIComboBox\* startmenu::options\_res [private]**

Definition at line 79 of file startmenu.h.

Referenced by init(), and OnEvent().

**9.20.4.25 stringc startmenu::path [private]**

Definition at line 57 of file startmenu.h.

Referenced by init(), and OnEvent().

**9.20.4.26 stringc startmenu::pathConfig [private]**

Definition at line 58 of file startmenu.h.

Referenced by init(), and saveConfig().

**9.20.4.27 stringc startmenu::pathMain [private]**

Definition at line 60 of file startmenu.h.

Referenced by init(), loadDesc(), and saveConfig().

**9.20.4.28 stringc startmenu::pathStart [private]**

Definition at line 59 of file startmenu.h.

Referenced by init(), and saveConfig().

**9.20.4.29 unsigned int startmenu::progressing [private]**

Definition at line 86 of file startmenu.h.

Referenced by progress(), and startmenu().

**9.20.4.30 bool startmenu::q [private]**

Definition at line 51 of file startmenu.h.

Referenced by OnEvent(), and update().

**9.20.4.31 ISceneManager\* startmenu::smgr [private]**

Definition at line 38 of file startmenu.h.

Referenced by init().

**9.20.4.32 bool startmenu::start [private]**

Definition at line 62 of file startmenu.h.

Referenced by init(), OnEvent(), and update().

**9.20.4.33 IGUIImage\* startmenu::starting [private]**

Definition at line 71 of file startmenu.h.

Referenced by init(), and startCoeur().

**9.20.4.34 IGUIImage\* startmenu::starting\_bar [private]**

Definition at line 84 of file startmenu.h.

Referenced by init(), and progress().

**9.20.4.35 IGUIStaticText\* startmenu::starting\_text [private]**

Definition at line 83 of file startmenu.h.

Referenced by init(), and progress().

**9.20.4.36 std::vector<stringw>\* startmenu::starting\_txt [private]**

Definition at line 85 of file startmenu.h.

Referenced by init(), progress(), startmenu(), and ~startmenu().

**9.20.4.37 bool startmenu::stencilbuffer [private]**

Definition at line 45 of file startmenu.h.

Referenced by init(), saveConfig(), and startmenu().

**9.20.4.38 bool startmenu::vsync [private]**

Definition at line 46 of file startmenu.h.

Referenced by `init()`, `saveConfig()`, and `startmenu()`.

**9.20.4.39 IGUIImage\* startmenu::welcome [private]**

Definition at line 68 of file startmenu.h.

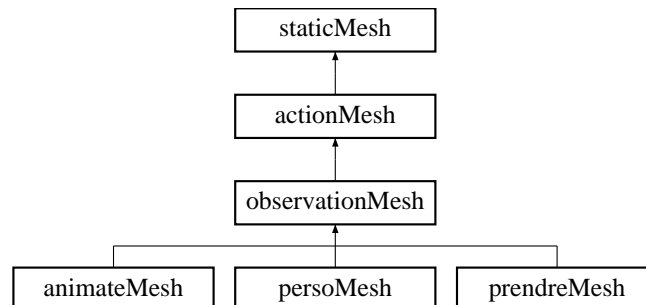
Referenced by `init()`, `OnEvent()`, and `startCoeur()`.

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

- [include/startmenu.h](#)
- [source/startmenu.cpp](#)

## 9.21 staticMesh Class Reference

`#include <staticMesh.h>`Inheritance diagram for staticMesh::



### Public Member Functions

- [staticMesh \(\)](#)
- [~staticMesh \(\)](#)
- virtual int [init](#) (int, IAnimatedMeshSceneNode \*)
- virtual int [update](#) ()
- virtual int [close](#) ()
- IAnimatedMeshSceneNode \* [getNode](#) ()
- int [getId](#) ()
- int [setVisible](#) (bool)

### Private Attributes

- IAnimatedMeshSceneNode \* [node](#)
- int [id](#)

### 9.21.1 Detailed Description

Definition at line 2 of file staticMesh.h.

### 9.21.2 Constructor & Destructor Documentation

#### 9.21.2.1 staticMesh::staticMesh ()

Definition at line 3 of file staticMesh.cpp.

#### 9.21.2.2 staticMesh::~~staticMesh ()

Definition at line 8 of file staticMesh.cpp.



### 9.21.3 Member Function Documentation

#### 9.21.3.1 int staticMesh::close () [virtual]

Definition at line 26 of file staticMesh.cpp.

#### 9.21.3.2 int staticMesh::getId ()

Definition at line 36 of file staticMesh.cpp.

References id.

Referenced by BOOST\_PYTHON\_MODULE(), actionMesh::init(), and interface::OnEvent().

#### 9.21.3.3 IAnimatedMeshSceneNode \* staticMesh::getNode ()

Definition at line 31 of file staticMesh.cpp.

References node.

Referenced by lieu::addAnimateMesh(), lieu::addObservationMesh(), lieu::addPrendreMesh(), lieu::addStaticMesh(), coeur::anim(), lieu::init(), actionMesh::init(), interface::OnEvent(), and interface::update().

#### 9.21.3.4 int staticMesh::init (int *i*, IAnimatedMeshSceneNode \* *n*) [virtual]

Definition at line 13 of file staticMesh.cpp.

References node.

Referenced by lieu::addStaticMesh().

#### 9.21.3.5 int staticMesh::setVisible (bool *v*)

Definition at line 41 of file staticMesh.cpp.

