

Dementia

1.0

Generat per Doxygen 1.8.4

Dv Mai 31 2013 02:39:41



# Índex

<b>1</b>	<b>Índex d'Espais de Noms</b>	<b>1</b>
1.1	Llista dels Espais de Noms	1
<b>2</b>	<b>Índex Jeràrquic</b>	<b>3</b>
2.1	Jerarquia de Classes	3
<b>3</b>	<b>Índex de Classes</b>	<b>5</b>
3.1	Llista de Classes	5
<b>4</b>	<b>Índex de Fitxers</b>	<b>7</b>
4.1	Llista dels Fitxers	7
<b>5</b>	<b>Documentació de l'Espai de Noms</b>	<b>9</b>
5.1	Referència de l'Espai de Noms Dementia	9
5.1.1	Descripció Detallada	10
5.1.2	Documentació de les Funcions	10
5.1.2.1	CompileShaderFromFile	10
5.1.2.2	MainWndProc	10
<b>6</b>	<b>Documentació de les Classes</b>	<b>13</b>
6.1	Referència de la Classe Dementia::Camera	13
6.1.1	Descripció Detallada	14
6.1.2	Documentació del Constructor i el Destructor	14
6.1.2.1	Camera	14
6.1.2.2	~Camera	15
6.1.3	Documentació de les Funcions Membre	15
6.1.3.1	GetAspect	15
6.1.3.2	GetFarWindowHeight	15
6.1.3.3	GetFarWindowWidth	15
6.1.3.4	GetFarZ	15
6.1.3.5	GetFovX	15
6.1.3.6	GetFovY	16
6.1.3.7	GetLook	16

6.1.3.8	GetNearWindowHeight	16
6.1.3.9	GetNearWindowWidth	16
6.1.3.10	GetNearZ	17
6.1.3.11	GetPosition	17
6.1.3.12	GetRight	17
6.1.3.13	GetUp	18
6.1.3.14	LookAt	18
6.1.3.15	Pitch	18
6.1.3.16	Proj	18
6.1.3.17	RotateY	19
6.1.3.18	SetLens	19
6.1.3.19	SetPosition	19
6.1.3.20	Strafe	19
6.1.3.21	UpdateViewMatrix	20
6.1.3.22	View	20
6.1.3.23	ViewProj	20
6.1.3.24	Walk	21
6.2	Referència de l'Estructura Dementia::DX11::ConstantBuffer	21
6.2.1	Descripció Detallada	22
6.2.2	Documentació de les Dades Membre	22
6.2.2.1	gDirLight	22
6.2.2.2	gEyePosW	22
6.2.2.3	gMaterial	22
6.2.2.4	gPointLight	22
6.2.2.5	gSpotLight	22
6.2.2.6	mProjection	22
6.2.2.7	mView	22
6.2.2.8	mWorld	22
6.2.2.9	pad	23
6.3	Referència de l'Estructura Dementia::DirectionalLight	23
6.3.1	Descripció Detallada	23
6.3.2	Documentació del Constructor i el Destructor	23
6.3.2.1	DirectionalLight	23
6.3.2.2	DirectionalLight	24
6.3.3	Documentació de les Dades Membre	24
6.3.3.1	Ambient	24
6.3.3.2	Diffuse	24
6.3.3.3	Direction	24
6.3.3.4	Pad	24
6.3.3.5	Specular	24

6.4	Referència de la Classe Dementia::DX11	24
6.4.1	Descripció Detallada	25
6.4.2	Documentació del Constructor i el Destructor	25
6.4.2.1	DX11	25
6.4.2.2	~DX11	25
6.4.3	Documentació de les Funcions Membre	26
6.4.3.1	close	26
6.4.3.2	drawScene	26
6.4.3.3	DX11::msgProc	27
6.4.3.4	getBackgroundColor	28
6.4.3.5	getCamera	28
6.4.3.6	getDevice	28
6.4.3.7	onResize	28
6.4.3.8	setBackgroundColor	29
6.4.3.9	updateScene	29
6.5	Referència de la Classe Dementia::Entity	30
6.5.1	Descripció Detallada	31
6.5.2	Documentació de les Enumeracions Membre	31
6.5.2.1	TYPE	31
6.5.3	Documentació del Constructor i el Destructor	32
6.5.3.1	Entity	32
6.5.3.2	~Entity	32
6.5.4	Documentació de les Funcions Membre	32
6.5.4.1	getId	32
6.5.4.2	getType	32
6.6	Referència de la Classe Dementia::GameTimer	33
6.6.1	Descripció Detallada	33
6.6.2	Documentació del Constructor i el Destructor	33
6.6.2.1	GameTimer	33
6.6.3	Documentació de les Funcions Membre	33
6.6.3.1	getDeltaTime	33
6.6.3.2	getGameTime	34
6.6.3.3	reset	34
6.6.3.4	start	35
6.6.3.5	stop	35
6.6.3.6	tick	35
6.7	Referència de la Classe Dementia::Geometry	36
6.7.1	Descripció Detallada	36
6.7.2	Documentació del Constructor i el Destructor	36
6.7.2.1	Geometry	36

6.7.2.2	~Geometry	37
6.7.3	Documentació de les Dades Membre	37
6.7.3.1	ib	37
6.7.3.2	Indices	37
6.7.3.3	vb	37
6.7.3.4	Vertices	37
6.8	Referència de la Classe Dementia::GeometryFactory	37
6.8.1	Descripció Detallada	38
6.8.2	Documentació del Constructor i el Destructor	38
6.8.2.1	GeometryFactory	38
6.8.2.2	~GeometryFactory	38
6.8.3	Documentació de les Funcions Membre	38
6.8.3.1	CreateBox	38
6.8.3.2	CreateCylinder	38
6.8.3.3	CreateFullscreenQuad	38
6.8.3.4	CreateGeosphere	38
6.8.3.5	CreateGrid	39
6.8.3.6	CreateSphere	39
6.9	Referència de la Classe Dementia::InputManager	39
6.9.1	Descripció Detallada	40
6.9.2	Documentació del Constructor i el Destructor	40
6.9.2.1	InputManager	40
6.9.2.2	~InputManager	40
6.9.3	Documentació de les Funcions Membre	40
6.9.3.1	getKeyboard	40
6.9.3.2	getMouse	40
6.9.3.3	processEvent	40
6.10	Referència de la Classe Dementia::Keyboard	41
6.10.1	Descripció Detallada	42
6.10.2	Documentació del Constructor i el Destructor	42
6.10.2.1	Keyboard	42
6.10.2.2	~Keyboard	42
6.10.3	Documentació de les Funcions Membre	42
6.10.3.1	isKeyPressed	42
6.10.3.2	setKeyPressed	43
6.10.3.3	setKeyReleased	43
6.11	Referència de la Classe Dementia::Light	43
6.11.1	Descripció Detallada	44
6.11.2	Documentació de les Enumeracions Membre	45
6.11.2.1	LIGHT_TYPE	45

6.11.3 Documentació del Constructor i el Destructor . . . . .	45
6.11.3.1 Light . . . . .	45
6.11.3.2 ~Light . . . . .	45
6.12 Referència de l'Estructura Dementia::Material . . . . .	45
6.12.1 Descripció Detallada . . . . .	46
6.12.2 Documentació del Constructor i el Destructor . . . . .	46
6.12.2.1 Material . . . . .	46
6.12.2.2 Material . . . . .	46
6.12.3 Documentació de les Dades Membre . . . . .	46
6.12.3.1 Ambient . . . . .	46
6.12.3.2 Diffuse . . . . .	47
6.12.3.3 Reflect . . . . .	47
6.12.3.4 Specular . . . . .	47
6.13 Referència de la Classe Dementia::MathHelper . . . . .	47
6.13.1 Descripció Detallada . . . . .	48
6.13.2 Documentació de les Funcions Membre . . . . .	48
6.13.2.1 AngleFromXY . . . . .	48
6.13.2.2 Clamp . . . . .	48
6.13.2.3 InverseTranspose . . . . .	49
6.13.2.4 Lerp . . . . .	49
6.13.2.5 Max . . . . .	49
6.13.2.6 Min . . . . .	50
6.13.2.7 RandF . . . . .	51
6.13.2.8 RandF . . . . .	51
6.13.2.9 RandHemisphereUnitVec3 . . . . .	53
6.13.2.10 RandUnitVec3 . . . . .	53
6.13.3 Documentació de les Dades Membre . . . . .	53
6.13.3.1 Infinity . . . . .	53
6.13.3.2 Pi . . . . .	53
6.14 Referència de la Classe Dementia::Mesh . . . . .	54
6.14.1 Descripció Detallada . . . . .	55
6.14.2 Documentació del Constructor i el Destructor . . . . .	55
6.14.2.1 Mesh . . . . .	55
6.14.2.2 Mesh . . . . .	55
6.14.2.3 Mesh . . . . .	55
6.14.2.4 ~Mesh . . . . .	56
6.14.3 Documentació de les Funcions Membre . . . . .	56
6.14.3.1 getGeometry . . . . .	56
6.14.3.2 getShader . . . . .	56
6.14.3.3 getTexture . . . . .	56

6.14.3.4	setGeometry	57
6.14.3.5	setShader	57
6.14.3.6	setTexture	57
6.15	Referència de la Classe Dementia::Mouse	58
6.15.1	Descripció Detallada	58
6.15.2	Documentació de les Enumeracions Membre	59
6.15.2.1	Button	59
6.15.3	Documentació del Constructor i el Destructor	59
6.15.3.1	Mouse	59
6.15.3.2	Mouse	59
6.15.3.3	~Mouse	59
6.15.4	Documentació de les Funcions Membre	59
6.15.4.1	getPosition	59
6.15.4.2	isButtonPressed	60
6.15.4.3	isMoved	60
6.15.4.4	setButtonPressed	60
6.15.4.5	setButtonReleased	60
6.15.4.6	setCenter	61
6.15.4.7	setPosition	61
6.15.4.8	updatePosition	61
6.16	Referència de la Classe Dementia::Node	62
6.16.1	Descripció Detallada	63
6.16.2	Documentació del Constructor i el Destructor	63
6.16.2.1	Node	63
6.16.2.2	Node	63
6.16.2.3	~Node	63
6.16.3	Documentació de les Funcions Membre	64
6.16.3.1	addChild	64
6.16.3.2	getAllNodes	64
6.16.3.3	getChild	64
6.16.3.4	getEntity	65
6.16.3.5	getId	65
6.16.3.6	getLocalPosition	65
6.16.3.7	getLocalRotation	65
6.16.3.8	getLocalScale	65
6.16.3.9	getParent	66
6.16.3.10	getWorldMatrix	66
6.16.3.11	getWorldPosition	66
6.16.3.12	getWorldRotation	66
6.16.3.13	getWorldScale	67



6.16.3.14 removeChild . . . . .	67
6.16.3.15 setEntity . . . . .	68
6.16.3.16 setLocalPosition . . . . .	68
6.16.3.17 setLocalRotation . . . . .	68
6.16.3.18 setLocalScale . . . . .	69
6.16.3.19 setParent . . . . .	69
6.17 Referència de l'Estructura Dementia::PointLight . . . . .	70
6.17.1 Descripció Detallada . . . . .	71
6.17.2 Documentació del Constructor i el Destructor . . . . .	71
6.17.2.1 PointLight . . . . .	71
6.17.2.2 PointLight . . . . .	71
6.17.3 Documentació de les Dades Membre . . . . .	72
6.17.3.1 Ambient . . . . .	72
6.17.3.2 Att . . . . .	72
6.17.3.3 Diffuse . . . . .	72
6.17.3.4 Pad . . . . .	72
6.17.3.5 Position . . . . .	72
6.17.3.6 Range . . . . .	72
6.17.3.7 Specular . . . . .	72
6.18 Referència de la Classe Dementia::SceneManager . . . . .	72
6.18.1 Descripció Detallada . . . . .	73
6.18.2 Documentació del Constructor i el Destructor . . . . .	74
6.18.2.1 SceneManager . . . . .	74
6.18.2.2 ~SceneManager . . . . .	74
6.18.3 Documentació de les Funcions Membre . . . . .	74
6.18.3.1 createDirectionalLight . . . . .	74
6.18.3.2 createMaterial . . . . .	74
6.18.3.3 createMesh . . . . .	75
6.18.3.4 createNode . . . . .	75
6.18.3.5 createPointLight . . . . .	76
6.18.3.6 createSpotLight . . . . .	76
6.18.3.7 createTexture . . . . .	76
6.18.3.8 getAllDirectionalLights . . . . .	77
6.18.3.9 getAllMeshes . . . . .	77
6.18.3.10 getAllNodes . . . . .	77
6.18.3.11 getAllPointLights . . . . .	78
6.18.3.12 getAllSpotLights . . . . .	78
6.18.3.13 getRootSceneNode . . . . .	78
6.18.3.14 removeNode . . . . .	78
6.19 Referència de la Classe Dementia::Shader . . . . .	79

6.19.1	Descripció Detallada	79
6.19.2	Documentació del Constructor i el Destructor	80
6.19.2.1	Shader	80
6.19.2.2	~Shader	80
6.19.3	Documentació de les Dades Membre	80
6.19.3.1	dx_constantBuffer	80
6.19.3.2	g_pPixelShader	80
6.19.3.3	g_pVertexLayout	80
6.19.3.4	g_pVertexShader	80
6.20	Referència de l'Estructura Dementia::SpotLight	80
6.20.1	Descripció Detallada	81
6.20.2	Documentació del Constructor i el Destructor	81
6.20.2.1	SpotLight	81
6.20.2.2	SpotLight	81
6.20.3	Documentació de les Dades Membre	82
6.20.3.1	Ambient	82
6.20.3.2	Att	82
6.20.3.3	Diffuse	82
6.20.3.4	Direction	82
6.20.3.5	Pad	82
6.20.3.6	Position	82
6.20.3.7	Range	82
6.20.3.8	Specular	83
6.20.3.9	Spot	83
6.21	Referència de la Classe Dementia::Texture	83
6.21.1	Descripció Detallada	83
6.21.2	Documentació del Constructor i el Destructor	83
6.21.2.1	Texture	83
6.21.2.2	~Texture	84
6.21.3	Documentació de les Dades Membre	84
6.21.3.1	g_pTextureRV	84
6.22	Referència de l'Estructura Dementia::Geometry::Vertex	84
6.22.1	Descripció Detallada	84
6.22.2	Documentació del Constructor i el Destructor	85
6.22.2.1	Vertex	85
6.22.2.2	Vertex	85
6.22.2.3	Vertex	85
6.22.3	Documentació de les Dades Membre	85
6.22.3.1	Normal	85
6.22.3.2	Position	86

6.22.3.3	TangentU	86
6.22.3.4	TexC	86
6.23	Referència de la Classe Dementia::Window	86
6.23.1	Descripció Detallada	87
6.23.2	Documentació del Constructor i el Destructor	87
6.23.2.1	Window	87
6.23.2.2	~Window	88
6.23.3	Documentació de les Funcions Membre	88
6.23.3.1	getBackgroundColor	88
6.23.3.2	getCamera	89
6.23.3.3	getDevice	89
6.23.3.4	getGameTimer	90
6.23.3.5	getHeight	90
6.23.3.6	getId	91
6.23.3.7	getInputManager	91
6.23.3.8	getSceneManager	91
6.23.3.9	getTitle	91
6.23.3.10	getWidth	92
6.23.3.11	getWindowHandle	92
6.23.3.12	init	93
6.23.3.13	isPaused	93
6.23.3.14	isVsync	93
6.23.3.15	isWindowed	93
6.23.3.16	msgProc	93
6.23.3.17	release	94
6.23.3.18	run	95
6.23.3.19	setBackgroundColor	96
6.23.3.20	setHeight	97
6.23.3.21	setTitle	97
6.23.3.22	setVsync	97
6.23.3.23	setWidth	97
6.23.3.24	setWindowed	97
<b>7</b>	<b>Documentació dels Fitxers</b>	<b>99</b>
7.1	Referència del Fitxer Include/Camera.h	99
7.2	Referència del Fitxer Include/Dementia.h	100
7.2.1	Documentació de les Definicions	101
7.2.1.1	HR	101
7.2.1.2	ReleaseCOM	101
7.2.1.3	UTILS_API	101

7.2.1.4	UTILS_EXPORT	101
7.2.1.5	WIN32_LEAN_AND_MEAN	101
7.2.2	Documentació de les Funcions	101
7.2.2.1	showErrorMessage	101
7.3	Referència del Fitxer Include/DX11.h	102
7.4	Referència del Fitxer Include/Entity.h	103
7.5	Referència del Fitxer Include/GameTimer.h	104
7.6	Referència del Fitxer Include/Geometry.h	104
7.7	Referència del Fitxer Include/GeometryFactory.h	105
7.8	Referència del Fitxer Include/InputManager.h	107
7.9	Referència del Fitxer Include/Keyboard.h	107
7.10	Referència del Fitxer Include/Light.h	108
7.11	Referència del Fitxer Include/MathHelper.h	109
7.12	Referència del Fitxer Include/Mesh.h	110
7.13	Referència del Fitxer Include/Mouse.h	111
7.14	Referència del Fitxer Include/Node.h	112
7.15	Referència del Fitxer Include/SceneManager.h	113
7.16	Referència del Fitxer Include/Shader.h	115
7.17	Referència del Fitxer Include/stdafx.h	116
7.17.1	Documentació de les Definicions	117
7.17.1.1	UTILS_EXPORT	117
7.17.1.2	WIN32_LEAN_AND_MEAN	117
7.18	Referència del Fitxer Include/targetver.h	117
7.19	Referència del Fitxer Include/Texture.h	117
7.20	Referència del Fitxer Include/Window.h	118
7.21	Referència del Fitxer src/Camera.cpp	120
7.22	Referència del Fitxer src/Dementia.cpp	120
7.22.1	Documentació de les Funcions	120
7.22.1.1	showErrorMessage	120
7.23	Referència del Fitxer src/dllmain.cpp	121
7.23.1	Documentació de les Funcions	121
7.23.1.1	DllMain	121
7.24	Referència del Fitxer src/DX11.cpp	121
7.25	Referència del Fitxer src/Entity.cpp	122
7.26	Referència del Fitxer src/GameTimer.cpp	122
7.27	Referència del Fitxer src/Geometry.cpp	123
7.28	Referència del Fitxer src/GeometryFactory.cpp	123
7.29	Referència del Fitxer src/InputManager.cpp	124
7.30	Referència del Fitxer src/Keyboard.cpp	125
7.31	Referència del Fitxer src/Light.cpp	125

7.32 Referència del Fitxer <code>src/MathHelper.cpp</code> . . . . .	126
7.33 Referència del Fitxer <code>src/Mesh.cpp</code> . . . . .	126
7.34 Referència del Fitxer <code>src/Mouse.cpp</code> . . . . .	127
7.35 Referència del Fitxer <code>src/Node.cpp</code> . . . . .	127
7.36 Referència del Fitxer <code>src/SceneManager.cpp</code> . . . . .	128
7.37 Referència del Fitxer <code>src/Shader.cpp</code> . . . . .	128
7.38 Referència del Fitxer <code>src/Texture.cpp</code> . . . . .	129
7.39 Referència del Fitxer <code>src/Window.cpp</code> . . . . .	130



# Capítol 1

## Índex d’Espais de Noms

### 1.1 Llista dels Espais de Noms

Aquests són tots els espais de noms amb breus descripcions:

#### [Dementia](#)

El espai de nom principal del framework. Totes les clases basiques o importants es troben aqui dins. . . . .





## Capítol 2

# Índex Jeràrquic

### 2.1 Jerarquia de Classes

Aquesta llista d'herència està ordenada toscament, però no completa, de forma alfabètica:

Dementia::Camera	13
Dementia::DX11::ConstantBuffer	21
Dementia::DirectionalLight	23
Dementia::DX11	24
Dementia::Entity	30
Dementia::Light	43
Dementia::Mesh	54
Dementia::GameTimer	33
Dementia::Geometry	36
Dementia::GeometryFactory	37
Dementia::InputManager	39
Dementia::Keyboard	41
Dementia::Material	45
Dementia::MathHelper	47
Dementia::Mouse	58
Dementia::Node	62
Dementia::PointLight	70
Dementia::SceneManager	72
Dementia::Shader	79
Dementia::SpotLight	80
Dementia::Texture	83
Dementia::Geometry::Vertex	84
Dementia::Window	86



## Capítol 3

# Índex de Classes

### 3.1 Llista de Classes

Aquestes són les classes, estructures, unions i interfícies acompanyades amb breus descripcions:

<a href="#">Dementia::Camera</a>	Una camara simple pensada per ser utilitzada com a Primera persona, per explorar la scena 3D	13
<a href="#">Dementia::DX11::ConstantBuffer</a>		21
<a href="#">Dementia::DirectionalLight</a>	Una llum que simular ser de tipus ambient, tots els objectes al scenari es veuen afectats per aquesta llum.	23
<a href="#">Dementia::DX11</a>		24
<a href="#">Dementia::Entity</a>	Clase base per a qualsevol altre classe que hagi de interactuar amb un node i el "Scene Manager"	30
<a href="#">Dementia::GameTimer</a>	S'encarrega de controlar tot el que te a veure amb el temps.	33
<a href="#">Dementia::Geometry</a>	Entenem per geometria el conjunt de vertex i index que calen per formar una figura geometrica.	36
<a href="#">Dementia::GeometryFactory</a>		37
<a href="#">Dementia::InputManager</a>	Clase encarregada de gestionar tots els dispositius de entrada entre l'usuari i el sistema operatiu	39
<a href="#">Dementia::Keyboard</a>	Clase encarregada del Taclat que fa de intermediari entre l'usuari i el Sistema Operatiu.	41
<a href="#">Dementia::Light</a>	La clase base de la llum.	43
<a href="#">Dementia::Material</a>	Entenem per <a href="#">Material</a> una estructura de dades on indiquem quin tipus de llum reflexa (color i intensitat) els objectes que tenen aquest material.	45
<a href="#">Dementia::MathHelper</a>		47
<a href="#">Dementia::Mesh</a>	Entenem per mesh la composicio de una geometria amb una textura i un shader.	54
<a href="#">Dementia::Mouse</a>	Clase encarregada del ratoli que fa de intermediari entre l'usuari i el Sistema Operatiu	58
<a href="#">Dementia::Node</a>	Un node es un objecte invisible que se encarrega de sapiguer on, amb quina escala, i amb quina rotacio s'ha de renderitzar qualsevol entitat	62
<a href="#">Dementia::PointLight</a>	Una llum de tipus punt, que irradia a tots els objectes que estiguin dins del seu radi d'accio per igual.	70
<a href="#">Dementia::SceneManager</a>	Clase encarregada de gestionar tot el relacionat amb tot el que surt per pantalla	72
<a href="#">Dementia::Shader</a>	Clase que conte tots elements necessaris per treballar amb els shaders.	79

[Dementia::SpotLight](#)

Una llum de tipus llinterna o focus, il·lumina només als objectes en els que la direcció apunti i estiguin dins del seu radi. . . . . 80

[Dementia::Texture](#)

Classe que conté tot el necessari per dibuixar les textures per pantalla. . . . . 83

[Dementia::Geometry::Vertex](#)

Conté tots els elements necessaris per fer tots els càlculs per pantalla (llums, etc). . . . . 84

[Dementia::Window](#)

Classe encarregada de gestionar tot el relacionat amb la finestra a nivell de Sistema Operatiu i conté tots els elements necessaris per utilitzar el framework . . . . . 86

## Capítol 4

# Índex de Fitxers

### 4.1 Llista dels Fitxers

Aquesta és la llista de tots els fitxers acompanyats amb breus descripcions:

Include/Camera.h	99
Include/Dementia.h	100
Include/DX11.h	102
Include/Entity.h	103
Include/GameTimer.h	104
Include/Geometry.h	104
Include/GeometryFactory.h	105
Include/InputManager.h	107
Include/Keyboard.h	107
Include/Light.h	108
Include/MathHelper.h	109
Include/Mesh.h	110
Include/Mouse.h	111
Include/Node.h	112
Include/SceneManager.h	113
Include/Shader.h	115
Include/stdafx.h	116
Include/targetver.h	117
Include/Texture.h	117
Include/Window.h	118
src/Camera.cpp	120
src/Dementia.cpp	120
src/dllmain.cpp	121
src/DX11.cpp	121
src/Entity.cpp	122
src/GameTimer.cpp	122
src/Geometry.cpp	123
src/GeometryFactory.cpp	123
src/InputManager.cpp	124
src/Keyboard.cpp	125
src/Light.cpp	125
src/MathHelper.cpp	126
src/Mesh.cpp	126
src/Mouse.cpp	127
src/Node.cpp	127
src/SceneManager.cpp	128
src/Shader.cpp	128
src/Texture.cpp	129

src/ <a href="#">Window.cpp</a> . . . . .	<a href="#">130</a>
---	---------------------

## Capítol 5

# Documentació de l'Espai de Noms

### 5.1 Referència de l'Espai de Noms Dementia

El espai de nom principal del framework. Totes les classes basiques o importants es troben aqui dins.

