

DON'T PANIC

3DMob: Grafica 3D su device mobili



Definizione di Prodotto

Informazioni sul documento

Versione	4.2.0
Redazione	Rampazzo Federico Cesarato Fabio Busato Luca
Verifica	Pezzutti Marco Lain Daniele
Responsabile	Basaglia Mattia
Uso	Esterno
Lista di distribuzione	Don't Panic Prof. Vardanega Tullio Prof. Cardin Riccardo

Descrizione

Architettura di dettaglio dell'applicazione 3DMob



Diario delle modifiche

Descrizione modifica	Autore	Ruolo	Data	Versione
Approvazione documento	Basaglia Mattia	Responsabile	2013-02-17	4.2.0
Verifica documento	Pezzutti Marco	Verificatore	2013-02-16	4.1.1
Verifica documento	Lain Daniele	Verificatore	2013-02-15	4.1.0
Appendice schemi file esportati	Sciarrone Riccardo	Progettista	2013-02-12	4.0.4
Tracciamento	Cesarato Fabio	Progettista	2013-02-10	4.0.3
Specifica componenti	Rampazzo Federico	Progettista	2013-02-09	4.0.2
Stesura standard di progetto	Busato Luca	Progettista	2013-02-08	4.0.1
Creazione scheletro del documento e stesura introduzione	Cesarato Fabio	Progettista	2013-02-07	4.0.0



Indice

1	Introduzione	1
1.1	Scopo del documento	1
1.2	Scopo del Prodotto	1
1.3	Glossario	1
1.4	Riferimenti	1
1.4.1	Normativi	1
1.4.2	Informativi	1
2	Standard di progetto	2
2.1	Standard di progettazione architettuale	2
2.2	Standard di documentazione del codice	2
2.3	Standard di denominazione di entità e relazioni	2
2.4	Standard di programmazione	2
2.5	Strumenti di lavoro	2
3	Specifica componenti	3
3.1	DDDMob	3
3.2	DDDMob::Model	3
3.3	DDDMob::Model::Commands	4
3.3.1	CommandTransform (abstract)	4
3.3.2	CommandAddLight (class)	5
3.3.3	CommandEditMesh (abstract)	6
3.3.4	CommandEditLight (abstract)	7
3.3.5	CommandScene (abstract)	8
3.3.6	CommandAddElement (abstract)	9
3.3.7	CommandPosition (class)	11
3.3.8	CommandScale (class)	12
3.3.9	CommandRotation (class)	14
3.3.10	CommandLightType (class)	16
3.3.11	CommandShininess (class)	17
3.3.12	CommandEmission (class)	20
3.3.13	CommandColor (abstract)	22
3.3.14	CommandColorEmission (class)	23
3.3.15	CommandColorSpecular (class)	24
3.3.16	CommandColorDiffuse (class)	26
3.3.17	CommandEditObject (class)	27
3.4	DDDMob::Model::CConverter	28
3.5	DDDMob::Model::CConverter::CSettingsModel	28
3.5.1	Setting (class)	28
3.5.2	DeviceLimit (class)	32
3.5.3	NumericPrecisionType (class)	33
3.6	DDDMob::Model::CConverter::CExportModel	34
3.6.1	SceneExporter (interface)	34
3.6.2	ExporterUBJSON (class)	35
3.6.3	ExporterXML (class)	38
3.6.4	ExporterMinifiedJSON (class)	41
3.6.5	ExporterJSON (class)	41



3.7	DDDMob::Model::CConverter::CLoaderModel	44
3.7.1	Scene (abstract)	45
3.7.2	SceneAdapter (class)	51
3.7.3	ImporterJson (class)	57
3.8	DDDMob::Controller	59
3.8.1	Controller (class)	59
3.9	DDDMob::View	65
3.9.1	FacadeView (class)	66
3.9.2	About (class)	70
3.10	DDDMob::View::CDockWidgets	71
3.10.1	Material (class)	71
3.10.2	Selector (class)	75
3.10.3	LightWidget (class)	77
3.10.4	Action (class)	79
3.10.5	Transformation (class)	80
3.11	DDDMob::View::CMainWindow	84
3.11.1	WindowWithMenu (class)	84
3.11.2	WindowWithDock (class)	86
3.11.3	WindowWithStatus (class)	87
3.11.4	Decorator (abstract)	88
3.11.5	MainWindow (interface)	90
3.11.6	View3D (class)	91
3.11.7	GLView (class)	93
3.12	DDDMob::View::CSettingsWindow	99
3.12.1	SettingsWindow (class)	99
3.13	DDDMob::View::CErrorWindow	101
3.13.1	ErrorMessageBox (class)	101
3.14	DDDMob::View::CWidgetElement	102
3.14.1	AxisSlider (class)	102
3.14.2	ColorPicker (class)	105
3.15	DDDMob::C3DObject	107
3.15.1	SceneObject (abstract)	108
3.15.2	Light (class)	115
3.15.3	Mesh (class)	118
3.15.4	LightType (class)	122
3.15.5	Texture (class)	123
3.15.6	Vertex (class)	125
3.15.7	Keyframe (class)	129
3.15.8	BoundingBox3D (class)	131
4	Diagrammi di sequenza	137
4.1	Importazione	137
4.2	Esportazione	138
4.3	Traslazione	139
A	Tracciamento	140
A.1	Tracciamento requisiti - classi	140
A.2	Tracciamento classi - requisiti	142
A.3	Tracciamento modulo - test	145



B	Schemi file esportati	152
B.1	JSONSchema	152
B.2	XMLSchema	154



Elenco delle tabelle

2	Tabella requisiti funzionali / classi	141
3	Tabella classi / requisiti	144
4	Tabella metodi / test unità	151



Elenco delle figure

1	Componente DDDMob	3
2	Componente DDDMob::Model	3
3	Componente DDDMob::Model::Commands	4
4	Classe CommandTransform	4
5	Classe CommandAddLight	5
6	Classe CommandEditMesh	6
7	Classe CommandEditLight	7
8	Classe CommandScene	8
9	Classe CommandAddElement	9
10	Classe CommandPosition	11
11	Classe CommandScale	12
12	Classe CommandRotation	14
13	Classe CommandLightType	16
14	Classe CommandShininess	17
15	Classe CommandEmission	20
16	Classe CommandColor	22
17	Classe CommandColorEmission	23
18	Classe CommandColorSpecular	24
19	Classe CommandColorDiffuse	26
20	Classe CommandEditObject	27
21	Componente DDDMob::Model::CConverter	28
22	Componente DDDMob::Model::CConverter::CSettingsModel	28
23	Classe Setting	28
24	Classe DeviceLimit	32
25	Classe NumericPrecisionType	33
26	Componente DDDMob::Model::CConverter::CExportModel	34
27	Classe SceneExporter	34
28	Classe ExporterUBJSON	35
29	Classe ExporterXML	38
30	Classe ExporterMinifiedJSON	41
31	Classe ExporterJSON	41
32	Componente DDDMob::Model::CConverter::CLoaderModel	44
33	Classe Scene	45
34	Classe SceneAdapter	51
35	Classe ImporterJson	57
36	Componente DDDMob::Controller	59
37	Classe Controller	59
38	Componente DDDMob::View	65
39	Classe FacadeView	66
40	Classe About	70
41	Componente DDDMob::View::CDockWidgets	71
42	Classe Material	71
43	Classe Selector	75
44	Classe LightWidget	77
45	Classe Action	79
46	Classe Transformation	80
47	Componente DDDMob::View::CMainWindow	84



48	Classe <code>WindowWithMenu</code>	84
49	Classe <code>WindowWithDock</code>	86
50	Classe <code>WindowWithStatus</code>	87
51	Classe <code>Decorator</code>	88
52	Classe <code>MainWindow</code>	90
53	Classe <code>View3D</code>	91
54	Classe <code>GLView</code>	93
55	Componente <code>DDDMob::View::CSettingsWindow</code>	99
56	Classe <code>SettingsWindow</code>	99
57	Componente <code>DDDMob::View::CErrorWindow</code>	101
58	Classe <code>ErrorMessageBox</code>	101
59	Componente <code>DDDMob::View::CWidgetElement</code>	102
60	Classe <code>AxisSlider</code>	102
61	Classe <code>ColorPicker</code>	105
62	Componente <code>DDDMob::C3DObject</code>	107
63	Classe <code>SceneObject</code>	108
64	Classe <code>Light</code>	115
65	Classe <code>Mesh</code>	118
66	Classe <code>LightType</code>	122
67	Classe <code>Texture</code>	123
68	Classe <code>Vertex</code>	125
69	Classe <code>Keyframe</code>	129
70	Classe <code>BoundingBox3D</code>	131
71	Diagrammi di sequenza per l'importazione	137
72	Diagrammi di sequenza per l'esportazione	138
73	Diagrammi di sequenza per la traslazione di un oggetto	139



1 Introduzione

1.1 Scopo del documento

Il seguente documento ha lo scopo di definire nel dettaglio la struttura del sistema 3DMob, approfondendo quanto già riportato nella *Specifica Tecnica*. Tale documento fornisce una struttura dettagliata e completa che viene utilizzata dai *programmatici* per le attività di codifica.

1.2 Scopo del Prodotto

Lo scopo del progetto è la realizzazione di un'applicazione in grado di convertire file prodotti da programmi di grafica 3D in file in formato JSON_G in grado di essere visualizzati su dispositivi mobile senza perdita di informazione. L'obiettivo è quello di semplificare il workflow attuale necessario a rendere compatibili i file.

1.3 Glossario

Al fine di evitare ogni ambiguità di linguaggio e massimizzare la comprensione dei documenti, i termini tecnici, di dominio, gli acronimi e le parole che necessitano di essere chiarite, sono riportate nel documento *Glossario v4.2.0*.

Ogni occorrenza di vocaboli presenti nel *Glossario* è marcata da una "G" maiuscola in pedice.

1.4 Riferimenti

1.4.1 Normativi

- **Specifica Tecnica:** *Specifica Tecnica v4.2.0*;
- **Analisi dei Requisiti:** *Analisi dei Requisiti v4.2.0*;
- **Norme di Progetto:** *Norme di Progetto v4.2.0*.

1.4.2 Informativi

- **Documentazione Qt_G per Segnali e Slot**
<http://qt-project.org/doc/qt-5.0/qtcore/signalsandslots.html>;
- **Documentazione di Assimp**
http://assimp.sourceforge.net/lib_html/index.html;
- **Documentazione Boost.PropertyTree**
http://www.boost.org/doc/libs/1_52_0/doc/html/property_tree.html.
- **Glossario:** *Glossario v4.2.0*.



2 Standard di progetto

2.1 Standard di progettazione architettuale

Gli standard di progettazione architettuale sono definiti nella *Specifica Tecnica v4.2.0*.

2.2 Standard di documentazione del codice

Gli standard per la scrittura della documentazione del codice sono definiti nelle *Norme di Progetto v4.2.0*.

2.3 Standard di denominazione di entità e relazioni

Tutti gli elementi definiti, siano essi package_G , classi, metodi o attributi, devono avere denominazioni chiare ed autoesplicative. Nel caso in cui il nome risulti essere lungo è preferibile anteporre la chiarezza alla lunghezza.

Sono ammesse abbreviazioni se:

- immediatamente comprensibili;
- non ambigue;
- sufficientemente contestualizzate.

Le regole tipografiche relativi ai nomi delle entità sono definiti nelle *Norme di Progetto v4.2.0*.

2.4 Standard di programmazione

Gli standard di programmazione sono definiti e descritti nelle *Norme di Progetto v4.2.0*.

2.5 Strumenti di lavoro

Gli strumenti da adottare e le procedure da seguire per utilizzarli correttamente durante la realizzazione del prodotto software sono definiti nelle *Norme di Progetto v4.2.0*.



3 Specifica componenti

3.1 DDDMob

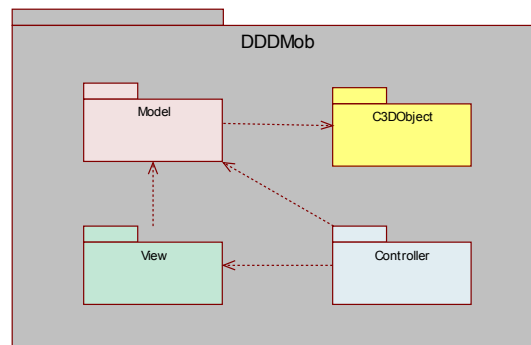


Figura 1: Componente DDDMob

3.2 DDDMob::Model

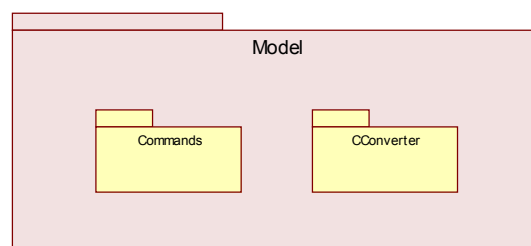


Figura 2: Componente DDDMob::Model



3.3 DDDMob::Model::Commands

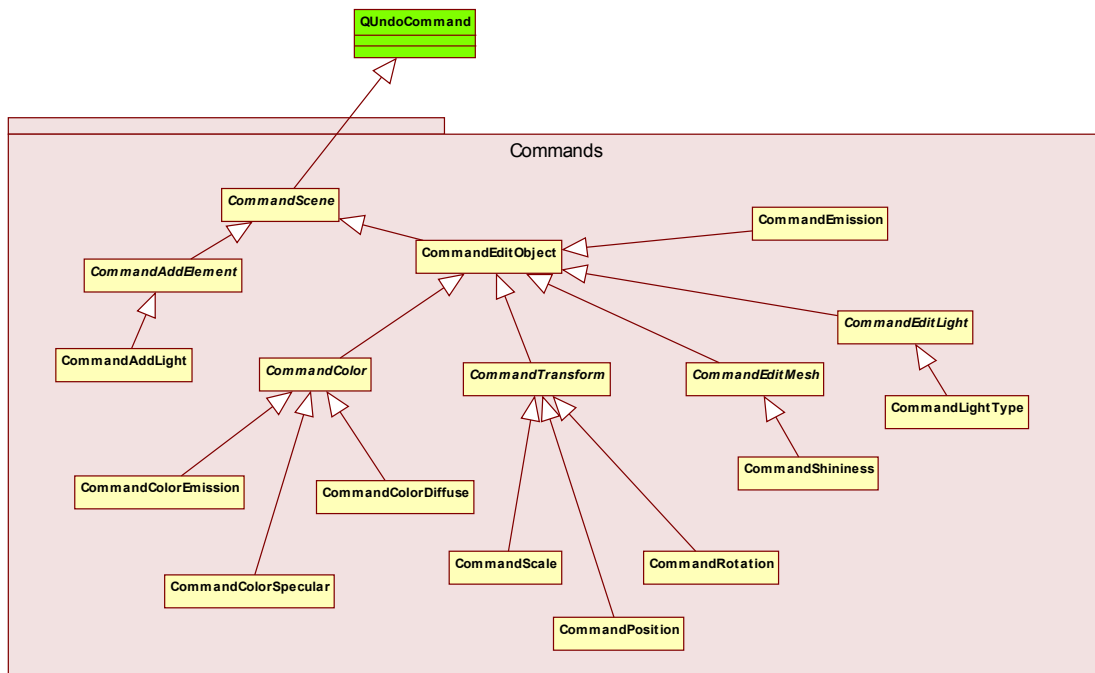


Figura 3: Componente DDDMob::Model::Commands

3.3.1 CommandTransform (abstract)

<i>CommandTransform</i>
#transformOriginal: QVector3D #transformNew: QVector3D
+ CommandTransform(scene: Scene*, sceneObject: SceneObject*, transformOriginal: QVector3D, transformNew: QVector3D)

Figura 4: Classe CommandTransform

3.3.1.1 Descrizione

Classe base per i comandi di trasformazione

3.3.1.2 Utilizzo

Viene utilizzata per applicare un generico parametro di trasformazione ad un oggetto della scena_G 3D, specificato poi nelle classi che ereditano da questa

3.3.1.3 Classi ereditate

- DDDMob :: Model :: Commands :: CommandEditObject.

3.3.1.4 Ereditata da

- DDDMob :: Model :: Commands :: CommandPosition;



- DDDMob :: Model :: Commands :: CommandScale;
- DDDMob :: Model :: Commands :: CommandRotation.

3.3.1.5 Attributi

Name

transformOriginal;

Accesso

protected;

Tipo

QVector3D;

Descrizione

Rappresenta la trasformazione originale prima di aver effettuato la nuova trasformazione.

Name

transformNew;

Accesso

protected;

Tipo

QVector3D;

Descrizione

Rappresenta la nuova trasformazione effettuata.

3.3.1.6 Metodi

CommandTransform

Accesso

public;

Descrizione

Costruttore di CommandTransform.

3.3.2 CommandAddLight (class)

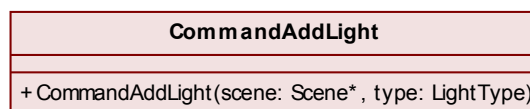


Figura 5: Classe CommandAddLight

3.3.2.1 Descrizione

Classe che rappresenta il comando di aggiunta di una luce con i parametri di default alla scena_G 3D



3.3.2.2 Utilizzo

Viene utilizzata per gestire i Signal_G riguardanti la creazione di una luce ed invocare i corretti metodi del Model

3.3.2.3 Classi ereditate

- $\text{DDDMob} :: \text{Model} :: \text{Commands} :: \text{CommandAddElement}$.

3.3.2.4 Metodi

CommandAddLight

Accesso

public;

Descrizione

costruttore di CommandAddLight.

3.3.3 CommandEditMesh (abstract)

<i>CommandEditMesh</i>
+ CommandEditMesh(scene: Scene*, sceneObject: Mesh*) - getMesh(): Mesh*

Figura 6: Classe CommandEditMesh

3.3.3.1 Descrizione

Classe che rappresenta il comando di modifica di una mesh

3.3.3.2 Utilizzo

Viene utilizzata per gestire i Signal_G riguardanti la modifica di una mesh ed invocare i corretti metodi del Model

3.3.3.3 Classi ereditate

- $\text{DDDMob} :: \text{Model} :: \text{Commands} :: \text{CommandEditObject}$.

3.3.3.4 Ereditata da

- $\text{DDDMob} :: \text{Model} :: \text{Commands} :: \text{CommandShininess}$.

3.3.3.5 Metodi

CommandEditMesh

Accesso

public;

Descrizione

costruttore per CommandEditMesh.



getMesh

Accesso

private;

Tipo di ritorno

Mesh*;

Descrizione

getter della Mesh.

3.3.4 CommandEditLight (abstract)

<i>CommandEditLight</i>
+ CommandEditLight(scene: Scene*, sceneObject: Light*) #getLight(): Light*

Figura 7: Classe CommandEditLight

3.3.4.1 Descrizione

Classe che rappresenta il comando di modifica della luce selezionata

3.3.4.2 Utilizzo

Viene utilizzata per gestire i Signal_G riguardanti la modifica della luce selezionata ed invocare i corretti metodi del Model

3.3.4.3 Classi ereditate

- DDDMob :: Model :: Commands :: CommandEditObject.

3.3.4.4 Ereditata da

- DDDMob :: Model :: Commands :: CommandLightType.

3.3.4.5 Metodi

CommandEditLight

Accesso

public;

Descrizione

Costruttore di CommandEditLight.

getLight

Accesso

protected;

Tipo di ritorno

Light*;



Descrizione

getter di CommandEditLight.

3.3.5 CommandScene (abstract)

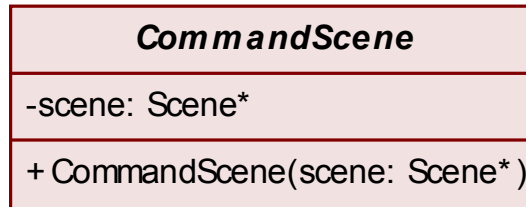


Figura 8: Classe CommandScene

3.3.5.1 Descrizione

Classe che rappresenta un generico comando da eseguire sulla scena_G. È il padre di tutti i possibili componenti concrete command del Design Pattern_G Command

3.3.5.2 Utilizzo

Viene utilizzata per applicare un generico comando da eseguire sulla scena_G 3D, specificato poi nelle classi che ereditano da questa

3.3.5.3 Classi ereditate

- QUndoCommand.

3.3.5.4 Ereditata da

- DDDMob :: Model :: Commands :: CommandAddElement;
- DDDMob :: Model :: Commands :: CommandEditObject.

3.3.5.5 Attributi

Name

scene;

Accesso

private;

Tipo

Scene*;

Descrizione

Riferimento alla scena_G del modello.



3.3.5.6 Metodi

CommandScene

Accesso

public;

Descrizione

Costruttore di CommandScene.

3.3.6 CommandAddElement (abstract)

<i>CommandAddElement</i>
#name: QString #object: SceneObject*
+ CommandAddElement(scene: Scene*, name: QString, object: SceneObject*) + undo(): void + redo(): void

Figura 9: Classe CommandAddElement

3.3.6.1 Descrizione

Classe che rappresenta un comando di aggiunta di un elemento alla scena_G 3D

3.3.6.2 Utilizzo

Viene utilizzata per gestire i Signal_G riguardanti l'aggiunta di un elemento ed invocare i corretti metodi del Model

3.3.6.3 Classi ereditate

- DDDMob :: Model :: Commands :: CommandScene.

3.3.6.4 Ereditata da

- DDDMob :: Model :: Commands :: CommandAddLight.

3.3.6.5 Attributi

Name

name;

Accesso

protected;

Tipo

QString;

Descrizione

Nome dell'elemento.

Name

object;



Accesso

protected;

Tipo

SceneObject*;

Descrizione

Oggetto della scena_G.

3.3.6.6 Metodi

CommandAddElement

Accesso

public;

Descrizione

costruttore CommandAddElement.

undo

Accesso

public;

Tipo di ritorno

void;

Descrizione

Annulla il comando;

Note

- Questo metodo è stato ridefinito.

redo

Accesso

public;

Tipo di ritorno

void;

Descrizione

Ripristina il comando annullato;

Note

- Questo metodo è stato ridefinito.