References node.

Referenced by BOOST\_PYTHON\_MODULE().

#### 9.21.3.6 int staticMesh::update () [virtual]

Definition at line 21 of file staticMesh.cpp.

### 9.21.4 Member Data Documentation

#### 9.21.4.1 int staticMesh::id [private]

Definition at line 18 of file staticMesh.h.

Referenced by getId().

#### 9.21.4.2 IAnimatedMeshSceneNode\* staticMesh::node [private]

Definition at line 17 of file staticMesh.h.

Referenced by getNode(), init(), and setVisible().

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

- [include/staticMesh.h](#)
- [source/staticMesh.cpp](#)

# Chapter 10

## File Documentation

### 10.1 doc/example/chap.xml File Reference

#### 10.1.1 Detailed Description

#### 10.1.2 Description

A chapter contain all the item that the user can handle in his inventory. For example : paper, pen, keys, magazine, tools, books, ladder... The [item](#) are common to all location, that's why they are in the chapter.

The last thing a chapter handle is the first location you are when begin the chapter(the changing of location are handled by other way)

#### 10.1.3 Xml

So now the xml,

I assume that you know how general xml are formatted(tags system like html)

The xml should begin with a line like that :

```
<?xml version="1.0"?>
```

Next there is three types of element possible :

- *chapter* : this is the main element it takes as attribute :
  - *nbr* : the number of [item](#) in the chapter

```
<chapter nbr="2">
```

- *item* : create an element it takes as attribute :
  - *id* : the id of the [item](#)
  - *texture* : the texture path of the [item](#)
  - *combi* the : id of the [item](#) it can be combine with(-1 if none)
  - *combiResult* : the result of a combinaison(-1 if none)

```
<item id="1" texture="icon/ladder.png" combi="-1" combiResult="-1">
```

- *lieu* : reference to the first location, it takes as attribute :
  - *id* : the id of the [lieu](#)

```
<lieu id="0"/>
```

### 10.1.4 Example

To see a complete example just look at the source code of this file

The example should create 2 items : a key and a ladder(according to the texture name)

And the chapter should begin at location 3

Definition in file [chap.xml](#).

## 10.2 include/actionMesh.h File Reference

Definition of [actionMesh](#) class and its derivate class.

### Classes

- class [actionMesh](#)  
*Class containing a mesh who can launched an action when it is clicked, derive from [staticMesh](#).*
- class [observationMesh](#)  
*Class containing a mesh who can be looked at, derive from [actionMesh](#).*
- class [prendreMesh](#)  
*Class containing a mesh who can be taken, derive from [observationMesh](#).*
- class [animateMesh](#)  
*Class containing a mesh who can launched an animation, derive from [observationMesh](#).*
- class [persoMesh](#)  
*Class containing a mesh representing a person, what mean you can speak to it.*

### Enumerations

- enum [ACTION\\_TYPE](#) {  
    [ACTION\\_AUCUNE](#) = 0, [ACTION\\_OBSERVER](#), [ACTION\\_PRENDRE](#), [ACTION\\_ANIMATE](#),  
    [ACTION\\_PERSO](#) }  
*Enum all the [actionMesh](#) type.*

#### 10.2.1 Detailed Description

Definition of [actionMesh](#) class and its derivate class. Contain [actionMesh](#), [observationMesh](#), [prendreMesh](#), [animateMesh](#) definitions.

Definition in file [actionMesh.h](#).

#### 10.2.2 Enumeration Type Documentation

##### 10.2.2.1 enum ACTION\_TYPE

Enum all the [actionMesh](#) type.

##### Enumerator:

**[ACTION\\_AUCUNE](#)** The mesh has no action.

**[ACTION\\_OBSERVER](#)** The mesh can be looked at.

**[ACTION\\_PRENDRE](#)** The mesh can be taken.

***ACTION\_ANIMATE*** The mesh can launch an animation.

***ACTION\_PERSO*** The mesh is a perso you can speak to.

Definition at line 29 of file actionMesh.h.

## 10.3 include/base.h File Reference

Definition of [base](#) class.

### Classes

- class [base](#)

*Class from whom most of the other class derive from, derive from IEventReceiver.*

### 10.3.1 Detailed Description

Definition of [base](#) class.

Definition in file [base.h](#).

## 10.4 include/chap.h File Reference

Definition of class [chap](#).

### Classes

- class [chap](#)

*Class containing all the information about a chapter, derive from [base](#).*

### 10.4.1 Detailed Description

Definition of class [chap](#).

Definition in file [chap.h](#).



## 10.5 include/coeur.h File Reference

Definition of class [chap](#).

### Classes

- class [coeur](#)

*Main class containing all important call to the irrlicht and a lot of utilities, derive from [base](#).*

### 10.5.1 Detailed Description

Definition of class [chap](#).

Definition in file [coeur.h](#).

## 10.6 include/console.h File Reference

Definition of class [console](#).

### Classes

- class [console](#)

*This class handle all the suff of the [console](#).*

### 10.6.1 Detailed Description

Definition of class [console](#).

Definition in file [console.h](#).