#### Classes

- class [Camera](#)  
*Una camara simple pensada per ser utilitzada com a Primera persona, per explorar la scena 3D.*
- class [DX11](#)
- class [Entity](#)  
*Clase base per a qualsevol altre classe que hagi de interactuar amb un node i el "Scene Manager"*
- class [GameTimer](#)  
*S'encarrega de controlar tot el que te a veure amb el temps.*
- class [Geometry](#)  
*Entenem per geometria el conjunt de vertex i index que calen per formar una figura geometrica.*
- class [GeometryFactory](#)
- class [InputManager](#)  
*Clase encarregada de gestionar tots els dispositius de entrada entre l'usuari i el sistema operatiu*
- class [Keyboard](#)  
*Classe encarregada del Taclat que fa de intermediari entre l'usuari i el Sistema Operatiu.*
- class [Light](#)  
*La classe base de la llum.*
- struct [DirectionalLight](#)  
*Una llum que simular ser de tipus ambient, tots els objectes al scenari es veuen afectats per aquesta llum.*
- struct [PointLight](#)  
*Una llum de tipus punt, que irradia a tots els objectes que estiguin dins del seu radi d'accio per igual.*
- struct [SpotLight](#)  
*Una llum de tipus llinterneta o focus, ilumina només als objectes en els que la direcció apunti i estiguin dins del seu radi.*
- struct [Material](#)  
*Entenem per [Material](#) una estructura de dades on indiquem quin tipus de llum reflexa (color i intensitat) els objectes que tenen aquest material.*
- class [MathHelper](#)
- class [Mesh](#)  
*Entenem per mesh la composició de una geometria amb una textura i un shader.*
- class [Mouse](#)

*Classe encarregada del ratolí que fa de intermediari entre l'usuari i el Sistema Operatiu*

- class [Node](#)

*Un node es un objecte invisible que se encarrega de sapiguer on, amb quina escala, i amb quina rotació s'ha de renderitzar qualsevol entitat.*

- class [SceneManager](#)

*Classe encarregada de gestionar tot el relacionat amb tot el que surt per pantalla.*

- class [Shader](#)

*Classe que conte tots elements necessaris per treballar amb els shaders.*

- class [Texture](#)

*Classe que conte tot el necessari per dibuixar les textures per pantalla.*

- class [Window](#)

*Classe encarregada de gestionar tot el relacionat amb la finestra a nivell de Sistema Operatiu i conte tots el elements necessaris per utilitzar el framework.*

## Funcions

- HRESULT [CompileShaderFromFile](#) (const wchar\_t \*szFileName, LPCSTR szEntryPoint, LPCSTR szShaderModel, ID3DBlob \*\*ppBlobOut)
- LRESULT CALLBACK [MainWndProc](#) (HWND hwnd, UINT msg, WPARAM wParam, LPARAM lParam)

### 5.1.1 Descripció Detallada

El espai de nom principal del framework. Totes les classes bàsiques o importants es troben aquí dins. namespace: [Dementia](#)

### 5.1.2 Documentació de les Funcions

#### 5.1.2.1 HRESULT Dementia::CompileShaderFromFile ( const wchar\_t \* szFileName, LPCSTR szEntryPoint, LPCSTR szShaderModel, ID3DBlob \*\* ppBlobOut )

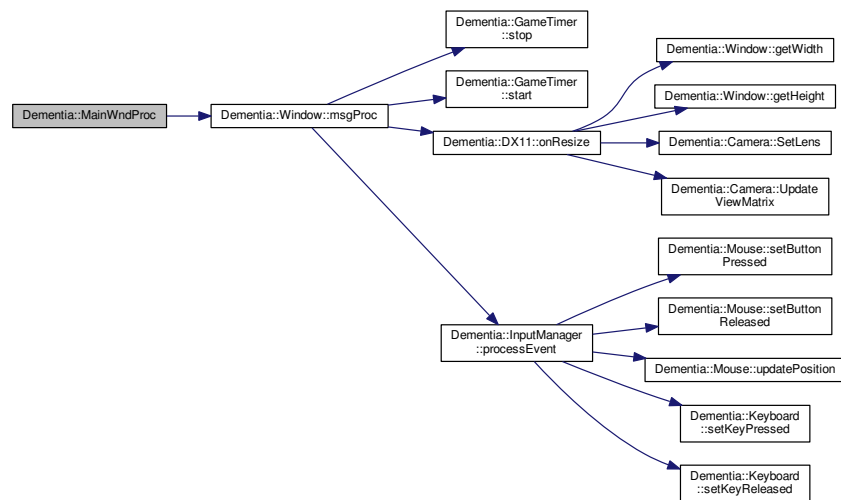
Definició a la línia 449 del fitxer DX11.cpp.

#### 5.1.2.2 LRESULT CALLBACK Dementia::MainWndProc ( HWND hwnd, UINT msg, WPARAM wParam, LPARAM lParam )

Definició a la línia 6 del fitxer Window.cpp.



Gràfic de crides d'aquesta funció:





## Capítol 6

# Documentació de les Classes

### 6.1 Referència de la Classe Dementia::Camera

Una camara simple pensada per ser utilitzada com a Primera persona, per explorar la scena 3D.

```
#include <Camera.h>
```

#### Mètodes públics

- [Camera](#) ()  
*Crea una camara buida a la posicio X:0.0f Y:0.0f Z:-15.0f .*
- [~Camera](#) ()  
*Destructor. Allibera automaticament tots els recursos.*
- XMVECTOR [GetPosition](#) () const  
*Obte la posicio on esta situada la camara.*
- void [SetPosition](#) (XMVECTOR position)  
*Cambia la posicio de la camara a la nova posicio.*
- XMVECTOR [GetRight](#) () const  
*Obte el vector dret de la camara.*
- XMVECTOR [GetUp](#) () const  
*Obte el vector d'alt de la camara.*
- XMVECTOR [GetLook](#) () const  
*Obte la direccio cap a on mira la camara.*
- float [GetNearZ](#) () const  
*Obte la distancia minima a partir de la que es renderitzara en el eix Z .*
- float [GetFarZ](#) () const  
*Obte la distancia maxima a partir de la que es deixa de renderitzar en el eix Z.*
- float [GetAspect](#) () const  
*Obte la relacio d'aspecte en el que esta configurada la camara. Es la divisio del ample entre la alcada de la finestra.*
- float [GetFovY](#) () const  
*Gets fov y coordinate.*
- float [GetFovX](#) () const  
*Gets fov x coordinate.*
- float [GetNearWindowWidth](#) () const
- float [GetNearWindowHeight](#) () const
- float [GetFarWindowWidth](#) () const
- float [GetFarWindowHeight](#) () const
- void [SetLens](#) (float fovY, float aspect, float zn, float zf)

*Configura les lents de la camara.*

- void **LookAt** (FXMVECTOR pos, FXMVECTOR target, FXMVECTOR worldUp)  
*Defineix l'espai de camera amb la posicio on es troba la camara, cap a on ha de mirar i el vector que apunta cap a d'alt.*
- XMATRIX **View** () const  
*Obte la matriu de vista (View Matrix).*
- XMATRIX **Proj** () const  
*Obte la matriu de projeccio (Projection Matrix).*
- XMATRIX **ViewProj** () const  
*Obte la multiplicacio de la matriu de vista (View Matrix) per la matriu de projeccio (Projection Matrix).*
- void **Strafe** (float d)  
*Es mou de forma lateral la distancia indicada en el eix de les X.*
- void **Walk** (float d)  
*Es mou cap a endavant o endarrera en el eix de les Z.*
- void **Pitch** (float angle)  
*Rota la camara en el eix de les X.*
- void **RotateY** (float angle)  
*Rota la camara en el eix de les Y.*
- void **UpdateViewMatrix** ()  
*Actualitza la matriu de vista (View Matrix). Crida a aquest metode, despres de fer tots els canvis a la camara, per a que s'actualitzi.*

### 6.1.1 Descripció Detallada

Una camara simple pensada per ser utilitzada com a Primera persona, per explorar la scena 3D.

- Les cordenades de la camara son relatives al systema de espai del mon (World Space).
- per tant es pot obtenir la Matriu de vista (View Matrix).
- La camara es pot configurar amb les mesures de projeccio per tant es pot obtenir la
- Matriu de projeccio (Projection Matrix).

Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

Definició a la línia 19 del fitxer Camera.h.

### 6.1.2 Documentació del Constructor i el Destructor

#### 6.1.2.1 Dementia::Camera::Camera ( )

Crea una camara buida a la posicio X:0.0f Y:0.0f Z:-15.0f .

Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

Definició a la línia 7 del fitxer Camera.cpp.

Gràfic de crides d'aquesta funció:



### 6.1.2.2 Dementia::Camera::~~Camera ( )

Destructor. Allibera automaticament tots els recursos.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

Definició a la línia 16 del fitxer Camera.cpp.

## 6.1.3 Documentació de les Funcions Membre

### 6.1.3.1 float Dementia::Camera::GetAspect ( ) const

Obte la relacio d'aspecte en el que esta configurada la camara. Es la divisio del ample entre la alcada de la finestra.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

Retorna

La relacio d'aspecte.

Definició a la línia 55 del fitxer Camera.cpp.

### 6.1.3.2 float Dementia::Camera::GetFarWindowHeight ( ) const

Definició a la línia 86 del fitxer Camera.cpp.

### 6.1.3.3 float Dementia::Camera::GetFarWindowWidth ( ) const

Definició a la línia 81 del fitxer Camera.cpp.

### 6.1.3.4 float Dementia::Camera::GetFarZ ( ) const

Obte la distancia maxima a partir de la que es deixa de renderitzar en el eix Z.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

Retorna

La distancia maxima a partir de la que es deixa de renderitzar en el eix Z.

Definició a la línia 50 del fitxer Camera.cpp.

### 6.1.3.5 float Dementia::Camera::GetFovX ( ) const

Gets fov x coordinate.

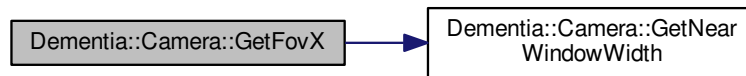
Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

Retorna

The fov x coordinate.

Definició a la línia 65 del fitxer Camera.cpp.

Gràfic de crides d'aquesta funció:



#### 6.1.3.6 float Dementia::Camera::GetFovY ( ) const

Gets fov y coordinate.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

##### Retorna

The fov y coordinate.

Definició a la línia 60 del fitxer Camera.cpp.

#### 6.1.3.7 XMVECTOR Dementia::Camera::GetLook ( ) const

Obte la direccio cap a on mira la camara.

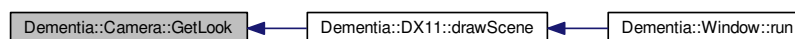
Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

##### Retorna

La direccio cap a on mira la camara.

Definició a la línia 40 del fitxer Camera.cpp.

Gràfic de crides a aquesta funció:



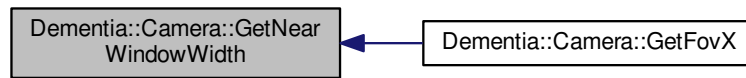
#### 6.1.3.8 float Dementia::Camera::GetNearWindowHeight ( ) const

Definició a la línia 76 del fitxer Camera.cpp.

#### 6.1.3.9 float Dementia::Camera::GetNearWindowWidth ( ) const

Definició a la línia 71 del fitxer Camera.cpp.

Gràfic de crides a aquesta funció:



#### 6.1.3.10 float Dementia::Camera::GetNearZ ( ) const

Obte la distancia minima a partir de la que es renderitzara en el eix Z .

Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

##### Retorna

La distancia minima a partir de la que es renderitzara en el eix Z.

Definició a la línia 45 del fitxer Camera.cpp.

#### 6.1.3.11 XMVECTOR Dementia::Camera::GetPosition ( ) const

Obte la posicio on esta situada la camara.

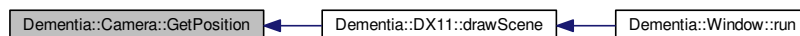
Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

##### Retorna

La posicio on esta situada la camara.

Definició a la línia 20 del fitxer Camera.cpp.

Gràfic de crides a aquesta funció:



#### 6.1.3.12 XMVECTOR Dementia::Camera::GetRight ( ) const

Obte el vector dret de la camara.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

##### Retorna

El vector dret de la camara.

Definició a la línia 30 del fitxer Camera.cpp.

#### 6.1.3.13 XMVECTOR Dementia::Camera::GetUp ( ) const

Obte el vector d'alt de la camara.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

Retorna

El vector d'alt de la camara.

Definició a la línia 35 del fitxer Camera.cpp.

#### 6.1.3.14 void Dementia::Camera::LookAt ( FXMVECTOR pos, FXMVECTOR target, FXMVECTOR worldUp )

Defineix l'espai de camera amb la posicio on es troba la camara, cap a on ha de mirar i el vector que apunta cap a d'alt.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

Paràmetres

<i>pos</i>	La posicio de la camara.
<i>target</i>	La direccio en la que mira la camara.
<i>worldUp</i>	El vector que apunta cap a d'alt.

Definició a la línia 105 del fitxer Camera.cpp.

#### 6.1.3.15 void Dementia::Camera::Pitch ( float angle )

Rota la camara en el eix de les X.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

Paràmetres

<i>angle</i>	El angle que es rota.
--------------	-----------------------

Definició a la línia 142 del fitxer Camera.cpp.

#### 6.1.3.16 XMMATRIX Dementia::Camera::Proj ( ) const

Obte la matriu de projeccio (Projection Matrix).

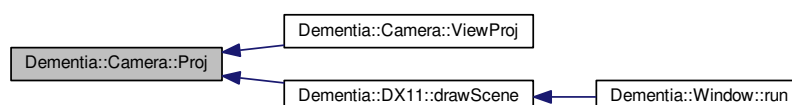
Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

Retorna

Retorna la matriu de projeccio.

Definició a la línia 120 del fitxer Camera.cpp.

Gràfic de crides a aquesta funció:





6.1.3.17 void Dementia::Camera::RotateY ( float *angle* )

Rota la camara en el eix de les Y.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

## Paràmetres

<i>angle</i>	El angle que es rota.
--------------	-----------------------

Definició a la línia 151 del fitxer Camera.cpp.

6.1.3.18 void Dementia::Camera::SetLens ( float *fovY*, float *aspect*, float *zn*, float *zf* )

Configura les lents de la camara.

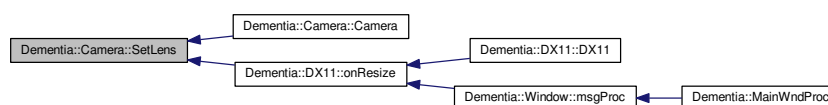
Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

## Paràmetres

<i>fovY</i>	El angle de la obertura de visio.
<i>aspect</i>	La relacio d'aspecte.
<i>zn</i>	La distancia minima a partir de la que es comencara a renderitzar.
<i>zf</i>	La distancia maxima fins a on es renderitzara.

Definició a la línia 91 del fitxer Camera.cpp.

Gràfic de crides a aquesta funció:

6.1.3.19 void Dementia::Camera::SetPosition ( XMVECTOR *position* )

Cambia la posicio de la camara a la nova posicio.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

## Paràmetres

<i>position</i>	La nova posicio on es situara la camara.
-----------------	--

Definició a la línia 25 del fitxer Camera.cpp.

6.1.3.20 void Dementia::Camera::Strafe ( float *d* )

Es mou de forma lateral la distancia indicada en el eix de les X.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

## Paràmetres

<i>d</i>	La distancia a moures.
----------	------------------------

Definició a la línia 130 del fitxer Camera.cpp.

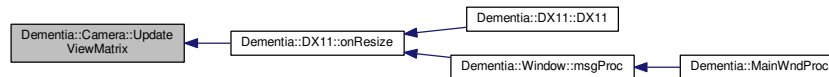
### 6.1.3.21 void Dementia::Camera::UpdateViewMatrix ( )

Actualitza la matriu de vista (View Matrix). Crida a aquest metode, despres de fer tots els canvis a la camara, per a que s'actualitzi.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

Definició a la línia 162 del fitxer Camera.cpp.

Gràfic de crides a aquesta funció:



### 6.1.3.22 XMATRIX Dementia::Camera::View ( ) const

Obte la matriu de vista (View Matrix).

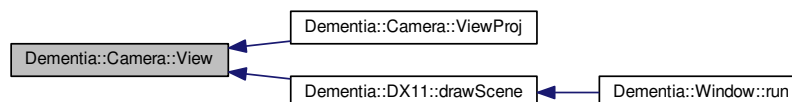
Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

Retorna

Retorna la matriu de vista.

Definició a la línia 115 del fitxer Camera.cpp.

Gràfic de crides a aquesta funció:



### 6.1.3.23 XMATRIX Dementia::Camera::ViewProj ( ) const

Obte la multiplicacio de la matriu de vista (View Matrix) per la matriu de projeccio (Projection Matrix).

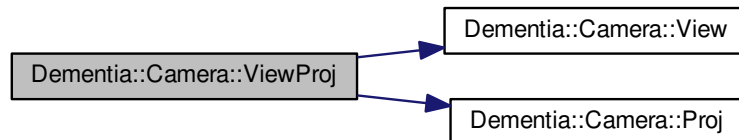
Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

Retorna

La multiplicació de la matriu de vista per la matriu de projecció.

Definició a la línia 125 del fitxer Camera.cpp.

Gràfic de crides d'aquesta funció:



#### 6.1.3.24 void Dementia::Camera::Walk ( float d )

Es mou cap a endavant o endarrera en el eix de les Z.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

Paràmetres

<i>d</i>	La distancia a moures.
----------	------------------------

Definició a la línia 136 del fitxer Camera.cpp.

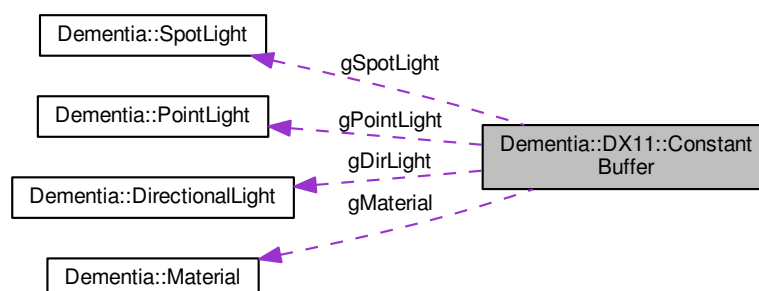
La documentació d'aquesta classe es va generar a partir dels següents fitxers:

- Include/[Camera.h](#)
- src/[Camera.cpp](#)

## 6.2 Referència de l'Estructura Dementia::DX11::ConstantBuffer

```
#include <DX11.h>
```

Diagrama de col·laboració per a Dementia::DX11::ConstantBuffer:



## Atributs Públics

- XMMATRIX [mWorld](#)
- XMMATRIX [mView](#)
- XMMATRIX [mProjection](#)
- [DirectionalLight](#) [gDirLight](#)
- [PointLight](#) [gPointLight](#)
- [SpotLight](#) [gSpotLight](#)
- [Material](#) [gMaterial](#)
- XMFLOAT3 [gEyePosW](#)
- float [pad](#)

### 6.2.1 Descripció Detallada

Definició a la línia 41 del fitxer DX11.h.

### 6.2.2 Documentació de les Dades Membre

#### 6.2.2.1 [DirectionalLight](#) Dementia::DX11::ConstantBuffer::gDirLight

Definició a la línia 46 del fitxer DX11.h.

#### 6.2.2.2 [XMFLOAT3](#) Dementia::DX11::ConstantBuffer::gEyePosW

Definició a la línia 50 del fitxer DX11.h.

#### 6.2.2.3 [Material](#) Dementia::DX11::ConstantBuffer::gMaterial

Definició a la línia 49 del fitxer DX11.h.

#### 6.2.2.4 [PointLight](#) Dementia::DX11::ConstantBuffer::gPointLight

Definició a la línia 47 del fitxer DX11.h.

#### 6.2.2.5 [SpotLight](#) Dementia::DX11::ConstantBuffer::gSpotLight

Definició a la línia 48 del fitxer DX11.h.

#### 6.2.2.6 [XMMATRIX](#) Dementia::DX11::ConstantBuffer::mProjection

Definició a la línia 45 del fitxer DX11.h.

#### 6.2.2.7 [XMMATRIX](#) Dementia::DX11::ConstantBuffer::mView

Definició a la línia 44 del fitxer DX11.h.

#### 6.2.2.8 [XMMATRIX](#) Dementia::DX11::ConstantBuffer::mWorld

Definició a la línia 43 del fitxer DX11.h.

## 6.2.2.9 float Dementia::DX11::ConstantBuffer::pad

Definició a la línia 51 del fitxer DX11.h.

La documentació d'aquesta estructura es va generar a partir del següent fitxer:

- Include/DX11.h

## 6.3 Referència de l'Estructura Dementia::DirectionalLight

Una llum que simular ser de tipus ambient, tots els objectes al scenari es veuen afectats per aquesta llum.

```
#include <Light.h>
```

### Mètodes públics

- [DirectionalLight](#) ()  
*Constructor per defecte.*
- [DirectionalLight](#) (XMFLOAT4 ambient, XMFLOAT4 diffuse, XMFLOAT4 specular, XMFLOAT3 direction, float pad)  
*Constructor de la llum.*

### Atributs Públics

- XMFLOAT4 [Ambient](#)  
*El color de la llum ambient i la seva intensitat (R,G,B, Intensitat).*
- XMFLOAT4 [Diffuse](#)  
*El color de la llum que reflexan els objectes i la seva intensitat (R,G,B, Intensitat).*
- XMFLOAT4 [Specular](#)  
*El color de la llum del brillantor dels objectes i la seva intensitat (R,G,B, Intensitat).*
- XMFLOAT3 [Direction](#)  
*La direcció des de on ve la llum.*
- float [Pad](#)  
*Un valor sense cap funcionalitat, que només serveix per que la estructura estigui alienada en 16bytes.*

### 6.3.1 Descripció Detallada

Una llum que simular ser de tipus ambient, tots els objectes al scenari es veuen afectats per aquesta llum.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

Definició a la línia 57 del fitxer Light.h.

### 6.3.2 Documentació del Constructor i el Destructor

#### 6.3.2.1 Dementia::DirectionalLight::DirectionalLight ( ) [inline]

Constructor per defecte.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

Definició a la línia 65 del fitxer Light.h.