3.3.7 CommandPosition (class)

CommandPosition
<pre>+ CommandPosition(scene: Scene*, sceneObject: SceneObject*, transformNew: QVector3D) + id(): int + redo(): void + undo(): void + mergeWith(command: const QUndoCommand*): bool</pre>

Figura 10: Classe CommandPosition

3.3.7.1 Descrizione

Classe che rappresenta un comando per cambiare la posizione dell'oggetto della scena_G 3D selezionato

3.3.7.2 Utilizzo

Viene utilizzata per gestire i Signal_G riguardanti la modifica della posizione di un oggetto ed invocare i corretti metodi del Model

3.3.7.3 Classi ereditate

- DDDMob :: Model :: Commands :: CommandTransform.

3.3.7.4 Metodi

CommandPosition

Accesso

public;

Descrizione

Costruttore di CommandPosition.

id

Accesso

public;

Tipo di ritorno

int;

Descrizione

Restituisce l'id del comando;

Note

- Deve essere esplicitamente marcati come costanti;
- Questo metodo è stato ridefinito.



redo

Accesso

public;

Tipo di ritorno

void;

Descrizione

Metodo che esegue nuovamente l'ultimo comando annullato;

Note

- Questo metodo è stato ridefinito.

undo

Accesso

public;

Tipo di ritorno

void;

Descrizione

Metodo che annulla l'ultimo comando eseguito;

Note

- Questo metodo è stato ridefinito.

mergeWith

Accesso

public;

Tipo di ritorno

bool;

Descrizione

Metodo che prova ad eseguire un merge_G del comando corrente con quello passato come parametro;

Note

- Questo metodo è stato ridefinito.

3.3.8 CommandScale (class)

CommandScale
+ CommandScale(scene: Scene*, sceneObject: SceneObject*, transformNew: QVector3D) + redo(): void + undo(): void + id(): int + mergeWith(command: const QUndoCommand*): bool

Figura 11: Classe CommandScale

3.3.8.1 Descrizione

Classe che rappresenta un comando per cambiare la dimensione dell'oggetto della scena_G 3D selezionato



3.3.8.2 Utilizzo

Viene utilizzata per gestire i Signal_G riguardanti la dimensione di un oggetto ed invocare i corretti metodi del Model

3.3.8.3 Classi ereditate

- $\text{DDDMob} :: \text{Model} :: \text{Commands} :: \text{CommandTransform}$.

3.3.8.4 Metodi

CommandScale

Accesso

public;

Descrizione

Costruttore di CommandScale.

redo

Accesso

public;

Tipo di ritorno

void;

Descrizione

Metodo che esegue nuovamente l'ultimo comando annullato;

Note

- Questo metodo è stato ridefinito.

undo

Accesso

public;

Tipo di ritorno

void;

Descrizione

Metodo che annulla l'ultimo comando eseguito;

Note

- Questo metodo è stato ridefinito.



id

Accesso

public;

Tipo di ritorno

int;

Descrizione

Restituisce l'id del comando;

Note

- Questo metodo è stato ridefinito.

mergeWith

Accesso

public;

Tipo di ritorno

bool;

Descrizione

Metodo che prova ad eseguire un merge_G del comando corrente con quello passato come parametro;

Note

- Questo metodo è stato ridefinito.

3.3.9 CommandRotation (class)

CommandRotation
+ CommandRotation(scene: Scene*, sceneObject: SceneObject*, transformNew: QVector3D) + id(): int + redo(): void + undo(): void + mergeWith(command: const QUndoCommand*): bool

Figura 12: Classe CommandRotation

3.3.9.1 Descrizione

Classe che rappresenta un comando per cambiare la rotazione del modello

3.3.9.2 Utilizzo

3.3.9.3 Classi ereditate

- DDDMob :: Model :: Commands :: CommandTransform.

3.3.9.4 Metodi

CommandRotation

Accesso

public;

Descrizione

Costruttore di CommandRotation.



id

Accesso

public;

Tipo di ritorno

int;

Descrizione

Restituisce l'id del comando;

Note

- Deve essere esplicitamente marcati come costanti;
- Questo metodo è stato ridefinito.

redo

Accesso

public;

Tipo di ritorno

void;

Descrizione

Metodo che esegue nuovamente l'ultimo comando annullato;

Note

- Questo metodo è stato ridefinito.

undo

Accesso

public;

Tipo di ritorno

void;

Descrizione

Metodo che annulla l'ultimo comando eseguito;

Note

- Questo metodo è stato ridefinito.

mergeWith

Accesso

public;

Tipo di ritorno

bool;

Descrizione

Metodo che prova ad eseguire un merge_G del comando corrente con quello passato come parametro;

Note

- Questo metodo è stato ridefinito.



3.3.10 CommandLightType (class)

CommandLightType
-oldType: Light Type -newType: Light Type
+ CommandLightType(scene: Scene*, light: Light*, type: LightType) + undo(): void + redo(): void

Figura 13: Classe CommandLightType

3.3.10.1 Descrizione

Classe che rappresenta un comando per cambiare il tipo della luce selezionata

3.3.10.2 Utilizzo

Viene utilizzata per gestire i Signal_G riguardanti la modifica della tipologia di una luce ed invocare i corretti metodi del Model

3.3.10.3 Classi ereditate

- DDDMob :: Model :: Commands :: CommandEditLight.

3.3.10.4 Attributi

Name

oldType;

Accesso

private;

Tipo

LightType;

Descrizione

vecchio tipo di luce.

Name

newType;

Accesso

private;

Tipo

LightType;

Descrizione

Nuovo tipo di luce.



3.3.10.5 Metodi

CommandLightType

Accesso

public;

Descrizione

Costruttore di CommandLightType.

undo

Accesso

public;

Tipo di ritorno

void;

Descrizione

Elimina il comando;

Note

- Questo metodo è stato ridefinito.

redo

Accesso

public;

Tipo di ritorno

void;

Descrizione

Ripristina il comando annullato;

Note

- Questo metodo è stato ridefinito.

3.3.11 CommandShininess (class)

CommandShininess
-oldShininess: double -newShininess: double
+ CommandShininess(scene: Scene*, object: Mesh*, Shininess: double) + undo(): void + redo(): void + id(): int + mergeWith(command: const QUndoCommand *): bool

Figura 14: Classe CommandShininess

3.3.11.1 Descrizione

Classe che rappresenta il comando che cambia il valore di lucentezza della mesh



3.3.11.2 Utilizzo

Viene utilizzata per gestire i Signal_G riguardanti la modifica della lucentezza di una mesh ed invocare i corretti metodi del Model

3.3.11.3 Classi ereditate

- DDDMob :: Model :: Commands :: CommandEditMesh.

3.3.11.4 Attributi

Name

oldShininess;

Accesso

private;

Tipo

double;

Descrizione

Vecchio valore di lucentezza.

Name

newShininess;

Accesso

private;

Tipo

double;

Descrizione

Nuovo valore di lucentezza.

3.3.11.5 Metodi

CommandShininess

Accesso

public;

Descrizione

costruttore CommandShininess.

undo

Accesso

public;

Tipo di ritorno

void;

Descrizione

Annulla il comando;

Note

- Questo metodo è stato ridefinito.



redo

Accesso

public;

Tipo di ritorno

void;

Descrizione

ripristina il comando annullato;

Note

- Questo metodo è stato ridefinito.

id

Accesso

public;

Tipo di ritorno

int;

Descrizione

Ritorna l'id del comando;

Note

- Deve essere esplicitamente marcati come costanti;
- Questo metodo è stato ridefinito.

mergeWith

Accesso

public;

Tipo di ritorno

bool;

Descrizione

Metodo che prova ad eseguire un merge_G del comando corrente con quello passato come parametro;

Note

- Questo metodo è stato ridefinito.



3.3.12 CommandEmission (class)

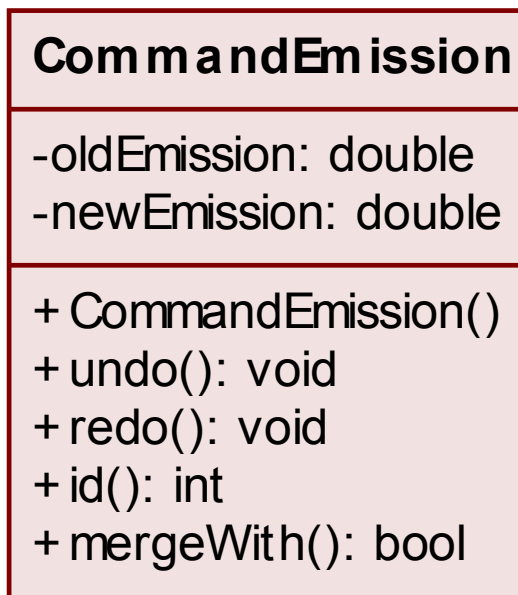


Figura 15: Classe CommandEmission

3.3.12.1 Descrizione

Classe che rappresenta il comando per cambiare il valore della quantità di emissione del materiale di un oggetto

3.3.12.2 Utilizzo

Viene utilizzata per gestire i Signal_G riguardanti la quantità di emissione del materiale di un oggetto ed invocare i corretti metodi del Model

3.3.12.3 Classi ereditate

- DDDMob :: Model :: Commands :: CommandEditObject.

3.3.12.4 Attributi

Name

oldEmission;

Accesso

private;

Tipo

double;

Descrizione

Valore del colore di emissione prima della modifica.

Name

newEmission;



Accesso

private;

Tipo

double;

Descrizione

Valore del colore di emissione dopo la modifica.

3.3.12.5 Metodi

CommandEmission

Accesso

public;

Descrizione

Costruttore di CommandEmission.

undo

Accesso

public;

Tipo di ritorno

void;

Descrizione

Metodo che annulla l'ultimo comando eseguito;

Note

- Questo metodo è stato ridefinito.

redo

Accesso

public;

Tipo di ritorno

void;

Descrizione

Metodo che esegue nuovamente l'ultimo comando annullato;

Note

- Questo metodo è stato ridefinito.



id

Accesso

public;

Tipo di ritorno

int;

Descrizione

Restituisce l'id del comando;

Note

- Deve essere esplicitamente marcati come costanti;
- Questo metodo è stato ridefinito.

mergeWith

Accesso

public;

Tipo di ritorno

bool;

Descrizione

Metodo che prova ad eseguire un merge_G del comando corrente con quello passato come parametro;

Note

- Questo metodo è stato ridefinito.

3.3.13 CommandColor (abstract)

CommandColor
#colorOld: QColor #colorNew: QColor
+CommandColor(out scene: Scene*, out object: SceneObject*, colorOld: QColor, colorNew: QColor)

Figura 16: Classe CommandColor

3.3.13.1 Descrizione

Classe che rappresenta il comando per cambiare le caratteristiche legate al colore dell'oggetto

3.3.13.2 Utilizzo

Viene utilizzata per gestire i Signal_G riguardanti il colore dell'oggetto

3.3.13.3 Classi ereditate

- DDDMob :: Model :: Commands :: CommandEditObject.



3.3.13.4 Ereditata da

- DDDMob :: Model :: Commands :: CommandColorEmission;
- DDDMob :: Model :: Commands :: CommandColorSpecular;
- DDDMob :: Model :: Commands :: CommandColorDiffuse.

3.3.13.5 Attributi

Name

colorOld;

Accesso

protected;

Tipo

QColor;

Descrizione

Colore vecchio dell'oggetto.

Name

colorNew;

Accesso

protected;

Tipo

QColor;

Descrizione

Rappresenta il colore nuovo dell'oggetto.

3.3.13.6 Metodi

CommandColor

Accesso

public;

Descrizione

Costruttore della classe CommandColor.

3.3.14 CommandColorEmission (class)

CommandColorEmission
+ CommandColorEmission(out scene: Scene*, out object: SceneObject*, color: QColor) + undo(): void + redo(): void

Figura 17: Classe CommandColorEmission

3.3.14.1 Descrizione

Classe che rappresenta il comando per cambiare il colore di emissione dell'oggetto



3.3.14.2 Utilizzo

Viene utilizzata per gestire i Signal_G riguardanti il colore di emissione dell'oggetto

3.3.14.3 Classi ereditate

- $\text{DDDMob} :: \text{Model} :: \text{Commands} :: \text{CommandColor}$.

3.3.14.4 Metodi

CommandColorEmission

Accesso

public;

Descrizione

Costruttore della classe `CommandColorEmission`.

undo

Accesso

public;

Tipo di ritorno

void;

Descrizione

Ripristina la modifica precedente;

Note

- Questo metodo è stato ridefinito.

redo

Accesso

public;

Tipo di ritorno

void;

Descrizione

Ripristina l'ultima modifica annullata;

Note

- Questo metodo è stato ridefinito.

3.3.15 CommandColorSpecular (class)

CommandColorSpecular
+ CommandColorSpecular(out scene: Scene*, out object: SceneObject*, color: QColor) + redo(): void + undo(): void

Figura 18: Classe `CommandColorSpecular`



3.3.15.1 Descrizione

Classe che rappresenta il comando per cambiare il colore di diffusione_G dell'oggetto

3.3.15.2 Utilizzo

Viene utilizzata per gestire i Signal_G riguardanti il colore di diffusione_G dell'oggetto

3.3.15.3 Classi ereditate

- DDDMob :: Model :: Commands :: CommandColor.

3.3.15.4 Metodi

CommandColorSpecular

Accesso

public;

Descrizione

Costruttore della classe CommandColorSpecular.

redo

Accesso

public;

Tipo di ritorno

void;

Descrizione

Ripristina l'ultima modifica annullata;

Note

- Questo metodo è stato ridefinito.

undo

Accesso

public;

Tipo di ritorno

void;

Descrizione

Ripristina la modifica precedente;

Note

- Questo metodo è stato ridefinito.



3.3.16 CommandColorDiffuse (class)

CommandColorDiffuse
+ CommandColorDiffuse(out scene: Scene*, out object: SceneObject*, color: QColor) + redo(): void + undo(): void

Figura 19: Classe CommandColorDiffuse

3.3.16.1 Descrizione

Comando per cambiare il colore di diffusione_G dell'oggetto

3.3.16.2 Utilizzo

Viene utilizzata per gestire i Signal_G riguardanti il colore di diffusione_G dell'oggetto

3.3.16.3 Classi ereditate

- DDDMob :: Model :: Commands :: CommandColor.

3.3.16.4 Metodi

CommandColorDiffuse

Accesso

public;

Descrizione

Costruttore della classe CommandColorDiffuse.

redo

Accesso

public;

Tipo di ritorno

void;

Descrizione

Ripristina l'ultima modifica annullata;

Note

- Questo metodo è stato ridefinito.

undo

Accesso

public;

Tipo di ritorno

void;

Descrizione

Ripristina la modifica precedente;

**Note**

- Questo metodo è stato ridefinito.

3.3.17 CommandEditObject (class)

CommandEditObject
-sceneObject: SceneObject*
+ CommandEditObject(scene: Scene*, sceneObject: SceneObject*)

Figura 20: Classe CommandEditObject

3.3.17.1 Descrizione

Classe che rappresenta il comando di modifica di un oggetto

3.3.17.2 Utilizzo

Viene utilizzata per gestire i Signal_G riguardanti la modifica di un oggetto ed invocare i corretti metodi del Model

3.3.17.3 Classi ereditate

- $\text{DDDMob} :: \text{Model} :: \text{Commands} :: \text{CommandScene}$.

3.3.17.4 Ereditata da

- $\text{DDDMob} :: \text{Model} :: \text{Commands} :: \text{CommandTransform}$;
- $\text{DDDMob} :: \text{Model} :: \text{Commands} :: \text{CommandEditMesh}$;
- $\text{DDDMob} :: \text{Model} :: \text{Commands} :: \text{CommandEditLight}$;
- $\text{DDDMob} :: \text{Model} :: \text{Commands} :: \text{CommandEmission}$;
- $\text{DDDMob} :: \text{Model} :: \text{Commands} :: \text{CommandColor}$.

3.3.17.5 Attributi**Name**

sceneObject;

Accesso

private;

Tipo

SceneObject*;

Descrizione

Riferimento all'oggetto della scena $_G$.



3.3.17.6 Metodi

CommandEditObject

Accesso

public;

Descrizione

Costruttore di CommandEditObject.

3.4 DDDMob::Model::CConverter

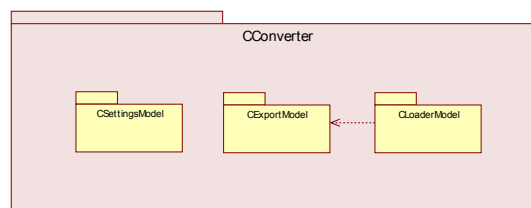


Figura 21: Componente DDDMob::Model::CConverter

3.5 DDDMob::Model::CConverter::CSettingsModel

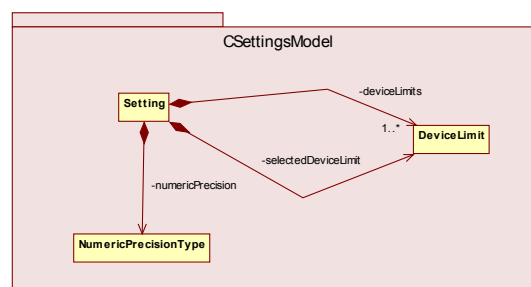


Figura 22: Componente DDDMob::Model::CConverter::CSettingsModel

3.5.1 Setting (class)

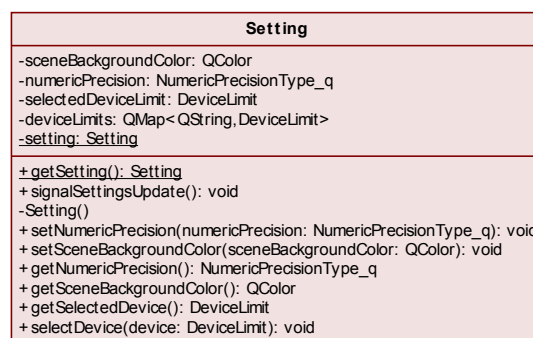


Figura 23: Classe Setting



3.5.1.1 Descrizione

Classe implementata con il Design Pattern_G Singleton che contiene tutte le impostazioni del programma

3.5.1.2 Utilizzo

Viene utilizzata per modificare e recuperare i limiti di importazione, precisione di esportazione e colore di sfondo. Riceve richieste di modifica di alcuni elementi o di restituzione dei dati, e notifica alla vista quando gli stessi sono disponibili o aggiornati

3.5.1.3 Attributi

Name

sceneBackgroundColor;

Accesso

private;

Tipo

QColor;

Descrizione

Colore di sfondo della scena_G.

Name

numericPrecision;

Accesso

private;

Tipo

NumericPrecisionType;

Descrizione

Precisione numerica dell'oggetto esportato.

Name

selectedDeviceLimit;

Accesso

private;

Tipo

DeviceLimit;

Descrizione

Limiti del dispositivo selezionato.

Name

deviceLimits;

Accesso

private;



Tipo

QMap<QString,DeviceLimit>;

Descrizione

Limiti dei dispositivi predefiniti.

Name

setting;

Accesso

private;

Tipo

Setting;

Descrizione

Riferimento a se stesso per l'accesso protetto.

3.5.1.4 Metodi

getSetting

Accesso

public;

Tipo di ritorno

Setting;

Descrizione

Ritorna l'istanza dell'oggetto Setting in modo protetto, secondo il Design Pattern_G Singleton;

Note

- Deve essere un metodo statico.

signalSettingsUpdate

Accesso

signal_G;

Tipo di ritorno

void;

Descrizione

Notifica l'avvenuto aggiornamento delle impostazioni.

Setting

Accesso

private;

Descrizione

Costruttore privato dell'oggetto Setting.



setNumericPrecision

Accesso

public;

Tipo di ritorno

void;

Descrizione

Imposta la precisione numerica dei dati che si desidera per l'esportazione della scena_G.

setSceneBackgroundColor

Accesso

public;

Tipo di ritorno

void;

Descrizione

Imposta il colore di sfondo dell'anteprima della scena_G.

getNumericPrecision

Accesso

public;

Tipo di ritorno

NumericPrecisionType;

Descrizione

Ritorna la precisione numerica correntemente impostata.

getSceneBackgroundColor

Accesso

public;

Tipo di ritorno

QColor;

Descrizione

Ritorna il colore di sfondo dell'anteprima della scena_G correntemente impostato.

getSelectedDevice

Accesso

public;

Tipo di ritorno

DeviceLimit&;

Descrizione

Ritorna un riferimento al dispositivo selezionato.



selectDevice

Accesso

public;

Tipo di ritorno

void;

Descrizione

Seleziona un device come device correntemente selezionato.

3.5.2 DeviceLimit (class)

DeviceLimit
-lightNumber: int -textureSize: int -deviceName: QString
+ DeviceLimit(lightNumber: int, textureSize: int): void

Figura 24: Classe DeviceLimit

3.5.2.1 Descrizione

Classe che rappresenta i limiti di un dispositivo

3.5.2.2 Utilizzo

Viene utilizzata per salvare i dati che rappresentano i limiti di esportazione relativi a un dispositivo

3.5.2.3 Attributi

Name

lightNumber;

Accesso

private;

Tipo

int;

Descrizione

Massimo numero di luci ammesse nella scena_G.

Name

textureSize;

Accesso

private;

Tipo

int;



Descrizione

Massima dimensione delle texture ammissibile.

Name

deviceName;

Accesso

private;

Tipo

QString;

Descrizione

Il nome del dispositivo del quale si vogliono rappresentare i limiti di OpenGL ES_G 2.0.

3.5.2.4 Metodi

DeviceLimit

Accesso

public;

Tipo di ritorno

void;

Descrizione

Costruttore della classe DeviceLimit.