## 10.7 include/credit.h File Reference

### Classes

- class [credit](#)

## 10.8 include/game.h File Reference

Main header file. #include <vector>

#include <iostream>

#include <fstream>

#include <string>

#include <irrlicht.h>

#include <boost/python.hpp>

#include <boost/python/class.hpp>

#include <boost/python/module.hpp>

#include <boost/python/def.hpp>

#include <base.h>

#include <logger.h>

#include <pyInt.h>

#include <item.h>

#include <staticMesh.h>

#include <actionMesh.h>

#include <local.h>

#include <inventaire.h>

#include <interface.h>

#include <console.h>

#include <lieu.h>

#include <chap.h>

#include <credit.h>

#include <coeur.h>

### 10.8.1 Detailed Description

Main header file. Contain all the include to the headers of:

STL : vector

irrlicht : irrlicht.h

irrklang : irrklang.h

The Merry Turnip : [base.h](#), [logger.h](#), [item.h](#), [staticMesh.h](#), [actionMesh.h](#), [local.h](#), [inventaire.h](#), [interface.h](#), [console.h](#), [lieu.h](#), [chap.h](#), [credit.h](#), [coeur.h](#)

Definition in file [game.h](#).

## 10.9 include/interface.h File Reference

### Classes

- class [interface](#)

## 10.10 include/inventaire.h File Reference

### Classes

- class [inventaire](#)

### Enumerations

- enum [INVENTAIRE\\_POS](#) { [INVENTAIRE\\_POS\\_GAUCHE](#) = 0, [INVENTAIRE\\_POS\\_DROITE](#), [INVENTAIRE\\_POS\\_BAS](#), [INVENTAIRE\\_POS\\_HAUT](#) }

### 10.10.1 Enumeration Type Documentation

#### 10.10.1.1 enum INVENTAIRE\_POS

##### Enumerator:

*INVENTAIRE\_POS\_GAUCHE*

*INVENTAIRE\_POS\_DROITE*

*INVENTAIRE\_POS\_BAS*

*INVENTAIRE\_POS\_HAUT*

Definition at line 2 of file `inventaire.h`.

## 10.11 include/item.h File Reference

### Classes

- class [item](#)

## 10.12 include/lieu.h File Reference

### Classes

- class [lieu](#)



## 10.13 include/local.h File Reference

### Classes

- class [localItem](#)
- class [local](#)

## 10.14 include/logger.h File Reference

### Classes

- class [logger](#)

## 10.15 include/pyInt.h File Reference

Definition of class [pyInt](#).

### Classes

- class [pyInt](#)  
*Class containing all the stuff to use python.*

### 10.15.1 Detailed Description

Definition of class [pyInt](#).

Definition in file [pyInt.h](#).

## 10.16 include/startmenu.h File Reference

### Classes

- class [startmenu](#)

## 10.17 include/staticMesh.h File Reference

### Classes

- class [staticMesh](#)

## 10.18 source/actionMesh.cpp File Reference

Implement [actionMesh](#) class and its derivate class. `#include <game.h>`

### 10.18.1 Detailed Description

Implement [actionMesh](#) class and its derivate class. Contain [actionMesh](#), [observationMesh](#), [prendreMesh](#), [animateMesh](#) implementations.

Definition in file [actionMesh.cpp](#).

## 10.19 source/base.cpp File Reference

Implement [base](#) class. `#include <game.h>`

### 10.19.1 Detailed Description

Implement [base](#) class.

Definition in file [base.cpp](#).

## 10.20 source/chap.cpp File Reference

Implement [chap](#) class. `#include <game.h>`

### 10.20.1 Detailed Description

Implement [chap](#) class.

Definition in file [chap.cpp](#).



## 10.21 source/coeur.cpp File Reference

Implement [coeur](#) class. `#include <game.h>`  
`#include <startmenu.h>`

### 10.21.1 Detailed Description

Implement [coeur](#) class. Contain all member of class [coeur](#).

#### Todo

- Remplacer les vector path par des listes indexé de la STL
- Creer repertoire log
- Use the event system to launch personal event(like changing [chap](#), animation, script,...)
- Add in the xml a line to say the path of the texture folder and ad it to irrlicht filesytem

#### Bug

- Not respond when launched fullscreen

Definition in file [coeur.cpp](#).

## 10.22 source/console.cpp File Reference

```
#include <game.h>
#include <iostream>
```

## 10.23 source/credit.cpp File Reference

```
#include <game.h>
```

## 10.24 source/interface.cpp File Reference

```
#include <game.h>
```

## 10.25 source/inventaire.cpp File Reference

```
#include <game.h>
```

## 10.26 source/item.cpp File Reference

```
#include <game.h>
```

## 10.27 source/lieu.cpp File Reference

```
#include <game.h>
```

## 10.28 source/local.cpp File Reference

```
#include <game.h>
```



## 10.29 source/logger.cpp File Reference

```
#include <game.h>
#include <iostream>
#include <fstream>
#include <string>
#include <sys/types.h>
#include <sys/stat.h>
```

## 10.30 source/main.cpp File Reference

```
#include <game.h>
#include <startmenu.h>
#include <iostream>
#include <unistd.h>
```

### Functions

- `int main (int argc, char *argv[])`  
*/todo Choose the language by system config*

#### 10.30.1 Function Documentation

##### 10.30.1.1 `int main (int argc, char * argv[])`

*/todo* Choose the language by system config

Definition at line 8 of file main.cpp.

References `startmenu::init()`, and `startmenu::update()`.