**6.3.2.2 Dementia::DirectionalLight::DirectionalLight** ( *XMFLOAT4 ambient*, *XMFLOAT4 diffuse*, *XMFLOAT4 specular*, *XMFLOAT3 direction*, *float pad* ) [inline]

Constructor de la llum.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

Paràmetres

<i>ambient</i>	El color de la llum ambient i la seva intensitat (R,G,B, Intensitat).
<i>diffuse</i>	El color de la llum que reflexan els objectes i la seva intensitat (R,G,B, Intensitat).
<i>specular</i>	El color de la llum del brillantor dels objectes i la seva intensitat (R,G,B, Intensitat).
<i>direction</i>	La direcció des de on ve la llum.
<i>pad</i>	Un valor sense cap funcionalitat, que només serveix per que la estructura estigui alienada en 16bytes.

Definició a la línia 82 del fitxer Light.h.

### 6.3.3 Documentació de les Dades Membre

#### 6.3.3.1 XMFLOAT4 Dementia::DirectionalLight::Ambient

El color de la llum ambient i la seva intensitat (R,G,B, Intensitat).

Definició a la línia 92 del fitxer Light.h.

#### 6.3.3.2 XMFLOAT4 Dementia::DirectionalLight::Diffuse

El color de la llum que reflexan els objectes i la seva intensitat (R,G,B, Intensitat).

Definició a la línia 95 del fitxer Light.h.

#### 6.3.3.3 XMFLOAT3 Dementia::DirectionalLight::Direction

La direcció des de on ve la llum.

Definició a la línia 101 del fitxer Light.h.

#### 6.3.3.4 float Dementia::DirectionalLight::Pad

Un valor sense cap funcionalitat, que només serveix per que la estructura estigui alienada en 16bytes.

Definició a la línia 104 del fitxer Light.h.

#### 6.3.3.5 XMFLOAT4 Dementia::DirectionalLight::Specular

El color de la llum del brillantor dels objectes i la seva intensitat (R,G,B, Intensitat).

Definició a la línia 98 del fitxer Light.h.

La documentació d'aquesta estructura es va generar a partir del següent fitxer:

- Include/[Light.h](#)

## 6.4 Referència de la Classe Dementia::DX11

```
#include <DX11.h>
```

## Classes

- struct [ConstantBuffer](#)

## Mètodes públics

- [DX11](#) ([Dementia::Window](#) \*window)
- virtual [~DX11](#) (void)
- void [setBackgroundColor](#) (XMVECTOR backgroundcolor)
- XMVECTOR [getBackgroundColor](#) ()
- [Dementia::Camera](#) \* [getCamera](#) ()
- void [close](#) ()
- void [updateScene](#) (float delta)
- void [drawScene](#) ()
- void [onResize](#) ()
- ID3D11Device \* [getDevice](#) ()
- HRESULT [DX11::msgProc](#) (UINT msg, WPARAM wParam, LPARAM lParam)

### 6.4.1 Descripció Detallada

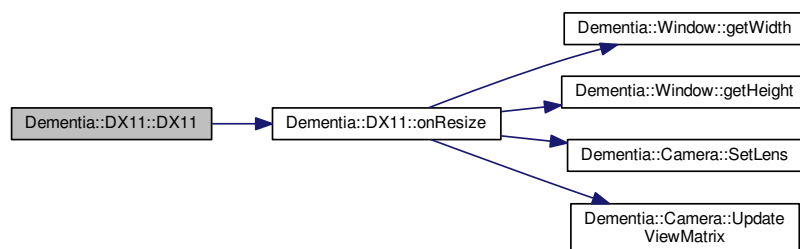
Definició a la línia 15 del fitxer DX11.h.

### 6.4.2 Documentació del Constructor i el Destructor

#### 6.4.2.1 Dementia::DX11::DX11 ( Dementia::Window \* window )

Definició a la línia 6 del fitxer DX11.cpp.

Gràfic de crides d'aquesta funció:



#### 6.4.2.2 Dementia::DX11::~~DX11 ( void ) [virtual]

Definició a la línia 26 del fitxer DX11.cpp.

Gràfic de crides d'aquesta funció:



### 6.4.3 Documentació de les Funcions Membre

#### 6.4.3.1 void Dementia::DX11::close ( )

Definició a la línia 31 del fitxer DX11.cpp.

Gràfic de crides a aquesta funció:

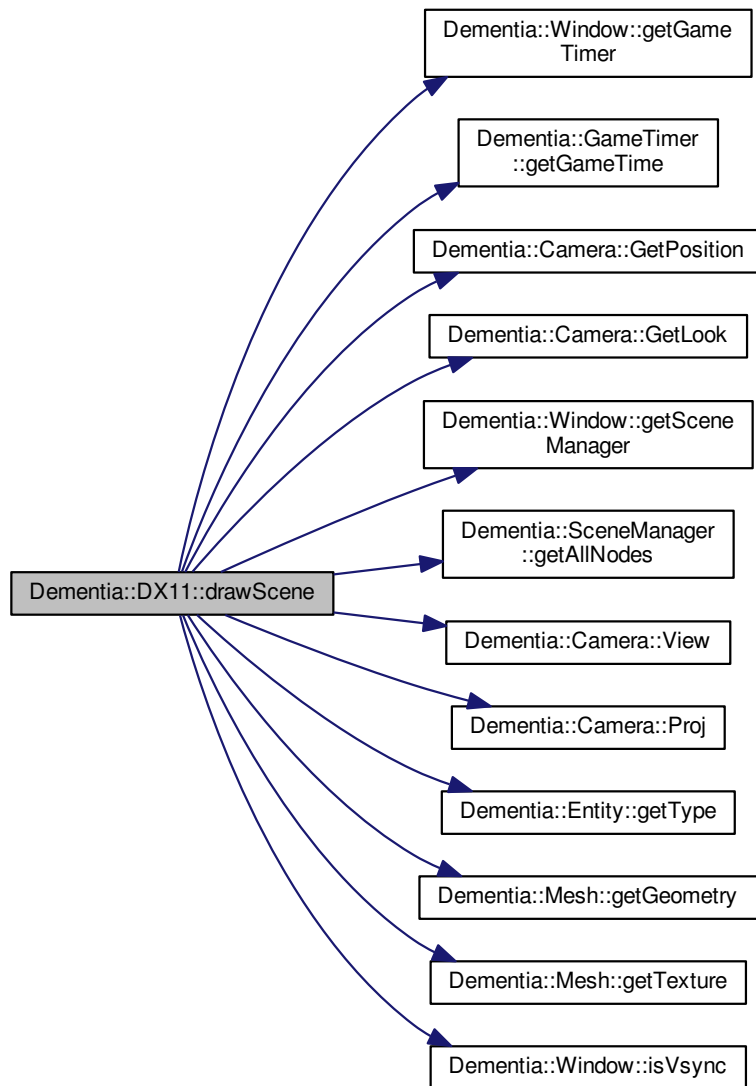


#### 6.4.3.2 void Dementia::DX11::drawScene ( )

Definició a la línia 363 del fitxer DX11.cpp.



Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



6.4.3.3 LRESULT Dementia::DX11::DX11::msgProc ( UINT msg, WPARAM wParam, LPARAM lParam )

#### 6.4.3.4 XMVECTOR Dementia::DX11::getBackgroundColor ( ) [inline]

Definició a la línia 23 del fitxer DX11.h.

Gràfic de crides a aquesta funció:



#### 6.4.3.5 Dementia::Camera\* Dementia::DX11::getCamera ( ) [inline]

Definició a la línia 24 del fitxer DX11.h.

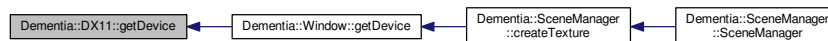
Gràfic de crides a aquesta funció:



#### 6.4.3.6 ID3D11Device \* Dementia::DX11::getDevice ( )

Definició a la línia 48 del fitxer DX11.cpp.

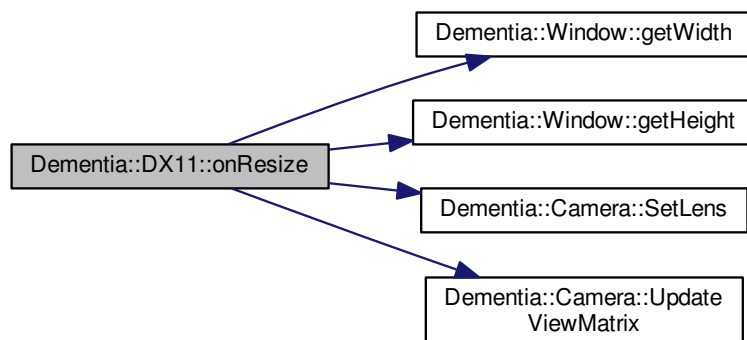
Gràfic de crides a aquesta funció:



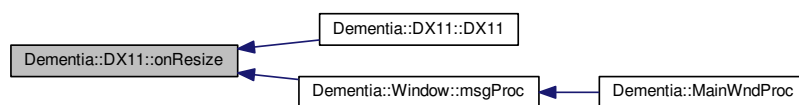
#### 6.4.3.7 void Dementia::DX11::onResize ( )

Definició a la línia 266 del fitxer DX11.cpp.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



#### 6.4.3.8 void Dementia::DX11::setBackgroundColor ( XMVECTOR *backgroundcolor* ) [inline]

Definició a la línia 22 del fitxer DX11.h.

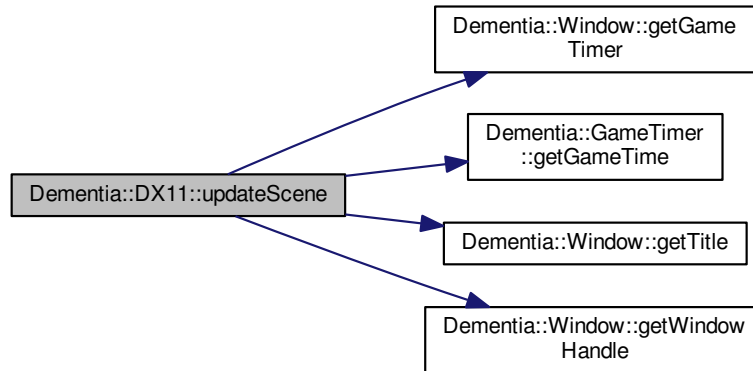
Gràfic de crides a aquesta funció:



#### 6.4.3.9 void Dementia::DX11::updateScene ( float *delta* )

Definició a la línia 338 del fitxer DX11.cpp.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



La documentació d'aquesta classe es va generar a partir dels següents fitxers:

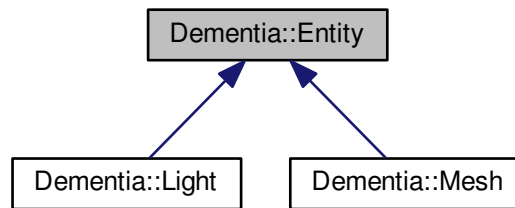
- [Include/DX11.h](#)
- [src/DX11.cpp](#)

## 6.5 Referència de la Classe Dementia::Entity

Clase base per a qualsevol altre classe que hagi de interactuar amb un node i el "Scene Manager"

```
#include <Entity.h>
```

Diagrama d'Herència per a Dementia::Entity:



### Tipus Públics

- enum `TYPE` { `Light`, `Mesh` }

*Els tipus de classe derivades que hi han. Això serveix per evitar haver de fer un `dynamic_cast`, que es molt costos a nivell de CPU.*

### Mètodes públics

- `Entity` (`TYPE` type)  
*Constructor per defecte que obliga a tots els objectes derivats dir de quin tipus son.*
- `~Entity` (void)  
*Destructor per defecte.*
- `UINT getId` ()  
*Obte l'identificador del objecte.*
- `TYPE getType` ()  
*Obte el tipus del qual es l'objecte.*

#### 6.5.1 Descripció Detallada

Clase base per a qualsevol altre classe que hagi de interactuar amb un node i el "Scene Manager"

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

Definició a la línia 13 del fitxer Entity.h.

#### 6.5.2 Documentació de les Enumeracions Membre

##### 6.5.2.1 enum Dementia::Entity::TYPE

Els tipus de classe derivades que hi han. Això serveix per evitar haver de fer un `dynamic_cast`, que es molt costos a nivell de CPU.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

Valors de les Enumeracions

**Light**

**Mesh**

Definició a la línia 25 del fitxer Entity.h.

### 6.5.3 Documentació del Constructor i el Destructor

#### 6.5.3.1 Dementia::Entity::Entity ( TYPE type )

Constructor per defecte que obliga a tots els objectes derivats dir de quin tipus son.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

Paràmetres

<i>type</i>	El tipus d'objecte derivat.
-------------	-----------------------------

Definició a la línia 4 del fitxer Entity.cpp.

#### 6.5.3.2 Dementia::Entity::~~Entity ( void )

Destructor per defecte.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

Definició a la línia 10 del fitxer Entity.cpp.

### 6.5.4 Documentació de les Funcions Membre

#### 6.5.4.1 UINT Dementia::Entity::getId ( ) [inline]

Obte l'identificador del objecte.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

Retorna

El identificador.

Definició a la línia 53 del fitxer Entity.h.

#### 6.5.4.2 TYPE Dementia::Entity::getType ( ) [inline]

Obte el tipus del qual es l'objecte.

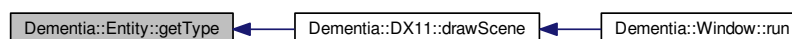
Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

Retorna

El tipus del objecte.

Definició a la línia 63 del fitxer Entity.h.

Gràfic de crides a aquesta funció:



La documentació d'aquesta classe es va generar a partir dels següents fitxers:

- Include/[Entity.h](#)
- src/[Entity.cpp](#)

## 6.6 Referència de la Classe Dementia::GameTimer

S'encarrega de controlar tot el que te a veure amb el temps.

```
#include <GameTimer.h>
```

### Mètodes públics

- `GameTimer ()`
- `float getTime () const`  
*Obte el total de temps que ha pasat des de que s'ha inicialitzat la finestra. El temps que ha pasat en pausa, no compta.*
- `float getDeltaTime () const`  
*Obte el temps que passa entre que es renderitza la imatge actual i la anterior.*
- `void reset ()`  
*Reinicia el temps de joc a 0.*
- `void start ()`  
*Comença a comptar el temps de joc. Si estava en stop continua des de l'últim punt.*
- `void stop ()`  
*Para de comptar el temps de joc. No posa el comptador a 0. Si es crida a al metode `start()` es retorna des de aquet punt.*
- `void tick ()`  
*Compta les imatges per segon. S'ha de cridar cada imatge*

### 6.6.1 Descripció Detallada

S'encarrega de controlar tot el que te a veure amb el temps.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 14/05/2013.

Definició a la línia 19 del fitxer GameTimer.h.

### 6.6.2 Documentació del Constructor i el Destructor

#### 6.6.2.1 Dementia::GameTimer::GameTimer ( )

Definició a la línia 5 del fitxer GameTimer.cpp.

### 6.6.3 Documentació de les Funcions Membre

#### 6.6.3.1 float Dementia::GameTimer::getDeltaTime ( ) const

Obte el temps que passa entre que es renderitza la imatge actual i la anterior.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 12/05/2013.

Retorna

El temps delta.

Definició a la línia 44 del fitxer GameTimer.cpp.

Gràfic de crides a aquesta funció:



#### 6.6.3.2 float Dementia::GameTimer::getGameTime ( ) const

Obte el total de temps que ha pasat des de que s'ha inicialitzat la finestra. El temps que ha pasat en pausa, no compta.

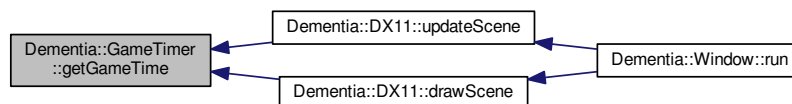
Alberto Lopez Sanchez i Ruben Bagan Benavides, 12/05/2013.

Retorna

El total de temps des de que s'ha inicialitzat la finestra.

Definició a la línia 16 del fitxer GameTimer.cpp.

Gràfic de crides a aquesta funció:



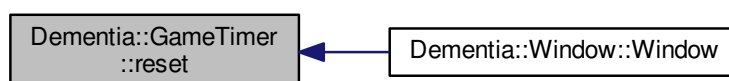
#### 6.6.3.3 void Dementia::GameTimer::reset ( )

Reinicia el temps de joc a 0.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 12/05/2013.

Definició a la línia 49 del fitxer GameTimer.cpp.

Gràfic de crides a aquesta funció:





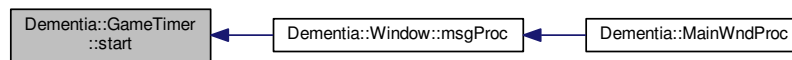
#### 6.6.3.4 void Dementia::GameTimer::start ( )

Comeca a comptar el temps de joc. Si estava en stop continua des de l'últim punt.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 12/05/2013.

Definició a la línia 60 del fitxer GameTimer.cpp.

Gràfic de crides a aquesta funció:



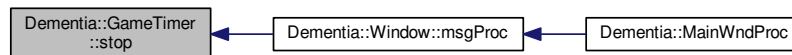
#### 6.6.3.5 void Dementia::GameTimer::stop ( )

Para de comptar el temps de joc. No posa el comptador a 0. Si es crida a al metode [start\(\)](#) es retorna des de aquet punt.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 12/05/2013.

Definició a la línia 82 del fitxer GameTimer.cpp.

Gràfic de crides a aquesta funció:



#### 6.6.3.6 void Dementia::GameTimer::tick ( )

Compta les imatges per segon. S'ha de cridar cada imatge

Alberto Lopez Sanchez i Ruben Bagan Benavides, 12/05/2013.

Definició a la línia 94 del fitxer GameTimer.cpp.

Gràfic de crides a aquesta funció:



La documentació d'aquesta classe es va generar a partir dels següents fitxers:

- Include/[GameTimer.h](#)
- src/[GameTimer.cpp](#)

## 6.7 Referència de la Classe Dementia::Geometry

Entenem per geometria el conjunt de vertex i index que calen per formar una figura geometrica.

```
#include <Geometry.h>
```

### Classes

- struct [Vertex](#)

*Conte tots els elements necessaris per fer tots els calculs per pantalla (llums, etc).*

### Mètodes públics

- [Geometry](#) ()

*Constructor per defecte.*

- [~Geometry](#) (void)

*Destructor.*

### Atributs Públics

- std::vector< [Vertex](#) > [Vertices](#)

*Els vertex de aquesta geometria.*

- std::vector< UINT > [Indices](#)

*Els indexos de aquesta geometria.*

- ID3D11Buffer \* [vb](#)

*Buffer de DirectX11 dels vertex.*

- ID3D11Buffer \* [ib](#)

*Buffer de DirectX11 dels indexos.*

### 6.7.1 Descripció Detallada

Entenem per geometria el conjunt de vertex i index que calen per formar una figura geometrica.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

Definició a la línia 16 del fitxer Geometry.h.

### 6.7.2 Documentació del Constructor i el Destructor

#### 6.7.2.1 Dementia::Geometry::Geometry ( )

Constructor per defecte.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

Definició a la línia 6 del fitxer Geometry.cpp.

### 6.7.2.2 Dementia::Geometry::~~Geometry ( void )

Destructor.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

Definició a la línia 11 del fitxer Geometry.cpp.

## 6.7.3 Documentació de les Dades Membre

### 6.7.3.1 ID3D11Buffer\* Dementia::Geometry::ib

Buffer de DirectX11 dels indexos.

Definició a la línia 121 del fitxer Geometry.h.

### 6.7.3.2 std::vector<UINT> Dementia::Geometry::Indices

Els indexos de aquesta geometria.

Definició a la línia 113 del fitxer Geometry.h.

### 6.7.3.3 ID3D11Buffer\* Dementia::Geometry::vb

Buffer de DirectX11 dels vertex.

Definició a la línia 117 del fitxer Geometry.h.

### 6.7.3.4 std::vector<Vertex> Dementia::Geometry::Vertices

Els vertex de aquesta geometria.

Definició a la línia 110 del fitxer Geometry.h.

La documentació d'aquesta classe es va generar a partir dels següents fitxers:

- Include/[Geometry.h](#)
- src/[Geometry.cpp](#)

## 6.8 Referència de la Classe Dementia::GeometryFactory

```
#include <GeometryFactory.h>
```

### Mètodes públics

- [GeometryFactory](#) ([Dementia::Window](#) \*window)
- virtual [~GeometryFactory](#) ()
- void [CreateBox](#) (float width, float height, float depth, [Dementia::Geometry](#) &geometry)
- void [CreateSphere](#) (float radius, UINT sliceCount, UINT stackCount, [Dementia::Geometry](#) &geometry)
- void [CreateGeosphere](#) (float radius, UINT numSubdivisions, [Dementia::Geometry](#) &geometry)
- void [CreateCylinder](#) (float bottomRadius, float topRadius, float height, UINT sliceCount, UINT stackCount, [Dementia::Geometry](#) &geometry)
- void [CreateGrid](#) (float width, float depth, UINT m, UINT n, [Dementia::Geometry](#) &geometry)
- void [CreateFullscreenQuad](#) ([Geometry](#) &geometry)

### 6.8.1 Descripció Detallada

Definició a la línia 10 del fitxer GeometryFactory.h.

### 6.8.2 Documentació del Constructor i el Destructor

#### 6.8.2.1 Dementia::GeometryFactory::GeometryFactory ( Dementia::Window \* window )

Definició a la línia 11 del fitxer GeometryFactory.cpp.

#### 6.8.2.2 Dementia::GeometryFactory::~~GeometryFactory ( ) [virtual]

summary> Crea una caixa centrada al origen de les dimensions donades. /summary>

Definició a la línia 17 del fitxer GeometryFactory.cpp.

### 6.8.3 Documentació de les Funcions Membre

#### 6.8.3.1 void Dementia::GeometryFactory::CreateBox ( float width, float height, float depth, Dementia::Geometry & geometry )

summary> Crea una esfera centrada al origen del seu radi. Els slices i stacks controlen el grau de tessellation. /summary>

Definició a la línia 22 del fitxer GeometryFactory.cpp.

#### 6.8.3.2 void Dementia::GeometryFactory::CreateCylinder ( float bottomRadius, float topRadius, float height, UINT sliceCount, UINT stackCount, Dementia::Geometry & geometry )

summary> Crea una graella del ample i profunditats especificats en el planol XZ centrat al origen de les dimensions donades. /summary>

Definició a la línia 374 del fitxer GeometryFactory.cpp.

#### 6.8.3.3 void Dementia::GeometryFactory::CreateFullscreenQuad ( Dementia::Geometry & geometry )

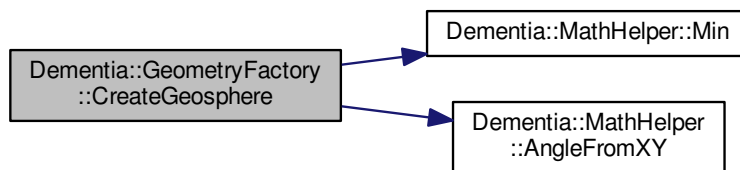
Definició a la línia 611 del fitxer GeometryFactory.cpp.

#### 6.8.3.4 void Dementia::GeometryFactory::CreateGeosphere ( float radius, UINT numSubdivisions, Dementia::Geometry & geometry )

summary> Crea un cilindre paral·lel al eix de les Y, centrat al seu origen. La part inferior i la part superior poden variar el radi, per fer figures diferents a cilindres de veritat, com per exemple un Con. Els slices i stacks controlen el grau de tessellation. /summary>

Definició a la línia 300 del fitxer GeometryFactory.cpp.

Gràfic de crides d'aquesta funció:



**6.8.3.5** void Dementia::GeometryFactory::CreateGrid ( float *width*, float *depth*, UINT *m*, UINT *n*, Dementia::Geometry & *geometry* )

summary> Crea un Quad que ocupa exactament tota la pantalla, ideal per a efectes de post renderitzat.  
/summary>

Definició a la línia 548 del fitxer GeometryFactory.cpp.