3.5.3 NumericPrecisionType (class)

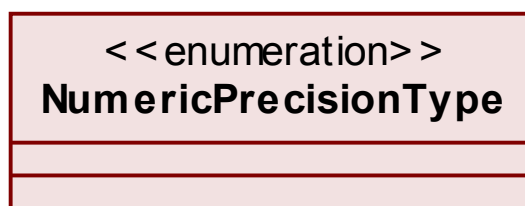


Figura 25: Classe NumericPrecisionType

3.5.3.1 Descrizione

Classe che contiene le diverse tipologie di precisione dei numeri

3.5.3.2 Utilizzo

Viene utilizzata come `Enumerate`



3.6 DDDMob::Model::CConverter::CExportModel

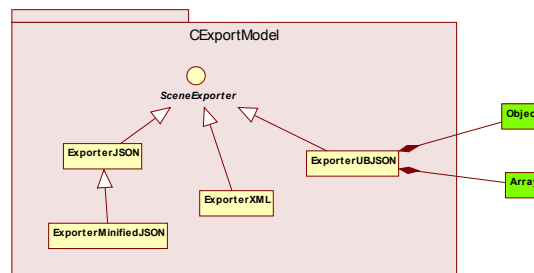


Figura 26: Componente DDDMob::Model::CConverter::CExportModel

3.6.1 SceneExporter (interface)

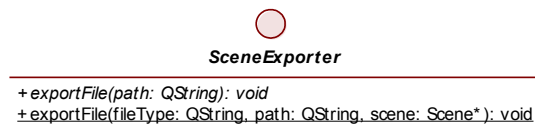


Figura 27: Classe SceneExporter

3.6.1.1 Descrizione

Interfaccia per la componente strategy del Design Pattern_G Strategy per la selezione dell'algoritmo di esportazione della scena_G tra tutti i disponibili

3.6.1.2 Utilizzo

Permette di selezionare dinamicamente ed in modo estensibile l'algoritmo di esportazione della scena_G e di conseguenza il formato del file salvato in uscita

3.6.1.3 Ereditata da

- DDDMob :: Model :: CConverter :: CExportModel :: ExporterUBJSON;
- DDDMob :: Model :: CConverter :: CExportModel :: ExporterXML;
- DDDMob :: Model :: CConverter :: CExportModel :: ExporterJSON.

3.6.1.4 Metodi

exportFile

Accesso

public;

Tipo di ritorno

void;

Descrizione

Esporta un file dato il percorso. Metodo virtuale ed astratto, deve essere ridefinito dalle classi che ereditano da questa, implementando la funzionalità di salvataggio;

**Note**

- Deve essere astratto;
- Questo metodo è stato ridefinito.

exportFile**Accesso**

public;

Tipo di ritorno

void;

Descrizione

Metodo statico che, dato il percorso del file da esportare e l'estensione del formato desiderato, istanzia la classe corretta per esportare la scena_G e ne chiama il metodo exportFile per realizzare la funzione richiesta;

Note

- Deve essere un metodo statico.

3.6.2 ExporterUBJSON (class)

ExporterUBJSON
-arrayUBJSON: Array -objectUBJSON: Object
+ exportFile(path: QString): void # build(scene: Scene*): QByteArray -serializeSceneObject(object: SceneObject*, objectMap: Object*): void -serializeMesh(out mesh: Mesh*, out objectMap: Object*): void -serializeLight(out light: Light*, out objectMap: Object*): void -serialize(vector: QVector3D, prec: NumericPrecisionType): Array* -serialize(vector: QVector2D, prec: NumericPrecisionType): Array* -serialize(list: QList<Keyframe>): Object*

Figura 28: Classe ExporterUBJSON

3.6.2.1 Descrizione

Classe che rappresenta un algoritmo di esportazione della scena_G in un file di formato UBJSON. È uno dei componenti concrete component del Design Pattern_G Strategy

3.6.2.2 Utilizzo

Viene utilizzata per creare un file UBJSON a partire dalla scena_G 3D

3.6.2.3 Classi ereditate

- DDDMob :: Model :: CConverter :: CExportModel :: SceneExporter.

3.6.2.4 Attributi**Name**

arrayUBJSON;

Accesso

private;



Tipo

Array;

Descrizione

Array contenente gli elementi della scena_G da serializzare.

Name

objectUBJSON;

Accesso

private;

Tipo

Object;

Descrizione

Oggetto contenente le animazioni della scena_G da serializzare.

3.6.2.5 Metodi

exportFile

Accesso

public;

Tipo di ritorno

void;

Descrizione

Esporta un file dato il percorso;

Note

- Questo metodo è stato ridefinito.

build

Accesso

protected;

Tipo di ritorno

QByteArray;

Descrizione

Crea lo stream_G di dati per il file UBJSON;

Note

- Deve essere esplicitamente marcati come costanti.



serializeSceneObject

Accesso

private;

Tipo di ritorno

void;

Descrizione

Funzione di utilità, aggiunge a ObjectMap gli attributi propri di SceneObject;

Note

- Deve essere esplicitamente marcati come costanti.

serializeMesh

Accesso

private;

Tipo di ritorno

void;

Descrizione

Funzione d'utilità, aggiunge a ObjectMap gli attributi propri dell'oggetto Mesh;

Note

- Deve essere esplicitamente marcati come costanti.

serializeLight

Accesso

private;

Tipo di ritorno

void;

Descrizione

Funzione d'utilità, aggiunge a ObjectMap gli attributi propri dell'oggetto Light;

Note

- Deve essere esplicitamente marcati come costanti.

serialize

Accesso

private;

Tipo di ritorno

Array*;

Descrizione

Funzione d'utilità, serializza un QVector3D e ritorna un Array della libreria UBJSON;

Note

- Deve essere esplicitamente marcati come costanti.



serialize

Accesso

private;

Tipo di ritorno

Array*;

Descrizione

Funzione d'utilità, serializza un QVector2D e ritorna un Array della libreria UBJSON;

Note

- Deve essere esplicitamente marcati come costanti.

serialize

Accesso

private;

Tipo di ritorno

Object*;

Descrizione

Funzione d'utilità, serializza una QList e ritorna un Object della libreria UBJSON;

Note

- Deve essere esplicitamente marcati come costanti.

3.6.3 ExporterXML (class)

ExporterXML
<pre> +exportFile(path: QString): void #build(out scene: Scene*, out file: QFile*): void -streamSceneObject(object: SceneObject, out writer: QXmlStreamWriter&): void -streamMesh(mesh: Mesh*, out writer: QXmlStreamWriter&): void -streamLight(light: Light*, out stream: QXmlStreamWriter&): void -stream(vector: QVector3D, prec: NumericPrecisionType_q, out stream: QXmlStreamWriter&): void -stream(vector: QVector2D, prec: NumericPrecisionType_q, out stream: QXmlStreamWriter&): void </pre>

Figura 29: Classe ExporterXML

3.6.3.1 Descrizione

Classe che rappresenta un algoritmo di esportazione della scena_G in un file di formato XML_G. È uno dei componenti concrete component del Design Pattern_G Strategy

3.6.3.2 Utilizzo

Viene utilizzata per creare un file XML_G a partire dalla scena_G 3D

3.6.3.3 Classi ereditate

- DDDMob :: Model :: CConverter :: CExportModel :: SceneExporter.



3.6.3.4 Metodi

exportFile

Accesso

public;

Tipo di ritorno

void;

Descrizione

Scrive lo stream_G di dati restituito dal metodo build in un file nel path specificato;

Note

- Deve essere virtuale;
- Questo metodo è stato ridefinito.

build

Accesso

protected;

Tipo di ritorno

void;

Descrizione

Crea lo stream_G di dati per il file in formato XML_G, e lo popola con i dati presenti nella scena_G secondo lo XMLSchema. Per serializzare gli oggetti della scena_G si appoggia ai metodi privati di utilità definiti;

Note

- Deve essere esplicitamente marcati come costanti.

streamSceneObject

Accesso

private;

Tipo di ritorno

void;

Descrizione

Dato uno sceneObject aggiunge i dati dell'oggetto allo stream_G da esportare nell'XML_G;

Note

- Deve essere esplicitamente marcati come costanti.



streamMesh

Accesso

private;

Tipo di ritorno

void;

Descrizione

Data una mesh aggiunge i dati dell'oggetto allo stream_G da esportare nell'XML_G;

Note

- Deve essere esplicitamente marcati come costanti.

streamLight

Accesso

private;

Tipo di ritorno

void;

Descrizione

Data una luce aggiunge i dati dell'oggetto allo stream_G da esportare nell'XML_G;

Note

- Deve essere esplicitamente marcati come costanti.

stream_G

Accesso

private;

Tipo di ritorno

void;

Descrizione

Metodo privato di utilità. Scrive su uno stream_G i campi dati di un vettore 3D nell'XML_G da esportare;

Note

- Deve essere esplicitamente marcati come costanti.

stream_G

Accesso

private;

Tipo di ritorno

void;

Descrizione

Metodo privato di utilità. Scrive su uno stream_G i campi dati di un vettore 2D nell'XML_G da esportare;

Note

- Deve essere esplicitamente marcati come costanti.



3.6.4 ExporterMinifiedJSON (class)

ExporterMinifiedJSON
+ exportFile(path: QString, out scene: Scene*): void

Figura 30: Classe ExporterMinifiedJSON

3.6.4.1 Descrizione

Classe che rappresenta un algoritmo di esportazione della scena_G in un file di formato JSON_G che contiene JSON_G minificato_G. È uno dei componenti concrete component del Design Pattern_G Strategy

3.6.4.2 Utilizzo

Viene utilizzata per creare un file JSON_G che contiene JSON_G minificato_G a partire dalla scena_G 3D

3.6.4.3 Classi ereditate

- DDDMob :: Model :: CConverter :: CExportModel :: ExporterJSON.

3.6.4.4 Metodi

exportFile

Accesso

public;

Tipo di ritorno

void;

Descrizione

Scrive lo stream_G di dati restituito dal metodo build in un file nel path specificato. È richiesto l'utilizzo del metodo build di ExporterJSON per ottenere il JSON_G da minificare;

Note

- Deve essere virtuale;
- Questo metodo è stato ridefinito.

3.6.5 ExporterJSON (class)

ExporterJSON
+ exportFile(path: QString, scene: CLoaderModel::Scene *): void # build(scene: CLoaderModel::Scene *): QByteArray - serializeSceneObject(object: C3DObject::SceneObject *, objectMap: QJsonObject &): void - serializeMesh(mesh: C3DObject::Mesh *, objectMap: QJsonObject &): void - serializeLight(light: C3DObject::Light *, objectMap: QJsonObject &): void - serialize(vector: QVector3D, prec: NumericPrecisionType_q): QJsonArray - serialize(vector: QVector2D, prec: NumericPrecisionType): QJsonArray - serialize(list: QList<Keyframe>): QJsonObject

Figura 31: Classe ExporterJSON



3.6.5.1 Descrizione

Classe che rappresenta un algoritmo di esportazione della scena_G in un file di formato JSON_G. È uno dei componenti concrete component del Design Pattern_G Strategy

3.6.5.2 Utilizzo

Viene utilizzata per creare un file JSON_G a partire dalla scena_G 3D

3.6.5.3 Classi ereditate

- DDDMob :: Model :: CConverter :: CExportModel :: SceneExporter.

3.6.5.4 Ereditata da

- DDDMob :: Model :: CConverter :: CExportModel :: ExporterMinifiedJSON.

3.6.5.5 Metodi

exportFile

Accesso

public;

Tipo di ritorno

void;

Descrizione

Scrive lo stream_G di dati restituito dal metodo build in un file nel path specificato;

Note

- Deve essere virtuale;
- Questo metodo è stato ridefinito.

build

Accesso

protected;

Tipo di ritorno

QByteArray;

Descrizione

Crea lo stream_G di dati per il file in formato JSON_G, e lo popola con i dati presenti nella scena_G secondo il JSONSchema. Per serializzare gli oggetti della scena_G si appoggia ai metodi privati di utilità definiti;

Note

- Deve essere esplicitamente marcati come costanti.



serializeSceneObject

Accesso

private;

Tipo di ritorno

void;

Descrizione

Metodo di utilità. Serializza i dati del SceneObject dato;

Note

- Deve essere esplicitamente marcati come costanti.

serializeMesh

Accesso

private;

Tipo di ritorno

void;

Descrizione

Metodo privato di utilità. Serializza le informazioni proprie di una Mesh in JSON_G e le aggiunge all'oggetto QJsonObject dato;

Note

- Deve essere esplicitamente marcati come costanti.

serializeLight

Accesso

private;

Tipo di ritorno

void;

Descrizione

Metodo privato di utilità. Serializza i valori propri di un oggetto Light in JSON_G e ne aggiunge i valori al QJsonObject dato;

Note

- Deve essere esplicitamente marcati come costanti.

serialize

Accesso

private;

Tipo di ritorno

QJsonArray;

Descrizione

Metodo privato di utilità. Serializza i campi dati di un vettore 3D in JSON_G;

Note

- Deve essere esplicitamente marcati come costanti.



serialize

Accesso

private;

Tipo di ritorno

QJsonArray;

Descrizione

Metodo privato di utilità. Serializza i campi dati di un vettore 2D in JSON_G ;

Note

- Deve essere esplicitamente marcati come costanti.

serialize

Accesso

private;

Tipo di ritorno

QJsonObject;

Descrizione

Metodo privato di utilità. Serializza le proprietà di un Keyframe_G in JSON_G ;

Note

- Deve essere esplicitamente marcati come costanti.

3.7 DDDMob::Model::CConverter::CLoaderModel

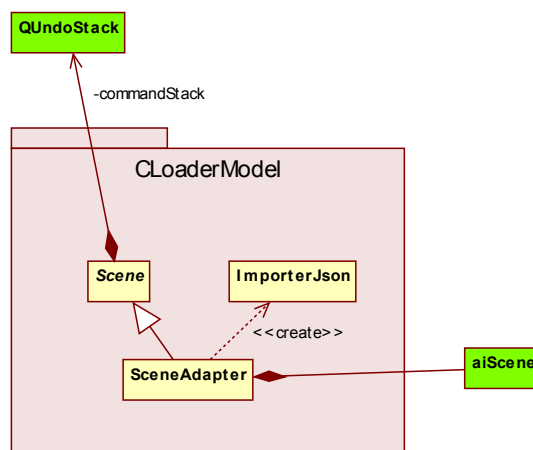


Figura 32: Componente DDDMob::Model::CConverter::CLoaderModel



3.7.1 Scene (abstract)

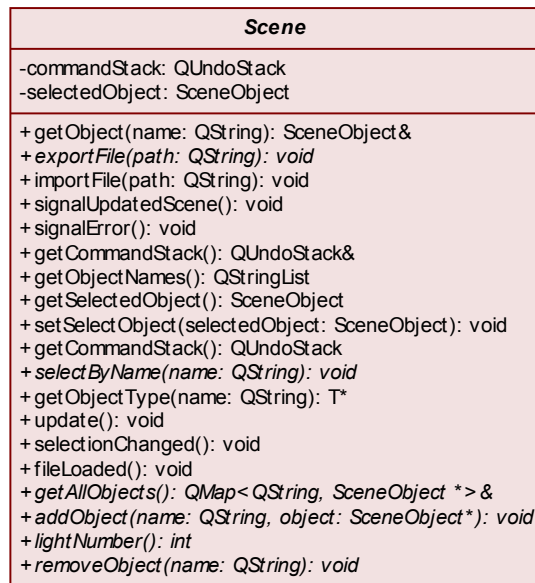


Figura 33: Classe Scene

3.7.1.1 Descrizione

Classe che rappresenta la scena_G 3D che contiene le mesh e le luci. È il componente receiver del Design Pattern_G Command, componente context del Design Pattern_G Strategy e componente target del Design Pattern_G Adapter. Internamente contiene la lista di comandi eseguiti sulla scena_G 3D

3.7.1.2 Utilizzo

Interfaccia per gestire le proprietà degli oggetti 3D. Permette la selezione di un oggetto della scena_G e la sua modifica. Contiene lo stato dei comandi eseguiti sulla scena_G così da poter annullare e ripristinare le modifiche effettuate. Permette l'esportazione in vari formati, selezionati grazie al Design Pattern_G Strategy, e l'importazione nei vari formati permessi dalla libreria esterna utilizzata

3.7.1.3 Ereditata da

- DDDMob :: Model :: CConverter :: CLoaderModel :: SceneAdapter.

3.7.1.4 Attributi

Name

commandStack;

Accesso

private;

Tipo

QUndoStack;



Descrizione

Lista delle azioni effettuate.

Name

selectedObject;

Accesso

private;

Tipo

SceneObject;

Descrizione

L'oggetto 3D attualmente selezionato.

3.7.1.5 Metodi

getObject

Accesso

public;

Tipo di ritorno

SceneObject&;

Descrizione

Ritorna il riferimento all'oggetto con il nome richiesto, o un riferimento nullo se l'oggetto non esiste;

Note

- Deve essere esplicitamente marcati come costanti;
- Questo metodo è stato ridefinito.

exportFile

Accesso

public;

Tipo di ritorno

void;

Descrizione

Esporta il file dato il percorso;

Note

- Deve essere astratto.

importFile

Accesso

public;

Tipo di ritorno

void;



Descrizione

Importa un file dato il percorso. È richiesto il caricamento del file grazie alla libreria esterna Assimp, e la creazione dei corretti SceneObject a partire dalle strutture dati di Assimp. È richiesto l'uso dei metodi privati di utilità buildObjects e buildLights per costruire i corretti oggetti di tipo Mesh e Light partendo dai dati restituiti da Assimp. È necessaria inoltre l'applicazione dei limiti di importazione della scena_G grazie all'invocazione del metodo importLimits;

Note

- Questo metodo è stato ridefinito.

signalUpdatedScene

Accesso

signal_G;

Tipo di ritorno

void;

Descrizione

Notifica l'avvenuto aggiornamento della scena_G;

Note

- Questo metodo è stato ridefinito.

signalError

Accesso

signal_G;

Tipo di ritorno

void;

Descrizione

Manda un segnale per indicare che vi è stato un errore interno alla scena_G;

Note

- Questo metodo è stato ridefinito.

getCommandStack

Accesso

public;

Tipo di ritorno

QUndoStack&;

Descrizione

Ritorna il riferimento all'oggetto contenente la lista delle operazioni effettuate;

Note

- Deve essere esplicitamente marcati come costanti;
- Questo metodo è stato ridefinito.



getObjectNames

Accesso

public;

Tipo di ritorno

QStringList;

Descrizione

Restituisce una lista con i nomi degli oggetti presenti nella scena_G;

Note

- Questo metodo è stato ridefinito.

getSelectedObject

Accesso

public;

Tipo di ritorno

SceneObject;

Descrizione

Ritorna l'oggetto selezionato.

setSelectObject

Accesso

public;

Tipo di ritorno

void;

Descrizione

Imposta l'oggetto selezionato.

getCommandStack

Accesso

public;

Tipo di ritorno

QUndoStack;

Descrizione

Ritorna lo stack dei comandi.

selectByName

Accesso

public;

Tipo di ritorno

void;



Descrizione

Imposta come selezionato l'oggetto con il nome specificato, o non modifica nulla se l'oggetto non è presente nella scena_G;

Note

- Deve essere astratto.

getObjectType

Accesso

public;

Tipo di ritorno

T*;

Descrizione

Ritorna il tipo dell'oggetto;

Note

- Deve essere esplicitamente marcati come costanti.

update

Accesso

public;

Tipo di ritorno

void;

Descrizione

Emette il signal_G di update.

selectionChanged

Accesso

public;

Tipo di ritorno

void;

Descrizione

Invia un segnale quando viene selezionato un nuovo oggetto.

fileLoaded

Accesso

public;

Tipo di ritorno

void;

Descrizione

Emette un segnale quando è stato caricato un file.



getAllObjects

Accesso

public;

Tipo di ritorno

QMap<QString, SceneObject *>&;

Descrizione

Ritorna una mappa associativa con nome e riferimento all'oggetto per tutti gli oggetti della scena_G;

Note

- Deve essere astratto;
- Deve essere esplicitamente marcati come costanti.

addObject

Accesso

public;

Tipo di ritorno

void;

Descrizione

Aggiunge un oggetto alla scena_G;

Note

- Deve essere astratto.

lightNumber

Accesso

public;

Tipo di ritorno

int;

Descrizione

Restituisce il numero di luci nella scena_G;

Note

- Deve essere astratto;
- Deve essere esplicitamente marcati come costanti.

removeObject

Accesso

public;

Tipo di ritorno

void;

**Descrizione**

Rimuove un oggetto dalla scena_G senza cancellarlo;

Note

- Deve essere astratto.

3.7.2 SceneAdapter (class)

SceneAdapter
-objects: QMap<QString,SceneObject*> -sceneAdaptee: aiScene*
+ SceneAdapter() + getObject(out name: QString): SceneObject * + exportFile(path: QString): void + importFile(path: QString): void + signalUpdatedScene(): void + signalError(): void + getObjectNames(): QStringList + addLight(): void + selectByName(name: QString): void -buildLights(lightList: aiLight**, lightNumber: int): void -buildMeshes(meshList: aiMesh**, meshNumber: int, materialList: aiMaterial**, materialNumber: int, animationList: aiAnimation**, animationNumber: int): void -meshCreator(mesh: aiMesh*, material: aiMaterial*): Mesh* -lightCreator(light: aiLight*): Light* -importLimits(): void + lightNumber(): int + removeObject(name: QString): void + addObject(name: QString, object: SceneObject*): void + getAllObjects(): QMap<QString, SceneObject *> &

Figura 34: Classe SceneAdapter

3.7.2.1 Descrizione

Classe che viene utilizzata come adattatore per la libreria esterna Assimp. Rappresenta il componente adapter del Design Pattern_G Adapter

3.7.2.2 Utilizzo

Viene utilizzata come adattatore tra quanto esposto dalla libreria esterna Assimp e la classe Scene

3.7.2.3 Classi ereditate

- DDDMob :: Model :: CConverter :: CLoaderModel :: Scene.