## 10.31 source/modPyIrr.cpp File Reference

Create the python module who wrap the irrlicht needed class. `#include <game.h>`

### Classes

- struct `cusString_python< cusString >`

### Functions

- `stringc reprPos (position2d< s32 > *pos)`
- `stringc reprDim (dimension2d< s32 > *dim)`
- `BOOST_PYTHON_MODULE (pyIrr)`

#### 10.31.1 Detailed Description

Create the python module who wrap the irrlicht needed class.

Definition in file `modPyIrr.cpp`.

#### 10.31.2 Function Documentation

##### 10.31.2.1 BOOST\_PYTHON\_MODULE (pyIrr)

Definition at line 64 of file `modPyIrr.cpp`.

References `reprDim()`, and `reprPos()`.

##### 10.31.2.2 stringc reprDim (dimension2d< s32 > \* *dim*)

Definition at line 59 of file `modPyIrr.cpp`.

Referenced by `BOOST_PYTHON_MODULE()`.

##### 10.31.2.3 stringc reprPos (position2d< s32 > \* *pos*)

Definition at line 54 of file `modPyIrr.cpp`.

Referenced by `BOOST_PYTHON_MODULE()`.

## 10.32 source/modPyMerry.cpp File Reference

Create the python module who wrap the game. `#include <game.h>`

### Functions

- [BOOST\\_PYTHON\\_MODULE](#) (pyMerry)

### 10.32.1 Detailed Description

Create the python module who wrap the game.

#### Todo

Ajouter des repr à tous les objets pour une intégration plus belle avec python cf `class_<item>`

Definition in file [modPyMerry.cpp](#).

### 10.32.2 Function Documentation

#### 10.32.2.1 BOOST\_PYTHON\_MODULE (pyMerry)

Problem with pure virtual function OnEvent

Definition at line 27 of file modPyMerry.cpp.

References `ACTION_ANIMATE`, `ACTION_AUCUNE`, `ACTION_OBSERVER`, `ACTION_PERSO`, `ACTION_PRENDRE`, `chap::addItem()`, `interface::addItemInventaire()`, `credit::addLogo()`, `credit::addText()`, `interface::afficherInventaire()`, `console::exit()`, `console::fps()`, `actionMesh::getAction()`, `coeur::getActionMesh()`, `animateMesh::getAnim()`, `animateMesh::getBegin()`, `local::getCaption()`, `actionMesh::getCombi()`, `item::getCombi()`, `coeur::getCombinaison()`, `local::getCombinaison()`, `item::getCombiResult()`, `coeur::getDim()`, `animateMesh::getEnd()`, `localItem::getId()`, `staticMesh::getId()`, `item::getId()`, `coeur::getItem()`, `prendreMesh::getItem()`, `chap::getItemFromId()`, `coeur::getObservationItem()`, `local::getObservationItem()`, `coeur::getObservationMesh()`, `local::getObservationMesh()`, `localItem::getPath()`, `coeur::getSelectedAction()`, `localItem::getText()`, `item::getTexture()`, `coeur::getTime()`, `coeur::getVersion()`, `console::help()`, `localItem::init()`, `item::init()`, `coeur::loadChap()`, `coeur::play()`, `console::position()`, `console::quit()`, `item::repr()`, `console::rotation()`, `coeur::setCaption()`, `interface::setImage()`, `interface::setImagePosition()`, `staticMesh::setVisible()`, `credit::start()`, `coeur::startCredit()`, `credit::stop()`, `console::target()`, and `console::write()`.

## 10.33 source/pyInt.cpp File Reference

Implement [pyInt](#) class. `#include <game.h>`

`#include <iostream>`

### Functions

- void [initpyIrr](#) ()
- void [initpyMerry](#) ()

#### 10.33.1 Detailed Description

Implement [pyInt](#) class.

Definition in file [pyInt.cpp](#).

#### 10.33.2 Function Documentation

##### 10.33.2.1 void [initpyIrr](#) ()

Referenced by [pyInt::init](#)().

##### 10.33.2.2 void [initpyMerry](#) ()

Referenced by [pyInt::init](#)().