**6.8.3.6** void Dementia::GeometryFactory::CreateSphere ( float *radius*, UINT *sliceCount*, UINT *stackCount*, Dementia::Geometry & *geometry* )

summary> Crea una geoesfera centrada al origen del seu radi. El depth controla el grau de tessellation.  
/summary>

Definició a la línia 107 del fitxer GeometryFactory.cpp.

La documentació d'aquesta classe es va generar a partir dels següents fitxers:

- Include/[GeometryFactory.h](#)
- src/[GeometryFactory.cpp](#)

## 6.9 Referència de la Classe Dementia::InputManager

Classe encarregada de gestionar tots els dispositius de entrada entre l'usuari i el sistema operatiu

```
#include <InputManager.h>
```

### Mètodes públics

- [InputManager](#) (void)  
*Constructor per defecte.*
- virtual [~InputManager](#) (void)  
*Destructor.*
- [Mouse](#) \* [getMouse](#) ()  
*Obte la instància del objecte "Mouse".*
- [Keyboard](#) \* [getKeyboard](#) ()  
*Obte la instància del objecte "Keyboard".*
- void [processEvent](#) (UINT msg, WPARAM wParam, LPARAM lParam)  
*Processa tots els events que tenen que pugin ser de entrada de l'usuari. No s'ha de cridar a aquesta funció a no ser que s'apigues el que estas fent.*

### 6.9.1 Descripció Detallada

Clase encarregada de gestionar tots els dispositius de entrada entre l'usuari i el sistema operatiu

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

Definició a la línia 17 del fitxer InputManager.h.

### 6.9.2 Documentació del Constructor i el Destructor

#### 6.9.2.1 Dementia::InputManager::InputManager ( void )

Constructor per defecte.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

Definició a la línia 5 del fitxer InputManager.cpp.

#### 6.9.2.2 Dementia::InputManager::~~InputManager ( void ) [virtual]

Destructor.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

Definició a la línia 11 del fitxer InputManager.cpp.

### 6.9.3 Documentació de les Funcions Membre

#### 6.9.3.1 Keyboard\* Dementia::InputManager::getKeyboard ( ) [inline]

Obte la instància del objecte "Keyboard".

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

**Retorna**

La instància del objecte "Keyboard".

Definició a la línia 57 del fitxer InputManager.h.

#### 6.9.3.2 Mouse\* Dementia::InputManager::getMouse ( ) [inline]

Obte la instància del objecte "Mouse".

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

**Retorna**

La instància del objecte "Mouse".

Definició a la línia 47 del fitxer InputManager.h.

#### 6.9.3.3 void Dementia::InputManager::processEvent ( UINT msg, WPARAM wParam, LPARAM lParam )

Processa tots els events que tenen que pugin ser de entrada de l'usuari. No s'ha de cridar a aquesta funció a no ser que s'apigués el que estas fent.

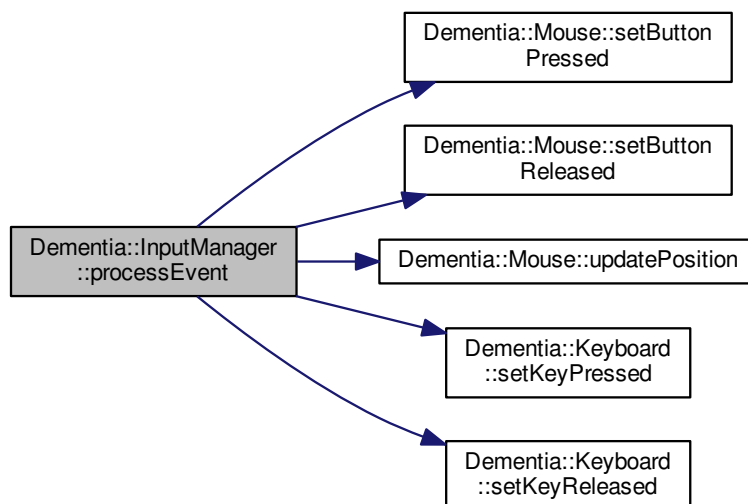
Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

## Paràmetres

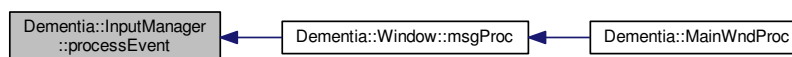
<i>msg</i>	El missatge a processar.
<i>wParam</i>	El camp wParam del missatge.
<i>lParam</i>	El camp lParam del missatge.

Definició a la línia 15 del fitxer InputManager.cpp.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



La documentació d'aquesta classe es va generar a partir dels següents fitxers:

- [Include/InputManager.h](#)
- [src/InputManager.cpp](#)

## 6.10 Referència de la Classe Dementia::Keyboard

Classe encarregada del Taclat que fa de intermediari entre l'usuari i el Sistema Operatiu.

```
#include <Keyboard.h>
```

### Mètodes públics

- [Keyboard](#) (void)

*Constructor per defecte.*

- virtual `~Keyboard` (void)

*Destructor.*

- void `setKeyPressed` (int key)

*Posa com pulsada la tecla pasada per parametre. Aquest metode no s'ha de utilitzar a no ser que sapigues que estas fent.*

- void `setKeyReleased` (int key)

*Posa com alliberada la telca pasada per parametre. Aquest metode no s'ha de utilitzar a no ser que sapigues que estas fent.*

- bool `isKeyPressed` (int key)

*Pregunta si la tecla pasada com a parametre esta pulsada.*

### 6.10.1 Descripció Detallada

Classe encarregada del Taclat que fa de intermediari entre l'usuari i el Sistema Operatiu.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 14/05/2013.

Definició a la línia 14 del fitxer Keyboard.h.

### 6.10.2 Documentació del Constructor i el Destructor

#### 6.10.2.1 Dementia::Keyboard::Keyboard ( void )

Constructor per defecte.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 14/05/2013.

Definició a la línia 6 del fitxer Keyboard.cpp.

#### 6.10.2.2 Dementia::Keyboard::~~Keyboard ( void ) [virtual]

Destructor.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 14/05/2013.

Definició a la línia 12 del fitxer Keyboard.cpp.

### 6.10.3 Documentació de les Funcions Membre

#### 6.10.3.1 bool Dementia::Keyboard::isKeyPressed ( int key )

Pregunta si la tecla pasada com a parametre esta pulsada.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 14/05/2013.

Paràmetres

<i>key</i>	The key.
------------	----------

Retorna

true si esta pulsada, false si no.

Definició a la línia 25 del fitxer Keyboard.cpp.



## 6.10.3.2 void Dementia::Keyboard::setKeyPressed ( int key )

Posa com pulsada la tecla pasada per parametre. Aquest metode no s'ha de utilitzar a no ser que sapigues que estas fent.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 14/05/2013.

## Paràmetres

<i>key</i>	La tecla que es marcara com pulsada.
------------	--------------------------------------

Definició a la línia 16 del fitxer Keyboard.cpp.

Gràfic de crides a aquesta funció:



## 6.10.3.3 void Dementia::Keyboard::setKeyReleased ( int key )

Posa com alliberada la telca pasada per parametre. Aquest metode no s'ha de utilitzar a no ser que sapigues que estas fent.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 14/05/2013.

## Paràmetres

<i>key</i>	La tecla que es marcara com alliberada.
------------	---

Definició a la línia 20 del fitxer Keyboard.cpp.

Gràfic de crides a aquesta funció:



La documentació d'aquesta classe es va generar a partir dels següents fitxers:

- Include/[Keyboard.h](#)
- src/[Keyboard.cpp](#)

## 6.11 Referència de la Classe Dementia::Light

La classe base de la llum.

```
#include <Light.h>
```

Diagrama d'Herència per a Dementia::Light:

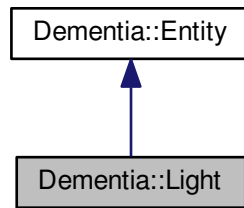
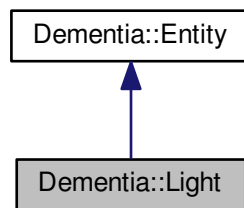


Diagrama de col·laboració per a Dementia::Light:



## Tipus Públics

- enum `LIGHT_TYPE` { `DIRECTIONAL_LIGHT`, `SPOT_LIGHT`, `POINT_LIGHT` }

*Els tipus de classes llum derivades que hi han. Això serveix per evitar haver de fer un `dynamic_cast`, que es molt costos a nivell de CPU.*

## Mètodes públics

- `Light` (void)  
*Constructor per defecte.*
- `~Light` (void)  
*Destructor.*

### 6.11.1 Descripció Detallada

La classe base de la llum.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

Definició a la línia 14 del fitxer Light.h.

### 6.11.2 Documentació de les Enumeracions Membre

#### 6.11.2.1 enum Dementia::Light::LIGHT\_TYPE

Els tipus de classes llum derivades que hi han. Això serveix per evitar haver de fer un `dynamic_cast`, que es molt costós a nivell de CPU.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

Valors de les Enumeracions

***DIRECTIONAL\_LIGHT***

***SPOT\_LIGHT***

***POINT\_LIGHT***

Definició a la línia 26 del fitxer `Light.h`.

### 6.11.3 Documentació del Constructor i el Destructor

#### 6.11.3.1 Dementia::Light::Light ( void )

Constructor per defecte.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

Definició a la línia 4 del fitxer `Light.cpp`.

#### 6.11.3.2 Dementia::Light::~~Light ( void )

Destructor.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

Definició a la línia 10 del fitxer `Light.cpp`.

La documentació d'aquesta classe es va generar a partir dels següents fitxers:

- [Include/Light.h](#)
- [src/Light.cpp](#)

## 6.12 Referència de l'Estructura Dementia::Material

Entenem per [Material](#) una estructura de dades on indiquem quin tipus de llum reflexa (color i intensitat) els objectes que tenen aquest material.

```
#include <Light.h>
```

### Mètodes públics

- [Material](#) ()  
*Constructor per defecte.*
- [Material](#) (XMFLOAT4 ambient, XMFLOAT4 diffuse, XMFLOAT4 specular, XMFLOAT4 reflect)  
*Constructor del material.*

## Atributs Públics

- XMFLOAT4 [Ambient](#)

*El color de la llum ambient que reflexa l'objecte i la seva intensitat.*

- XMFLOAT4 [Diffuse](#)

*El color de la llum que difusa l'objecte i la seva intensitat.*

- XMFLOAT4 [Specular](#)

*El color de la llum de la brillantor de l'objecte i la seva intensitat.*

- XMFLOAT4 [Reflect](#)

*El color de la llum que reflexa globalment l'objecte i la seva intensitat.*

### 6.12.1 Descripció Detallada

Entenem per [Material](#) una estructura de dades on indiquem quin tipus de llum reflexa (color i intensitat) els objectes que tenen aquest material.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

Definició a la línia 260 del fitxer Light.h.

### 6.12.2 Documentació del Constructor i el Destructor

#### 6.12.2.1 Dementia::Material::Material ( ) [inline]

Constructor per defecte.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

Definició a la línia 268 del fitxer Light.h.

#### 6.12.2.2 Dementia::Material::Material ( XMFLOAT4 ambient, XMFLOAT4 diffuse, XMFLOAT4 specular, XMFLOAT4 reflect ) [inline]

Constructor del material.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

Paràmetres

<i>ambient</i>	El color de la llum ambient que reflexa l'objecte i la seva intensitat.
<i>diffuse</i>	El color de la llum que difusa l'objecte i la seva intensitat.
<i>specular</i>	El color de la llum de la brillantor de l'objecte i la seva intensitat.
<i>reflect</i>	El color de la llum que reflexa globalment l'objecte i la seva intensitat.

Definició a la línia 284 del fitxer Light.h.

### 6.12.3 Documentació de les Dades Membre

#### 6.12.3.1 XMFLOAT4 Dementia::Material::Ambient

El color de la llum ambient que reflexa l'objecte i la seva intensitat.

Definició a la línia 294 del fitxer Light.h.

## 6.12.3.2 XMFLOAT4 Dementia::Material::Diffuse

El color de la llum que diffusa l'objecte i la seva intensitat.

Definició a la línia 297 del fitxer Light.h.

## 6.12.3.3 XMFLOAT4 Dementia::Material::Reflect

El color de la llum que reflexa globalment l'objecte i la seva intensitat.

Definició a la línia 303 del fitxer Light.h.

## 6.12.3.4 XMFLOAT4 Dementia::Material::Specular

El color de la llum de la brillantor de l'objecte i la seva intensitat.

Definició a la línia 300 del fitxer Light.h.

La documentació d'aquesta estructura es va generar a partir del següent fitxer:

- Include/[Light.h](#)

## 6.13 Referència de la Classe Dementia::MathHelper

```
#include <MathHelper.h>
```

## Mètodes Públics Estàtics

- static float [RandF](#) ()  
*Obte un numero aleatori float entre 0 i 1.*
- static float [RandF](#) (float a, float b)  
*Obte un numero aleatori float entre a i b.*
- template<typename T >  
static T [Min](#) (const T &a, const T &b)  
*Determina el minim entre els 2 parametres.*
- template<typename T >  
static T [Max](#) (const T &a, const T &b)  
*Determina el maxim entre els 2 parametres.*
- template<typename T >  
static T [Lerp](#) (const T &a, const T &b, float t)
- template<typename T >  
static T [Clamp](#) (const T &x, const T &low, const T &high)  
*Comproba que el valor x no es pasa ni per sobre ni per sota.*
- static float [AngleFromXY](#) (float x, float y)  
*Obte el angle polar del punt (x,y) en [0, 2\*PI].*
- static XMATRIX [InverseTranspose](#) (CXMATRIX M)  
*Obte la matriu inversa i transposada a la pasada per parametre.*
- static XMVECTOR [RandUnitVec3](#) ()
- static XMVECTOR [RandHemisphereUnitVec3](#) (XMVECTOR n)

## Atributs Públics Estàtics

- static const float **Infinity** = FLT\_MAX  
*Esta considerat com el maxim valor que pot agafar un float, canvia segons el sistema.*
- static const float **Pi** = 3.1415926535f  
*Valor predeterminat de pi.*

### 6.13.1 Descripció Detallada

Definició a la línia 7 del fitxer MathHelper.h.

### 6.13.2 Documentació de les Funcions Membre

#### 6.13.2.1 float Dementia::MathHelper::AngleFromXY ( float x, float y ) [static]

Obte el angle polar del punt (x,y) en [0, 2\*PI].

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

##### Paràmetres

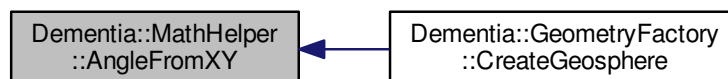
<i>x</i>	La coordenada X.
<i>y</i>	La coordenada Y.

##### Retorna

El angle calculat dels 2 punts.

Definició a la línia 8 del fitxer MathHelper.cpp.

Gràfic de crides a aquesta funció:



#### 6.13.2.2 template<typename T> static T Dementia::MathHelper::Clamp ( const T & x, const T & low, const T & high ) [inline], [static]

Comproba que el valor x no es pasa ni per sobre ni per sota.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

##### Template Parameters

<i>typename T</i>	Type of the typename t.
-------------------	-------------------------

## Paràmetres

<i>x</i>	El valor a comprobar.
<i>low</i>	El valor mínim.
<i>high</i>	El valor màxim.

## Retorna

Si el valor és més petit que el mínim retorna el mínim, si és més gran que el màxim retorna el màxim, si no és cap dels dos retorna el valor.

Definició a la línia 103 del fitxer MathHelper.h.

6.13.2.3 `static XMMATRIX Dementia::MathHelper::InverseTranspose ( CXMMATRIX M ) [inline],[static]`

Obté la matriu inversa i transposada a la passada per paràmetre.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

## Paràmetres

<i>M</i>	La matriu a processar.
----------	------------------------

## Retorna

La matriu invertida i transposada.

Definició a la línia 131 del fitxer MathHelper.h.

6.13.2.4 `template<typename T> static T Dementia::MathHelper::Lerp ( const T & a, const T & b, float t ) [inline],[static]`

Definició a la línia 82 del fitxer MathHelper.h.

6.13.2.5 `template<typename T> static T Dementia::MathHelper::Max ( const T & a, const T & b ) [inline],[static]`

Determina el màxim entre els 2 paràmetres.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

## Template Parameters

<i>typename T</i>	Type of the typename t.
-------------------	-------------------------

## Paràmetres

<i>a</i>	El límit de la comprovació.
<i>b</i>	El valor a comprobar.

## Retorna

Si el paràmetre B és més petit que el paràmetre a, retorna el paràmetre B, si no retorna el paràmetre A.

Definició a la línia 75 del fitxer MathHelper.h.

**6.13.2.6** `template<typename T > static T Dementia::MathHelper::Min ( const T & a, const T & b ) [inline],  
[static]`

Determina el mínim entre els 2 parametres.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.



## Template Parameters

<i>typename T</i>	Type of the typename t.
-------------------	-------------------------

## Paràmetres

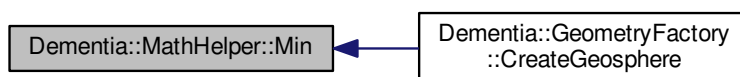
<i>a</i>	El limit de la comprobacio.
<i>b</i>	El valor a comprobar.

## Retorna

Si el parametre b es mes gran que el parametre a, retorna el parametre B, si no retorna el parametre A.

Definició a la línia 55 del fitxer MathHelper.h.

Gràfic de crides a aquesta funció:



## 6.13.2.7 static float Dementia::MathHelper::RandF ( ) [inline],[static]

Obte un numero aleatori float entre 0 i 1.

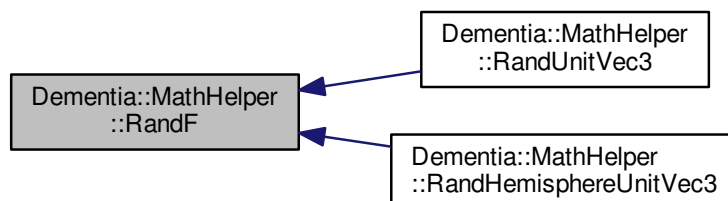
Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

## Retorna

El numero aleatori entre 0 i 1.

Definició a la línia 19 del fitxer MathHelper.h.

Gràfic de crides a aquesta funció:



## 6.13.2.8 static float Dementia::MathHelper::RandF ( float a, float b ) [inline],[static]

Obte un numero aleatori float entre a i b.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

## Paràmetres

<i>a</i>	El numero minim que retornara.
<i>b</i>	El numero maxim que retornara.

## Retorna

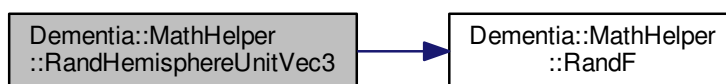
Un numero aleatori float entre a i b.

Definició a la línia 35 del fitxer MathHelper.h.

**6.13.2.9 XMVECTOR Dementia::MathHelper::RandHemisphereUnitVec3 ( XMVECTOR *n* ) [static]**

Definició a la línia 52 del fitxer MathHelper.cpp.

Gràfic de crides d'aquesta funció:

**6.13.2.10 XMVECTOR Dementia::MathHelper::RandUnitVec3 ( ) [static]**

Definició a la línia 30 del fitxer MathHelper.cpp.

Gràfic de crides d'aquesta funció:

**6.13.3 Documentació de les Dades Membre****6.13.3.1 const float Dementia::MathHelper::Infinity = FLT\_MAX [static]**

Esta considerat com el maxim valor que pot agafar un float, cambia segons el sistema.

Definició a la línia 145 del fitxer MathHelper.h.

**6.13.3.2 const float Dementia::MathHelper::Pi = 3.1415926535f [static]**

Valor predeterminat de pi.

Definició a la línia 148 del fitxer MathHelper.h.

La documentació d'aquesta classe es va generar a partir dels següents fitxers:

- Include/[MathHelper.h](#)
- src/[MathHelper.cpp](#)

## 6.14 Referència de la Classe Dementia::Mesh

Entenem per mesh la composició de una geometria amb una textura i un shader.

```
#include <Mesh.h>
```

Diagrama d'Herència per a Dementia::Mesh:

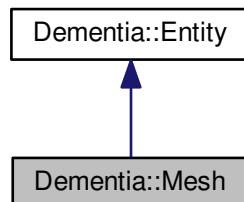
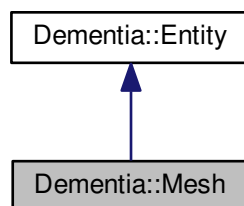


Diagrama de col·laboració per a Dementia::Mesh:



### Mètodes públics

- [Mesh](#) (void)  
*Constructor per defecte.*
- [Mesh](#) ([Dementia::Geometry](#) \*geometry)  
*Constructor amb només una geometria, es el basic.*
- [Mesh](#) ([Dementia::Geometry](#) \*geometry, [Dementia::Texture](#) \*texture, [Dementia::Shader](#) \*shader)  
*Constructor complet.*
- virtual [~Mesh](#) (void)

*Destructor.*

- void `setGeometry (Dementia::Geometry *geometry)`  
*Posa o canvia la geometria que contindra.*
- void `setTexture (Dementia::Texture *texture)`  
*Posa o canvia la textura que utilitzara*
- void `setShader (Dementia::Shader *shader)`  
*Posa o canvia el shader amb el que es dibuixara.*
- `Dementia::Geometry * getGeometry ()`  
*Obte la geometria.*
- `Dementia::Texture * getTexture ()`  
*Obte la textura.*
- `Dementia::Shader * getShader ()`  
*Obte el shader.*

## Additional Inherited Members

### 6.14.1 Descripció Detallada

Entenem per mesh la composicio de una geometria amb una textura i un shader.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

Definició a la línia 17 del fitxer Mesh.h.

### 6.14.2 Documentació del Constructor i el Destructor

#### 6.14.2.1 Dementia::Mesh::Mesh ( void )

Constructor per defecte.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

Definició a la línia 5 del fitxer Mesh.cpp.

#### 6.14.2.2 Dementia::Mesh::Mesh ( Dementia::Geometry \* geometry )

Constructor amb nomes una geometria, es el basic.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

Paràmetres

<i>geometry</i>	[in] La geometria que contindra.
-----------------	----------------------------------

Definició a la línia 13 del fitxer Mesh.cpp.

#### 6.14.2.3 Dementia::Mesh::Mesh ( Dementia::Geometry \* geometry, Dementia::Texture \* texture, Dementia::Shader \* shader )

Constructor complet.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

## Paràmetres

<i>geometry</i>	[in] La geometria que contindra.
<i>texture</i>	[in] La textura que ultilitzara.
<i>shader</i>	[in] El shader amb el que es dibuixara.

Definició a la línia 22 del fitxer Mesh.cpp.

#### 6.14.2.4 Dementia::Mesh::~~Mesh ( void ) [virtual]

Destructor.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

Definició a la línia 30 del fitxer Mesh.cpp.

### 6.14.3 Documentació de les Funcions Membre

#### 6.14.3.1 Dementia::Geometry\* Dementia::Mesh::getGeometry ( ) [inline]

Obte la geometria.

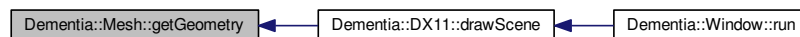
Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

##### Retorna

La geometria.

Definició a la línia 97 del fitxer Mesh.h.

Gràfic de crides a aquesta funció:



#### 6.14.3.2 Dementia::Shader\* Dementia::Mesh::getShader ( ) [inline]

Obte el shader.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

##### Retorna

El shader.

Definició a la línia 117 del fitxer Mesh.h.

#### 6.14.3.3 Dementia::Texture\* Dementia::Mesh::getTexture ( ) [inline]

Obte la textura.

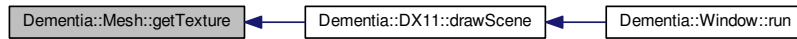
Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

**Retorna**

La textura.