3.7.2.4 Attributi**Name**

objects;

Accesso

private;

Tipo

QMap<QString,SceneObject*>;

Descrizione

Mappa tra i nomi e i riferimenti agli oggetti contenuti nella scena_G.



Name

sceneAdaptee;

Accesso

private;

Tipo

aiScene*;

Descrizione

Riferimento alla struttura dati di Assimp che rappresenta la scena_G da adattare.

3.7.2.5 Metodi

SceneAdapter

Accesso

public;

Descrizione

Costruttore della classe SceneAdapter. Inizializza una mappa vuota per il campo dati objects.

getObject

Accesso

public;

Tipo di ritorno

SceneObject *;

Descrizione

Ritorna il riferimento all'oggetto con il nome richiesto, o un riferimento nullo se l'oggetto non esiste;

Note

- Deve essere esplicitamente marcati come costanti;
- Questo metodo è stato ridefinito.

exportFile

Accesso

public;

Tipo di ritorno

void;

Descrizione

Dato il percorso, invoca il metodo statico di SceneExporter per la scelta dell'algoritmo di esportazione, fornendo un riferimento alla mappa degli oggetti della scena_G. Ottiene il salvataggio della scena_G nel percorso specificato, con il formato scelto dall'utente;

Note

- Questo metodo è stato ridefinito.



importFile

Accesso

public;

Tipo di ritorno

void;

Descrizione

Importa un file dato il percorso. È richiesto il caricamento del file grazie alla libreria esterna Assimp, e la creazione dei corretti SceneObject a partire dalle strutture dati di Assimp. È richiesto l'uso dei metodi privati di utilità buildObjects e buildLights per costruire i corretti oggetti di tipo Mesh e Light partendo dai dati restituiti da Assimp. È necessaria inoltre l'applicazione dei limiti di importazione della scena_G grazie all'invocazione del metodo importLimits;

Note

- Questo metodo è stato ridefinito.

signalUpdatedScene

Accesso

signal_G;

Tipo di ritorno

void;

Descrizione

Notifica l'avvenuto aggiornamento della scena_G;

Note

- Questo metodo è stato ridefinito.

signalError

Accesso

signal_G;

Tipo di ritorno

void;

Descrizione

Notifica un errore interno alla scena_G;

Note

- Questo metodo è stato ridefinito.



getObjectNames

Accesso

public;

Tipo di ritorno

QStringList;

Descrizione

Restituisce una lista con i nomi degli oggetti presenti nella scena_G;

Note

- Deve essere esplicitamente marcati come costanti;
- Questo metodo è stato ridefinito.

addLight

Accesso

public;

Tipo di ritorno

void;

Descrizione

Crea una nuova luce con i parametri di default, e la aggiunge alla scena_G con il nome 3DMobLight-

Note

- Questo metodo è stato ridefinito.

selectByName

Accesso

public;

Tipo di ritorno

void;

Descrizione

Imposta come selezionato l'oggetto con il nome specificato, o non modifica nulla se l'oggetto non è presente nella scena_G;

Note

- Questo metodo è stato ridefinito.

buildLights

Accesso

private;

Tipo di ritorno

void;

Descrizione

Costruisce tutti gli oggetti di tipo luce grazie al metodo di utilità lightCreator, applicando anche i limiti di importazione ad esse associati.



buildMeshes

Accesso

private;

Tipo di ritorno

void;

Descrizione

Costruisce tutte le Mesh presenti nella scena_G importata, grazie anche al metodo di utilità meshCreator, che crea un singolo oggetto Mesh.

meshCreator

Accesso

private;

Tipo di ritorno

Mesh*;

Descrizione

Data una luce nella struttura dati di Assimp, crea un oggetto Mesh e ne imposta i parametri corretti. Aggiunge poi il riferimento alla mesh creata nella mappa di oggetti della scena_G.

lightCreator

Accesso

private;

Tipo di ritorno

Light*;

Descrizione

Data una luce nella struttura dati di Assimp, crea un oggetto Light e ne imposta i parametri corretti. Aggiunge poi il riferimento alla luce creata nella mappa di oggetti della scena_G.

importLimits

Accesso

private;

Tipo di ritorno

void;

Descrizione

Applica i limiti di importazione riguardanti le texture ad ogni oggetto presente nella scena_G. È richiesto il ridimensionamento di ogni texture che ecceda i limiti massimi di dimensione in modo che vengano mantenute le proporzioni originarie e l'istogramma dei colori, il suo salvataggio in un nuovo file immagine che non sovrasciva il file precedente, e l'impostazione della nuova texture nella Mesh oggetto dell'applicazione dei limiti.



lightNumber

Accesso

public;

Tipo di ritorno

int;

Descrizione

Restituisce il numero di luci presenti nella scena_G;

Note

- Deve essere esplicitamente marcati come costanti;
- Questo metodo è stato ridefinito.

removeObject

Accesso

public;

Tipo di ritorno

void;

Descrizione

Rimuove un oggetto dalla scena_G senza cancellarlo;

Note

- Questo metodo è stato ridefinito.

addObject

Accesso

public;

Tipo di ritorno

void;

Descrizione

Aggiunge il riferimento dell'oggetto alla mappa di oggetti della scena_G con il nome passato come parametro;

Note

- Questo metodo è stato ridefinito.

getAllObjects

Accesso

public;

Tipo di ritorno

QMap<QString, SceneObject *>&;

**Descrizione**

Ritorna una mappa associativa con nome e riferimento all'oggetto per tutti gli oggetti della scena_G;

Note

- Deve essere esplicitamente marcati come costanti;
- Questo metodo è stato ridefinito.

3.7.3 ImporterJson (class)

ImporterJson
-path: QString
+importFile(path: QString path, scene: Scene*): void -readObject(tree: boost::property_tree::ptree&): C3DObject::SceneObject* -readLight(tree: boost::property_tree::ptree&): C3DObject::Light* -readMesh(tree: boost::property_tree::ptree&): C3DObject::Mesh* -readObjectProperties(tree: boost::property_tree::ptree&, object: C3DObject::SceneObject*): void

Figura 35: Classe ImporterJson

3.7.3.1 Descrizione

Classe che rappresenta un algoritmo di importazione della scena_G in un file di formato JSON_G

3.7.3.2 Utilizzo

Viene utilizzata per creare tutti gli oggetti della scena_G a partire da un file JSON_G

3.7.3.3 Attributi**Name**

path;

Accesso

private;

Tipo

QString;

Descrizione

Percorso del file da importare.

3.7.3.4 Metodi**importFile****Accesso**

public;

Tipo di ritorno

void;



Descrizione

Metodo che permette l'importazione del file nel percorso specificato nel parametro. Deve leggere il file ed estrarre le informazioni degli oggetti contenute all'interno, utilizzando anche i metodi privati di utilità definiti per creare i tipi di oggetto popolati correttamente.

readObject

Accesso

private;

Tipo di ritorno

C3DObject::SceneObject*;

Descrizione

Legge un oggetto dal file.

readLight

Accesso

private;

Tipo di ritorno

C3DObject::Light*;

Descrizione

Legge una legge dal file.

readMesh

Accesso

private;

Tipo di ritorno

C3DObject::Mesh*;

Descrizione

Legge una Mesh dal file.

readObjectProperties

Accesso

private;

Tipo di ritorno

void;

Descrizione

Legge le proprietà di un oggetto.



3.8 DDDMob::Controller

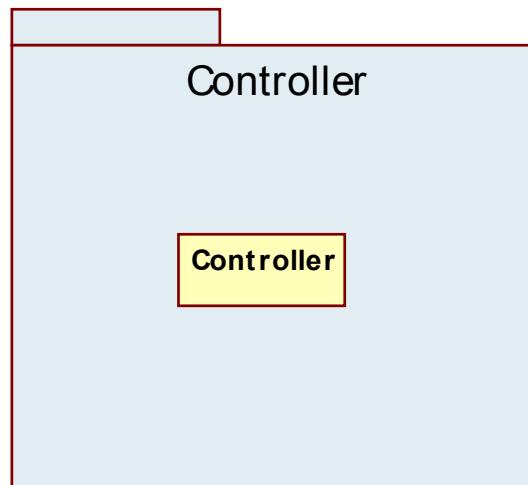


Figura 36: Componente DDDMob::Controller

3.8.1 Controller (class)

Controller
<u>-sceneModel: Scene*</u> <u>-controller: Controller*</u>
+ slotRotation(vector: QVector3D): void + slotScale(vector: QVector3D): void + slotPosition(vector: QVector3D): void + slotAddLight(): void + slotColorDiffuse(color: QColor): void + slotColorSpecular(color: QColor): void + slotColorEmission(color: QColor): void + slotSelectObject(objectName: QString): void + slotLightType(lightType: LightType): void + slotMeshShininess(shininess: double): void + slotEmission(emission: double): void + slotSettings(settings: Setting): void + slotOpen(path: QString): void + slotSave(path: QString, type: QString): void + slotViewAbout(): void + slotViewSave(): void + slotViewOpen(): void + slotViewHelp(): void + slotViewSettings(): void -Controller() <u>+ getController(): void</u> <u>+ getScene(): Scene*</u> + start3DMob(): void

Figura 37: Classe Controller



3.8.1.1 Descrizione

Classe che rappresenta il componente controller del Design Pattern_G MVC ed il componente client del Design Pattern_G Command

3.8.1.2 Utilizzo

Gestisce i Signal_G inviati dalla View ed agisce nel modo corretto. Genera i comandi da eseguire sulla scena_G 3D e memorizza tali comandi nel componente Scene del Model. Concorre nell'applicare le modifiche all'oggetto selezionato della scena_G 3D

3.8.1.3 Attributi

Name

sceneModel;

Accesso

private;

Tipo

Scene*;

Descrizione

Riferimento alla scena_G del modello.

Name

controller;

Accesso

private;

Tipo

Controller*;

Descrizione

Riferimento a se stesso per l'accesso protetto.

3.8.1.4 Metodi

slotRotation

Accesso

public;

Tipo di ritorno

void;

Descrizione

riceve la richiesta di rotazione.



slotScale

Accesso

public;

Tipo di ritorno

void;

Descrizione

riceve la richiesta di rotazione.

slotPosition

Accesso

public;

Tipo di ritorno

void;

Descrizione

Riceve la richiesta di spostamento.

slotAddLight

Accesso

public;

Tipo di ritorno

void;

Descrizione

riceve la richiesta di aggiunta luce.

slotColorDiffuse

Accesso

public;

Tipo di ritorno

void;

Descrizione

riceve la richiesta di cambio colore di diffusione_G.

slotColorSpecular

Accesso

public;

Tipo di ritorno

void;

Descrizione

riceve la richiesta di modifica del colore speculare_G.



slotColorEmission

Accesso

public;

Tipo di ritorno

void;

Descrizione

riceve la richiesta di modifica del colore di emissione.

slotSelectObject

Accesso

public;

Tipo di ritorno

void;

Descrizione

riceve la richiesta di modifica dell'oggetto selezionato.

slotLightType

Accesso

public;

Tipo di ritorno

void;

Descrizione

riceve la richiesta di modifica del tipo della luce.

slotMeshShininess

Accesso

public;

Tipo di ritorno

void;

Descrizione

riceve la richiesta di modifica della lucentezza del mesh.

slotEmission

Accesso

public;

Tipo di ritorno

void;

Descrizione

riceve la richiesta di modifica dell'emissione dell'oggetto.



slotSettings

Accesso

public;

Tipo di ritorno

void;

Descrizione

Riceve richieste sulle impostazioni del sistema.

slotOpen

Accesso

public;

Tipo di ritorno

void;

Descrizione

Riceve la richiesta di apertura di un file.

slotSave

Accesso

public;

Tipo di ritorno

void;

Descrizione

Riceve la richiesta di salvataggio della scena_G su file.

slotViewAbout

Accesso

public;

Tipo di ritorno

void;

Descrizione

Riceve il segnale per visualizzare le informazioni sul programma.

slotViewSave

Accesso

public;

Tipo di ritorno

void;

Descrizione

Richiede l'apertura della finestra di salvataggio.



slotViewOpen

Accesso

public;

Tipo di ritorno

void;

Descrizione

Richiede l'apertura della finestra di apertura file.

slotViewHelp

Accesso

public;

Tipo di ritorno

void;

Descrizione

Richiede l'apertura della finestra di aiuto.

slotViewSettings

Accesso

public;

Tipo di ritorno

void;

Descrizione

Richiede l'apertura della finestra di impostazione delle preferenze del programma.

Controller

Accesso

private;

Descrizione

Costruttore privato della classe Controller.

getController

Accesso

public;

Tipo di ritorno

void;

Descrizione

Restituisce un riferimento a se stesso in modo protetto;

Note

- Deve essere un metodo statico.



getScene

Accesso

public;

Tipo di ritorno

Scene*;

Descrizione

Ritorna il puntatore alla scena_G corrente;

Note

- Deve essere esplicitamente marcati come costanti.

start3DMob

Accesso

public;

Tipo di ritorno

void;

Descrizione

Avvia la finestra principale dell'applicazione.

3.9 DDDMob::View

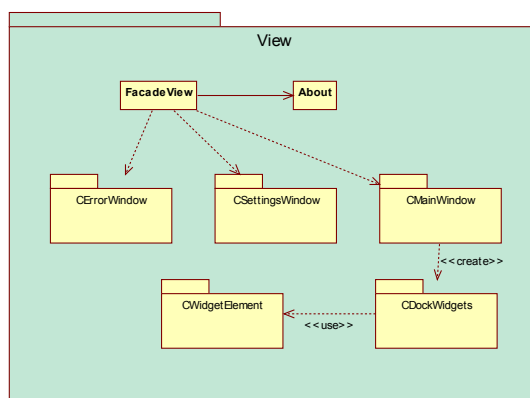


Figura 38: Componente DDDMob::View



3.9.1 FacadeView (class)

FacadeView
-settingsWindow: CSettingsWindow::SettingsWindow -mainWindow: CMainWindow::MainWindow <u>+getFacadeView: FacadeView&</u> -errorMessageBox: CErrorWindows::ErrorMessageBox -about: About
+getMainWindow(): QMainWindow * +showHelpWindow(): void +showSettings(): void +showAbout(): void +showMainWindow(scene: Scene*): void -FacadeView() +operator=(): void -FacadeView(facade: FacadeView const&) -operator=(facade: FacadeView const&): void +getErrorMessageBox(): ErrorMessageBox *

Figura 39: Classe FacadeView

3.9.1.1 Descrizione

Classe che rappresenta la Vista. È la componente facade del Design Pattern_G Facade, ed è implementata con il Design Pattern_G Singleton

3.9.1.2 Utilizzo

Viene utilizzata per accedere alla vista in modo protetto, in quanto espone i metodi necessari per l'interazione dall'esterno, nascondendo l'implementazione e la struttura

3.9.1.3 Attributi

Name

settingsWindow;

Accesso

private;

Tipo

CSettingsWindow::SettingsWindow;

Descrizione

Riferimento all'interfaccia per le impostazioni del programma.

Name

mainWindow;

Accesso

private;

Tipo

CMainWindow::MainWindow;

Descrizione

Riferimento alla finestra principale.



Name

getFacadeView;

Accesso

public;

Tipo

FacadeView&;

Descrizione

Ritorna il riferimento protetto alla Vista.

Name

errorMessageBox;

Accesso

private;

Tipo

CErrorWindows::ErrorMessageBox;

Descrizione

Riferimento protetto alla finestra di informazione di errore.

Name

about;

Accesso

private;

Tipo

About;

Descrizione

Riferimento alla finestra di dialogo con le informazioni sull'applicazione.

3.9.1.4 Metodi

getMainWindow

Accesso

public;

Tipo di ritorno

QMainWindow *;

Descrizione

Ritorna il riferimento protetto alla finestra principale;

Note

- Deve essere esplicitamente marcati come costanti.



showHelpWindow

Accesso

public;

Tipo di ritorno

void;

Descrizione

Mostra la finestra che permette l'accesso al sistema di aiuto.

showSettings

Accesso

public;

Tipo di ritorno

void;

Descrizione

Mostra la finestra con le informazioni sulle impostazioni del programma.

showAbout

Accesso

public;

Tipo di ritorno

void;

Descrizione

Mostra la finestra con le informazioni sul programma.

showMainWindow

Accesso

public;

Tipo di ritorno

void;

Descrizione

Mostra la finestra principale del programma.

FacadeView

Accesso

private;

Descrizione

Costruttore della classe.



operator=

Accesso

public;

Tipo di ritorno

void;

Descrizione

Ridefinizione dell'operatore di assegnazione.

FacadeView

Accesso

private;

Descrizione

Costrutture della classe.

operator=

Accesso

private;

Tipo di ritorno

void;

Descrizione

Ridefinizione dell'operatore =.

getErrorMessageBox

Accesso

public;

Tipo di ritorno

ErrorMessageBox *;

Descrizione

Ritorna un riferimento a una finestra di messaggio di errore;

Note

- Deve essere esplicitamente marcati come costanti.



3.9.2 About (class)



Figura 40: Classe About

3.9.2.1 Descrizione

Classe che rappresenta la finestra contenente le informazioni sul programma e sui suoi creatori

3.9.2.2 Utilizzo

Viene utilizzata per visualizzare le informazioni sul programma e sui suoi creatori, oltre che la versione del prodotto e delle librerie utilizzate

3.9.2.3 Classi ereditate

- QDialog.

3.9.2.4 Metodi

About

Accesso

public;

Descrizione

Costruttore di About.



3.10 DDDMob::View::CDockWidgets

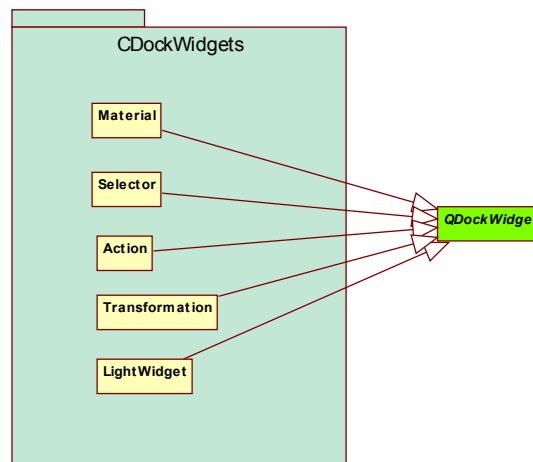


Figura 41: Componente DDDMob::View::CDockWidgets

3.10.1 Material (class)

Material
-materialDiffusionColor: ColorPicker -materialSpecularColor: MaterialColor -materialEmissionColor: MaterialColor -shininess: QSlider -emission: QSlider
+ setDiffuseColor(color: QColor): void + getDiffuseColor(): QColor + getSpecularColor(): QColor + getEmissionColor(): QColor + setEmissionColor(color: QColor): void + getShininess(): double + setShininess(value: double): void + Material(parent = 0: QWidget *) + setSpecularColor(color: QColor): void + update(): void

Figura 42: Classe Material

3.10.1.1 Descrizione

Classe che rappresenta il widget contenente le informazioni sul materiale

3.10.1.2 Utilizzo

Viene utilizzata per permettere di cambiare le caratteristiche del materiale dell'oggetto. Emette un Signal_G in seguito all'interazione dell'utente con l'elemento dell'interfaccia grafica



3.10.1.3 Classi ereditate

- QDockWidget.

3.10.1.4 Attributi

Name

materialDiffusionColor;

Accesso

private;

Tipo

ColorPicker;

Descrizione

Elemento grafico che permette di modificare il colore di diffusione_G del materiale.

Name

materialSpecularColor;

Accesso

private;

Tipo

MaterialColor;

Descrizione

Elemento grafico che permette di modificare il colore speculare_G del materiale.

Name

materialEmissionColor;

Accesso

private;

Tipo

MaterialColor;

Descrizione

Elemento grafico che permette di modificare il colore di emissione del materiale.

Name

shininess;

Accesso

private;

Tipo

QSlider;

Descrizione

Elemento grafico che permette di modificare il valore della luminosità del materiale.



Name

emission;

Accesso

private;

Tipo

QSlider;

Descrizione

Elemento grafico che permette di modificare il valore di emissione del materiale.

3.10.1.5 Metodi

setDiffuseColor

Accesso

public;

Tipo di ritorno

void;

Descrizione

Imposta il colore di diffusione_G;

Note

- Deve essere esplicitamente marcati come costanti.

getDiffuseColor

Accesso

public;

Tipo di ritorno

QColor;

Descrizione

Ritorna il colore di diffusione_G selezionato;

Note

- Deve essere esplicitamente marcati come costanti.

getSpecularColor

Accesso

public;

Tipo di ritorno

QColor;

Descrizione

Ritorna il colore speculare_G selezionato;

Note

- Deve essere esplicitamente marcati come costanti.



getEmissionColor

Accesso

public;

Tipo di ritorno

QColor;

Descrizione

Ritorna il colore di emissione selezionato;

Note

- Deve essere esplicitamente marcati come costanti.

setEmissionColor

Accesso

public;

Tipo di ritorno

void;

Descrizione

Imposta il colore di emissione;

Note

- Deve essere esplicitamente marcati come costanti.

getShininess

Accesso

public;

Tipo di ritorno

double;

Descrizione

Ritorna il valore di lucentezza del materiale;

Note

- Deve essere esplicitamente marcati come costanti.

setShininess

Accesso

public;

Tipo di ritorno

void;

Descrizione

Imposta il valore di lucentezza.



Material

Accesso

public;

Descrizione

Costruttore per la classe Material.

setSpecularColor

Accesso

public;

Tipo di ritorno

void;

Descrizione

Imposta il colore speculare_G.

update

Accesso

public;

Tipo di ritorno

void;

Descrizione

Metodo per aggiornare il materiale.