## 10.34 source/startmenu.cpp File Reference

```
#include <game.h>
#include <startmenu.h>
#include <iostream>
```

## 10.35 source/staticMesh.cpp File Reference

```
#include <game.h>
```

# Index

- ~actionMesh
  - actionMesh, [18](#)
- ~animateMesh
  - animateMesh, [22](#)
- ~base
  - base, [27](#)
- ~chap
  - chap, [31](#)
- ~coeur
  - coeur, [38](#)
- ~console
  - console, [51](#)
- ~credit
  - credit, [56](#)
- ~interface
  - interface, [63](#)
- ~inventaire
  - inventaire, [70](#)
- ~item
  - item, [74](#)
- ~lieu
  - lieu, [77](#)
- ~local
  - local, [82](#)
- ~localItem
  - localItem, [85](#)
- ~logger
  - logger, [87](#)
- ~observationMesh
  - observationMesh, [90](#)
- ~persoMesh
  - persoMesh, [93](#)
- ~prendreMesh
  - prendreMesh, [95](#)
- ~pyInt
  - pyInt, [98](#)
- ~startmenu
  - startmenu, [102](#)
- ~staticMesh
  - staticMesh, [110](#)
- action
  - actionMesh, [19](#)
- ACTION\_ANIMATE
  - actionMesh.h, [115](#)
- ACTION\_AUCUNE
  - actionMesh.h, [115](#)
- ACTION\_OBSERVER
  - actionMesh.h, [115](#)
- ACTION\_PERSO
  - actionMesh.h, [116](#)
- ACTION\_PRENDRE
  - actionMesh.h, [115](#)
- ACTION\_TYPE
  - actionMesh.h, [115](#)
- actionMesh, [17](#)
  - ~actionMesh, [18](#)
  - action, [19](#)
  - actionMesh, [18](#)
  - color, [20](#)
  - combi, [20](#)
  - getAction, [19](#)
  - getColor, [19](#)
  - getCombi, [19](#)
  - init, [19](#)
- actionMesh.h
  - ACTION\_ANIMATE, [115](#)
  - ACTION\_AUCUNE, [115](#)
  - ACTION\_OBSERVER, [115](#)
  - ACTION\_PERSO, [116](#)
  - ACTION\_PRENDRE, [115](#)
  - ACTION\_TYPE, [115](#)
- activate
  - credit, [58](#)
- activeIn
  - coeur, [44](#)
- activeScene
  - coeur, [44](#)
- addAnimateMesh
  - lieu, [78](#)
- addCombinaison
  - local, [82](#)
- addInventaire
  - interface, [63](#)
- addItem
  - chap, [31](#)
  - inventaire, [70](#)
- addItemInventaire
  - interface, [63](#)
- addLight



- coeur, 38
- addLogo
  - credit, 57
- addMesh
  - coeur, 38
- addObservationItem
  - local, 82
- addObservationMesh
  - lieu, 78
  - local, 82
- addParticleSystem
  - coeur, 39
- addPrendreMesh
  - lieu, 78
- addPyEnv
  - pyInt, 98
- addStaticMesh
  - lieu, 78
- addText
  - credit, 57
- afficher
  - inventaire, 70
- afficherInventaire
  - interface, 64
- anim
  - animateMesh, 23
  - coeur, 39
  - persoMesh, 93
- animateMesh, 21
  - ~animateMesh, 22
  - anim, 23
  - animateMesh, 22
  - begin, 23
  - end, 23
  - getAnim, 22
  - getBegin, 22
  - getEnd, 22
  - init, 23
- base, 25
  - ~base, 27
  - base, 26
  - close, 27
  - getId, 27
  - id, 28
  - init, 27
  - initialize, 28
  - isInit, 27
  - log, 28
  - logId, 29
  - mycore, 29
  - nameClass, 29
  - update, 28
- begin
  - animateMesh, 23
  - persoMesh, 93
- bits
  - coeur, 44
  - startmenu, 104
- BOOST\_PYTHON\_MODULE
  - modPyIrr.cpp, 145
  - modPyMerry.cpp, 146
- box
  - console, 54
- c
  - startmenu, 104
- cam
  - coeur, 45
- caption
  - local, 84
- chap, 30
  - ~chap, 31
  - addItem, 31
  - chap, 31
  - close, 31
  - getItemFromId, 32
  - init, 32
  - listItem, 33
  - OnEvent, 32
  - update, 33
- classic
  - interface, 66
- close
  - base, 27
  - chap, 31
  - coeur, 39
  - console, 52
  - credit, 57
  - interface, 64
  - inventaire, 70
  - lieu, 78
  - local, 82
  - logger, 88
  - pyInt, 98
  - startmenu, 102
  - staticMesh, 111
- closeFile
  - logger, 88
- co
  - coeur, 45
- coeur, 34
  - ~coeur, 38
  - activeIn, 44
  - activeScene, 44
  - addLight, 38
  - addMesh, 38
  - addParticleSystem, 39

- anim, 39
- bits, 44
- cam, 45
- close, 39
- co, 45
- coeur, 38
- console, 44
- curPos, 45
- currentChap, 45
- currentLieu, 45
- currentLocal, 45
- device, 45
- dim, 46
- dr, 46
- driver, 46
- env, 46
- fullscreen, 46
- getActionMesh, 39
- getCombinaison, 40
- getCurPos, 40
- getDim, 40
- getItem, 40
- getLogger, 40
- getObservationItem, 40
- getObservationMesh, 41
- getSelectedAction, 41
- getSelectedNode, 41
- getTexture, 41
- getTime, 41
- getVersion, 41
- in, 46
- init, 42
- lang, 46
- loadChap, 42
- loadLieu, 42
- loadLocal, 42
- metaSelector, 47
- mycredit, 47
- mylogger, 47
- OnEvent, 42
- pathChap, 47
- pathGame, 47
- pathLieu, 47
- pathLocal, 47
- pickNode, 43
- play, 43
- py, 48
- pyInt, 44
- q, 48
- selectedAction, 48
- setCaption, 43
- setSkyBox, 43
- smgr, 48
- startCredit, 43
- stencilbuffer, 48
- update, 44
- vsync, 48
- color
  - actionMesh, 20
- combi
  - actionMesh, 20
  - item, 75
- combinaison
  - local, 84
- combiResult
  - item, 75
- cons
  - console, 54
- console, 50
  - ~console, 51
  - box, 54
  - close, 52
  - coeur, 44
  - cons, 54
  - console, 51
  - device, 54
  - env, 54
  - exit, 52
  - FPS, 54
  - fps, 52
  - help, 52
  - history, 55
  - histPos, 55
  - init, 52
  - OnEvent, 53
  - position, 53
  - quit, 53
  - rotation, 53
  - target, 53
  - text, 55
  - tx, 55
  - update, 53
  - write, 54
- construct
  - cusString\_python, 60
- convert
  - cusString\_python, 60
- convertible
  - cusString\_python, 60
- createFile
  - logger, 88
- credit, 56
  - ~credit, 56
  - activate, 58
  - addLogo, 57
  - addText, 57
  - close, 57
  - credit, 56