Definició a la línia 107 del fitxer Mesh.h.

Gràfic de crides a aquesta funció:

**6.14.3.4 void Dementia::Mesh::setGeometry ( Dementia::Geometry \* *geometry* ) [inline]**

Posa o canvia la geometria que contindra.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

**Paràmetres**

<i>geometry</i>	[in] La geometria.
-----------------	--------------------

Definició a la línia 67 del fitxer Mesh.h.

**6.14.3.5 void Dementia::Mesh::setShader ( Dementia::Shader \* *shader* ) [inline]**

Posa o canvia el shader amb el que es dibuixara.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

**Paràmetres**

<i>shader</i>	[in] El shader.
---------------	-----------------

Definició a la línia 87 del fitxer Mesh.h.

**6.14.3.6 void Dementia::Mesh::setTexture ( Dementia::Texture \* *texture* ) [inline]**

Posa o canvia la textura que utilitzara

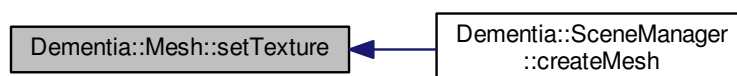
Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

**Paràmetres**

<i>texture</i>	[in] La textura.
----------------	------------------

Definició a la línia 77 del fitxer Mesh.h.

Gràfic de crides a aquesta funció:



La documentació d'aquesta classe es va generar a partir dels següents fitxers:

- Include/[Mesh.h](#)
- src/[Mesh.cpp](#)

## 6.15 Referència de la Classe Dementia::Mouse

Classe encarregada del ratolí que fa de intermediari entre l'usuari i el Sistema Operatiu

```
#include <Mouse.h>
```

### Tipus Públics

- enum [Button](#) {  
    [Right](#), [Left](#), [Middle](#), [Button4](#),  
    [Button5](#), [COUNT](#) }

*Els possibles buttons dels quals es pot monitoritzar el seu estat.*

### Mètodes públics

- [Mouse](#) (void)  
*Constructor per defecte.*
- [Mouse](#) (float x, float y)  
*Constructor que inicialitza el ratolí a una posició concreta. Al inicialitzar es mou el ratolí a la posició especificada.*
- virtual [~Mouse](#) (void)  
*Destructor.*
- XMFLOAT2 [getPosition](#) ()  
*Obté la posició actual on es troba el ratolí.*
- virtual void [setPosition](#) (XMFLOAT2 newPosition)  
*Posa el ratolí a una posició concreta de la pantalla.*
- void [updatePosition](#) (XMFLOAT2 newPosition)  
*Actualitza la variable interna de posició del ratolí de la classe. No mou el ratolí a la nova posició. Aquest mètode no s'ha de utilitzar a no ser que sapigues que estas fent.*
- void [setButtonPressed](#) ([Button](#) buttonPressed, XMFLOAT2 position)  
*Posa el boto pasat per parametre com pulsat.*
- void [setButtonReleased](#) ([Button](#) buttonReleased, XMFLOAT2 position)  
*Posa el boto pasat per parametre com a lliberat.*
- bool [isButtonPressed](#) ([Button](#) button)  
*Pregunta si el boto pasat per parametre esta pulsat.*
- bool [isMoved](#) ()  
*Pregunta si el ratolí s'ha mogut des de la ultima vegada que s'ha cridat al mètode getPosition.*
- void [setCenter](#) (XMFLOAT2 center)  
*L'indica el ratolí on es troba el centre de la finestra.*

### 6.15.1 Descripció Detallada

Classe encarregada del ratolí que fa de intermediari entre l'usuari i el Sistema Operatiu

Alberto Lopez Sanchez i Ruben Bagan Benavides, 14/05/2013.

Definició a la línia 15 del fitxer Mouse.h.



## 6.15.2 Documentació de les Enumeracions Membre

### 6.15.2.1 enum Dementia::Mouse::Button

Els possibles buttons dels quals es pot monitoritzar el seu estat.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 14/05/2013.

Valors de les Enumeracions

**Right** El Boto dret del ratoli.

**Left** El boto esquerra del ratoli.

**Middle** El boto central del ratoli.

**Button4** El Buto4 del ratoli, normalment es un dels buttons laterals.

**Button5** El Buto4 del ratoli, normalment es un dels buttons laterals.

**COUNT** El numero de elements que te el enum.

Definició a la línia 25 del fitxer Mouse.h.

## 6.15.3 Documentació del Constructor i el Destructor

### 6.15.3.1 Dementia::Mouse::Mouse ( void )

Constructor per defecte.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 14/05/2013.

Definició a la línia 6 del fitxer Mouse.cpp.

### 6.15.3.2 Dementia::Mouse::Mouse ( float x, float y )

Constructor que inicialitza el ratoli a una posicio concreta. Al inicialitzar es moua el ratoli a la posicio especificada.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 14/05/2013.

Paràmetres

x	La coordenada X.
y	La coordenada Y.

Definició a la línia 13 del fitxer Mouse.cpp.

### 6.15.3.3 Dementia::Mouse::~~Mouse ( void ) [virtual]

Destructor.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 14/05/2013.

Definició a la línia 20 del fitxer Mouse.cpp.

## 6.15.4 Documentació de les Funcions Membre

### 6.15.4.1 XMFLOAT2 Dementia::Mouse::getPosition ( )

Obte la posicio actual on es troba el ratoli.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 14/05/2013.

**Retorna**

La posicio.

Definició a la línia 68 del fitxer Mouse.cpp.

**6.15.4.2 bool Dementia::Mouse::isButtonPressed ( Button *button* )**

Pregunta si el boto pasat per parametre esta pulsat.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 14/05/2013.

**Paràmetres**

<i>button</i>	El boto per el qual preguntes.
---------------	--------------------------------

**Retorna**

true si el boto per el qual preguntes esta pulsat, false si no ho esta.

Definició a la línia 36 del fitxer Mouse.cpp.

**6.15.4.3 bool Dementia::Mouse::isMoved ( )**

Pregunta si el ratoli s'ha mogut des de la ultima vegada que s'ha cridat al metode getPosition.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 14/05/2013.

**Retorna**

true si s'ha mogut, false si no.

Definició a la línia 41 del fitxer Mouse.cpp.

**6.15.4.4 void Dementia::Mouse::setButtonPressed ( Button *buttonPressed*, XMFLOAT2 *position* )**

Posa el boto pasat per parametre com pulsat.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 14/05/2013.

**Paràmetres**

<i>buttonPressed</i>	El boto que es marcara com pulsat.
<i>position</i>	La posicio on s'ha pulsat el boto.

Definició a la línia 24 del fitxer Mouse.cpp.

Gràfic de crides a aquesta funció:

**6.15.4.5 void Dementia::Mouse::setButtonReleased ( Button *buttonReleased*, XMFLOAT2 *position* )**

Posa el boto pasat per parametre com a lliberat.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 14/05/2013.

## Paràmetres

<i>buttonReleased</i>	El boto que es marcara com a lliberat.
<i>position</i>	La posicio on s'ha alliberat el boto.

Definició a la línia 30 del fitxer Mouse.cpp.

Gràfic de crides a aquesta funció:



## 6.15.4.6 void Dementia::Mouse::setCenter ( XMFLOAT2 center )

L'indica el ratoli on es troba el centre de la finestra.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

## Paràmetres

<i>center</i>	El centre de la finestra.
---------------	---------------------------

Definició a la línia 57 del fitxer Mouse.cpp.

## 6.15.4.7 void Dementia::Mouse::setPosition ( XMFLOAT2 newPosition ) [virtual]

Posa el ratoli a una posicio concreta de la pantalla.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 14/05/2013.

## Paràmetres

<i>newPosition</i>	La nova posicio.
--------------------	------------------

Definició a la línia 63 del fitxer Mouse.cpp.

## 6.15.4.8 void Dementia::Mouse::updatePosition ( XMFLOAT2 newPosition )

Actualitza la variable interna de posicio del ratoli de la classe. No mou el ratoli a la nova posicio. Aquest metode no s'ha de utilitzar a no ser que sapigues que estas fent.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 14/05/2013.

## Paràmetres

<i>newPosition</i>	La nova posicio.
--------------------	------------------

Definició a la línia 46 del fitxer Mouse.cpp.

Gràfic de crides a aquesta funció:



La documentació d'aquesta classe es va generar a partir dels següents fitxers:

- Include/[Mouse.h](#)
- src/[Mouse.cpp](#)

## 6.16 Referència de la Classe Dementia::Node

Un node es un objecte invisible que se encarrega de sapiguer on, amb quina escala, i amb quina rotació s'ha de renderitzar qualsevol entitat.

```
#include <Node.h>
```

### Mètodes públics

- [Node](#) (UINT id)  
*Constructor per defecte.*
- [Node](#) (UINT id, XMVECTOR position, XMVECTOR scale, XMVECTOR rotation)  
*Constructor.*
- [~Node](#) (void)  
*Destructor.*
- UINT [getId](#) ()  
*Obte l'identificador del node.*
- XMVECTOR [getLocalPosition](#) ()  
*Obte la posició relativa al node pare.*
- XMVECTOR [getLocalScale](#) ()  
*Obte la escala relativa al node pare.*
- XMVECTOR [getLocalRotation](#) ()  
*Obte la rotació relativa al node pare.*
- XMVECTOR [getWorldPosition](#) ()  
*Obte la posició on es troba el node respecte al mon.*
- XMVECTOR [getWorldScale](#) ()  
*Obte la escala total respecte al mon.*
- XMVECTOR [getWorldRotation](#) ()  
*Obte la rotació total respecte al mon.*
- [Dementia::Node \\*](#) [getParent](#) ()  
*Obte el node pare. El Root Scene [Node](#) no te pare.*
- [Dementia::Entity \\*](#) [getEntity](#) ()  
*Obte la entitat associada a aquest node.*
- XMMATRIX [getWorldMatrix](#) (void)  
*Obte la matriu de mon (World Matrix) que es la multiplicació de les següents matrius:*
- void [getAllNodes](#) (std::vector< [Dementia::Node \\*](#) > \*nodes)  
*Obte un vector amb tots els nodes fills del node pasat per parametre. Evitar utilitzar aquesta funció, té un cost de CPU elevat.*
- void [setEntity](#) ([Dementia::Entity \\*](#)entity)  
*Posa o canvia la entitat que té aquest node.*
- void [setLocalPosition](#) (XMVECTOR position)  
*Posa o canvia la posició que tindrà aquest node respecte al node pare*
- void [setLocalScale](#) (XMVECTOR scale)  
*Posa o canvia la escala que tindrà aquest node respecte al node pare.*
- void [setLocalRotation](#) (XMVECTOR rotation)  
*Posa o canvia la rotació que tindrà aquest node respecte al node pare.*
- void [addChild](#) ([Dementia::Node \\*](#)child)  
*Afegeix un fill al node actual.*

- **Dementia::Node \* getChild** (UINT id)  
*Obte el fill a partir de l'id pasat per parametre.*
- **bool removeChild** (UINT id)  
*Eborra el fill que te l'identificador pasat per parametre.*
- **void setParent** (Dementia::Node \*parent)  
*Especifica quin sera el pare de aquest node, si ya en tenia un pare, se li notifica al pare que aquest node deixa de ser fill seu i se li notifica al nou pare que are te aquest fill.*

### 6.16.1 Descripció Detallada

Un node es un objecte invisible que se encarrega de sapiguer on, amb quina escala, i amb quina rotacio s'ha de renderitzar qualsevol entitat.

El sistema de nodes es gerarquic, això significa que un node pot tenir varis fills i només un pare, els nodes fills hereden tots els valors dels pares de forma transparent.

Exemple 1: El node pare te escala 5 i el node fill te escala 1, però el fill ya esta escalat a 5.

Exemple 2: El node pare te escala 5 i el node fill te escala 2, la escala total del node fill es la del pare mes la seva, es a dir 7.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

Definició a la línia 24 del fitxer Node.h.

### 6.16.2 Documentació del Constructor i el Destructor

#### 6.16.2.1 Dementia::Node::Node ( UINT id )

Constructor per defecte.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

Paràmetres

<i>id</i>	L'identificador del node.
-----------	---------------------------

Definició a la línia 5 del fitxer Node.cpp.

#### 6.16.2.2 Dementia::Node::Node ( UINT id, XMVECTOR position, XMVECTOR scale, XMVECTOR rotation )

Constructor.

Rodolfo, 29/05/2013.

Paràmetres

<i>id</i>	L'identificador del node.
<i>position</i>	La posicio a la que es troba.
<i>scale</i>	La escala que te.
<i>rotation</i>	La seva rotacio.

Definició a la línia 22 del fitxer Node.cpp.

#### 6.16.2.3 Dementia::Node::~~Node ( void )

Destructor.

Rodolfo, 29/05/2013.

Definició a la línia 39 del fitxer Node.cpp.

### 6.16.3 Documentació de les Funcions Membre

#### 6.16.3.1 void Dementia::Node::addChild ( Dementia::Node \* *child* )

Afegeix un fill al node actual.

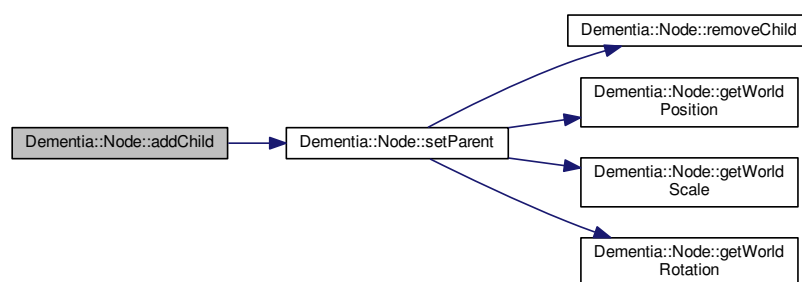
Rodolfo, 29/05/2013.

##### Paràmetres

<i>child</i>	[in] El fill que s'afegira al node.
--------------	-------------------------------------

Definició a la línia 130 del fitxer Node.cpp.

Gràfic de crides d'aquesta funció:



#### 6.16.3.2 void Dementia::Node::getAllNodes ( std::vector< Dementia::Node \* > \* *nodes* )

Obte un vector amb tots els nodes fills del node pasat per parametre. Evitar ultilitzar aquesta funcio, te un cost de CPU elevat.

Rodolfo, 29/05/2013.

##### Paràmetres

<i>nodes</i>	[in] El node del qual es buesquen tots els fills.
--------------	---

Definició a la línia 120 del fitxer Node.cpp.

#### 6.16.3.3 Dementia::Node \* Dementia::Node::getChild ( UINT *id* )

Obte el fill a partir de l'id pasat per parametre.

Rodolfo, 29/05/2013.

##### Paràmetres

<i>id</i>	L'ID del node a buscar.
-----------	-------------------------

##### Retorna

El fill si l'ha trobat, si no torna null.

Definició a la línia 136 del fitxer Node.cpp.

**6.16.3.4 Dementia::Entity\* Dementia::Node::getEntity ( ) [inline]**

Obte la entitat associada a aquest node.

Rodolfo, 29/05/2013.

**Retorna**

La entitat associada a aquest node, null si no en te cap.

Definició a la línia 148 del fitxer Node.h.

**6.16.3.5 UINT Dementia::Node::getId ( ) [inline]**

Obte l'identificador del node.

Rodolfo, 29/05/2013.

**Retorna**

L'identificador del node.

Definició a la línia 68 del fitxer Node.h.

**6.16.3.6 XMVECTOR Dementia::Node::getLocalPosition ( ) [inline]**

Obte la posicio relativa al node pare.

Rodolfo, 29/05/2013.

**Retorna**

La posicio relativa al node pare.

Definició a la línia 78 del fitxer Node.h.

**6.16.3.7 XMVECTOR Dementia::Node::getLocalRotation ( ) [inline]**

Obte la rotacio relativa al node pare.

Rodolfo, 29/05/2013.

**Retorna**

La rotacio relativa al node pare.

Definició a la línia 98 del fitxer Node.h.

**6.16.3.8 XMVECTOR Dementia::Node::getLocalScale ( ) [inline]**

Obte la escala relativa al node pare.

Rodolfo, 29/05/2013.

**Retorna**

La escala relativa al node pare.

Definició a la línia 88 del fitxer Node.h.

#### 6.16.3.9 Dementia::Node\* Dementia::Node::getParent ( ) [inline]

Obte el node pare. El Root Scene [Node](#) no te pare.

Rodolfo, 29/05/2013.

##### Retorna

El node pare, null si no te pare.

Definició a la línia 138 del fitxer Node.h.

#### 6.16.3.10 XMATRIX Dementia::Node::getWorldMatrix ( void )

Obte la matriu de mon (World Matrix) que es la multiplicacio de les següents matrius:

- Translacio (posicio on es troba el node respecte el mon)
- Scala (la scala total del node respecte el mon)
- Rotacio (la rotacio total del node respecte el mon)

Rodolfo, 29/05/2013.

##### Retorna

La matriu de mon (World Matrix).

Definició a la línia 96 del fitxer Node.cpp.

#### 6.16.3.11 XMVECTOR Dementia::Node::getWorldPosition ( ) [inline]

Obte la posicio on es troba el node respecte al mon.

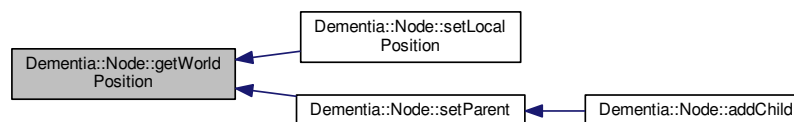
Rodolfo, 29/05/2013.

##### Retorna

La posicio on es troba el node respecte al mon.

Definició a la línia 108 del fitxer Node.h.

Gràfic de crides a aquesta funció:



#### 6.16.3.12 XMVECTOR Dementia::Node::getWorldRotation ( ) [inline]

Obte la rotacio total respecte al mon.

Rodolfo, 29/05/2013.

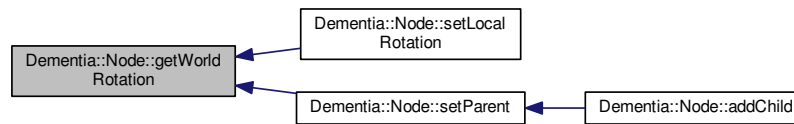


**Retorna**

La rotació total respecte al mon.

Definició a la línia 128 del fitxer Node.h.

Gràfic de crides a aquesta funció:



### 6.16.3.13 XMVECTOR Dementia::Node::getWorldScale ( ) [inline]

Obte la scala total respecte al mon.

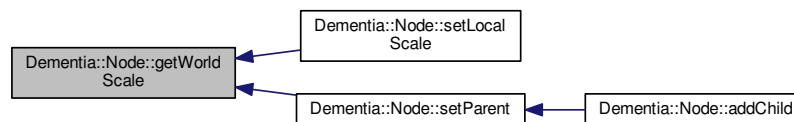
Rodolfo, 29/05/2013.

**Retorna**

La scala total respecte al mon.

Definició a la línia 118 del fitxer Node.h.

Gràfic de crides a aquesta funció:



### 6.16.3.14 bool Dementia::Node::removeChild ( UINT id )

Esborra el fill que te l'identificador pasat per parametre.

Rodolfo, 29/05/2013.

**Paràmetres**

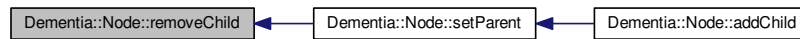
<i>id</i>	L'ID del node a esborrar.
-----------	---------------------------

**Retorna**

true si s'ha esborrat, false si no s'ha trobat.

Definició a la línia 147 del fitxer Node.cpp.

Gràfic de crides a aquesta funció:



#### 6.16.3.15 void Dementia::Node::setEntity ( Dementia::Entity \* *entity* ) [inline]

Posa o canvia la entitat que té aquest node.

Rodolfo, 29/05/2013.

##### Paràmetres

<i>entity</i>	[in] La entitat que tindrà aquest node.
---------------	---

Definició a la línia 186 del fitxer Node.h.

#### 6.16.3.16 void Dementia::Node::setLocalPosition ( XMVECTOR *position* )

Posa o canvia la posició que tindrà aquest node respecte al node pare

Rodolfo, 29/05/2013.

##### Paràmetres

<i>position</i>	La posició respecte al node pare.
-----------------	-----------------------------------

Definició a la línia 43 del fitxer Node.cpp.

Gràfic de crides d'aquesta funció:



#### 6.16.3.17 void Dementia::Node::setLocalRotation ( XMVECTOR *rotation* )

Posa o canvia la rotació que tindrà aquest node respecte al node pare.

Rodolfo, 29/05/2013.

##### Paràmetres

<i>rotation</i>	La rotació del node.
-----------------	----------------------

Definició a la línia 63 del fitxer Node.cpp.

Gràfic de crides d'aquesta funció:



#### 6.16.3.18 void Dementia::Node::setLocalScale ( XMVECTOR *scale* )

Posa o cambia la scala que tindra aquest node respecte al node pare.

Rodolfo, 29/05/2013.

Paràmetres

<i>scale</i>	La scala del node.
--------------	--------------------

Definició a la línia 53 del fitxer Node.cpp.

Gràfic de crides d'aquesta funció:



#### 6.16.3.19 void Dementia::Node::setParent ( Dementia::Node \* *parent* )

Especifica quin sera el pare de aquet node, si ya en tenia un pare, se li notifica al pare que aquet node deixa de ser fill seu i se li notifica al nou pare que are te aquest fill.

Aquesta funcio es privada porque nomes els propis nodes han de poder fer això.

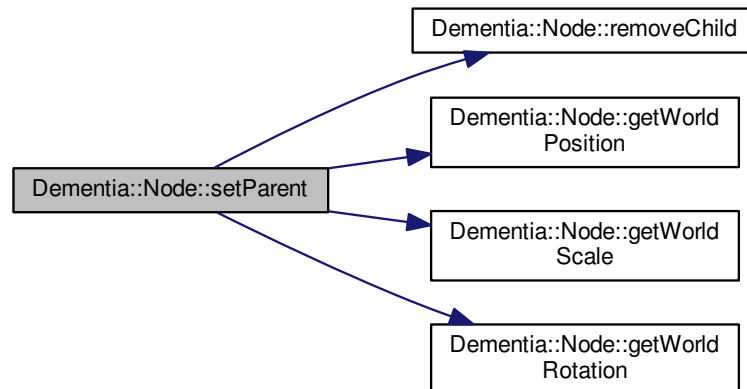
Rodolfo, 29/05/2013.

Paràmetres

<i>parent</i>	[in] El seu pare.
---------------	-------------------

Definició a la línia 101 del fitxer Node.cpp.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



La documentació d'aquesta classe es va generar a partir dels següents fitxers:

- [Include/Node.h](#)
- [src/Node.cpp](#)

## 6.17 Referència de l'Estructura Dementia::PointLight

Una llum de tipus punt, que irradia a tots els objectes que estiguin dins del seu radi d'acció per igual.

```
#include <Light.h>
```

### Mètodes públics

- [PointLight \(\)](#)  
*Constructor per defecte.*
- [PointLight](#) (XMFLOAT4 ambient, XMFLOAT4 diffuse, XMFLOAT4 specular, XMFLOAT3 position, float range, XMFLOAT3 att, float pad)  
*Constructor de la llum.*

## Atributs Públics

- XMFLOAT4 [Ambient](#)  
*El color de la llum ambient i la seva intensitat (R,G,B, Intensitat).*
- XMFLOAT4 [Diffuse](#)  
*El color de la llum que reflexan els objectes i la seva intensitat (R,G,B, Intensitat).*
- XMFLOAT4 [Specular](#)  
*El color de la llum del brillantor dels objectes i la seva intensitat (R,G,B, Intensitat).*
- XMFLOAT3 [Position](#)  
*La posició on es troba la llum.*
- float [Range](#)  
*El radi de distància fins a on arriba la llum.*
- XMFLOAT3 [Att](#)  
*La atenuació que va guanyant segons la distància.*
- float [Pad](#)  
*Un valor sense cap funcionalitat, que només serveix per que la estructura estigui alienada en 16bytes.*

### 6.17.1 Descripció Detallada

Una llum de tipus punt, que irradia a tots els objectes que estiguin dins del seu radi d'acció per igual.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

Definició a la línia 115 del fitxer Light.h.