3.10.2 Selector (class)

Selector
-objectSelector: QListWidget
+ Selector(parent: QWidget*) + updateObjects(scene: const Scene*): void + autoUpdateObjects(): void + updateSelection(): void

Figura 43: Classe Selector

3.10.2.1 Descrizione

Classe che rappresenta il widget contenente il selettore di oggetti

3.10.2.2 Utilizzo

Viene utilizzata per selezionare un oggetto della scena_G. Emette un Signal_G in seguito all'interazione dell'utente con l'elemento dell'interfaccia grafica



3.10.2.3 Classi ereditate

- QDockWidget.

3.10.2.4 Attributi

Name

objectSelector;

Accesso

private;

Tipo

QListWidget;

Descrizione

Elemento grafico che permette di selezionare un oggetto.

3.10.2.5 Metodi

Selector

Accesso

public;

Descrizione

Costruttore della classe Selector.

updateObjects

Accesso

public;

Tipo di ritorno

void;

Descrizione

Metodo che aggiorna la lista degli oggetti presenti nella scena_G.

autoUpdateObjects

Accesso

public;

Tipo di ritorno

void;

Descrizione

Metodo che aggiorna automaticamente gli elementi presenti nella scena_G.

**updateSelection****Accesso**

public;

Tipo di ritorno

void;

Descrizione

Metodo che recupera l'elemento selezionato nella scena_G e lo aggiorna nella lista di oggetti.

3.10.3 LightWidget (class)

LightWidget
-addLight: QPushButton -omni: QRadioButton -spot: QRadioButton
+ LightWidget(parent = 0: QWidget *) + getLightType(): void + update(): void + lightTypeChanged(): void

Figura 44: Classe LightWidget

3.10.3.1 Descrizione

Classe che rappresenta il widget che permette di interagire con le luci della scena_G e di aggiungerne di nuove

3.10.3.2 Utilizzo

Viene utilizzata per interagire con le luci della scena_G o aggiungerne una nuova. Emette un Signal_G in seguito all'interazione dell'utente con l'elemento dell'interfaccia grafica

3.10.3.3 Classi ereditate

- QDockWidget.

3.10.3.4 Attributi**Name**

addLight;

Accesso

private;

Tipo

QPushButton;



Descrizione

Elemento grafico che permette di aggiungere una luce.

Name

omni;

Accesso

private;

Tipo

QRadioButton;

Descrizione

Oggetto grafico per gestire la tipologia della luce.

Name

spot;

Accesso

private;

Tipo

QRadioButton;

Descrizione

Elemento grafico per la gestione della luce di tipo spot.

3.10.3.5 Metodi

LightWidget

Accesso

public;

Descrizione

Costruttore della classe LightWidget.

getLightType

Accesso

public;

Tipo di ritorno

void;

Descrizione

Ritorna il tipo di luce;

Note

- Deve essere esplicitamente marcati come costanti.



update

Accesso

public;

Tipo di ritorno

void;

Descrizione

Aggiorna il tipo di luce.

lightTypeChanged

Accesso

signal_G;

Tipo di ritorno

void;

Descrizione

Metodo eseguito al cambio del tipo di luce.

3.10.4 Action (class)

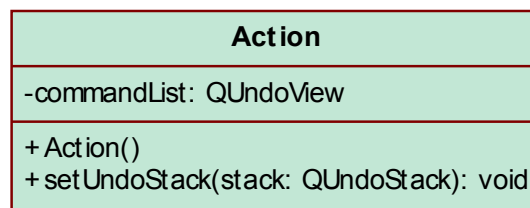


Figura 45: Classe Action

3.10.4.1 Descrizione

Classe che rappresenta il widget che consente di visualizzare le azioni effettuate, di annullarle e ripristinarle

3.10.4.2 Utilizzo

Viene utilizzata per visualizzare le azioni effettuate, annullarle e ripristinarle. Emette un Signal_G in seguito all'interazione dell'utente con l'elemento dell'interfaccia grafica

3.10.4.3 Classi ereditate

- QDockWidget.

3.10.4.4 Attributi

Name

commandList;



Accesso

private;

Tipo

QUndoView;

Descrizione

Elemento grafico che permette di visualizzare le operazioni effettuate.

3.10.4.5 Metodi

Action

Accesso

public;

Descrizione

Costruttore della classe.

setUndoStack

Accesso

public;

Tipo di ritorno

void;

Descrizione

Imposta il range di comandi memorizzati.

3.10.5 Transformation (class)

Transformation
-scale: CWidgetElement::AxisSlider -rotation: CWidgetElement::AxisSlider -translation: CWidgetElement::AxisSlider
+ getPosition(): QVector3D + getRotation(): QVector3D + getScale(): QVector3D + setPosition(position: QVector3D): void + setRotation(rotation: QVector3D): void + setScale(scale: QVector3D): void + update(): void + scaleChanged(scale: QVector3D): void + positionChanged(position: QVector3D): void + rotationChanged(rotation: QVector3D): void

Figura 46: Classe Transformation

3.10.5.1 Descrizione

Classe che rappresenta un widget contenente gli slider per modificare alcune caratteristiche dell'oggetto



3.10.5.2 Utilizzo

Viene utilizzata per permettere di applicare una trasformazione alla scena_G. Emette un Signal_G in seguito all'interazione dell'utente con l'elemento dell'interfaccia grafica

3.10.5.3 Classi ereditate

- QDockWidget.

3.10.5.4 Attributi

Name

scale;

Accesso

private;

Tipo

CWidgetElement::AxisSlider;

Descrizione

Elemento grafico che permette di modificare la dimensione.

Name

rotation;

Accesso

private;

Tipo

CWidgetElement::AxisSlider;

Descrizione

Elemento grafico che permette di modificare la rotazione dell'elemento.

Name

translation;

Accesso

private;

Tipo

CWidgetElement::AxisSlider;

Descrizione

Elemento grafico che permette di modificare la traslazione dell'elemento.

3.10.5.5 Metodi

getPosition

Accesso

public;

Tipo di ritorno

QVector3D;



Descrizione

Ritorna il vettore rappresentante la posizione indicata;

Note

- Deve essere esplicitamente marcati come costanti.

getRotation

Accesso

public;

Tipo di ritorno

QVector3D;

Descrizione

Ritorna il vettore rappresentante la rotazione indicata;

Note

- Deve essere esplicitamente marcati come costanti.

getScale

Accesso

public;

Tipo di ritorno

QVector3D;

Descrizione

Ritorna il vettore rappresentante la dimensione indicata;

Note

- Deve essere esplicitamente marcati come costanti.

setPosition

Accesso

public;

Tipo di ritorno

void;

Descrizione

Imposta i valori visualizzati riguardanti la posizione in base al vettore passato.

setRotation

Accesso

public;

Tipo di ritorno

void;

Descrizione

Imposta i valori visualizzati riguardanti la rotazione in base al vettore passato.



setScale

Accesso

public;

Tipo di ritorno

void;

Descrizione

Imposta i valori visualizzati riguardanti la dimensione in base al vettore passato.

update

Accesso

public;

Tipo di ritorno

void;

Descrizione

Aggiorna gli AxisSlider a seconda del modello.

scaleChanged

Accesso

signal_G;

Tipo di ritorno

void;

Descrizione

E' stata cambiata la scala.

positionChanged

Accesso

signal_G;

Tipo di ritorno

void;

Descrizione

E' cambiata la posizione.

rotationChanged

Accesso

signal_G;

Tipo di ritorno

void;

Descrizione

E' cambiata la rotazione.



3.11 DDDMob::View::CMainWindow

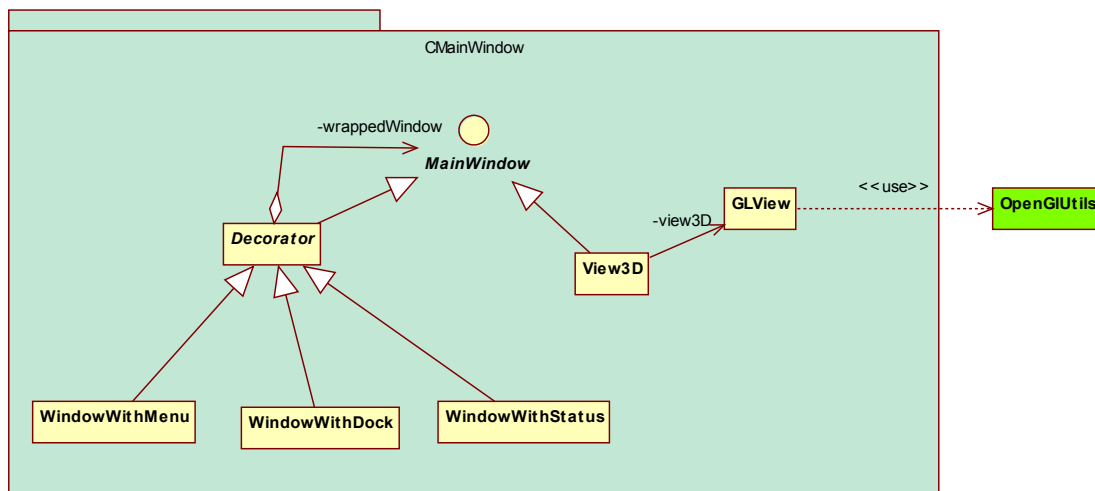


Figura 47: Componente DDDMob::View::CMainWindow

3.11.1 WindowWithMenu (class)

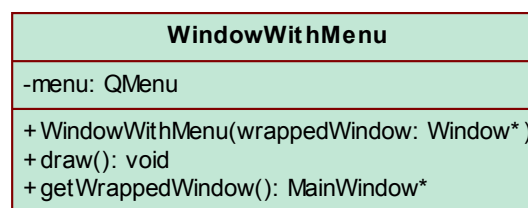


Figura 48: Classe WindowWithMenu

3.11.1.1 Descrizione

Classe che decora la finestra principale. Rappresenta uno dei componenti concrete decorator del Design Pattern_G Decorator

3.11.1.2 Utilizzo

Viene utilizzata per aggiungere un menù alla finestra

3.11.1.3 Classi ereditate

- QMenuBar;
- DDDMob :: View :: CMainWindow :: Decorator.

3.11.1.4 Attributi

Name

menu;



Accesso

private;

Tipo

QMenu;

Descrizione

Menù nella barra dei menù.

3.11.1.5 Metodi

WindowWithMenu

Accesso

public;

Descrizione

Costruttore della classe WindowWithMenu.

draw

Accesso

public;

Tipo di ritorno

void;

Descrizione

Metodo per rendere visibile la finestra;

Note

- Questo metodo è stato ridefinito.

getWrappedWindow

Accesso

public;

Tipo di ritorno

MainWindow*;

Descrizione

Metodo che ritorna un riferimento alla View3D;

Note

- Questo metodo è stato ridefinito.



3.11.2 WindowWithDock (class)

WindowWithDock
-dockWidgets: QDockWidget
+ WindowWithDock(wrappedWindow: Window*) + draw(): void + getWrappedWindow(): MainWindow*

Figura 49: Classe WindowWithDock

3.11.2.1 Descrizione

Classe che decora la finestra principale. Rappresenta uno dei componenti concrete decorator del Design Pattern_G Decorator

3.11.2.2 Utilizzo

Decoratore che aggiunge elementi ancorabili alla finestra

3.11.2.3 Classi ereditate

- DDDMob :: View :: CMainWindow :: Decorator.

3.11.2.4 Attributi

Name

dockWidgets;

Accesso

private;

Tipo

QDockWidget;

Descrizione

rappresenta i dockWidgets presenti nella finestra.

3.11.2.5 Metodi

WindowWithDock

Accesso

public;

Descrizione

Costruttore della classe WindowWithDock.

draw

Accesso

public;

**Tipo di ritorno**

void;

Descrizione

Metodo che disegna la finestra applicando i dockWidgets;

Note

- Questo metodo è stato ridefinito.

getWrappedWindow**Accesso**

public;

Tipo di ritorno

MainWindow*;

Descrizione

Metodo che ritorna la View3D;

Note

- Questo metodo è stato ridefinito.

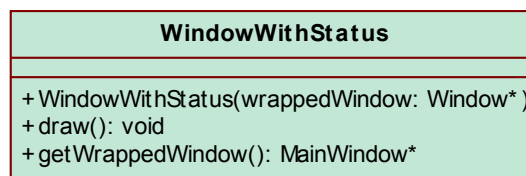
3.11.3 WindowWithStatus (class)

Figura 50: Classe WindowWithStatus

3.11.3.1 Descrizione

Classe che decora la finestra principale. Rappresenta uno dei componenti concrete decorator del Design Pattern_G Decorator

3.11.3.2 Utilizzo

Viene utilizzata per aggiungere ad una finestra la barra di stato

3.11.3.3 Classi ereditate

- QStatusBar;
- DDDMob :: View :: CMainWindow :: Decorator.



3.11.3.4 Metodi

WindowWithStatus

Accesso

public;

Descrizione

Costruttore della classe WindowWithStatus.

draw

Accesso

public;

Tipo di ritorno

void;

Descrizione

Metodo per rendere visibile la finestra;

Note

- Questo metodo è stato ridefinito.

getWrappedWindow

Accesso

public;

Tipo di ritorno

MainWindow*;

Descrizione

Metodo che ritorna un riferimento alla View3D;

Note

- Questo metodo è stato ridefinito.

3.11.4 Decorator (abstract)

<i>Decorator</i>
-wrappedWindow: MainWindow*
+ Decorator(wrappedWindow: MainWindow*) + getWrappedWindow(): MainWindow*

Figura 51: Classe Decorator

3.11.4.1 Descrizione

Classe astratta che rappresenta le varie decorazioni applicabili all'interfaccia utente. Rappresenta il componente decorator del Design Pattern_G Decorator



3.11.4.2 Utilizzo

Classe che rappresenta le possibili decorazioni applicabili all'interfaccia utente, specificate poi dalle classi che ereditano da questa

3.11.4.3 Classi ereditate

- DDDMob :: View :: CMainWindow :: MainWindow.

3.11.4.4 Ereditata da

- DDDMob :: View :: CMainWindow :: WindowWithMenu;
- DDDMob :: View :: CMainWindow :: WindowWithDock;
- DDDMob :: View :: CMainWindow :: WindowWithStatus.

3.11.4.5 Attributi

Name

wrappedWindow;

Accesso

private;

Tipo

MainWindow*;

Descrizione

Puntatore alla composizione di finestre che si vuole decorare con la classe.

3.11.4.6 Metodi

Decorator**Accesso**

public;

Descrizione

Costruttore della classe Decorator.

getWrappedWindow**Accesso**

public;

Tipo di ritorno

MainWindow*;

Descrizione

Metodo che ritorna la View3D;

Note

- Deve essere virtuale;
- Deve essere esplicitamente marcati come costanti;
- Questo metodo è stato ridefinito.



3.11.5 MainWindow (interface)



MainWindow

+draw(): void
+getWrappedWindow(): MainWindow*

Figura 52: Classe MainWindow

3.11.5.1 Descrizione

Interfaccia che rappresenta la finestra principale del programma, ottenuta dalla vista 3D decorata con le decorazioni offerte dalla classe Decorator. Rappresenta il componente component del Design Pattern_G Decorator

3.11.5.2 Utilizzo

3.11.5.3 Ereditata da

- DDDMob :: View :: CMainWindow :: Decorator;
- DDDMob :: View :: CMainWindow :: View3D.

3.11.5.4 Metodi

draw

Accesso

public;

Tipo di ritorno

void;

Descrizione

Metodo per rendere visibile la finestra;

Note

- Deve essere astratto.

getWrappedWindow

Accesso

public;

Tipo di ritorno

MainWindow*;

Descrizione

Metodo che ritorna un riferimento alla View3D;

Note

- Deve essere astratto.



3.11.6 View3D (class)

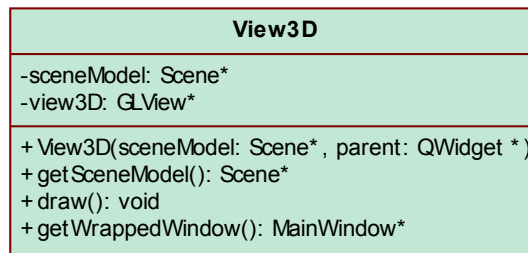


Figura 53: Classe View3D

3.11.6.1 Descrizione

Classe che rappresenta la vista contenente la scena_G 3D. È il componente concrete component del Design Pattern_G Decorator

3.11.6.2 Utilizzo

Viene utilizzata per visualizzare l'anteprima della scena_G 3D del Model. Manda Signal_G di richiesta di aggiornamento dei dati di visualizzazione della scena_G 3D al Model. Può leggere i dati aggiornati direttamente dal Model dopo aver ricevuto un Signal_G di aggiornamento

3.11.6.3 Classi ereditate

- QMainWindow;
- DDDMob :: View :: CMainWindow :: MainWindow.

3.11.6.4 Attributi

Name

sceneModel;

Accesso

private;

Tipo

Scene*;

Descrizione

Riferimento alla scena_G del Modello.

Name

view3D;

Accesso

private;

Tipo

GLView*;

Descrizione

Finestra di visualizzazione della scena_G 3D.



3.11.6.5 Metodi

View3D

Accesso

public;

Descrizione

Costruttore di View3D.

getSceneModel

Accesso

public;

Tipo di ritorno

Scene*;

Descrizione

Ritorna un riferimento costante alla Scena_G;

Note

- Deve essere esplicitamente marcati come costanti.

draw

Accesso

public;

Tipo di ritorno

void;

Descrizione

Metodo per rendere visibile la finestra;

Note

- Questo metodo è stato ridefinito.

getWrappedWindow

Accesso

public;

Tipo di ritorno

MainWindow*;

Descrizione

Metodo che ritorna un riferimento alla View3D;

Note

- Questo metodo è stato ridefinito.



3.11.7 GLView (class)

GLView
<div><div>-sceneModel: Scene* sceneModel -cameraPosition: QVector3D -cameraRotation: QVector3D -lastPos: QPointF -background: QColor background -movementConstraint: Movement Constraint -farClippingPlane: float -nearClippingPlane: float</div><div>+ paintGL(): void + GLView(sceneModel: Scene*, parent = 0: QWidget *): explicit + getSceneModel(): Scene* + initializeObjects(): void # updateBackgroundColor(): void # initializeGL(): void # paintGL(): void # resizeGL(width: int, height: int): void # mousePressEvent(event: QMouseEvent *): void # mouseMoveEvent(event: QMouseEvent *event): void # mouseReleaseEvent(event: QMouseEvent *): void # wheelEvent(event: QWheelEvent *): void # keyPressEvent(event: QKeyEvent *): void # keyReleaseEvent(event: QKeyEvent *): void -renderAxes(): void + selectionChanged(): void</div></div>

Figura 54: Classe GLView

3.11.7.1 Descrizione

Classe che esegue il rendering_G della scena_G

3.11.7.2 Utilizzo

3.11.7.3 Attributi

Name

sceneModel;

Accesso

private;

Tipo

Scene* sceneModel;

Descrizione

Modello della scena_G.

Name

cameraPosition;

Accesso

private;

Tipo

QVector3D;

Descrizione

centro di rotazione della camera_G.



Name

cameraRotation;

Accesso

private;

Tipo

QVector3D;

Descrizione

Angoli di rotazione della camera_G.

Name

lastPos;

Accesso

private;

Tipo

QPointF;

Descrizione

Tracciamento della posizione del mouse.

Name

background;

Accesso

private;

Tipo

QColor background;

Descrizione

Colore di sfondo.

Name

movementConstraint;

Accesso

private;

Tipo

MovementConstraint;

Descrizione

Tipologia di movimento.

Name

farClippingPlane;

Accesso

private;

Tipo

float;



Descrizione

Distanza del clipping plane per il tronco di visualizzazione.

Name

nearClippingPlane;

Accesso

private;

Tipo

float;

Descrizione

Distanza del clipping plane per il tronco di visualizzazione.

3.11.7.4 Metodi

paintGL

Accesso

public;

Tipo di ritorno

void;

Descrizione

Aggiorna la scena_G;

Note

- Questo metodo è stato ridefinito.

GLView

Accesso

public;

Tipo di ritorno

explicit;

Descrizione

Costruttore di GLView.

getSceneModel

Accesso

public;

Tipo di ritorno

Scene*;

Descrizione

Ritorna il modello della scena_G;

Note

- Deve essere esplicitamente marcati come costanti.



initializeObjects

Accesso

public;

Tipo di ritorno

void;

Descrizione

Inizializza gli oggetti della scena_G.

updateBackgroundColor

Accesso

protected;

Tipo di ritorno

void;

Descrizione

Aggiorna il colore di sfondo.

initializeGL

Accesso

protected;

Tipo di ritorno

void;

Descrizione

Inizializza OpenGL_G;

Note

- Questo metodo è stato ridefinito.

paintGL

Accesso

protected;

Tipo di ritorno

void;

Descrizione

Disegna la scena_G OpenGL_G;

Note

- Questo metodo è stato ridefinito.



resizeGL

Accesso

protected;

Tipo di ritorno

void;

Descrizione

Ridimensiona la scena_G;

Note

- Questo metodo è stato ridefinito.

mousePressEvent

Accesso

protected;

Tipo di ritorno

void;

Descrizione

Metodo eseguito alla pressione di un tasto del mouse;

Note

- Questo metodo è stato ridefinito.

mouseMoveEvent

Accesso

protected;

Tipo di ritorno

void;

Descrizione

Metodo eseguito al movimento del mouse;

Note

- Questo metodo è stato ridefinito.

mouseReleaseEvent

Accesso

protected;

Tipo di ritorno

void;

Descrizione

Metodo eseguito al rilascio di un tasto premuto sul mouse;

Note

- Questo metodo è stato ridefinito.



wheelEvent

Accesso

protected;

Tipo di ritorno

void;

Descrizione

Metodo eseguito alla rotazione della rotellina del mouse;

Note

- Questo metodo è stato ridefinito.

keyPressEvent

Accesso

protected;

Tipo di ritorno

void;

Descrizione

Metodo eseguito alla pressione di un tasto della tastiera;

Note

- Questo metodo è stato ridefinito.

keyReleaseEvent

Accesso

protected;

Tipo di ritorno

void;

Descrizione

Metodo eseguito al rilascio di un tasto premuto sulla tastiera;

Note

- Questo metodo è stato ridefinito.

renderAxes

Accesso

private;

Tipo di ritorno

void;

Descrizione

Esegue il render sugli assi.



selectionChanged

Accesso

`signalG;`

Tipo di ritorno

`void;`

Descrizione

Cambio selezione.