- env, 58
- init, 57
- logos, 58
- mainFont, 58
- newY, 58
- OnEvent, 57
- sound, 59
- start, 57
- stop, 58
- texts, 59
- update, 58
- y, 59
- curAction
  - interface, 66
- curDefault
  - interface, 66
- curLook
  - interface, 66
- curPos
  - coeur, 45
  - startmenu, 105
- currentChap
  - coeur, 45
- currentLieu
  - coeur, 45
- currentLocal
  - coeur, 45
- curTake
  - interface, 66
- cusString\_python, 60
  - construct, 60
  - convert, 60
  - convertible, 60
  - cusString\_python, 60
  - cusString\_python, 60
- device
  - coeur, 45
  - console, 54
  - startmenu, 105
- dict
  - pyInt, 100
- dim
  - coeur, 46
  - startmenu, 105
- doc/example/chap.xml, 113
- dr
  - coeur, 46
  - startmenu, 105
- driver
  - coeur, 46
  - startmenu, 105
- emCur
  - interface, 66
- end
  - animateMesh, 23
  - persoMesh, 93
- env
  - coeur, 46
  - console, 54
  - credit, 58
  - interface, 66
  - inventaire, 72
  - startmenu, 105
- eval
  - pyInt, 98
- exec
  - pyInt, 99
- exit
  - console, 52
- files
  - logger, 88
- FPS
  - console, 54
- fps
  - console, 52
- fullscreen
  - coeur, 46
  - startmenu, 105
- game
  - startmenu, 105
- game\_img
  - startmenu, 105
- game\_name
  - startmenu, 106
- game\_path
  - startmenu, 106
- game\_text
  - startmenu, 106
- getAction
  - actionMesh, 19
- getActionMesh
  - coeur, 39
  - lieu, 79
- getAnim
  - animateMesh, 22
  - persoMesh, 93
- getBegin
  - animateMesh, 22
  - persoMesh, 93
- getCaption
  - local, 82
- getColor
  - actionMesh, 19
- getCombi

- actionMesh, 19
- item, 74
- getCombinaison
  - coeur, 40
  - local, 83
- getCombiResult
  - item, 74
- getCurPos
  - coeur, 40
  - startmenu, 102
- getDim
  - coeur, 40
- getEnd
  - animateMesh, 22
  - persoMesh, 93
- getFileSystem
  - startmenu, 103
- getFreePlace
  - interface, 64
  - inventaire, 70
- getId
  - base, 27
  - item, 75
  - localItem, 85
  - staticMesh, 111
- getImage
  - inventaire, 70
- getInventaire
  - interface, 64
- getItem
  - coeur, 40
  - inventaire, 70, 71
  - prendreMesh, 95
- getItemFromId
  - chap, 32
- getLogger
  - coeur, 40
- getNode
  - staticMesh, 111
- getNumItem
  - inventaire, 71
- getObservationItem
  - coeur, 40
  - local, 83
- getObservationMesh
  - coeur, 41
  - local, 83
- getPath
  - localItem, 85
- getSelectedAction
  - coeur, 41
- getSelectedNode
  - coeur, 41
- getStaticMeshFromId
  - lieu, 79
- getText
  - localItem, 85
- getTexture
  - coeur, 41
  - item, 75
- getTime
  - coeur, 41
  - startmenu, 103
- getVersion
  - coeur, 41
  - startmenu, 103
- handle
  - interface, 66
- help
  - console, 52
- history
  - console, 55
- histPos
  - console, 55
- id
  - base, 28
  - item, 75
  - localItem, 86
  - staticMesh, 111
- image
  - interface, 66
  - inventaire, 72
- images
  - inventaire, 72
- in
  - coeur, 46
- include/actionMesh.h, 115
- include/base.h, 117
- include/chap.h, 118
- include/coeur.h, 119
- include/console.h, 120
- include/credit.h, 121
- include/game.h, 122
- include/interface.h, 123
- include/inventaire.h, 124
- include/item.h, 125
- include/lieu.h, 126
- include/local.h, 127
- include/logger.h, 128
- include/pyInt.h, 129
- include/startmenu.h, 130
- include/staticMesh.h, 131
- init
  - actionMesh, 19
  - animateMesh, 23
  - base, 27