### 6.17.2 Documentació del Constructor i el Destructor

#### 6.17.2.1 Dementia::PointLight::PointLight ( ) [inline]

Constructor per defecte.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

Definició a la línia 123 del fitxer Light.h.

#### 6.17.2.2 Dementia::PointLight::PointLight ( XMFLOAT4 *ambient*, XMFLOAT4 *diffuse*, XMFLOAT4 *specular*, XMFLOAT3 *position*, float *range*, XMFLOAT3 *att*, float *pad* ) [inline]

Constructor de la llum.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

#### Paràmetres

<i>ambient</i>	El color de la llum ambient i la seva intensitat (R,G,B, Intensitat).
<i>diffuse</i>	El color de la llum que reflexan els objectes i la seva intensitat (R,G,B, Intensitat).
<i>specular</i>	El color de la llum del brillantor dels objectes i la seva intensitat (R,G,B, Intensitat).
<i>position</i>	La posició on es troba la llum.
<i>range</i>	El radi de distància fins a on arriba la llum.
<i>att</i>	La atenuació que va guanyant segons la distància.
<i>pad</i>	Un valor sense cap funcionalitat, que només serveix per que la estructura estigui alienada en 16bytes.

Definició a la línia 142 del fitxer Light.h.

### 6.17.3 Documentació de les Dades Membre

#### 6.17.3.1 XMFLOAT4 Dementia::PointLight::Ambient

El color de la llum ambient i la seva intensitat (R,G,B, Intensitat).

Definició a la línia 154 del fitxer Light.h.

#### 6.17.3.2 XMFLOAT3 Dementia::PointLight::Att

La atenuació que va guanyant segons la distància.

Definició a la línia 169 del fitxer Light.h.

#### 6.17.3.3 XMFLOAT4 Dementia::PointLight::Diffuse

El color de la llum que reflexen els objectes i la seva intensitat (R,G,B, Intensitat).

Definició a la línia 157 del fitxer Light.h.

#### 6.17.3.4 float Dementia::PointLight::Pad

Un valor sense cap funcionalitat, que només serveix per que la estructura estigui alienada en 16bytes.

Definició a la línia 172 del fitxer Light.h.

#### 6.17.3.5 XMFLOAT3 Dementia::PointLight::Position

La posició on es troba la llum.

Definició a la línia 163 del fitxer Light.h.

#### 6.17.3.6 float Dementia::PointLight::Range

El radi de distància fins a on arriba la llum.

Definició a la línia 166 del fitxer Light.h.

#### 6.17.3.7 XMFLOAT4 Dementia::PointLight::Specular

El color de la llum del brillantor dels objectes i la seva intensitat (R,G,B, Intensitat).

Definició a la línia 160 del fitxer Light.h.

La documentació d'aquesta estructura es va generar a partir del següent fitxer:

- Include/[Light.h](#)

## 6.18 Referència de la Classe Dementia::SceneManager

Classe encarregada de gestionar tot el relacionat amb tot el que surt per pantalla.

```
#include <SceneManager.h>
```

## Mètodes públics

- [SceneManager](#) ([Dementia::Window](#) \*window)  
*Constructor.*
- virtual [~SceneManager](#) (void)  
*Destructor.*
- [Dementia::Node](#) \* [createNode](#) (XMVECTOR position, XMVECTOR scale, XMVECTOR rotation)  
*Crea un node.*
- [Dementia::Mesh](#) \* [createMesh](#) ([Dementia::Geometry](#) \*geometry)  
*Crea una mesh.*
- [Dementia::Texture](#) \* [createTexture](#) (std::wstring path)  
*Crea un objecte textura.*
- [DirectionalLight](#) \* [createDirectionalLight](#) (XMFLOAT4 ambient, XMFLOAT4 diffuse, XMFLOAT4 specular, XMFLOAT3 direction)  
*Creaa una llum direccional.*
- [SpotLight](#) \* [createSpotLight](#) (XMFLOAT4 ambient, XMFLOAT4 diffuse, XMFLOAT4 specular, XMFLOAT3 position, float range, XMFLOAT3 direction, float spot, XMFLOAT3 att)  
*Crea una llum flash/llinterna.*
- [PointLight](#) \* [createPointLight](#) (XMFLOAT4 ambient, XMFLOAT4 diffuse, XMFLOAT4 specular, XMFLOAT3 position, float range, XMFLOAT3 att)  
*Crea una llum de punt.*
- [Material](#) \* [createMaterial](#) (XMFLOAT4 ambient, XMFLOAT4 diffuse, XMFLOAT4 specular, XMFLOAT4 re-  
flect)  
*Crea un material.*
- [Dementia::Node](#) \* [getRootSceneNode](#) (void)  
*Obte el node arrel de la escena.*
- std::vector< [Dementia::Mesh](#) \* > \* [getAllMeshes](#) ()  
*Obte tots els models que hi ha al escenari.*
- std::vector< [Dementia::Node](#) \* > \* [getAllNodes](#) ()  
*Obte tots els nodes que hi ha al escenari.*
- std::vector  
< [Dementia::DirectionalLight](#) \* > \* [getAllDirectionalLights](#) ()  
*Obte totes les llums direccionals que hi ha.*
- std::vector  
< [Dementia::SpotLight](#) \* > \* [getAllSpotLights](#) ()  
*Obte totes les llums de tipus flash/llinterna.*
- std::vector  
< [Dementia::PointLight](#) \* > \* [getAllPointLights](#) ()  
*Obte totes les llums de tipus punt.*
- bool [removeNode](#) (UINT id)  
*Esborra el node amb aquest ID.*

## 6.18.1 Descripció Detallada

Clase encarregada de gestionar tot el relacionat amb tot el que surt per pantalla.

Crea i destrueix els objectes que es retornen al usuari de manera transparent.

Rodolfo, 29/05/2013.

Definició a la línia 22 del fitxer SceneManager.h.

## 6.18.2 Documentació del Constructor i el Destructor

### 6.18.2.1 Dementia::SceneManager::SceneManager ( Dementia::Window \* window )

Constructor.

Rodolfo, 29/05/2013.

Paràmetres

<i>window</i>	[in] Rep la finestra que el crea.
---------------	-----------------------------------

Definició a la línia 4 del fitxer SceneManager.cpp.

Gràfic de crides d'aquesta funció:



### 6.18.2.2 Dementia::SceneManager::~SceneManager ( void ) [virtual]

Destructor.

Rodolfo, 29/05/2013.

Definició a la línia 11 del fitxer SceneManager.cpp.

## 6.18.3 Documentació de les Funcions Membre

### 6.18.3.1 DirectionalLight \* Dementia::SceneManager::createDirectionalLight ( XMFLOAT4 ambient, XMFLOAT4 diffuse, XMFLOAT4 specular, XMFLOAT3 direction )

Creaa una llum direccional.

Rodolfo, 29/05/2013.

Paràmetres

<i>ambient</i>	El color de la llum ambient que reflexa l'objecte i la seva intensitat.
<i>diffuse</i>	El color de la llum que difusa l'objecte i la seva intensitat.
<i>specular</i>	El color de la llum de la brillantor de l'objecte i la seva intensitat..
<i>direction</i>	La direccio cap a on apunta la llum.

Retorna

La llum direccional.

Definició a la línia 65 del fitxer SceneManager.cpp.

### 6.18.3.2 Material \* Dementia::SceneManager::createMaterial ( XMFLOAT4 ambient, XMFLOAT4 diffuse, XMFLOAT4 specular, XMFLOAT4 reflect )

Crea un material.

Rodolfo, 29/05/2013.



## Paràmetres

<i>ambient</i>	El color de la llum ambient que reflexa l'objecte i la seva intensitat.
<i>diffuse</i>	El color de la llum que difusa l'objecte i la seva intensitat.
<i>specular</i>	El color de la llum de la brillantor de l'objecte i la seva intensitat.
<i>reflect</i>	El color de la llum que reflexa globalment l'objecte i la seva intensitat.

## Retorna

El material.

Definició a la línia 89 del fitxer SceneManager.cpp.

## 6.18.3.3 Dementia::Mesh \* Dementia::SceneManager::createMesh ( Dementia::Geometry \* geometry )

Crea una mesh.

Rodolfo, 29/05/2013.

## Paràmetres

<i>geometry</i>	[in] La geometria associada.
-----------------	------------------------------

## Retorna

La geometria creada.

Definició a la línia 47 del fitxer SceneManager.cpp.

Gràfic de crides d'aquesta funció:



## 6.18.3.4 Dementia::Node \* Dementia::SceneManager::createNode ( XMVECTOR position, XMVECTOR scale, XMVECTOR rotation )

Crea un node.

Rodolfo, 29/05/2013.

## Paràmetres

<i>position</i>	La posicio local del node.
<i>scale</i>	La escala local del node.
<i>rotation</i>	La rotacio local del node.

## Retorna

El node creat.

Definició a la línia 39 del fitxer SceneManager.cpp.

#### 6.18.3.5 **PointLight** \* Dementia::SceneManager::createPointLight ( XMFLOAT4 *ambient*, XMFLOAT4 *diffuse*, XMFLOAT4 *specular*, XMFLOAT3 *position*, float *range*, XMFLOAT3 *att* )

Crea una llum de punt.

Rodolfo, 29/05/2013.

##### Paràmetres

<i>ambient</i>	El color de la llum ambient que reflexa l'objecte i la seva intensitat.
<i>diffuse</i>	El color de la llum que diffusa l'objecte i la seva intensitat.
<i>specular</i>	El color de la llum de la brillantor de l'objecte i la seva intensitat.
<i>position</i>	La posició des de on irradia la llum.
<i>range</i>	El radi de distància fins a on arriba la llum.
<i>att</i>	La atenuació de la llum segons la distància.

##### Retorna

La llum de punt.

Definició a la línia 81 del fitxer SceneManager.cpp.

#### 6.18.3.6 **SpotLight** \* Dementia::SceneManager::createSpotLight ( XMFLOAT4 *ambient*, XMFLOAT4 *diffuse*, XMFLOAT4 *specular*, XMFLOAT3 *position*, float *range*, XMFLOAT3 *direction*, float *spot*, XMFLOAT3 *att* )

Crea una llum flash/l'linterna

Rodolfo, 29/05/2013.

##### Paràmetres

<i>ambient</i>	El color de la llum ambient que reflexa l'objecte i la seva intensitat.
<i>diffuse</i>	El color de la llum que diffusa l'objecte i la seva intensitat.
<i>specular</i>	El color de la llum de la brillantor de l'objecte i la seva intensitat.
<i>position</i>	La posició des de on irradia la llum.
<i>range</i>	El radi de distància fins a on arriba la llum.
<i>direction</i>	La direcció cap a on apunta la llum.
<i>spot</i>	El exponent utilitzat per controlar la grandària del con de la llum.
<i>att</i>	La atenuació de la llum segons la distància.

##### Retorna

La llum flash/l'linterna.

Definició a la línia 73 del fitxer SceneManager.cpp.

#### 6.18.3.7 **Dementia::Texture** \* Dementia::SceneManager::createTexture ( std::wstring *path* )

Crea un objecte textura.

Rodolfo, 29/05/2013.

##### Paràmetres

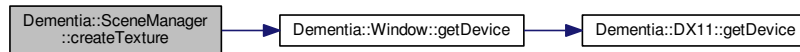
<i>path</i>	La ruta on es troba la textura que es carregarà.
-------------	--

**Retorna**

El objecte textura.

Definició a la línia 56 del fitxer SceneManager.cpp.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



#### 6.18.3.8 `std::vector<Dementia::DirectionalLight*>* Dementia::SceneManager::getAllDirectionalLights ( )` `[inline]`

Obte totes les llums direccionals que hi ha.

Rodolfo, 29/05/2013.

**Retorna**

Un vector amb totes les llums direccionals que hi ha.

Definició a la línia 189 del fitxer SceneManager.h.

#### 6.18.3.9 `std::vector<Dementia::Mesh*>* Dementia::SceneManager::getAllMeshes ( )` `[inline]`

Obte tots els models que hi ha al escenari.

Rodolfo, 29/05/2013.

**Retorna**

Un vector amb tots els models que hi ha al escenari.

Definició a la línia 169 del fitxer SceneManager.h.

#### 6.18.3.10 `std::vector<Dementia::Node*>* Dementia::SceneManager::getAllNodes ( )` `[inline]`

Obte tots els nodes que hi ha al escenari.

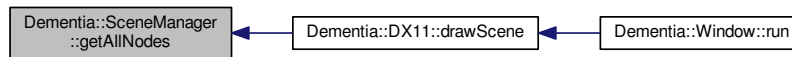
Rodolfo, 29/05/2013.

**Retorna**

Un vector amb tots els nodes que hi ha al escenari.

Definició a la línia 179 del fitxer SceneManager.h.

Gràfic de crides a aquesta funció:



#### 6.18.3.11 `std::vector<Dementia::PointLight*>* Dementia::SceneManager::getAllPointLights ( ) [inline]`

Obte totes les llums de tipus punt.

Rodolfo, 29/05/2013.

**Retorna**

Un vector amb totes les llums de tipus punt.

Definició a la línia 209 del fitxer SceneManager.h.

#### 6.18.3.12 `std::vector<Dementia::SpotLight*>* Dementia::SceneManager::getAllSpotLights ( ) [inline]`

Obte totes les llums de tipus flash/linternas.

Rodolfo, 29/05/2013.

**Retorna**

Un vector amb totes les llums de tipus flash/linternas.

Definició a la línia 199 del fitxer SceneManager.h.

#### 6.18.3.13 `Dementia::Node* Dementia::SceneManager::getRootSceneNode ( void ) [inline]`

Obte el node arrel de la escena.

Rodolfo, 29/05/2013.

**Retorna**

El node arrel de la escena.

Definició a la línia 159 del fitxer SceneManager.h.

#### 6.18.3.14 `bool Dementia::SceneManager::removeNode ( UINT id )`

Esborra el node amb aquest ID.

Rodolfo, 30/05/2013.

## Paràmetres

<i>id</i>	El identificador.
-----------	-------------------

## Retorna

true si l'esborra, false si no.

Definició a la línia 96 del fitxer SceneManager.cpp.

La documentació d'aquesta classe es va generar a partir dels següents fitxers:

- Include/[SceneManager.h](#)
- src/[SceneManager.cpp](#)

## 6.19 Referència de la Classe Dementia::Shader

Clase que conte tots elements necessaris per treballar amb els shaders.

```
#include <Shader.h>
```

## Mètodes públics

- [Shader](#) (void)  
*Constructor per defecte.*
- [~Shader](#) (void)  
*Destructor.*

## Atributs Públics

- ID3D11InputLayout \* [g\\_pVertexLayout](#)  
*Interfície de directx que conte totes les dades sobre com es la disposicio dels vertexs que utilitzarem per dibuixar per pantalla*
- ID3D11VertexShader \* [g\\_pVertexShader](#)  
*Interfície de DirectX que conte el Vertex [Shader](#).*
- ID3D11PixelShader \* [g\\_pPixelShader](#)  
*Interfície de DirectX que conte el Pixel [Shader](#).*
- ID3D11Buffer \* [dx\\_constantBuffer](#)  
*Interfície de DirectX que conte un Buffer amb la estructura de dades que se li pasa al shader per actualitzar els valors de dibuixat.*

### 6.19.1 Descripció Detallada

Clase que conte tots elements necessaris per treballar amb els shaders.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

Definició a la línia 15 del fitxer Shader.h.

## 6.19.2 Documentació del Constructor i el Destructor

### 6.19.2.1 Dementia::Shader::Shader ( void )

Constructor per defecte.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

Definició a la línia 5 del fitxer Shader.cpp.

### 6.19.2.2 Dementia::Shader::~~Shader ( void )

Destructor.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

Definició a la línia 13 del fitxer Shader.cpp.

## 6.19.3 Documentació de les Dades Membre

### 6.19.3.1 ID3D11Buffer\* Dementia::Shader::dx\_constantBuffer

Interfície de DirectX que conte un Buffer amb la estructura de dades que se li pasa al shader per actualitzar els valors de dibuixat.

Definició a la línia 47 del fitxer Shader.h.

### 6.19.3.2 ID3D11PixelShader\* Dementia::Shader::g\_pPixelShader

Interfície de DirectX que conte el Pixel [Shader](#).

Definició a la línia 44 del fitxer Shader.h.

### 6.19.3.3 ID3D11InputLayout\* Dementia::Shader::g\_pVertexLayout

Interfície de directx que conte totes les dades sobre com es la disposicio dels vertexs que ultilitzarem per dibuixar per pantalla

Definició a la línia 38 del fitxer Shader.h.

### 6.19.3.4 ID3D11VertexShader\* Dementia::Shader::g\_pVertexShader

Interfície de DirectX que conte el Vertex [Shader](#).

Definició a la línia 41 del fitxer Shader.h.

La documentació d'aquesta classe es va generar a partir dels següents fitxers:

- Include/[Shader.h](#)
- src/[Shader.cpp](#)

## 6.20 Referència de l'Estructura Dementia::SpotLight

Una llum de tipus llinerna o focus, ilumina nomes als objectes en els que la direccio apunti i estiguin dins del seu radi.

```
#include <Light.h>
```

## Mètodes públics

- [SpotLight](#) ()  
*Constructor per defecte.*
- [SpotLight](#) (XMFLOAT4 ambient, XMFLOAT4 diffuse, XMFLOAT4 specular, XMFLOAT3 position, float range, XMFLOAT3 direction, float spot, XMFLOAT3 att, float pad)  
*Constructor de la llum.*

## Atributs Públics

- XMFLOAT4 [Ambient](#)  
*El color de la llum ambient i la seva intensitat (R,G,B, Intensitat).*
- XMFLOAT4 [Diffuse](#)  
*El color de la llum que reflexan els objectes i la seva intensitat (R,G,B, Intensitat).*
- XMFLOAT4 [Specular](#)  
*El color de la llum del brillantor dels objectes i la seva intensitat (R,G,B, Intensitat).*
- XMFLOAT3 [Position](#)  
*La posició des de on irradia la llum.*
- float [Range](#)  
*El radi de distància fins a on arriba la llum.*
- XMFLOAT3 [Direction](#)  
*La direcció cap a on apunta la llum.*
- float [Spot](#)  
*El exponent utilitzat per controlar la grandia del con de la llum.*
- XMFLOAT3 [Att](#)  
*La atenuació de la llum segons la distància.*
- float [Pad](#)  
*Un valor sense cap funcionalitat, que només serveix per que la estructura estigui alienada en 16bytes.*

### 6.20.1 Descripció Detallada

Una llum de tipus llinterna o focus, il·lumina només als objectes en els que la direcció apunti i estiguin dins del seu radi.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

Definició a la línia 183 del fitxer Light.h.

### 6.20.2 Documentació del Constructor i el Destructor

#### 6.20.2.1 Dementia::SpotLight::SpotLight ( ) [inline]

Constructor per defecte.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

Definició a la línia 191 del fitxer Light.h.

#### 6.20.2.2 Dementia::SpotLight::SpotLight ( XMFLOAT4 ambient, XMFLOAT4 diffuse, XMFLOAT4 specular, XMFLOAT3 position, float range, XMFLOAT3 direction, float spot, XMFLOAT3 att, float pad ) [inline]

Constructor de la llum.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

## Paràmetres

<i>ambient</i>	El color de la llum ambient i la seva intensitat (R,G,B, Intensitat).
<i>diffuse</i>	El color de la llum que reflexan els objectes i la seva intensitat (R,G,B, Intensitat).
<i>specular</i>	El color de la llum del brillantor dels objectes i la seva intensitat (R,G,B, Intensitat).
<i>position</i>	La posició des de on irradia la llum.
<i>range</i>	El radi de distància fins a on arriba la llum.
<i>direction</i>	La direcció cap a on apunta la llum.
<i>spot</i>	El exponent utilitzat per controlar la grandària del con de la llum.
<i>att</i>	La atenuació de la llum segons la distància.
<i>pad</i>	Un valor sense cap funcionalitat, que només serveix per que la estructura estigui alienada en 16bytes.

Definició a la línia 212 del fitxer Light.h.

### 6.20.3 Documentació de les Dades Membre

#### 6.20.3.1 XMFLOAT4 Dementia::SpotLight::Ambient

El color de la llum ambient i la seva intensitat (R,G,B, Intensitat).

Definició a la línia 226 del fitxer Light.h.

#### 6.20.3.2 XMFLOAT3 Dementia::SpotLight::Att

La atenuació de la llum segons la distància.

Definició a la línia 247 del fitxer Light.h.

#### 6.20.3.3 XMFLOAT4 Dementia::SpotLight::Diffuse

El color de la llum que reflexan els objectes i la seva intensitat (R,G,B, Intensitat).

Definició a la línia 229 del fitxer Light.h.

#### 6.20.3.4 XMFLOAT3 Dementia::SpotLight::Direction

La direcció cap a on apunta la llum.

Definició a la línia 241 del fitxer Light.h.

#### 6.20.3.5 float Dementia::SpotLight::Pad

Un valor sense cap funcionalitat, que només serveix per que la estructura estigui alienada en 16bytes.

Definició a la línia 250 del fitxer Light.h.

#### 6.20.3.6 XMFLOAT3 Dementia::SpotLight::Position

La posició des de on irradia la llum.

Definició a la línia 235 del fitxer Light.h.

#### 6.20.3.7 float Dementia::SpotLight::Range

El radi de distància fins a on arriba la llum.



Definició a la línia 238 del fitxer Light.h.

#### 6.20.3.8 XMFLOAT4 Dementia::SpotLight::Specular

El color de la llum del brillantor dels objectes i la seva intensitat (R,G,B, Intensitat).

Definició a la línia 232 del fitxer Light.h.

#### 6.20.3.9 float Dementia::SpotLight::Spot

El exponent utilitzat per controlar la grandia del con de la llum.

Definició a la línia 244 del fitxer Light.h.

La documentació d'aquesta estructura es va generar a partir del següent fitxer:

- Include/[Light.h](#)

## 6.21 Referència de la Classe Dementia::Texture

Clase que conte tot el necessari per dibuixar les textures per pantalla.

```
#include <Texture.h>
```

### Mètodes públics

- [Texture](#) (void)  
*Constructor per defecte.*
- virtual [~Texture](#) (void)  
*Destructor.*

### Atributs Públics

- ID3D11ShaderResourceView \* [g\\_pTextureRV](#)  
*La Interficie de DirectX encarregada de treballar amb la textura.*

#### 6.21.1 Descripció Detallada

Clase que conte tot el necessari per dibuixar les textures per pantalla.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

Definició a la línia 13 del fitxer Texture.h.

#### 6.21.2 Documentació del Constructor i el Destructor

##### 6.21.2.1 Dementia::Texture::Texture ( void )

Constructor per defecte.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

Definició a la línia 5 del fitxer Texture.cpp.

#### 6.21.2.2 Dementia::Texture::~~Texture ( void ) [virtual]

Destructor.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

Definició a la línia 11 del fitxer Texture.cpp.

### 6.21.3 Documentació de les Dades Membre

#### 6.21.3.1 ID3D11ShaderResourceView\* Dementia::Texture::g\_pTextureRV

La Interficie de DirectX encarregada de treballar amb la textura.

Definició a la línia 35 del fitxer Texture.h.