3.12 DDDMob::View::CSettingsWindow

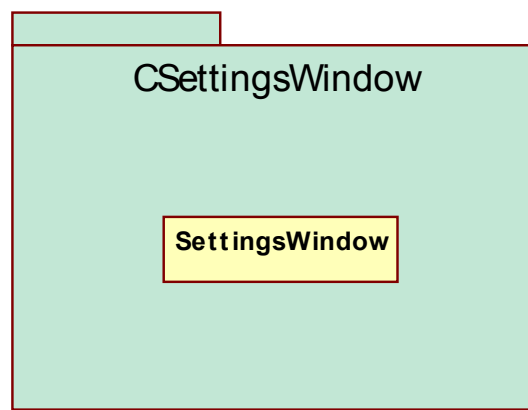


Figura 55: Componente DDDMob::View::CSettingsWindow

3.12.1 SettingsWindow (class)

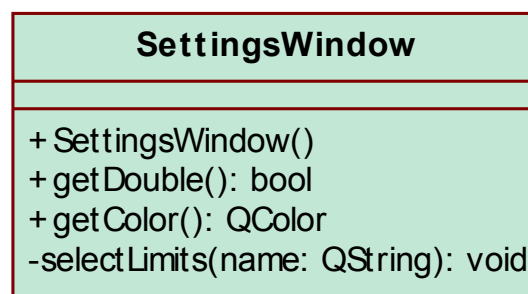


Figura 56: Classe SettingsWindow

3.12.1.1 Descrizione

Classe che rappresenta la finestra per modificare le impostazioni del programma

3.12.1.2 Utilizzo

Viene utilizzata per visualizzare le impostazioni del programma. La sua apertura deriva dalla ricezione di un `SignalG` da parte del Controller. Manda `SignalG` di richiesta di aggiornamento dei dati di configurazione al Model. Può leggere i dati aggiornati direttamente dal Model dopo aver ricevuto un `SignalG` di aggiornamento



3.12.1.3 Classi ereditate

- QWindow.

3.12.1.4 Metodi

SettingsWindow

Accesso

public;

Descrizione

Costruttore della classe.

getDouble

Accesso

public;

Tipo di ritorno

bool;

Descrizione

Ritorna il settaggio numerico dell'applicazione;

Note

- Deve essere esplicitamente marcati come costanti.

getColor

Accesso

public;

Tipo di ritorno

QColor;

Descrizione

Ritorna il colore di sfondo;

Note

- Deve essere esplicitamente marcati come costanti.

selectLimits

Accesso

private;

Tipo di ritorno

void;

Descrizione

Seleziona un settaggio.



3.13 DDDMob::View::CErrorWindow

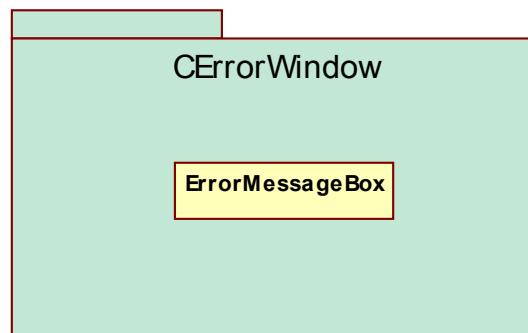


Figura 57: Componente DDDMob::View::CErrorWindow

3.13.1 ErrorMessageBox (class)

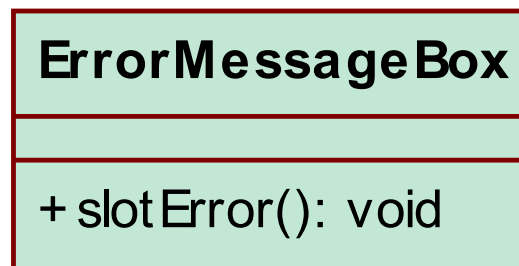


Figura 58: Classe ErrorMessageBox

3.13.1.1 Descrizione

Classe che rappresenta una finestra che riporta messaggi d'errore per l'utente

3.13.1.2 Utilizzo

La sua apertura deriva dalla ricezione di un Signal_G da parte del Model, generato nel caso in cui si verifichi una inconsistenza durante l'azione del Model. Il Signal_G include un codice di errore e visualizza il relativo messaggio di errore. Il messaggio di errore verrà recuperato mediante QtLinguist

3.13.1.3 Metodi

slotError

Accesso

public;

Tipo di ritorno

void;

Descrizione

riceve il codice di errore.



3.14 DDDMob::View::CWidgetElement

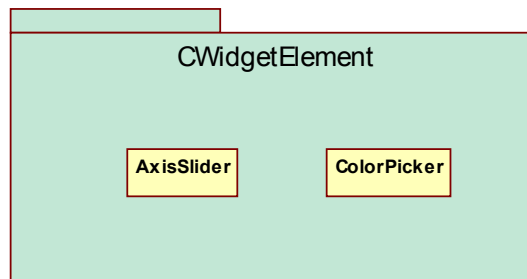


Figura 59: Componente DDDMob::View::CWidgetElement

3.14.1 AxisSlider (class)

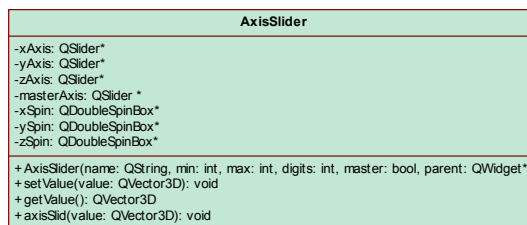


Figura 60: Classe AxisSlider

3.14.1.1 Descrizione

Classe che rappresenta il widget per modificare dei valori sugli assi x, y, z

3.14.1.2 Utilizzo

Viene utilizzata per inviare il signal_G opportuno al controller per la modifica dei valori

3.14.1.3 Classi ereditate

- QGroupBox.

3.14.1.4 Attributi

Name

xAxis;

Accesso

private;

Tipo

QSlider*;

Descrizione

slider che permette di cambiare le coordinate lungo l'asse x del mesh.

Name

yAxis;



Accesso

private;

Tipo

QSlider*;

Descrizione

slider che permette di cambiare le coordinate lungo l'asse y del mesh.

Name

zAxis;

Accesso

private;

Tipo

QSlider*;

Descrizione

slider che permette di cambiare le coordinate lungo l'asse z del mesh.

Name

masterAxis;

Accesso

private;

Tipo

QSlider *;

Descrizione

Slider che controlla gli altri.

Name

xSpin;

Accesso

private;

Tipo

QDoubleSpinBox*;

Descrizione

Elemento di input per controllare in modo preciso il valore di xAxis.

Name

ySpin;

Accesso

private;

Tipo

QDoubleSpinBox*;

Descrizione

Elemento di input per controllare in modo preciso il valore di yAxis.



Name

zSpin;

Accesso

private;

Tipo

QDoubleSpinBox*;

Descrizione

Elemento di input per controllare in modo preciso il valore di zAxis.

3.14.1.5 Metodi

AxisSlider

Accesso

public;

Descrizione

Costruttore di AxisSlider.

setValue

Accesso

public;

Tipo di ritorno

void;

Descrizione

Imposta i valori degli slider.

getValue

Accesso

public;

Tipo di ritorno

QVector3D;

Descrizione

Ritorna un vettore contenente il valore rappresentato dagli slider;

Note

- Deve essere esplicitamente marcati come costanti.

axisSlid

Accesso

signal_G;

Tipo di ritorno

void;

Descrizione

Segnale emesso quando il valore degli slider cambia.



3.14.2 ColorPicker (class)

ColorPicker
-colorDialog: QColorDialog
-updateColor(): void -updateColorFromDialog(): void +ColorPicker(parent = nullptr: QWidget*) +ColorPicker(label: QString, parent = nullptr: QWidget*) +setColor(color: QColor): void +getColor(): QColor #mousePressEvent(event: QMouseEvent *): void #changeEvent(event: QEvent *): void +colorCanged(color: QColor): void

Figura 61: Classe ColorPicker

3.14.2.1 Descrizione

Classe che rappresenta il widget che consente di selezionare un colore.

3.14.2.2 Utilizzo

Viene utilizzata per permette di modificare il colore selezionato. Modifiche al colore selezionato provocano un signal_G appropriato.

3.14.2.3 Classi ereditate

- QPushButton.

3.14.2.4 Attributi

Name

colorDialog;

Accesso

private;

Tipo

QColorDialog;

Descrizione

form per la scelta del colore.

3.14.2.5 Metodi

updateColor

Accesso

private;

Tipo di ritorno

void;

Descrizione

Aggiorna il widget.



updateColorFromDialog

Accesso

private;

Tipo di ritorno

void;

Descrizione

Aggiorna la finestra di dialogo.

ColorPicker

Accesso

public;

Descrizione

Costruttore dell'oggetto.

ColorPicker

Accesso

public;

Descrizione

Costruttore dell'oggetto.

setColor

Accesso

public;

Tipo di ritorno

void;

Descrizione

Imposta il colore nel widget.

getColor

Accesso

public;

Tipo di ritorno

QColor;

Descrizione

Restituisce il colore selezionato nel widget;

Note

- Deve essere esplicitamente marcati come costanti.



mouseReleaseEvent

Accesso

protected;

Tipo di ritorno

void;

Descrizione

Metodo eseguito quando viene rilasciato il tasto del mouse premuto.

changeEvent

Accesso

protected;

Tipo di ritorno

void;

Descrizione

Metodo eseguito quando viene cambiato il colore nel widget.

colorCanged

Accesso

signal_G;

Tipo di ritorno

void;

Descrizione

Segnale emesso quando viene cambiato il colore del widget.

3.15 DDDMob::C3DObject

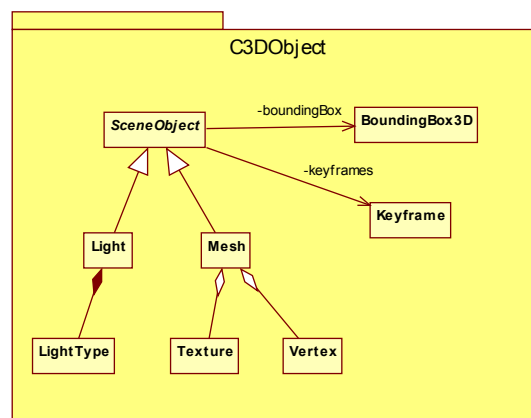


Figura 62: Componente DDDMob::C3DObject



3.15.1 SceneObject (abstract)

SceneObject
#position: QVector3D #rotation: QVector3D #scale: QVector3D #specularColor: QColor #diffuseColor: QColor #emissionColor: QColor #emission: double #boundingBox: BoundingBox3D #keyframes: QList< Keyframe>
+getDiffuseColor(): QColor +getEmission(): double +getEmissionColor(): QColor +getPosition(): QVector3D +getRotation(): QVector3D +getScale(): QVector3D +getSpecularColor(): QColor +setDiffuseColor(diffusionColor: QColor): void +setEmission(emission: double): void +setEmissionColor(emissionColor: QColor): void +setPosition(position: QVector3D): void +setRotation(rotation: QVector3D): void +setScale(scale: QVector3D): void +setSpecularColor(specularColor: QColor): void +transformation(): QMatrix4x4 +initialize(): void +render(scaleFactor: double): void +SceneObject() #renderInternal(): void +getBoundingBox(): BoundingBox3D +getKeyframes(): QList< Keyframe> +pushKeyframe(frame: Keyframe): void

Figura 63: Classe SceneObject

3.15.1.1 Descrizione

Classe astratta che rappresenta gli oggetti che è possibile trovare nella scena_G 3D

3.15.1.2 Utilizzo

Mette a disposizione i metodi per la modifica degli attributi e per le trasformazioni all'oggetto.

3.15.1.3 Ereditata da

- DDDMob :: C3DObject :: Light;
- DDDMob :: C3DObject :: Mesh.

3.15.1.4 Attributi

Name

position;



Accesso

protected;

Tipo

QVector3D;

Descrizione

Posizione.

Name

rotation;

Accesso

protected;

Tipo

QVector3D;

Descrizione

Rotazione.

Name

scale;

Accesso

protected;

Tipo

QVector3D;

Descrizione

Ridimensionamento.

Name

specularColor;

Accesso

protected;

Tipo

QColor;

Descrizione

Colore speculare_G.

Name

diffuseColor;

Accesso

protected;

Tipo

QColor;

Descrizione

Colore di diffusione_G.



Name

emissionColor;

Accesso

protected;

Tipo

QColor;

Descrizione

Colore di emissione.

Name

emission;

Accesso

protected;

Tipo

double;

Descrizione

Quantità di emissione nell'intervallo $[0,1]$.

Name

boundingBox_G;

Accesso

protected;

Tipo

BoundingBox3D;

Descrizione

Parallelepipedo di volume minimo che contiene completamente la parte visibile dell'oggetto.

Name

keyframes;

Accesso

protected;

Tipo

QList<Keyframe_G >;

Descrizione

Sequenza di elementi di animazione.

3.15.1.5 Metodi

getDiffuseColor

Accesso

public;



Tipo di ritorno

QColor;

Descrizione

Restituisce il colore di diffusione $_G$.

getEmission

Accesso

public;

Tipo di ritorno

double;

Descrizione

Ritorna l'intensità di emissione;

Note

- Deve essere esplicitamente marcati come costanti.

getEmissionColor

Accesso

public;

Tipo di ritorno

QColor;

Descrizione

Restituisce il colore emissione dell'oggetto;

Note

- Deve essere esplicitamente marcati come costanti.

getPosition

Accesso

public;

Tipo di ritorno

QVector3D;

Descrizione

Ritorna la posizione dell'oggetto.

getRotation

Accesso

public;

Tipo di ritorno

QVector3D;

Descrizione

Ritorna il parametro di rotazione dell'oggetto.



getScale

Accesso

public;

Tipo di ritorno

QVector3D;

Descrizione

Ritorna la dimensione dell'oggetto.

getSpecularColor

Accesso

public;

Tipo di ritorno

QColor;

Descrizione

Ritorna il colore speculare_G dell'oggetto.

setDiffuseColor

Accesso

public;

Tipo di ritorno

void;

Descrizione

Imposta il colore di diffusione_G dell'oggetto.

setEmission

Accesso

public;

Tipo di ritorno

void;

Descrizione

Imposta il valore di emissione dell'oggetto.

setEmissionColor

Accesso

public;

Tipo di ritorno

void;

Descrizione

Imposta il colore di emissione dell'oggetto.



setPosition

Accesso

public;

Tipo di ritorno

void;

Descrizione

Imposta la posizione in X, Y e Z dell'oggetto.

setRotation

Accesso

public;

Tipo di ritorno

void;

Descrizione

Imposta la rotazione dell'oggetto.

setScale

Accesso

public;

Tipo di ritorno

void;

Descrizione

Imposta la dimensione dell'oggetto.

setSpecularColor

Accesso

public;

Tipo di ritorno

void;

Descrizione

Imposta il colore speculare_G dell'oggetto.

transformation

Accesso

public;

Tipo di ritorno

QMatrix4x4;

Descrizione

Ritorna la matrice di trasformazione risultante da translation, rotation e scale;

Note

- Deve essere esplicitamente marcati come costanti.



initialize

Accesso

public;

Tipo di ritorno

void;

Descrizione

Prepara l'oggetto per il rendering_G OpenGL_G allocando eventuali risorse necessarie.

render

Accesso

public;

Tipo di ritorno

void;

Descrizione

Applica le trasformazioni ed esegue il rendering_G.

SceneObject

Accesso

public;

Descrizione

Costruttore di default della classe SceneObject.

renderInternal

Accesso

protected;

Tipo di ritorno

void;

Descrizione

Esegue il rendering_G in OpenGL_G;

Note

- Deve essere astratto;
- Deve essere esplicitamente marcati come costanti.

getBoundingBox

Accesso

public;

Tipo di ritorno

BoundingBox3D;



Descrizione

Ritorna il parallelepipedo di volume minimo che contiene completamente la parte visibile dell'oggetto;

Note

- Deve essere esplicitamente marcati come costanti.

getKeyframes

Accesso

public;

Tipo di ritorno

QList<Keyframe_G>;

Descrizione

Restituisce la lista dei Keyframe_G dell'oggetto;

Note

- Deve essere esplicitamente marcati come costanti.

pushKeyframe

Accesso

public;

Tipo di ritorno

void;

Descrizione

Aggiunge un keyframe_G.

3.15.2 Light (class)

Light
-index: GLint -type: LightType
+ getLightType(): LightType + Light(wrappedObject: QGLSceneNode&) + setLightType(lightType: LightType): void #renderInternal(): void + operator=(light: const Light&): Light& + initialize(): void + Light(name-changed: LightType): void

Figura 64: Classe Light

3.15.2.1 Descrizione

Classe che rappresenta una luce presente nella scena_G 3D



3.15.2.2 Utilizzo

Mette a disposizione i metodi per la modifica degli attributi propri della luce

3.15.2.3 Classi ereditate

- DDDMob :: C3DObject :: SceneObject.

3.15.2.4 Attributi

Name

index;

Accesso

private;

Tipo

GLint;

Descrizione

ritorna la luce OpenGL_G dato l'indice.

Name

type;

Accesso

private;

Tipo

LightType;

Descrizione

Tipo della luce.

3.15.2.5 Metodi

getLightType

Accesso

public;

Tipo di ritorno

LightType;

Descrizione

Ritorna il tipo di luce;

Note

- Deve essere esplicitamente marcati come costanti.

Light

Accesso

public;

Descrizione

Costruttore di Light.



setLightType

Accesso

public;

Tipo di ritorno

void;

Descrizione

Imposta il tipo di luce.

renderInternal

Accesso

protected;

Tipo di ritorno

void;

Descrizione

Esegue il rendering_G in OpenGL_G;

Note

- Deve essere esplicitamente marcati come costanti;
- Questo metodo è stato ridefinito.

operator=

Accesso

public;

Tipo di ritorno

Light&;

Descrizione

Ridefinizione operatore =.

initialize

Accesso

public;

Tipo di ritorno

void;

Descrizione

Inizializzazione della classe;

Note

- Questo metodo è stato ridefinito.



Light

Accesso

public;

Tipo di ritorno

void;

Descrizione

Costruttore della classe.

3.15.3 Mesh (class)

Mesh
-geometry: QVector<Face> -shininess: double -texture: Texture* -Face: std::array<Vertex,3>
+getShininess(): double +setShininess(shininess: double): void +Mesh(name-changed: const QVector<Face> &) #renderInternal(): void #updateBoundingBox(): void +initialize(): void +setGeometry(geometry: QVector<Face> &): void +getGeometry(): const QVector<Face> & +addFace(face: const Face&): void +addFace(v1: const Vertex&, v2: const Vertex&, v3: const Vertex&): void +getTexture(): Texture* +setTexture(t: Texture*): void +operator==(o: const Mesh&): bool

Figura 65: Classe Mesh

3.15.3.1 Descrizione

Classe che rappresenta una mesh poligonale presente nella scena_G 3D

3.15.3.2 Utilizzo

Mette a disposizione i metodi per la modifica degli attributi propri delle mesh

3.15.3.3 Classi ereditate

- DDDMob :: C3DObject :: SceneObject.

3.15.3.4 Attributi

Name

geometry;

Accesso

private;

Tipo

QVector<Face>;

Descrizione

Vettore di facce.



Name

shininess;

Accesso

private;

Tipo

double;

Descrizione

Lucentezza.

Name

texture;

Accesso

private;

Tipo

Texture*;

Descrizione

texture.

Name

Face;

Accesso

private;

Tipo

std::array<Vertex,3>;

Descrizione

Rappresenta una faccia della mesh.

3.15.3.5 Metodi

getShininess

Accesso

public;

Tipo di ritorno

double;

Descrizione

Ritorna la lucentezza della Mesh.

setShininess

Accesso

public;

Tipo di ritorno

void;



Descrizione

Modifica la lucentezza.

Mesh

Accesso

public;

Descrizione

Costruttore di Mesh.

renderInternal

Accesso

protected;

Tipo di ritorno

void;

Descrizione

Metodo per eseguire il render;

Note

- Deve essere esplicitamente marcati come costanti;
- Questo metodo è stato ridefinito.

updateBoundingBox

Accesso

protected;

Tipo di ritorno

void;

Descrizione

Aggiorna BoundingBox_G .

initialize

Accesso

public;

Tipo di ritorno

void;

Descrizione

Inizializzazione della classe;

Note

- Questo metodo è stato ridefinito.



setGeometry

Accesso

public;

Tipo di ritorno

void;

Descrizione

Imposta la geometria dell'oggetto.

getGeometry

Accesso

public;

Tipo di ritorno

const QVector<Face>&;

Descrizione

Restituisce il vettore delle facce dell'oggetto;

Note

- Deve essere esplicitamente marcati come costanti.

addFace

Accesso

public;

Tipo di ritorno

void;

Descrizione

Aggiunge una faccia all'oggetto.

addFace

Accesso

public;

Tipo di ritorno

void;

Descrizione

Aggiunge una faccia all'oggetto.

getTexture

Accesso

public;

Tipo di ritorno

Texture*;



Descrizione

Ritorna la texture dell'oggetto;

Note

- Deve essere esplicitamente marcati come costanti.

setTexture

Accesso

public;

Tipo di ritorno

void;

Descrizione

Imposta la texture dell'oggetto.

operator==

Accesso

public;

Tipo di ritorno

bool;

Descrizione

Ridefinizione dell'operatore ==;

Note

- Deve essere esplicitamente marcati come costanti.