- chap, 32
- coeur, 42
- console, 52
- credit, 57
- interface, 64
- inventaire, 71
- item, 75
- lieu, 79
- local, 83
- localItem, 86
- logger, 88
- observationMesh, 91
- persoMesh, 93
- prendreMesh, 95
- pyInt, 99
- startmenu, 103
- staticMesh, 111
- initialize
  - base, 28
- initpyIrr
  - pyInt.cpp, 147
- initpyMerry
  - pyInt.cpp, 147
- interface, 62
  - ~interface, 63
  - addInventaire, 63
  - addItemInventaire, 63
  - afficherInventaire, 64
  - classic, 66
  - close, 64
  - curAction, 66
  - curDefault, 66
  - curLook, 66
  - curTake, 66
  - emCur, 66
  - env, 66
  - getFreePlace, 64
  - getInventaire, 64
  - handle, 66
  - image, 66
  - init, 64
  - interface, 63
  - inv, 67
  - mainFont, 67
  - OnEvent, 64
  - pafCur, 67
  - psCur, 67
  - pselectedInv, 67
  - pselectedItem, 67
  - selectedInv, 67
  - selectedItem, 67
  - setImage, 65
  - setImagePosition, 65
  - soundFind, 68
  - speak, 65
  - time, 68
  - txt, 68
  - update, 65
  - v, 68
- inv
  - interface, 67
- inventaire, 69
  - ~inventaire, 70
  - addItem, 70
  - afficher, 70
  - close, 70
  - env, 72
  - getFreePlace, 70
  - getImage, 70
  - getItem, 70, 71
  - getNumItem, 71
  - image, 72
  - images, 72
  - init, 71
  - inventaire, 70
  - it, 72
  - maxItem, 72
  - OnEvent, 71
  - position, 72
  - removeItem, 71
  - update, 71
  - volume, 72
  - vtot, 72
- inventaire.h
  - INVENTAIRE\_POS\_BAS, 124
  - INVENTAIRE\_POS\_DROITE, 124
  - INVENTAIRE\_POS\_GAUCHE, 124
  - INVENTAIRE\_POS\_HAUT, 124
  - INVENTAIRE\_POS, 124
  - INVENTAIRE\_POS\_BAS
    - inventaire.h, 124
  - INVENTAIRE\_POS\_DROITE
    - inventaire.h, 124
  - INVENTAIRE\_POS\_GAUCHE
    - inventaire.h, 124
  - INVENTAIRE\_POS\_HAUT
    - inventaire.h, 124
  - INVENTAIRE\_POS
    - inventaire.h, 124
- isInit
  - base, 27
- it
  - inventaire, 72
  - prendreMesh, 96
- item, 74
  - ~item, 74
  - combi, 75
  - combiResult, 75

- getCombi, 74
  - getCombiResult, 74
  - getId, 75
  - getTexture, 75
  - id, 75
  - init, 75
  - item, 74
  - repr, 75
  - texture, 76
- lang
  - coeur, 46
  - startmenu, 106
- langMenu
  - startmenu, 106
- lieu, 77
  - ~lieu, 77
  - addAnimateMesh, 78
  - addObservationMesh, 78
  - addPrendreMesh, 78
  - addStaticMesh, 78
  - close, 78
  - getActionMesh, 79
  - getStaticMeshFromId, 79
  - init, 79
  - lieu, 77
  - listAction, 80
  - listStatic, 80
  - OnEvent, 79
  - update, 79
- listAction
  - lieu, 80
- listItem
  - chap, 33
- listStatic
  - lieu, 80
- loadChap
  - coeur, 42
- loadDesc
  - startmenu, 103
- loadLieu
  - coeur, 42
- loadLocal
  - coeur, 42
- local, 81
  - ~local, 82
  - addCombinaison, 82
  - addObservationItem, 82
  - addObservationMesh, 82
  - caption, 84
  - close, 82
  - combinaison, 84
  - getCaption, 82
  - getCombinaison, 83
  - getObservationItem, 83
  - getObservationMesh, 83
  - init, 83
  - local, 82
  - observationItem, 84
  - observationMesh, 84
  - OnEvent, 83
  - update, 83
- localItem, 85
  - ~localItem, 85
  - getId, 85
  - getPath, 85
  - getText, 85
  - id, 86
  - init, 86
  - localItem, 85
  - path, 86
  - text, 86
- log
  - base, 28
  - startmenu, 103
- logger, 87
  - ~logger, 87
  - close, 88
  - closeFile, 88
  - createFile, 88
  - files, 88
  - init, 88
  - logger, 87
  - OnEvent, 88
  - write, 88
- logId
  - base, 29
- logos
  - credit, 58
- main
  - main.cpp, 144
- main.cpp
  - main, 144
- main\_module
  - pyInt, 100
- mainFont
  - credit, 58
  - interface, 67
- maxItem
  - inventaire, 72
- menus
  - startmenu, 106
- menusIn
  - startmenu, 106
- menusOut
  - startmenu, 106
- metaSelector

- coeur, 47
- modPyIrr.cpp
  - BOOST\_PYTHON\_MODULE, 145
  - reprDim, 145
  - reprPos, 145
- modPyMerry.cpp
  - BOOST\_PYTHON\_MODULE, 146
- mycore
  - base, 29
- mycredit
  - coeur, 47
- mylogger
  - coeur, 47
- nameClass
  - base, 29
- newY
  - credit, 58
- node
  - staticMesh, 111
- observationItem
  - local, 84
- observationMesh, 90
  - ~observationMesh, 90
  - init, 91
  - local, 84
  - observationMesh, 90
- OnEvent
  - chap, 32
  - coeur, 42
  - console, 53
  - credit, 57
  - interface, 64
  - inventaire, 71
  - lieu, 79
  - local, 83
  - logger, 88
  - pyInt, 99
  - startmenu, 103
- options
  - startmenu, 106
- options\_dv
  - startmenu, 107
- options\_fs
  - startmenu, 107
- options\_lang
  - startmenu, 107
- options\_res
  - startmenu, 107
- pafCur
  - interface, 67
- path
  - localItem, 86
  - startmenu, 107
- pathChap
  - coeur, 47
- pathConfig
  - startmenu, 107
- pathGame
  - coeur, 47
- pathLieu
  - coeur, 47
- pathLocal
  - coeur, 47
- pathMain
  - startmenu, 107
- pathStart
  - startmenu, 107
- persoMesh, 92
  - ~persoMesh, 93
  - anim, 93
  - begin, 93
  - end, 93
  - getAnim, 93
  - getBegin, 93
  - getEnd, 93
  - init, 93
  - persoMesh, 93
- pickNode
  - coeur, 43
- play
  - coeur, 43
- position
  - console, 53
  - inventaire, 72
- prendreMesh, 94
  - ~prendreMesh, 95
  - getItem, 95
  - init, 95
  - it, 96
  - prendreMesh, 95
- progress
  - startmenu, 104
- progressing
  - startmenu, 107
- psCur
  - interface, 67
- pselectedInv
  - interface, 67
- pselectedItem
  - interface, 67
- py
  - coeur, 48
- pyInt, 97
  - ~pyInt, 98
  - addPyEnv, 98