La documentació d'aquesta classe es va generar a partir dels següents fitxers:

- Include/[Texture.h](#)
- src/[Texture.cpp](#)

## 6.22 Referència de l'Estructura Dementia::Geometry::Vertex

Conte tots els elements necessaris per fer tots els calculs per pantalla (llums, etc).

```
#include <Geometry.h>
```

### Mètodes públics

- [Vertex](#) ()  
*Constructor per defecte.*
- [Vertex](#) (const XMFLOAT3 &p, const XMFLOAT3 &n, const XMFLOAT3 &t, const XMFLOAT2 &uv)  
*Constructor.*
- [Vertex](#) (float px, float py, float pz, float nx, float ny, float nz, float tx, float ty, float tz, float u, float v)  
*Constructor.*

### Atributs Públics

- XMFLOAT3 [Position](#)  
*La posicio del vertex relativa al seu espai local.*
- XMFLOAT3 [Normal](#)  
*El normal del vertex.*
- XMFLOAT3 [TangentU](#)  
*La tangent del vertex.*
- XMFLOAT2 [TexC](#)  
*Les cordenades de textura.*

#### 6.22.1 Descripció Detallada

Conte tots els elements necessaris per fer tots els calculs per pantalla (llums, etc).

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

Definició a la línia 27 del fitxer Geometry.h.

## 6.22.2 Documentació del Constructor i el Destructor

### 6.22.2.1 Dementia::Geometry::Vertex::Vertex ( ) [inline]

Constructor per defecte.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

Definició a la línia 35 del fitxer Geometry.h.

### 6.22.2.2 Dementia::Geometry::Vertex::Vertex ( const XMFLOAT3 & p, const XMFLOAT3 & n, const XMFLOAT3 & t, const XMFLOAT2 & uv ) [inline]

Constructor.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

Paràmetres

<i>p</i>	La posició del vertex relativa al seu espai local.
<i>n</i>	La direcció cap a on mira aquest vertex.
<i>t</i>	La tangent d'aquest vertex.
<i>uv</i>	Les coordenades de la textura com s'ha d'aplicar fins a aquest vertex.

Definició a la línia 48 del fitxer Geometry.h.

### 6.22.2.3 Dementia::Geometry::Vertex::Vertex ( float px, float py, float pz, float nx, float ny, float nz, float tx, float ty, float tz, float u, float v ) [inline]

Constructor.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 29/05/2013.

Paràmetres

<i>px</i>	Camp X de la posició.
<i>py</i>	Camp Y de la posició.
<i>pz</i>	Camp Z de la posició.
<i>nx</i>	Camp X del "normal".
<i>ny</i>	Camp Y del "normal".
<i>nz</i>	Camp Z del "normal".
<i>tx</i>	Camp X de la tangent.
<i>ty</i>	Camp Y de la tangent.
<i>tz</i>	Camp Z de la tangent.
<i>u</i>	Camp U de la coordenada de textura.
<i>v</i>	Camp V de la coordenada de textura.

Definició a la línia 69 del fitxer Geometry.h.

## 6.22.3 Documentació de les Dades Membre

### 6.22.3.1 XMFLOAT3 Dementia::Geometry::Vertex::Normal

El normal del vertex.

Definició a la línia 81 del fitxer Geometry.h.

### 6.22.3.2 XMFLOAT3 Dementia::Geometry::Vertex::Position

La posició del vertex relativa al seu espai local.

Definició a la línia 78 del fitxer Geometry.h.

### 6.22.3.3 XMFLOAT3 Dementia::Geometry::Vertex::TangentU

La tangent del vertex.

Definició a la línia 84 del fitxer Geometry.h.

### 6.22.3.4 XMFLOAT2 Dementia::Geometry::Vertex::TexC

Les coordenades de textura.

Definició a la línia 87 del fitxer Geometry.h.

La documentació d'aquesta estructura es va generar a partir del següent fitxer:

- Include/[Geometry.h](#)

## 6.23 Referència de la Classe Dementia::Window

Classe encarregada de gestionar tot el relacionat amb la finestra a nivell de Sistema Operatiu i conte tots els elements necessaris per utilitzar el framework.

```
#include <Window.h>
```

### Mètodes públics

- [Window](#) (UINT id, UINT width, UINT height, std::wstring title, bool vsync, bool windowed)  
*Constructor. Per inicialitzar la finestra crida al mètode [init\(\)](#).*
- virtual [~Window](#) ()  
*Destructor. Allibera tots els recursos automàticament.*
- void [init](#) ()  
*Inicialitza la finestra i tots els seus components. La crea i la mostra per pantalla. També inicialitza la implementació gràfica (DirectX o OpenGL).*
- void [release](#) (void)  
*Allibera tots els recursos de la finestra. no cal cridar-lo. Es crida automàticament quan es destrueix la finestra.*
- bool [run](#) (int \*exitCode)  
*Executa el bucle principal de la finestra. S'ha de fer en bucle per mantenir la execució.*
- void [setBackgroundColor](#) (XMVECTOR backgroundColor)  
*Cambia el color de fons de la finestra.*
- void [setWindowed](#) (bool state)  
*Posa la finestra a mode finestra o a pantalla completa.*
- void [setHeight](#) (UINT height)  
*Cambia la altura de la finestra.*
- void [setWidth](#) (UINT width)  
*Cambia l'ample de la finestra.*
- void [setTitle](#) (std::wstring title)  
*Cambia el títol de la finestra.*
- void [setVsync](#) (bool state)

- Cambia l'estat de la sincronització vertical.*
- XMVECTOR `getBackgroundColor ()`  
*Retorna el color de fons de la finestra.*
- bool `isWindowed ()`  
*Pregunta si la finestra està en mode finestra o a pantalla completa.*
- bool `isVsync ()`  
*Pregunta si la sincronització vertical està activada.*
- bool `isPaused ()`  
*Pregunta si la finestra està pausada. La finestra es pausa automàticament quan perd el focus i es torna a reanudar quan torna a tenir el focus.*
- UINT `getHeight ()`  
*Obté la alçada de la finestra.*
- UINT `getWidth ()`  
*Obté la amplada de la finestra.*
- UINT `getId ()`  
*Obté l'identificador de la finestra.*
- std::wstring `getTitle ()`  
*Obté el títol de la finestra.*
- HWND `getWindowHandle ()`  
*Obté el "handle" de la finestra.*
- Dementia::InputManager \* `getInputManager ()`  
*Obté el Input Manager.*
- Dementia::GameTimer \* `getGameTimer ()`  
*Obté el Game Timer.*
- Dementia::Camera \* `getCamera ()`  
*Obté la càmera.*
- Dementia::SceneManager \* `getSceneManager ()`  
*Obté el "Scene Manager"*
- ID3D11Device \* `getDevice ()`  
*Obté el Dispositiu Graphic DirectX 11.*
- HRESULT `msgProc (UINT msg, WPARAM wParam, LPARAM lParam)`  
*Sencarrega de processar els missatges que envia el sistema operatiu a la finestra.*

### 6.23.1 Descripció Detallada

Classe encarregada de gestionar tot el relacionat amb la finestra a nivell de Sistema Operatiu i conté tots els elements necessaris per utilitzar el framework.

*Cada finestra es considera com un escenari totalment independent.*

Alberto Lopez Sanchez i Ruben Bagan Benavides, 12/05/2013.

Definició a la línia 40 del fitxer Window.h.

### 6.23.2 Documentació del Constructor i el Destructor

#### 6.23.2.1 Dementia::Window::Window ( UINT id, UINT width, UINT height, std::wstring title, bool vsync, bool windowed )

Constructor. Per inicialitzar la finestra crida al mètode `init()`.

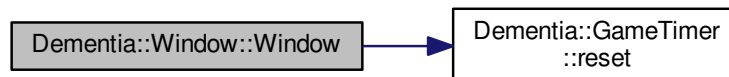
Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

**Paràmetres**

<i>id</i>	El identificador de la finestra. Ha de ser unic.
<i>width</i>	El ample.
<i>height</i>	La alcada.
<i>title</i>	El titol que veura l'usuari.
<i>vsync</i>	Indica si el refresc de la finestra estara sincronitzat amb el refresc del monitor.
<i>windowed</i>	Indica si la finestra es creara a pantalla completa o no.

Definició a la línia 32 del fitxer Window.cpp.

Gràfic de crides d'aquesta funció:



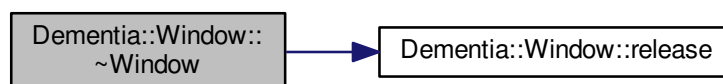
#### 6.23.2.2 Dementia::Window::~~Window ( void ) [virtual]

Destructor. Allibera tots els recursos automaticament.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

Definició a la línia 59 del fitxer Window.cpp.

Gràfic de crides d'aquesta funció:



### 6.23.3 Documentació de les Funcions Membre

#### 6.23.3.1 XMVECTOR Dementia::Window::getBackgroundColor ( )

Retorna el color de fons de la finestra.

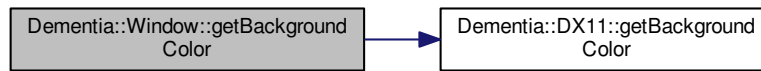
Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

**Retorna**

El color de fons.

Definició a la línia 278 del fitxer Window.cpp.

Gràfic de crides d'aquesta funció:



#### 6.23.3.2 Dementia::Camera \* Dementia::Window::getCamera ( )

Obte la camara.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

Retorna

El objecte [Camera](#).

Definició a la línia 288 del fitxer Window.cpp.

Gràfic de crides d'aquesta funció:



#### 6.23.3.3 ID3D11Device \* Dementia::Window::getDevice ( )

Obte el Dispositiu Graphic DirectX 11.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

Retorna

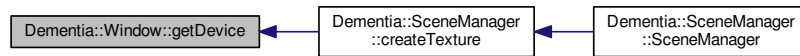
Retorna el objecte ID3D11Device de la implementacio grafica.

Definició a la línia 283 del fitxer Window.cpp.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



#### 6.23.3.4 Dementia::GameTimer\* Dementia::Window::getGameTimer ( ) [inline]

Obte el Game Timer.

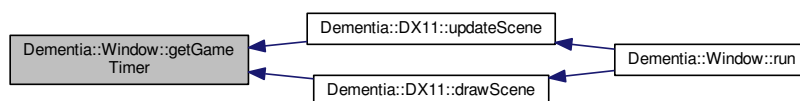
Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

**Retorna**

El objecte Game Timer.

Definició a la línia 277 del fitxer Window.h.

Gràfic de crides a aquesta funció:



#### 6.23.3.5 UINT Dementia::Window::getHeight ( ) [inline]

Obte la alcada de la finestra.

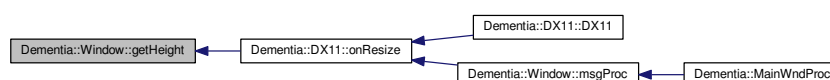
Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

**Retorna**

La alcada de la finestra.

Definició a la línia 217 del fitxer Window.h.

Gràfic de crides a aquesta funció:





**6.23.3.6** `UINT Dementia::Window::getId ( ) [inline]`

Obte l'identificador de la finestra.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

**Retorna**

L'identificador de la finestra.

Definició a la línia 237 del fitxer Window.h.

**6.23.3.7** `Dementia::InputManager* Dementia::Window::getInputManager ( ) [inline]`

Obte el Input Manager.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

**Retorna**

El objecte Input Manager.

Definició a la línia 267 del fitxer Window.h.

**6.23.3.8** `Dementia::SceneManager * Dementia::Window::getSceneManager ( )`

Obte el "Scene Manager"

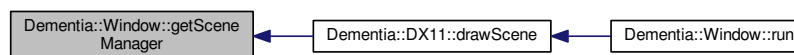
Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

**Retorna**

El objecte Scene Manager.

Definició a la línia 293 del fitxer Window.cpp.

Gràfic de crides a aquesta funció:

**6.23.3.9** `std::wstring Dementia::Window::getTitle ( ) [inline]`

Obte el títol de la finestra.

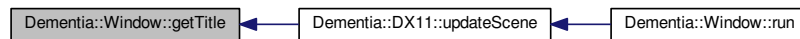
Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

**Retorna**

El títol de la finestra.

Definició a la línia 247 del fitxer Window.h.

Gràfic de crides a aquesta funció:



### 6.23.3.10 UINT Dementia::Window::getWidth ( ) [inline]

Obté la amplada de la finestra.

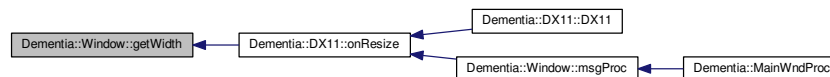
Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

**Retorna**

La amplada de la finestra.

Definició a la línia 227 del fitxer Window.h.

Gràfic de crides a aquesta funció:



### 6.23.3.11 HWND Dementia::Window::getWindowHandle ( ) [inline]

Obté el "handle" de la finestra.

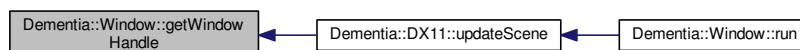
Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

**Retorna**

El "handle" de la finestra.

Definició a la línia 257 del fitxer Window.h.

Gràfic de crides a aquesta funció:



**6.23.3.12 void Dementia::Window::init ( )**

Inicialitza la finestra i tots els seus components. La crea i la mostra per pantalla. També inicialitza la implementació gràfica (DirectX o OpenGL).

Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

Definició a la línia 50 del fitxer Window.cpp.

**6.23.3.13 bool Dementia::Window::isPaused ( ) [inline]**

Pregunta si la finestra està pausada. La finestra es pausa automàticament quan perd el focus i es torna a reanudar quan torna a tenir el focus.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

**Retorna**

true si està pausada. false si no ho està.

Definició a la línia 207 del fitxer Window.h.

**6.23.3.14 bool Dementia::Window::isVsync ( ) [inline]**

Pregunta si la sincronització vertical està activada.

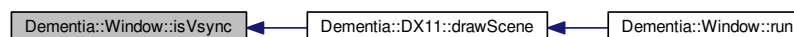
Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

**Retorna**

true si la sincronització vertical està activada, false si no ho està.

Definició a la línia 196 del fitxer Window.h.

Gràfic de crides a aquesta funció:

**6.23.3.15 bool Dementia::Window::isWindowed ( ) [inline]**

Pregunta si la finestra està en mode finestra o a pantalla completa.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

**Retorna**

true si està en mode finestra, false si està a pantalla completa.

Definició a la línia 186 del fitxer Window.h.

**6.23.3.16 LRESULT Dementia::Window::msgProc ( UINT msg, WPARAM wParam, LPARAM lParam )**

Sencarrega de processar els missatges que envia el sistema operatiu a la finestra.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

## Paràmetres

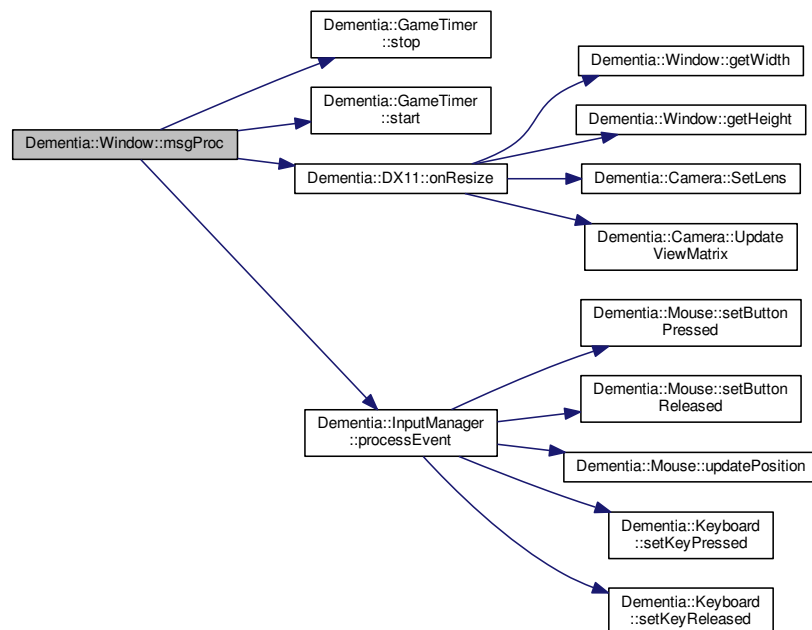
<i>msg</i>	El message.
<i>wParam</i>	El camp wParam del missatge.
<i>lParam</i>	El camp lParam del missatge.

## Retorna

El resultat del procesament del missatge.

Definició a la línia 150 del fitxer Window.cpp.

Gràfic de crides d'aquesta funció:



Gràfic de crides a aquesta funció:



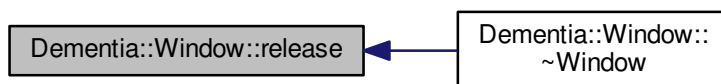
## 6.23.3.17 void Dementia::Window::release ( void )

Allibera tots els recursos de la finestra. no cal cridar-lo. Es crida automaticament quan es destrueix la finestra.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

Definició a la línia 64 del fitxer Window.cpp.

Gràfic de crides a aquesta funció:



#### 6.23.3.18 `bool Dementia::Window::run ( int * exitCode )`

Executa el bucle principal de la finestra. S'ha de fer en bucle per mantenir la execució.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

##### Paràmetres

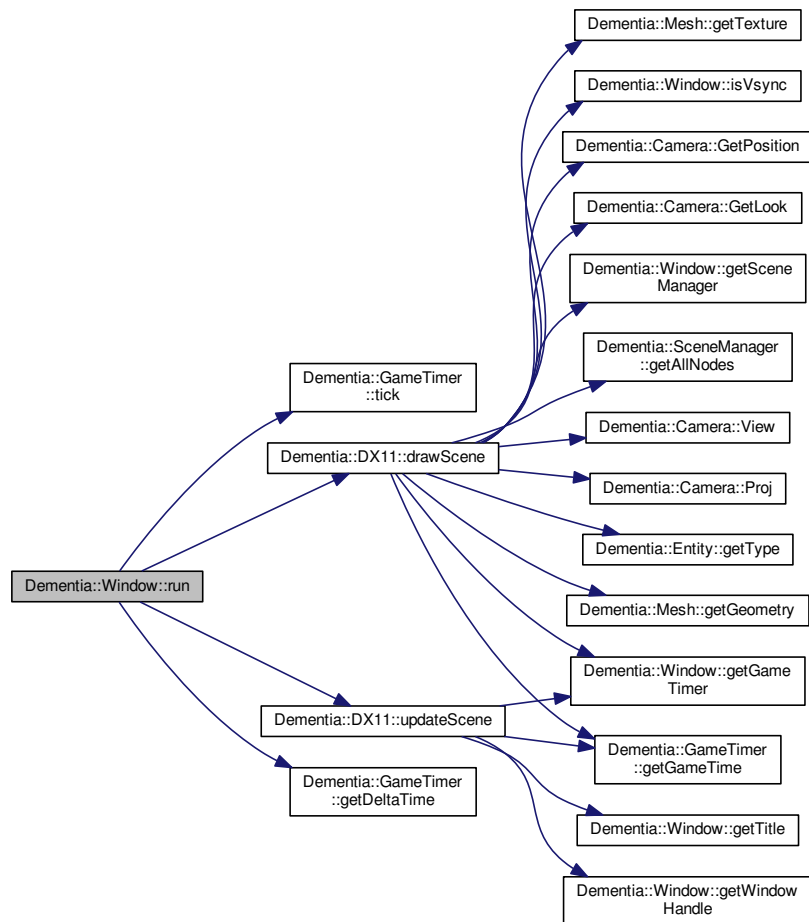
<i>exitCode</i>	[out] Retorna el codi de sortida de la finestra. Tan si ha terminat correctament com si no.
-----------------	---

##### Retorna

true si es continua executant. false si s'ha acabat la execució.

Definició a la línia 113 del fitxer Window.cpp.

Gràfic de crides d'aquesta funció:



#### 6.23.3.19 void Dementia::Window::setBackgroundColor ( XMVECTOR *backgroundColor* )

Cambia el color de fons de la finestra.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

##### Paràmetres

<i>backgroundColor</i>	El color de fons.
------------------------	-------------------

Definició a la línia 273 del fitxer Window.cpp.

Gràfic de crides d'aquesta funció:



6.23.3.20 void Dementia::Window::setHeight ( UINT *height* ) [inline]

Cambia la altura de la finestra.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

Paràmetres

<i>height</i>	La nova altura.
---------------	-----------------

Definició a la línia 136 del fitxer Window.h.

6.23.3.21 void Dementia::Window::setTitle ( std::wstring *title* ) [inline]

Cambia el titol de la finestra.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

Paràmetres

<i>title</i>	El nou titol.
--------------	---------------

Definició a la línia 156 del fitxer Window.h.

6.23.3.22 void Dementia::Window::setVsync ( bool *state* ) [inline]

Cambia l'estat de la sincronitzacio vertical.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

Paràmetres

<i>state</i>	true activa la sincronitzacio vertical, false la desactiva.
--------------	---

Definició a la línia 166 del fitxer Window.h.

6.23.3.23 void Dementia::Window::setWidth ( UINT *width* ) [inline]

Cambia l'ample de la finestra.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

Paràmetres

<i>width</i>	El nou ample.
--------------	---------------

Definició a la línia 146 del fitxer Window.h.

6.23.3.24 void Dementia::Window::setWindowed ( bool *state* ) [inline]

Posa la finestra a mode finestra o a pantalla completa.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

Paràmetres

<i>state</i>	true per mode finestra. false per mode a pantalla completa
--------------	--

Definició a la línia 126 del fitxer Window.h.

La documentació d'aquesta classe es va generar a partir dels següents fitxers:

- Include/[Window.h](#)
- src/[Window.cpp](#)





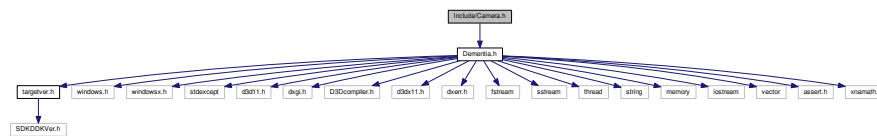
## Capítol 7

# Documentació dels Fitxers

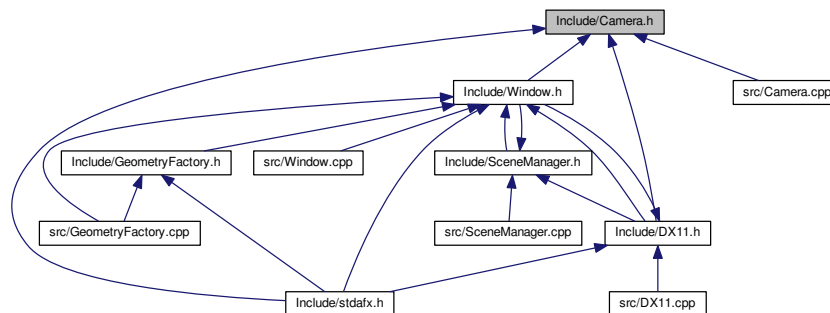
### 7.1 Referència del Fitxer Include/Camera.h

```
#include "Dementia.h"
```

Inclou el graf de dependències per a Camera.h:



Aquest gràfic mostra quins fitxers inclouen, de forma directa o indirecta, aquest fitxer:



### Classes

- class `Dementia::Camera`

*Una camara simple pensada per ser utilitzada com a Primera persona, per explorar la scena 3D.*

### Espais de Noms

- `Dementia`

*El espai de nom principal del framework. Totes les classes basiques o importants es troben aqui dins.*

## Constant Groups

- [Dementia](#)

*El espai de nom principal del framework. Totes les classes basiques o importants es troben aqui dins.*

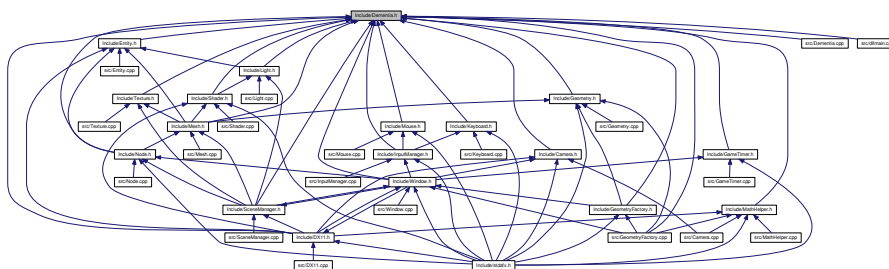
## 7.2 Referència del Fitxer Include/Dementia.h

```
#include "targetver.h"
#include <windows.h>
#include <windowsx.h>
#include <stdexcept>
#include <d3d11.h>
#include <dxgi.h>
#include <D3Dcompiler.h>
#include <d3dx11.h>
#include <dxerr.h>
#include <fstream>
#include <sstream>
#include <thread>
#include <string>
#include <memory>
#include <iostream>
#include <vector>
#include <assert.h>
#include <xnamath.h>
```

Inclou el graf de dependències per a Dementia.h:



Aquest gràfic mostra quins fitxers inclouen, de forma directa o indirecta, aquest fitxer:



## Definicions

- `#define UTILS_EXPORT`  
*Definiciones globales del motor.*
- `#define WIN32_LEAN_AND_MEAN`
- `#define UTILS_API __declspec(dllexport)`
- `#define HR(x) (x)`
- `#define ReleaseCOM(x) { if(x){ x->Release();x = 0; } }`

Macro per a alliberar els Objectes de tipus COM. \*\*\*\*\*

## Funcions

- void [showErrorMessage](#) (DWORD errorCode)

Funcio que es crida quan es produeix un error i tel mostra en l'idioma que te configurat l'usuari.