3.15.4 LightType (class)

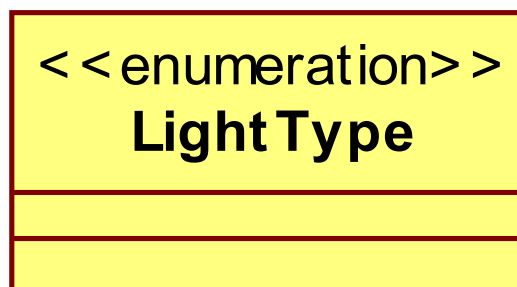


Figura 66: Classe LightType

3.15.4.1 Descrizione

Definisce il tipo di luce

3.15.4.2 Utilizzo

Viene utilizzato come tipo di enumerazione



3.15.5 Texture (class)

Texture
-path: QString -texture: QImage -openglID: GLuint
+ Texture(path: QString) + getPath(): QString + getTexture(): QImage + setPath(path: QString): void + initialize(): void + render(): void

Figura 67: Classe Texture

3.15.5.1 Descrizione

Classe che rappresenta una texture per un oggetto della scena_G

3.15.5.2 Utilizzo

Classe utilizzata per la rappresentazione di una texture per un oggetto della scena_G, definita con il percorso al file immagine e con una rappresentazione interna dell'immagine grazie alla classe QImage

3.15.5.3 Attributi

Name

path;

Accesso

private;

Tipo

QString;

Descrizione

Path della Texture.

Name

texture;

Accesso

private;

Tipo

QImage;



Descrizione

Immagine della Texture.

Name

openglID;

Accesso

private;

Tipo

GLuint;

Descrizione

Id in openGL_G .

3.15.5.4 Metodi

Texture

Accesso

public;

Descrizione

Costruttore per Texture, che si occupa di impostare il percorso fornito e caricare dinamicamente in una QImage l'immagine trovata.

getPath

Accesso

public;

Tipo di ritorno

QString;

Descrizione

Ritorna il percorso del file immagine della texture;

Note

- Deve essere esplicitamente marcati come costanti.

getTexture

Accesso

public;

Tipo di ritorno

QImage;

Descrizione

Ritorna il file immagine della texture caricato dinamicamente come QImage;

Note

- Deve essere esplicitamente marcati come costanti.



setPath

Accesso

public;

Tipo di ritorno

void;

Descrizione

Imposta il percorso del file immagine della texture.

initialize

Accesso

public;

Tipo di ritorno

void;

Descrizione

Inizializza l'oggetto per il rendering_G di OpenGL_G.

render

Accesso

public;

Tipo di ritorno

void;

Descrizione

Esegue il render della texture;

Note

- Deve essere esplicitamente marcati come costanti.

3.15.6 Vertex (class)

Vertex
-position: QVector3D -normal: QVector3D -uv: QVector2D -color: QColor
+ Vertex(position= QVector3D(0,0,0): QVector3D, normal = QVector3D(0,0,0): QVector3D, uv = QVector2D(0,0): QVector2D, name-changed: QColor) + getPosition(): QVector3D + getNormal(): QVector3D + getColor(): QColor + getUv(): QVector2D + setPosition(p: QVector3D): void + setNormal(p: QVector3D): void + setColor(p: QColor): void + setUv(p: QVector2D): void + render(): void + operator==(v: const Vertex&): bool

Figura 68: Classe Vertex

3.15.6.1 Descrizione

Classe che rappresenta un vertice di un oggetto



3.15.6.2 Utilizzo

Viene usata per fornire informazioni utili proprie dei vertici, quali la posizione, le normali e l'UV

3.15.6.3 Attributi

Name

position;

Accesso

private;

Tipo

QVector3D;

Descrizione

posizione con x,y,z.

Name

normal;

Accesso

private;

Tipo

QVector3D;

Descrizione

normale in x,y,z.

Name

uv;

Accesso

private;

Tipo

QVector2D;

Descrizione

Coordinate nello spazio della texture.

Name

color;

Accesso

private;

Tipo

QColor;

Descrizione

Colore dell'oggetto.



3.15.6.4 Metodi

Vertex

Accesso

public;

Descrizione

Costruttore dell'oggetto Vertex.

getPosition

Accesso

public;

Tipo di ritorno

QVector3D;

Descrizione

Ritorna la posizione del vertice;

Note

- Deve essere esplicitamente marcati come costanti.

getNormal

Accesso

public;

Tipo di ritorno

QVector3D;

Descrizione

Ritorna la normale del vertice;

Note

- Deve essere esplicitamente marcati come costanti.

getColor

Accesso

public;

Tipo di ritorno

QColor;

Descrizione

Torna il colore del vertice;

Note

- Deve essere esplicitamente marcati come costanti.



getUv

Accesso

public;

Tipo di ritorno

QVector2D;

Descrizione

Restituisce le coordinate nello spazio della texture;

Note

- Deve essere esplicitamente marcati come costanti.

setPosition

Accesso

public;

Tipo di ritorno

void;

Descrizione

Imposta la posizione del vertice.

setNormal

Accesso

public;

Tipo di ritorno

void;

Descrizione

Imposta i valori della normale.

setColor

Accesso

public;

Tipo di ritorno

void;

Descrizione

Imposta il colore del vertice.

setUv

Accesso

public;

Tipo di ritorno

void;

Descrizione

Imposta le coordinate nello spazio della texture.



render

Accesso

public;

Tipo di ritorno

void;

Descrizione

Esegue il render del vertice;

Note

- Deve essere esplicitamente marcati come costanti.

operator==

Accesso

public;

Tipo di ritorno

bool;

Descrizione

Redefinizione dell'operatore ==;

Note

- Deve essere esplicitamente marcati come costanti.

3.15.7 Keyframe (class)

Keyframe
-time: double -position: QVector3D
+ Keyframe(time: double, position: QVector3D) + getTime(): double + setTime(time: double): void + getPosition(): QVector3D + setPosition(position: QVector3D): void

Figura 69: Classe Keyframe

3.15.7.1 Descrizione

Classe che rappresenta un keyframe_G nella scena_G 3D

3.15.7.2 Utilizzo

Viene utilizzata per contenere le informazioni di un dato keyframe_G , tempo e posizione, per uno degli oggetti della scena_G



3.15.7.3 Attributi

Name

time;

Accesso

private;

Tipo

double;

Descrizione

Durata del keyframe_G .

Name

position;

Accesso

private;

Tipo

QVector3D;

Descrizione

Posizione del keyframe_G .

3.15.7.4 Metodi

Keyframe_G

Accesso

public;

Descrizione

Costruttore dell'oggetto keyform.

getTime

Accesso

public;

Tipo di ritorno

double;

Descrizione

Ritorna la durata del keyframe_G .

setTime

Accesso

public;

Tipo di ritorno

void;

Descrizione

Imposta la durata del keyframe_G .



getPosition

Accesso

public;

Tipo di ritorno

QVector3D;

Descrizione

Ritorna la posizione del keyframe_G.

setPosition

Accesso

public;

Tipo di ritorno

void;

Descrizione

Imposta la posizione del keyframe_G.

3.15.8 BoundingBox3D (class)

BoundingBox3D
-m_min: QVector3D -m_max: QVector3D -empty: bool
+ BoundingBox3D() + BoundingBox3D(p: QVector3D) + center(): QVector3D + size(): QVector3D + max(): QVector3D + min(): QVector3D + setMax(max: QVector3D): void + setMin(min: QVector3D): void + clear(): void + isEmpty(): bool + include(p: QVector3D): void + include(bb: BoundingBox3D): void + contains(p: QVector3D): bool + intersects(A: QVector3D, B: QVector3D): bool + scale(s: QVector3D): BoundingBox3D + translate(s: QVector3D s): BoundingBox3D

Figura 70: Classe BoundingBox3D

3.15.8.1 Descrizione

Oggetto Bounding Box_G



3.15.8.2 Utilizzo

3.15.8.3 Attributi

Name

m_min;

Accesso

private;

Tipo

QVector3D;

Descrizione

Vertice minore.

Name

m_max;

Accesso

private;

Tipo

QVector3D;

Descrizione

Vertice massimo.

Name

empty;

Accesso

private;

Tipo

bool;

Descrizione

Bounding Box vuoto.

3.15.8.4 Metodi

BoundingBox3D

Accesso

public;

Descrizione

Costruttore di default di BoundingBox3D.

BoundingBox3D

Accesso

public;

Descrizione

Costruttore di BoundingBox3D.



center

Accesso

public;

Tipo di ritorno

QVector3D;

Descrizione

Restituisce il centro del BoundingBox_G ;

Note

- Deve essere esplicitamente marcati come costanti.

size

Accesso

public;

Tipo di ritorno

QVector3D;

Descrizione

Restituisce altezza, larghezza, lunghezza della BoundingBox_G ;

Note

- Deve essere esplicitamente marcati come costanti.

max

Accesso

public;

Tipo di ritorno

QVector3D;

Descrizione

Ritorna il vertice massimo della BoundingBox_G ;

Note

- Deve essere esplicitamente marcati come costanti.

min

Accesso

public;

Tipo di ritorno

QVector3D;

Descrizione

Ritorna il vertice minimo della BoundingBox_G ;

Note

- Deve essere esplicitamente marcati come costanti.



setMax

Accesso

public;

Tipo di ritorno

void;

Descrizione

Imposta il vertice massimo del BoundingBox_G .

setMin

Accesso

public;

Tipo di ritorno

void;

Descrizione

Imposta il vertice minimo del BoundingBox_G .

clear

Accesso

public;

Tipo di ritorno

void;

Descrizione

Azzera il BoundingBox_G .

isEmpty

Accesso

public;

Tipo di ritorno

bool;

Descrizione

Ritorna true se il BoundingBox_G è vuoto;

Note

- Deve essere esplicitamente marcati come costanti.

include

Accesso

public;

Tipo di ritorno

void;

Descrizione

Espanse il BoundingBox_G fino ad includere il punto (x,y,z) .



include

Accesso

public;

Tipo di ritorno

void;

Descrizione

Espande il BoundingBox_G fino a includere il BoundingBox_G passato per parametro.

contains

Accesso

public;

Tipo di ritorno

bool;

Descrizione

Ritorna true se il BoundingBox_G contiene il punto (x,y,z) ;

Note

- Deve essere esplicitamente marcati come costanti.

intersects

Accesso

public;

Tipo di ritorno

bool;

Descrizione

Restituisce true se il BoundingBox_G interseca quello passato per parametro;

Note

- Deve essere esplicitamente marcati come costanti.

scale

Accesso

public;

Tipo di ritorno

BoundingBox3D ;

Descrizione

Ritorna un BoundingBox_G scalato;

Note

- Deve essere esplicitamente marcati come costanti.



translate

Accesso

public;

Tipo di ritorno

BoundingBox3D;

Descrizione

Trasla il BoundingBox_G ;

Note

- Deve essere esplicitamente marcati come costanti.



4 Diagrammi di sequenza

Vengono qui riportati i diagrammi di sequenza delle operazioni principali dell'applicazione.

4.1 Importazione

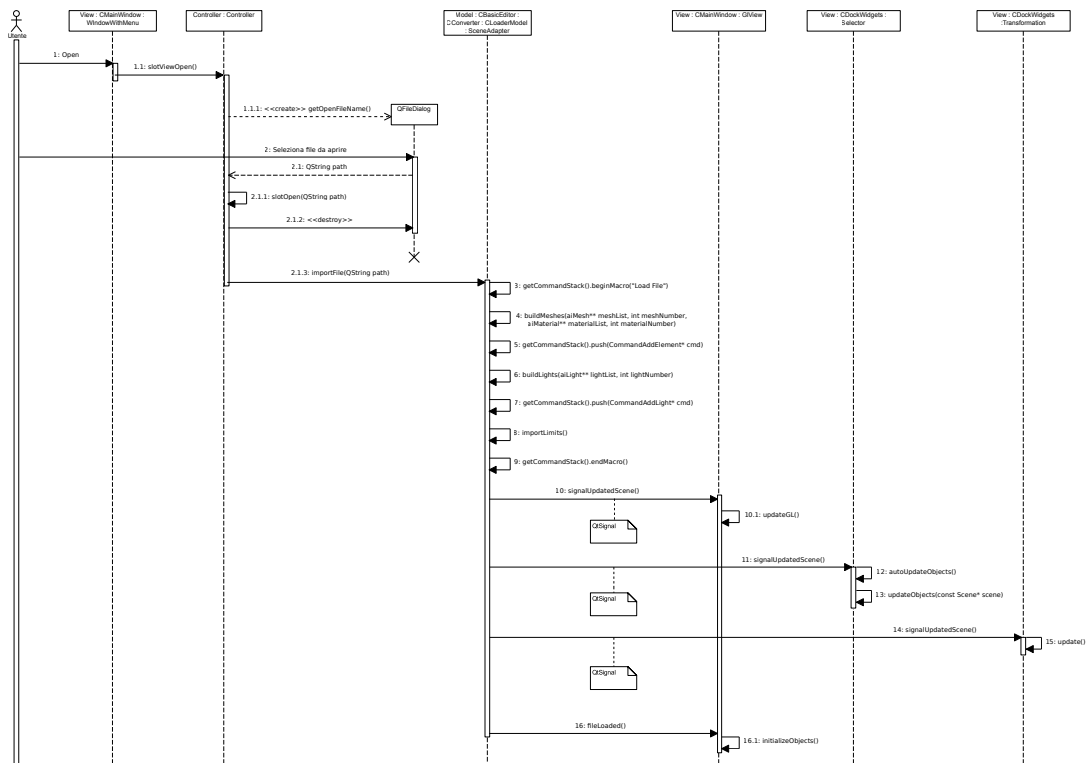


Figura 71: Diagrammi di sequenza per l'importazione



4.2 Esportazione

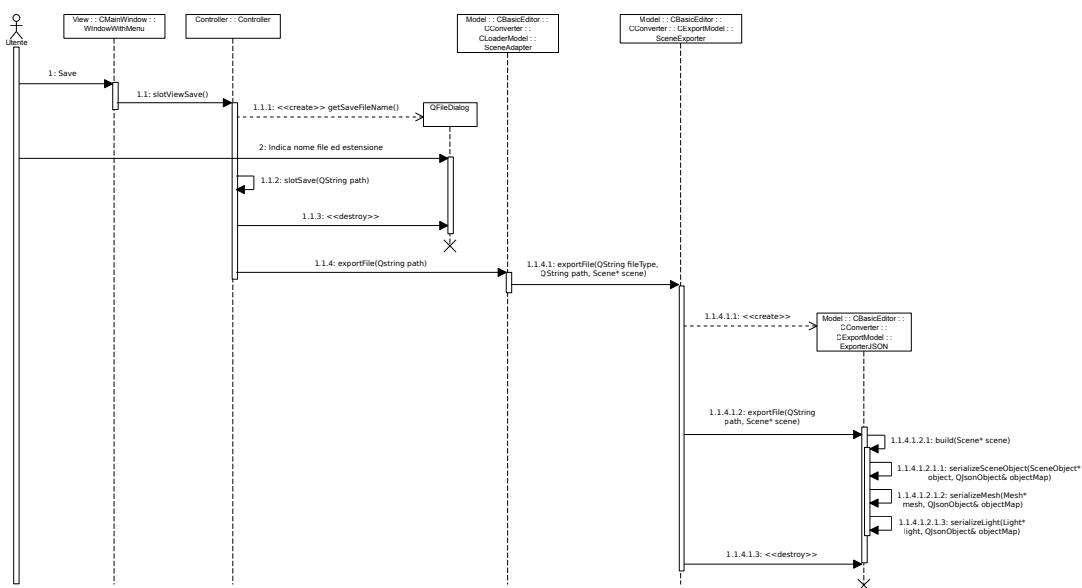


Figura 72: Diagrammi di sequenza per l'esportazione



4.3 Traslazione

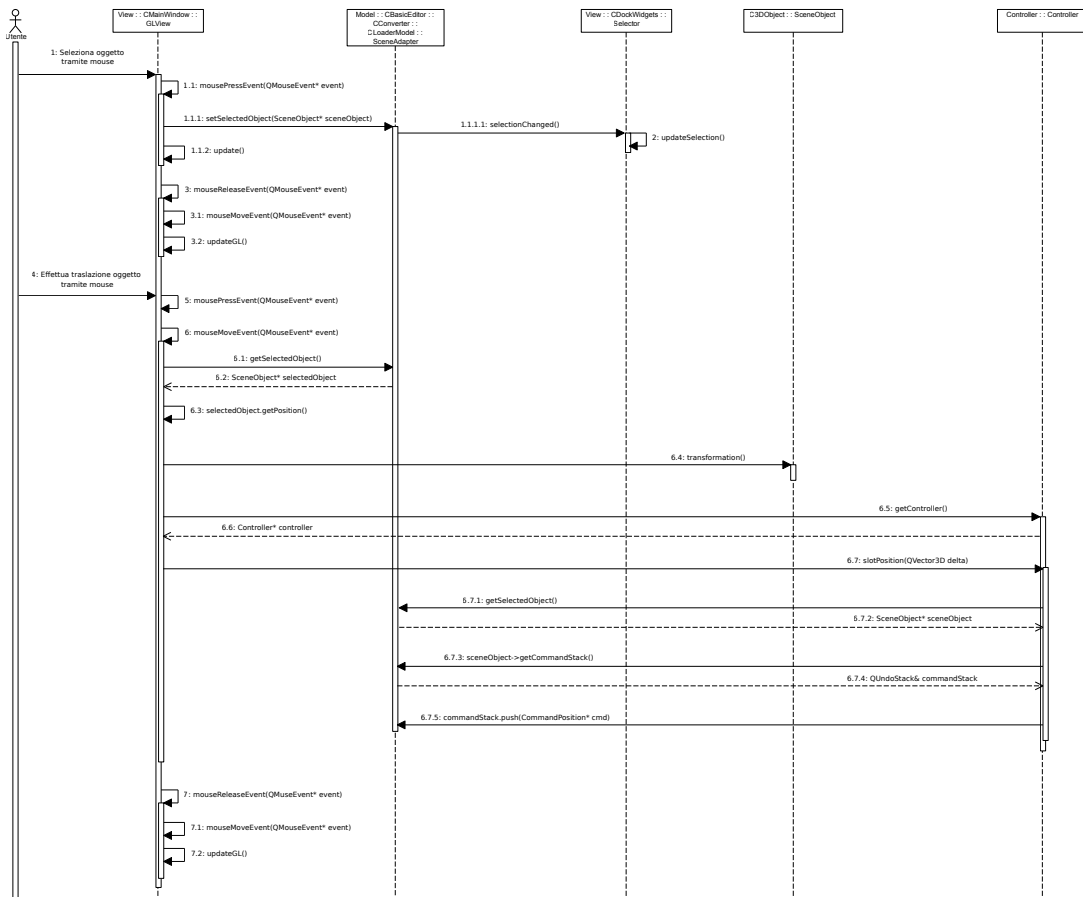


Figura 73: Diagrammi di sequenza per la traslazione di un oggetto



A Tracciamento

A.1 Tracciamento requisiti - classi

Requisito	Classi
R0F1	SceneExporter
↳ R0F1.1	ExporterJSON ExporterMinifiedJSON ExporterXML
↳ R0F1.2	NumericPrecisionType
↳ R0F1.3	SceneExporter
↳ R0F1.3.1	Vertex SceneExporter
↳ R0F1.3.2	Texture SceneExporter
↳ R0F1.3.3	SceneExporter
↳ R0F1.3.4	SceneExporter
↳ R1F1.3.5	Keyframe _G SceneExporter
↳ R0F1.4	ExporterJSON
↳ R0F1.4.1	ExporterJSON
↳ R2F1.5	ExporterXML
↳ R2F1.5.1	ExporterXML
↳ R2F1.6	DeviceLimit Setting
↳ R2F1.6.1	Setting
↳ R2F1.6.2	Setting
↳ R1F1.6.3	Setting
↳ R2F1.7	ExporterUBJSON
R2F7	MainWindow FacadeView
↳ R2F7.1	GLView
↳ R2F7.1.1	GLView
↳ R1F7.1.1.1	Setting QGLView
↳ R2F7.1.2	GLView
↳ R2F7.1.2.1	GLView
↳ R2F7.1.2.2	GLView
↳ R2F7.1.2.3	GLView
↳ R2F7.2	ErrorMessageBox
↳ R2F7.2.1	ErrorMessageBox
↳ R2F7.2.2	ErrorMessageBox
↳ R2F7.2.3	ErrorMessageBox
↳ R2F7.3	FacadeView
↳ R2F7.3.1	FacadeView
↳ R2F7.3.2	FacadeView
↳ R2F7.4	About
R0F8	FacadeView
↳ R0F8.1	SceneAdapter
↳ R2F8.2	ImporterJson
↳ R1F8.3	SceneObject CommandScene GLView
↳ R2F8.3.1	Transformation
↳ R2F8.3.1.1	SceneObject CommandEditObject
↳ R1F8.3.1.1.1	CommandRotation GLView
↳ R1F8.3.1.1.2	CommandPosition GLView



↪	R1F8.3.1.1.3	CommandScale GLView
↪	R1F8.3.1.1.4	CommandEmission Material
↘	R1F8.3.1.1.4.1	CommandColorSpecular Material
↘	R1F8.3.1.1.4.2	CommandColorDiffuse Material
↪	R1F8.3.1.1.5	Material
↘	R1F8.3.1.1.5.1	
↳	R2F8.3.1.2	CommandEditLight LightWidget
↪	R1F8.3.1.2.1	CommandColor CommandEmission Material
↪	R1F8.3.1.2.2	Material
↪	R1F8.3.1.2.3	CommandEditLight LightWidget
↪	R0F8.3.1.2.4	Transformation
↪	R2F8.3.1.2.5	Transformation
↳	R2F8.3.2	SceneAdapter CommandScene Action
↳	R0F8.3.3	GLView
↳	R1F8.3.4	CommandAddLight LightWidget
↳	R2F8.3.4.1	CommandAddLight LightWidget
↳	R1F8.3.5	BoundingBox3D Selector GLView
↳	R1F8.4	SceneAdapter
↳	R1F8.4.1	SceneAdapter

Tabella 2: Tabella requisiti funzionali / classi



A.2 Tracciamento classi - requisiti

Nella tabella sottostante sono presenti delle celle vuote in corrispondenza di alcune classi concrete, a causa del tracciamento dei requisiti all'interfaccia che espone i loro metodi.