- close, 98
- coeur, 44
- dict, 100
- eval, 98
- exec, 99
- init, 99
- main\_module, 100
- OnEvent, 99
- pyInt, 98
- update, 100
- pyInt.cpp
  - initpyIrr, 147
  - initpyMerry, 147
- q
  - coeur, 48
  - startmenu, 108
- quit
  - console, 53
- removeItem
  - inventaire, 71
- repr
  - item, 75
- reprDim
  - modPyIrr.cpp, 145
- reprPos
  - modPyIrr.cpp, 145
- rotation
  - console, 53
- saveConfig
  - startmenu, 104
- selectedAction
  - coeur, 48
- selectedInv
  - interface, 67
- selectedItem
  - interface, 67
- setCaption
  - coeur, 43
  - startmenu, 104
- setImage
  - interface, 65
- setImagePosition
  - interface, 65
- setSkyBox
  - coeur, 43
- setVisible
  - staticMesh, 111
- smgr
  - coeur, 48
  - startmenu, 108
- sound
  - credit, 59
  - soundFind
    - interface, 68
  - source/actionMesh.cpp, 132
  - source/base.cpp, 133
  - source/chap.cpp, 134
  - source/coeur.cpp, 135
  - source/console.cpp, 136
  - source/credit.cpp, 137
  - source/interface.cpp, 138
  - source/inventaire.cpp, 139
  - source/item.cpp, 140
  - source/lieu.cpp, 141
  - source/local.cpp, 142
  - source/logger.cpp, 143
  - source/main.cpp, 144
  - source/modPyIrr.cpp, 145
  - source/modPyMerry.cpp, 146
  - source/pyInt.cpp, 147
  - source/startmenu.cpp, 148
  - source/staticMesh.cpp, 149
  - speak
    - interface, 65
  - start
    - credit, 57
    - startmenu, 108
  - startCoeur
    - startmenu, 104
  - startCredit
    - coeur, 43
  - starting
    - startmenu, 108
  - starting\_bar
    - startmenu, 108
  - starting\_text
    - startmenu, 108
  - starting\_txt
    - startmenu, 108
  - startmenu, 101
    - ~startmenu, 102
    - bits, 104
    - c, 104
    - close, 102
    - curPos, 105
    - device, 105
    - dim, 105
    - dr, 105
    - driver, 105
    - env, 105
    - fullscreen, 105
    - game, 105
    - game\_img, 105
    - game\_name, 106
    - game\_path, 106



- game\_text, 106
- getCurPos, 102
- getFileSystem, 103
- getTime, 103
- getVersion, 103
- init, 103
- lang, 106
- langMenu, 106
- loadDesc, 103
- log, 103
- menus, 106
- menusIn, 106
- menusOut, 106
- OnEvent, 103
- options, 106
- options\_dv, 107
- options\_fs, 107
- options\_lang, 107
- options\_res, 107
- path, 107
- pathConfig, 107
- pathMain, 107
- pathStart, 107
- progress, 104
- progressing, 107
- q, 108
- saveConfig, 104
- setCaption, 104
- smgr, 108
- start, 108
- startCoeur, 104
- starting, 108
- starting\_bar, 108
- starting\_text, 108
- starting\_txt, 108
- startmenu, 102
- stencilbuffer, 108
- update, 104
- vsync, 108
- welcome, 109
- staticMesh, 110
  - ~staticMesh, 110
  - close, 111
  - getId, 111
  - getNode, 111
  - id, 111
  - init, 111
  - node, 111
  - setVisible, 111
  - staticMesh, 110
  - update, 111
- stencilbuffer
  - coeur, 48
  - startmenu, 108
- stop
  - credit, 58
- target
  - console, 53
- text
  - console, 55
  - localItem, 86
- texts
  - credit, 59
- texture
  - item, 76
- time
  - interface, 68
- tx
  - console, 55
- txt
  - interface, 68
- update
  - base, 28
  - chap, 33
  - coeur, 44
  - console, 53
  - credit, 58
  - interface, 65
  - inventaire, 71
  - lieu, 79
  - local, 83
  - pyInt, 100
  - startmenu, 104
  - staticMesh, 111
- v
  - interface, 68
- volume
  - inventaire, 72
- vsync
  - coeur, 48
  - startmenu, 108
- vtot
  - inventaire, 72
- welcome
  - startmenu, 109
- write
  - console, 54
  - logger, 88
- XML Scripts, 15
- y
  - credit, 59