Classe	Requisiti
Scene	
SceneAdapter	R0F8.1 R1F8.4 R1F8.4.1 R2F8.3.2
SceneExporter	R0F1 R0F1.3 R0F1.3.1 R0F1.3.2 R0F1.3.3 R0F1.3.4 R1F1.3.5
ExporterUBJSON	R2F1.7
ExporterXML	R0F1.1 R2F1.5 R2F1.5.1
ExporterMinifiedJSON	R0F1.1
Controller	
CommandTransform	
CommandAddLight	R1F8.3.4 R2F8.3.4.1
SceneObject	R1F8.3 R2F8.3.1.1
Light	
Mesh	
Setting	R1F1.6.3 R1F7.1.1.1 R2F1.6 R2F1.6.1 R2F1.6.2
AxisSlider	
ColorPicker	
LightType	
Material	R1F8.3.1.1.4 R1F8.3.1.1.4.1 R1F8.3.1.1.4.2 R1F8.3.1.1.5 R1F8.3.1.2.1 R1F8.3.1.2.2
Selector	R1F8.3.5



LightWidget	R1F8.3.1.2.3 R1F8.3.4 R2F8.3.1.2 R2F8.3.4.1
Action	R2F8.3.2
Transformation	R0F8.3.1.2.4 R2F8.3.1 R2F8.3.1.2.5
WindowWithMenu	
WindowWithDock	
WindowWithStatus	
Decorator	
MainWindow	R2F7
View3D	
DeviceLimit	R2F1.6
CommandEditMesh	
CommandEditLight	R1F8.3.1.2.3 R2F8.3.1.2
SettingsWindow	
CommandScene	R1F8.3 R2F8.3.2
CommandAddElement	
FacadeView	R0F8 R2F7 R2F7.3 R2F7.3.1 R2F7.3.2
ErrorMessageBox	R2F7.2 R2F7.2.1 R2F7.2.2 R2F7.2.3
About	R2F7.4
NumericPrecisionType	R0F1.2
CommandPosition	R1F8.3.1.1.2
CommandScale	R1F8.3.1.1.3
CommandRotation	R1F8.3.1.1.1
CommandLightType	
CommandShininess	
CommandEmission	R1F8.3.1.1.4 R1F8.3.1.2.1
CommandColor	R1F8.3.1.2.1
CommandColorEmission	
CommandColorSpecular	R1F8.3.1.1.4.1
CommandColorDiffuse	R1F8.3.1.1.4.2
CommandEditObject	R2F8.3.1.1



GLView	R0F8.3.3 R1F8.3 R1F8.3.1.1.1 R1F8.3.1.1.2 R1F8.3.1.1.3 R1F8.3.5 R2F7.1 R2F7.1.1 R2F7.1.2 R2F7.1.2.1 R2F7.1.2.2 R2F7.1.2.3
ImporterJson	R2F8.2
Texture	R0F1.3.2
Vertex	R0F1.3.1
Keyframe _G	R1F1.3.5
BoundingBox3D	R1F8.3.5
ExporterJSON	R0F1.1 R0F1.4 R0F1.4.1

Tabella 3: Tabella classi / requisiti

**A.3 Tracciamento modulo - test**

Metodo	Test
CommandTransform::CommandTransform()	TU30
CommandAddLight::CommandAddLight()	TU19
CommandEditMesh::CommandEditMesh()	TU30
CommandEditMesh::getMesh()	TU31
CommandEditLight::CommandEditLight()	TU30
CommandEditLight::getLight()	TU31
CommandScene::CommandScene()	TU30
CommandAddElement::CommandAddElement()	TU19
CommandAddElement::undo()	TU19
CommandAddElement::redo()	TU19
CommandPosition::CommandPosition()	TU25
CommandPosition::id()	TU30
CommandPosition::redo()	TU25
CommandPosition::undo()	TU25
CommandPosition::mergeWith()	TU25
CommandScale::CommandScale()	TU27
CommandScale::redo()	TU27
CommandScale::undo()	TU27
CommandScale::id()	TU30
CommandScale::mergeWith()	TU27
CommandRotation::CommandRotation()	TU26
CommandRotation::id()	TU30
CommandRotation::redo()	TU26
CommandRotation::undo()	TU26
CommandRotation::mergeWith()	TU26
CommandLightType::CommandLightType()	TU24
CommandLightType::undo()	TU24
CommandLightType::redo()	TU24
CommandShininess::CommandShininess()	TU28
CommandShininess::undo()	TU28
CommandShininess::redo()	TU28
CommandShininess::id()	TU30
CommandShininess::mergeWith()	TU28
CommandEmission::CommandEmission()	TU23
CommandEmission::undo()	TU23
CommandEmission::redo()	TU23
CommandEmission::id()	TU30
CommandEmission::mergeWith()	TU23
CommandColor::CommandColor()	TU30
CommandColorEmission::CommandColorEmission()	TU21
CommandColorEmission::undo()	TU21
CommandColorEmission::redo()	TU21
CommandColorSpecular::CommandColorSpecular()	TU22
CommandColorSpecular::redo()	TU22



CommandColorSpecular::undo()	TU22
CommandColorDiffuse::CommandColorDiffuse()	TU20
CommandColorDiffuse::redo()	TU20
CommandColorDiffuse::undo()	TU20
CommandEditObject::CommandEditObject()	TU30
Setting::getSetting()	TU16
Setting::signalSettingsUpdate()	TU30
Setting::Setting()	TU16
Setting::setNumericPrecision()	TU32
Setting::setSceneBackgroundColor()	TU32
Setting::getNumericPrecision()	TU32
Setting::getSceneBackgroundColor()	TU32
Setting::getSelectedDevice()	TU32
Setting::selectDevice()	TU15
DeviceLimit::DeviceLimit()	TU15
SceneExporter::exportFile()	
SceneExporter::exportFile()	TU14
ExporterUBJSON::exportFile()	TU35
ExporterUBJSON::build()	TU35
ExporterUBJSON::serializeSceneObject()	TU35
ExporterUBJSON::serializeMesh()	TU35
ExporterUBJSON::serializeLight()	TU35
ExporterUBJSON::serialize()	TU35
ExporterUBJSON::serialize()	TU35
ExporterUBJSON::serialize()	TU35
ExporterXML::exportFile()	TU33
ExporterXML::build()	TU33
ExporterXML::streamSceneObject()	TU33
ExporterXML::streamMesh()	TU33
ExporterXML::streamLight()	TU33
ExporterXML::stream _G ()	TU33
ExporterXML::stream _G ()	TU33
ExporterMinifiedJSON::exportFile()	TU30
ExporterJSON::exportFile()	TU34
ExporterJSON::build()	TU34
ExporterJSON::serializeSceneObject()	TU34
ExporterJSON::serializeMesh()	TU34
ExporterJSON::serializeLight()	TU34
ExporterJSON::serialize()	TU34
ExporterJSON::serialize()	TU34
ExporterJSON::serialize()	TU34
Scene::getObject()	TU32
Scene::exportFile()	
Scene::importFile()	
Scene::signalUpdatedScene()	TU30
Scene::signalError()	TU30
Scene::getCommandStack()	TU32



Scene::getObjectNames()	TU32
Scene::getSelectedObject()	TU31
Scene::setSelectObject()	TU31
Scene::getCommandStack()	TU31
Scene::selectByName()	
Scene::getObjectType()	TU31
Scene::update()	TU30
Scene::selectionChanged()	
Scene::fileLoaded()	
Scene::getAllObjects()	TU31
Scene::addObject()	
Scene::lightNumber()	
Scene::removeObject()	
SceneAdapter::SceneAdapter()	
SceneAdapter::getObject()	TU31
SceneAdapter::exportFile()	TU2
SceneAdapter::importFile()	TU1
SceneAdapter::signalUpdatedScene()	
SceneAdapter::signalError()	
SceneAdapter::getObjectNames()	
SceneAdapter::addLight()	TU3
SceneAdapter::selectByName()	TU29
SceneAdapter::buildLights()	TU1
SceneAdapter::buildMeshes()	TU1
SceneAdapter::meshCreator()	TU1
SceneAdapter::lightCreator()	TU1
SceneAdapter::importLimits()	TU1
SceneAdapter::lightNumber()	TU1
SceneAdapter::removeObject()	TU3
SceneAdapter::addObject()	TU1
SceneAdapter::getAllObjects()	TU1
ImporterJson::importFile ()	
ImporterJson::readObject()	
ImporterJson::readLight()	
ImporterJson::readMesh()	
ImporterJson::readObjectProperties()	
Controller::slotRotation()	TU30
Controller::slotScale()	TU30
Controller::slotPosition()	TU30
Controller::slotAddLight()	TU30
Controller::slotColorDiffuse()	TU30
Controller::slotColorSpecular()	TU30
Controller::slotColorEmission()	TU30
Controller::slotSelectObject()	TU30
Controller::slotLightType()	TU30
Controller::slotMeshShininess()	TU30
Controller::slotEmission()	TU30



Controller::slotSettings()	TU30
Controller::slotOpen()	TU30
Controller::slotSave()	TU30
Controller::slotViewAbout()	TU30
Controller::slotViewSave()	TU30
Controller::slotViewOpen()	TU30
Controller::slotViewHelp()	TU30
Controller::slotViewSettings()	TU30
Controller::Controller()	TU30
Controller::getController()	TU32
Controller::getScene()	TU31
Controller::start3DMob()	TU30
FacadeView::getMainWindow()	TU32
FacadeView::showHelpWindow()	
FacadeView::showSettings()	
FacadeView::showAbout()	
FacadeView::showMainWindow()	
FacadeView::FacadeView()	
FacadeView::operator=()	
FacadeView::FacadeView()	
FacadeView::operator=()	
FacadeView::getErrorMessageBox()	
About::About()	
Material::setDiffuseColor()	TU32
Material::getDiffuseColor()	TU32
Material::getSpecularColor()	TU32
Material::getEmissionColor()	TU32
Material::setEmissionColor()	TU31
Material::getShininess()	TU31
Material::setShininess()	TU31
Material::Material()	
Material::setSpecularColor ()	TU31
Material::update()	
Selector::Selector()	
Selector::updateObjects()	
Selector::autoUpdateObjects()	
Selector::updateSelection()	
LightWidget::LightWidget()	TU30
LightWidget::getLightType()	TU31
LightWidget::update()	
LightWidget::lightTypeChanged()	
Action::Action()	
Action::setUndoStack()	TU31
Transformation::getPosition()	TU31
Transformation::getRotation()	TU31
Transformation::getScale()	TU31
Transformation::setPosition()	TU31



Transformation::setRotation()	TU31
Transformation::setScale()	TU31
Transformation::update()	
Transformation::scaleChanged()	
Transformation::positionChanged()	
Transformation::rotationChanged()	
WindowWithMenu::WindowWithMenu()	TU30
WindowWithMenu::draw()	
WindowWithMenu::getWrappedWindow()	TU31
WindowWithDock::WindowWithDock()	TU30
WindowWithDock::draw()	TU30
WindowWithDock::getWrappedWindow()	TU31
WindowWithStatus::WindowWithStatus()	TU30
WindowWithStatus::draw()	
WindowWithStatus::getWrappedWindow()	TU31
Decorator::Decorator()	
Decorator::getWrappedWindow()	TU31
MainWindow::draw()	
MainWindow::getWrappedWindow()	TU31
View3D::View3D()	TU30
View3D::getSceneModel()	TU31
View3D::draw()	
View3D::getWrappedWindow()	TU31
GLView::paintGL()	
GLView::GLView()	
GLView::getSceneModel()	TU31
GLView::initializeObjects()	
GLView::updateBackgroundColor()	
GLView::initializeGL()	
GLView::paintGL()	
GLView::resizeGL()	
GLView::mousePressEvent()	
GLView::mouseMoveEvent()	
GLView::mouseReleaseEvent()	
GLView::wheelEvent()	
GLView::keyPressEvent()	
GLView::keyReleaseEvent()	
GLView::renderAxes()	
GLView::selectionChanged()	
SettingsWindow::SettingsWindow()	
SettingsWindow::getDouble()	TU32
SettingsWindow::getColor()	TU32
SettingsWindow::selectLimits()	
ErrorMessageBox::slotError()	TU30
AxisSlider::AxisSlider()	
AxisSlider::setValue()	TU32
AxisSlider::getValue()	TU32



AxisSlider::axisSlid()	
ColorPicker::updateColor()	
ColorPicker::updateColorFromDialog()	
ColorPicker::ColorPicker()	
ColorPicker::ColorPicker()	
ColorPicker::setColor()	TU32
ColorPicker::getColor()	TU32
ColorPicker::mouseReleaseEvent()	
ColorPicker::changeEvent()	
ColorPicker::colorCanged()	
SceneObject::getDiffuseColor()	TU32
SceneObject::getEmission()	TU32
SceneObject::getEmissionColor()	TU32
SceneObject::getPosition()	TU32
SceneObject::getRotation()	TU32
SceneObject::getScale()	TU32
SceneObject::getSpecularColor()	TU32
SceneObject::setDiffuseColor()	TU8
SceneObject::setEmission()	TU7
SceneObject::setEmissionColor()	TU17
SceneObject::setPosition()	TU6
SceneObject::setRotation()	TU4
SceneObject::setScale()	TU5
SceneObject::setSpecularColor()	TU18
SceneObject::transformation()	TU4
SceneObject::initialize()	
SceneObject::render()	
SceneObject::SceneObject()	
SceneObject::renderInternal()	
SceneObject::getBoundingBox()	TU32
SceneObject::getKeyframes()	TU32
SceneObject::pushKeyframe()	
Light::getLightType()	TU32
Light::Light()	
Light::setLightType()	TU11
Light::renderInternal()	
Light::operator=()	
Light::initialize()	
Light::Light()	
Mesh::getShininess()	TU32
Mesh::setShininess()	TU12
Mesh::Mesh()	
Mesh::renderInternal()	
Mesh::updateBoundingBox()	
Mesh::initialize()	
Mesh::setGeometry()	TU31
Mesh::getGeometry()	TU31



Mesh::addFace()	
Mesh::addFace()	
Mesh::getTexture()	TU31
Mesh::setTexture()	TU31
Mesh::operator== ()	
Texture::Texture()	
Texture::getPath()	TU31
Texture:: getTexture()	TU31
Texture::setPath()	TU31
Texture::initialize()	
Texture::render()	
Vertex::Vertex()	
Vertex::getPosition()	TU31
Vertex::getNormal()	TU31
Vertex::getColor()	TU31
Vertex::getUv()	TU31
Vertex::setPosition()	TU31
Vertex::setNormal()	TU31
Vertex::setColor()	TU31
Vertex::setUv()	TU31
Vertex::render()	
Vertex::operator== ()	
Keyframe _G ::Keyframe _G ()	
Keyframe _G ::getTime()	TU31
Keyframe _G ::setTime()	TU31
Keyframe _G ::getPosition()	TU31
Keyframe _G ::setPosition()	TU31
BoundingBox3D::BoundingBox3D()	
BoundingBox3D::BoundingBox3D()	
BoundingBox3D::center()	
BoundingBox3D::size()	
BoundingBox3D::max()	
BoundingBox3D::min()	
BoundingBox3D::setMax()	TU31
BoundingBox3D::setMin()	TU32
BoundingBox3D::clear()	
BoundingBox3D::isEmpty()	
BoundingBox3D::include()	
BoundingBox3D::include()	
BoundingBox3D::contains()	
BoundingBox3D::intersects()	
BoundingBox3D::scale()	
BoundingBox3D::translate()	

Tabella 4: Tabella metodi / test unità



B Schemi file esportati

B.1 JSONSchema

Viene riportato il JSONSchema che i file JSON_G esportati rispettano.

Tale schema permette di definire la struttura del file JSON_G esportati e di validarli.

```
{
  "$schema": "http://\glo{json}-schema.org/draft-04/schema#",
  "title": "3DMob",
  "description": "Schema del file \glo{JSON} esportato da 3DMob",
  "type": "object",
  "items": {
    "title": "3DMob",
    "type": "object",
    "properties": {
      ".*": {
        "description": "Mesh o Luce della \glo{scena}",
        "type": "array",
        "items": {
          "type": "object",
          "properties": {
            "object": {
              "description": "Oggetto 3D con caratteristiche condivise tra mesh e luce",
              "type": "object",
              "properties": {
                "position": {
                  "type": "array",
                  "items": {
                    "type": "number"
                  },
                  "minItems": 3,
                  "maxItems": 3
                },
                "rotation": {
                  "type": "array",
                  "items": {
                    "type": "number"
                  },
                  "minItems": 3,
                  "maxItems": 3
                },
                "scale": {
                  "type": "array",
                  "items": {
                    "type": "number"
                  },
                  "minItems": 3,
                  "maxItems": 3
                },
                "diffuseColor": {
                  "type": "string"
                },
                "emission": {
                  "type": "number"
                }
              }
            }
          }
        }
      }
    }
  }
}
```




```

    },
    "emissionColor":{
      "type": "string"
    },
    "specularColor":{
      "type": "string"
    },
    "keyframes":{
      "description": "animazioni presenti sull'oggetto 3D",
      "type": "array",
      "items": {
        "title" : "animazione presente sulla \glo{scena}",
        "type" : "object",
        "properties" : {
          "position": {
            "type": "array",
            "items": {
              "type": "number"
            },
            "minItems": 3,
            "maxItems": 3
          },
          "time":{
            "type": "number"
          }
        }
      }
    },
    "mesh":{
      "title" : "caratteristiche uniche della mesh",
      "type" : "object",
      "properties" : {
        "geometry":{
          "type": "array",
          "items": {
            "title" : "facce della mesh",
            "type" : "array",
            "items":{
              "type": "object",
              "properties":{
                "position": {
                  "type": "array",
                  "items": {
                    "type": "number"
                  },
                  "minItems": 3,
                  "maxItems": 3
                },
                "normal": {
                  "type": "array",
                  "items": {
                    "type": "number"
                  },
                  "minItems": 3,

```



```

        "maxItems": 3
    },
    "uv": {
        "type": "array",
        "items": {
            "type": "number"
        },
        "minItems": 3,
        "maxItems": 3
    },
    "color": {
        "type": "string"
    }
}
}
},
"shininess": {
    "type": "number"
},
"texture": {
    "type": "string"
}
},
"minItems": 0
},
"light": {
    "title" : "caratteristiche uniche della luce",
    "type" : "object",
    "properties" : {
        "lightType": {
            "type": "integer"
        }
    },
    "minItems": 0
}
}
}
},
"required": [".*"]
}
}

```

B.2 XMLSchema

Viene riportato il XMLSchema che i file XML_G esportati rispettano. Tale schema permette di definire la struttura del file XML_G esportati e di validarli.

```

<?\glo{xml} version="1.0" encoding="\glo{UTF-8}"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="3dMob"
xmlns="3dMob"
elementFormDefault="qualified">

```



```
<xs:element name="scene" type="Tscena" />

<xs:complexType name="Tscena">
  <xs:sequence>
    <xs:element name="mesh" type="Tmesh" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="light" type="Tlight" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="Tlight">
  <xs:sequence>
    <xs:element name="object" type="Tobject"/>
    <xs:element name="lightType" type="xs:integer"/>
    <xs:element name="keyframes" type="Tkeyframes"/>
  </xs:sequence>
  <xs:attribute name="name" type="xs:string"/>
</xs:complexType>

<xs:complexType name="Tmesh">
  <xs:sequence>
    <xs:element name="object" type="Tobject"/>
    <xs:element name="shininess" type="xs:integer"/>
    <xs:element name="texture" type="xs:string"/>
    <xs:element name="geometry" type="Tgeometry"/>
    <xs:element name="keyframes" type="Tkeyframes"/>
  </xs:sequence>
  <xs:attribute name="name" type="xs:integer"/>
</xs:complexType>

<xs:complexType name="Tobject">
  <xs:sequence>
    <xs:element name="diffuseColor" type="xs:string"/>
    <xs:element name="emission" type="xs:integer"/>
    <xs:element name="emissionColor" type="xs:string"/>
    <xs:element name="position" type="Tvector3D"/>
    <xs:element name="rotation" type="Tvector3D"/>
    <xs:element name="scale" type="Tvector3D"/>
    <xs:element name="specularColor" type="xs:string"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="Tvector3D">
  <xs:sequence>
    <xs:element name="x" type="xs:double"/>
    <xs:element name="y" type="xs:double"/>
    <xs:element name="z" type="xs:double"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="Tgeometry">
  <xs:sequence>
    <xs:element name="face" type="Tface" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="Tface">
```



```
<xs:sequence>
<xs:element name="vertex" type="Tvertex" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>

<xs:complexType name="Tvertex">
<xs:sequence>
<xs:element name="position" type="Tvector3D"/>
<xs:element name="normal" type="Tnormal"/>
<xs:element name="uv" type="Tuv"/>
<xs:element name="color" type="xs:string"/>
</xs:sequence>
</xs:complexType>

<xs:complexType name="Tnormal">
<xs:sequence>
<xs:element name="x" type="xs:double"/>
<xs:element name="y" type="xs:double"/>
<xs:element name="z" type="xs:double"/>
</xs:sequence>
</xs:complexType>

<xs:complexType name="Tuv">
<xs:sequence>
<xs:element name="x" type="xs:double"/>
<xs:element name="y" type="xs:double"/>
</xs:sequence>
</xs:complexType>

<xs:complexType name="Tkeyframes">
<xs:sequence>
<xs:element name="time" type="Tkeyframe" minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>

<xs:complexType name="Tkeyframe">
<xs:sequence>
<xs:element name="time" type="xs:double"/>
<xs:element name="position" type="Tvector3D"/>
</xs:sequence>
</xs:complexType>

</xs:schema>
```