### 7.2.1 Documentació de les Definicions

#### 7.2.1.1 #define HR( x )(x)

Definició a la línia 65 del fitxer Dementia.h.

#### 7.2.1.2 #define ReleaseCOM( x ) { if(x){ x->Release();x = 0; } }

Macro per a alliberar els Objectes de tipus COM. \*\*\*\*\*

Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

##### Paràmetres

x	The void to process.
---	----------------------

Definició a la línia 81 del fitxer Dementia.h.

#### 7.2.1.3 #define UTILS\_API \_\_declspec(dllexport)

Definició a la línia 42 del fitxer Dementia.h.

#### 7.2.1.4 #define UTILS\_EXPORT

Definiciones globales del motor.

Definició a la línia 5 del fitxer Dementia.h.

#### 7.2.1.5 #define WIN32\_LEAN\_AND\_MEAN

Definició a la línia 9 del fitxer Dementia.h.

### 7.2.2 Documentació de les Funcions

#### 7.2.2.1 void showErrorMessage ( DWORD errorCode )

Funcio que es crida quan es produeix un error i tel mostra en l'idioma que te configurat l'usuari.

Alberto Lopez Sanchez i Ruben Bagan Benavides, 28/05/2013.

##### Paràmetres

errorCode	The error code.
-----------	-----------------

Definició a la línia 6 del fitxer Dementia.cpp.

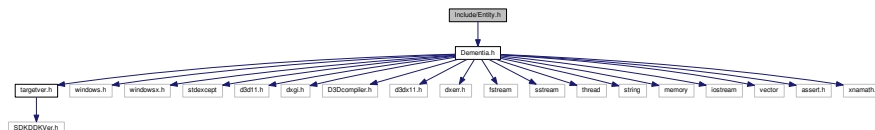


*El espai de nom principal del framework. Totes les classes basiques o importants es troben aqui dins.*

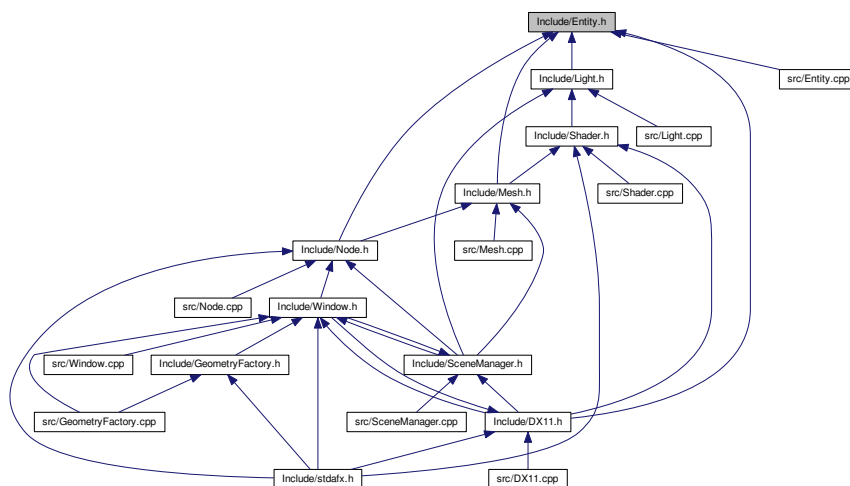
## 7.4 Referència del Fitxer Include/Entity.h

```
#include "Dementia.h"
```

Inclou el graf de dependències per a Entity.h:



Aquest gràfic mostra quins fitxers inclouen, de forma directa o indirecta, aquest fitxer:



## Classes

- class **Dementia::Entity**

*Clase base per a qualsevol altre classe que hagi de interactuar amb un node i el "Scene Manager"*

## Espais de Noms

- Dementia

*El espai de nom principal del framework. Totes les classes basiques o importants es troben aqui dins.*

## Constant Groups

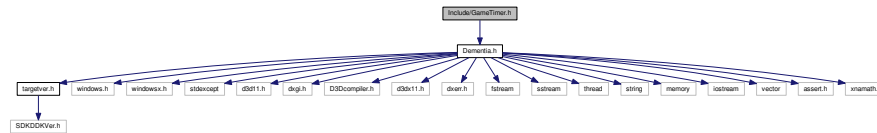
- Dementia

*El espai de nom principal del framework. Totes les classes basiques o importants es troben aqui dins.*

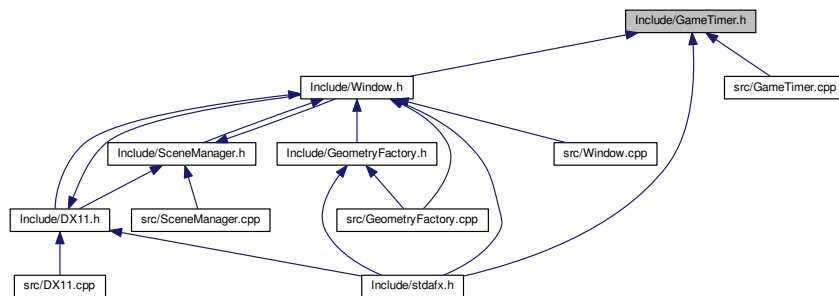
## 7.5 Referència del Fitxer Include/GameTimer.h

```
#include "Dementia.h"
```

Inclou el graf de dependències per a GameTimer.h:



Aquest gràfic mostra quins fitxers inclouen, de forma directa o indirecta, aquest fitxer:



## Classes

- class [Dementia::GameTimer](#)

*S'encarrega de controlar tot el que te a veure amb el temps.*

## Espais de Noms

- [Dementia](#)

*El espai de nom principal del framework. Totes les classes basiques o importants es troben aqui dins.*

## Constant Groups

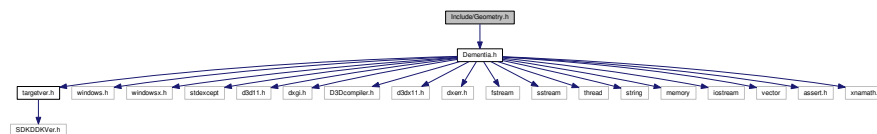
- [Dementia](#)

*El espai de nom principal del framework. Totes les classes basiques o importants es troben aqui dins.*

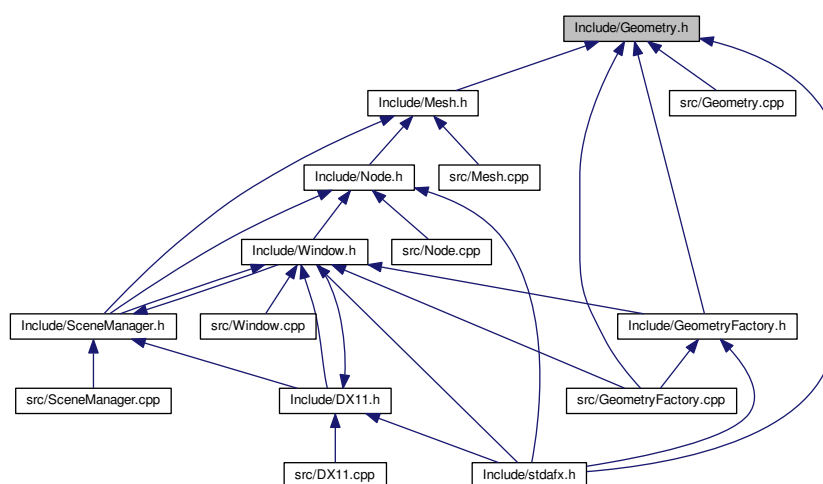
## 7.6 Referència del Fitxer Include/Geometry.h

```
#include "Dementia.h"
```

Inclou el graf de dependències per a Geometry.h:



Aquest gràfic mostra quins fitxers inclouen, de forma directa o indirecta, aquest fitxer:



## Classes

- class `Dementia::Geometry`  
*Entenem per geometria el conjunt de vertex i index que calen per formar una figura geometrica.*
- struct `Dementia::Geometry::Vertex`  
*Conte tots els elements necessaris per fer tots els calculs per pantalla (llums, etc).*

## Espais de Noms

- **Dementia**
- El espai de nom principal del framework. Totes les classes basiques o importants es troben aqui dins.*

## Constant Groups

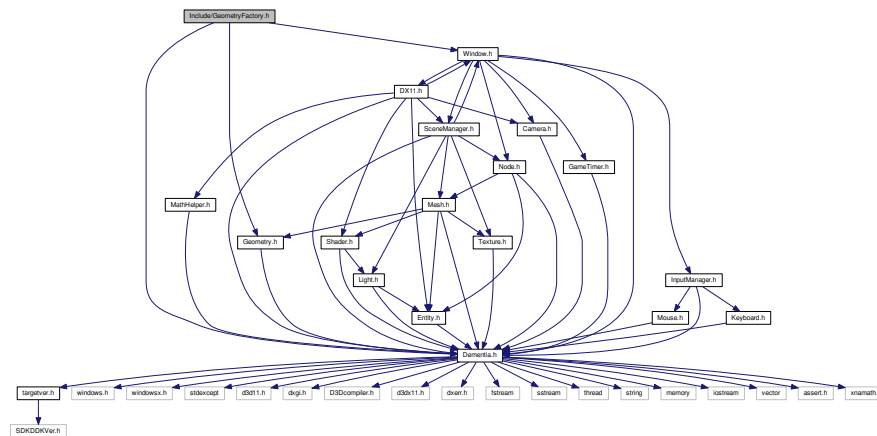
- **Dementia**
- El espai de nom principal del framework. Totes les classes basiques o importants es troben aqui dins.*

## 7.7 Referència del Fitxer Include/GeometryFactory.h

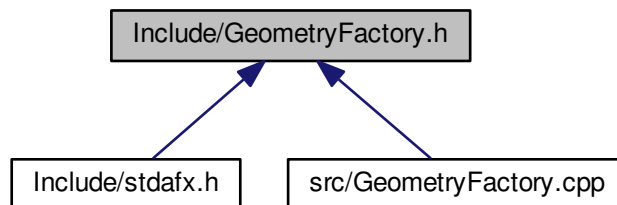
```
#include "Dementia.h"
```

```
#include "Geometry.h"
#include "Window.h"
```

Inclou el graf de dependències per a GeometryFactory.h:



Aquest gràfic mostra quins fitxers inclouen, de forma directa o indirecta, aquest fitxer:



## Classes

- class [Dementia::GeometryFactory](#)

## Espais de Noms

- [Dementia](#)

*El espai de nom principal del framework. Totes les classes basiques o importants es troben aqui dins.*

## Constant Groups

- [Dementia](#)

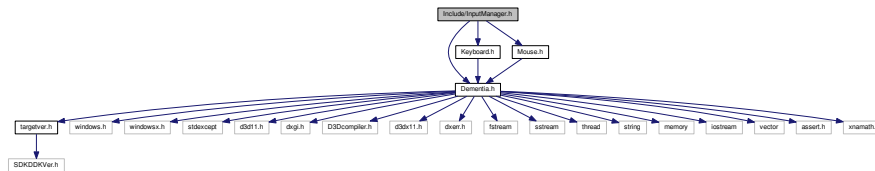
*El espai de nom principal del framework. Totes les classes basiques o importants es troben aqui dins.*



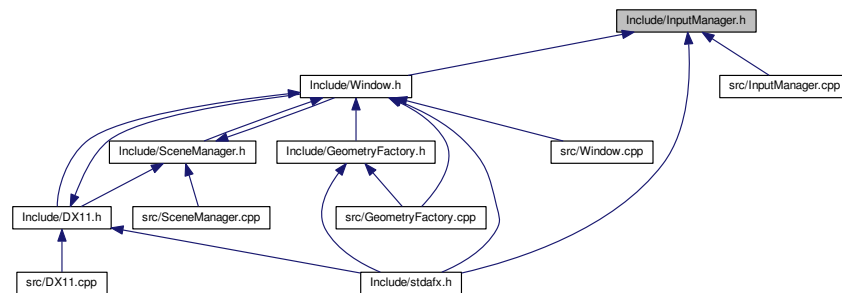
## 7.8 Referència del Fitxer Include/InputManager.h

```
#include "Dementia.h"
#include "Keyboard.h"
#include "Mouse.h"
```

Inclou el graf de dependències per a InputManager.h:



Aquest gràfic mostra quins fitxers inclouen, de forma directa o indirecta, aquest fitxer:



## Classes

- class `Dementia::InputManager`

*Clase encarregada de gestionar tots els dispositius de entrada entre l'usuari i el sistema operatiu*

## Espais de Noms

- `Dementia`

*El espai de nom principal del framework. Totes les classes basiques o importants es troben aqui dins.*

## Constant Groups

- `Dementia`

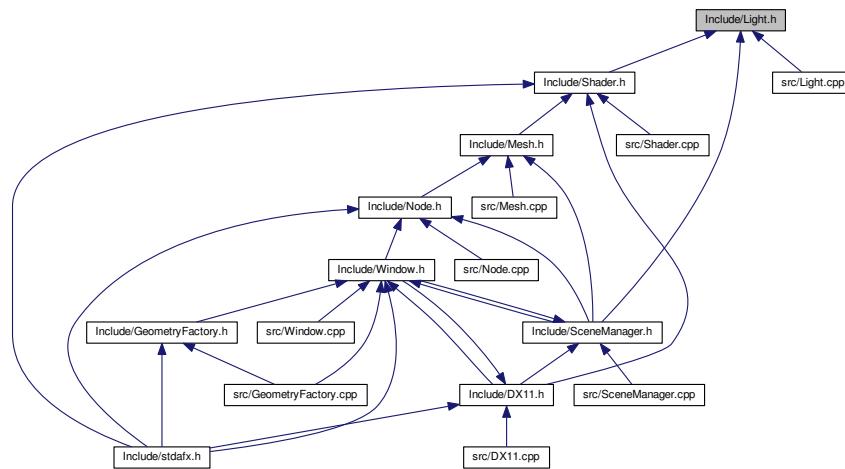
*El espai de nom principal del framework. Totes les classes basiques o importants es troben aqui dins.*

## 7.9 Referència del Fitxer Include/Keyboard.h

```
#include "Dementia.h"
```



Aquest gràfic mostra quins fitxers inclouen, de forma directa o indirecta, aquest fitxer:



## Classes

- class [Dementia::Light](#)  
*La classe base de la llum.*
- struct [Dementia::DirectionalLight](#)  
*Una llum que simular ser de tipus ambient, tots els objectes al scenari es veuen afectats per aquesta llum.*
- struct [Dementia::PointLight](#)  
*Una llum de tipus punt, que irradia a tots els objectes que estiguin dins del seu radi d'acció per igual.*
- struct [Dementia::SpotLight](#)  
*Una llum de tipus llinterna o focus, il·lumina només als objectes en els que la direcció apunti i estiguin dins del seu radi.*
- struct [Dementia::Material](#)  
*Entenem per [Material](#) una estructura de dades on indiquem quin tipus de llum reflexa (color i intensitat) els objectes que tenen aquest material.*

## Espais de Noms

- [Dementia](#)  
*El espai de nom principal del framework. Totes les classes bàsiques o importants es troben aquí dins.*

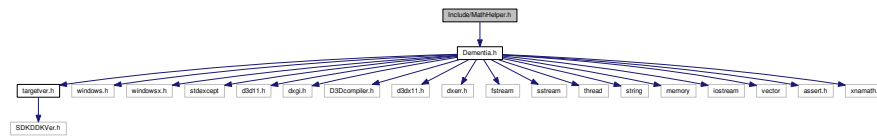
## Constant Groups

- [Dementia](#)  
*El espai de nom principal del framework. Totes les classes bàsiques o importants es troben aquí dins.*

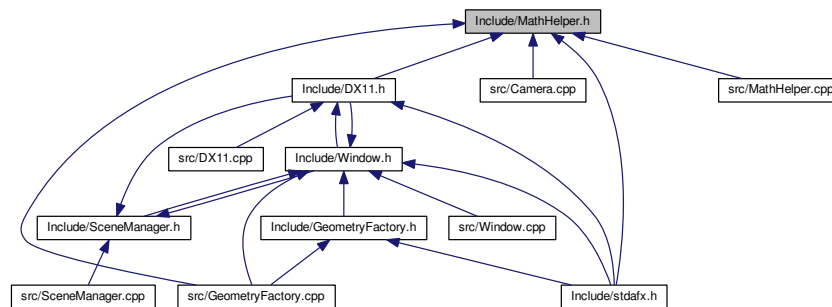
## 7.11 Referència del Fitxer Include/MathHelper.h

```
#include "Dementia.h"
```

Inclou el graf de dependències per a MathHelper.h:



Aquest gràfic mostra quins fitxers inclouen, de forma directa o indirecta, aquest fitxer:



## Classes

- class [Dementia::MathHelper](#)

## Espais de Noms

- [Dementia](#)

*El espai de nom principal del framework. Totes les classes basiques o importants es troben aqui dins.*

## Constant Groups

- [Dementia](#)

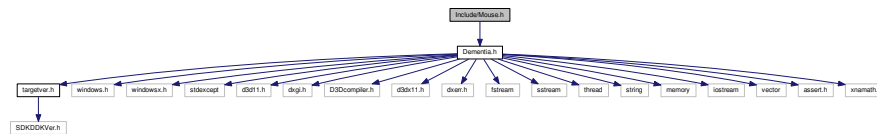
*El espai de nom principal del framework. Totes les classes basiques o importants es troben aqui dins.*

## 7.12 Referència del Fitxer Include/Mesh.h

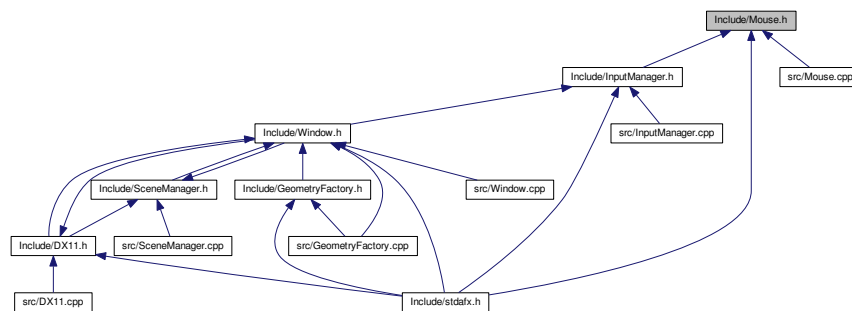
```
#include "Dementia.h"
#include "Texture.h"
#include "Shader.h"
#include "Geometry.h"
#include "Entity.h"
```



Inclou el graf de dependències per a Mouse.h:



Aquest gràfic mostra quins fitxers inclouen, de forma directa o indirecta, aquest fitxer:



## Classes

- class [Dementia::Mouse](#)

*Classe encarregada del ratolí que fa de intermediari entre l'usuari i el Sistema Operatiu*

## Espais de Noms

- [Dementia](#)

*El espai de nom principal del framework. Totes les classes bàsiques o importants es troben aquí dins.*

## Constant Groups

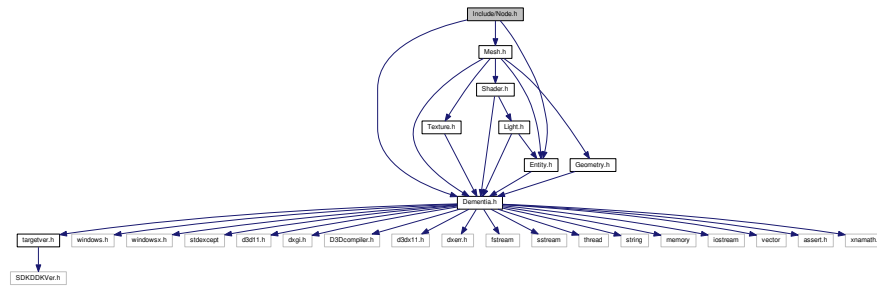
- [Dementia](#)

*El espai de nom principal del framework. Totes les classes bàsiques o importants es troben aquí dins.*

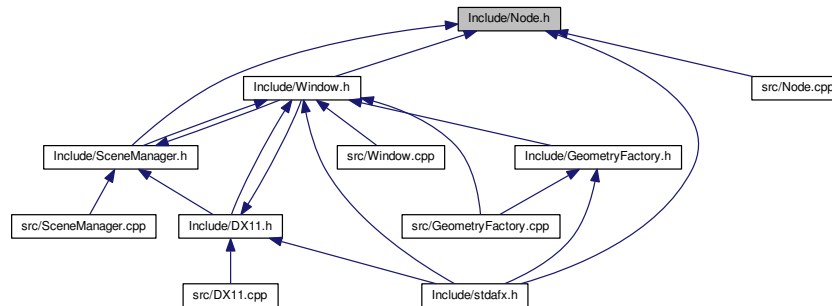
## 7.14 Referència del Fitxer Include/Node.h

```
#include "Dementia.h"
#include "Mesh.h"
#include "Entity.h"
```

Inclou el graf de dependències per a Node.h:



Aquest gràfic mostra quins fitxers inclouen, de forma directa o indirecta, aquest fitxer:



## Classes

- class [Dementia::Node](#)

*Un node es un objecte invisible que se encarrega de sapiguer on, amb quina escala, i amb quina rotació s'ha de renderitzar qualsevol entitat.*

## Espais de Noms

- [Dementia](#)

*El espai de nom principal del framework. Totes les classes bàsiques o importants es troben aquí dins.*

## Constant Groups

- [Dementia](#)

*El espai de nom principal del framework. Totes les classes bàsiques o importants es troben aquí dins.*

## 7.15 Referència del Fitxer Include/SceneManager.h

```
#include "Dementia.h"
```



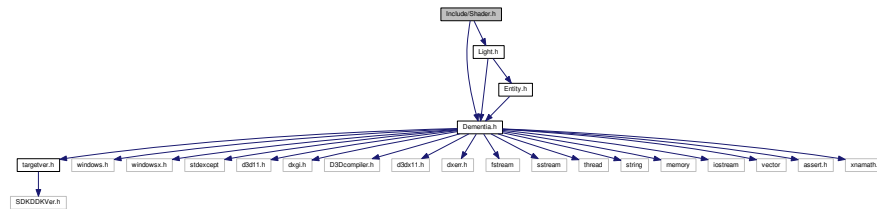


## 7.16 Referència del Fitxer Include/Shader.h

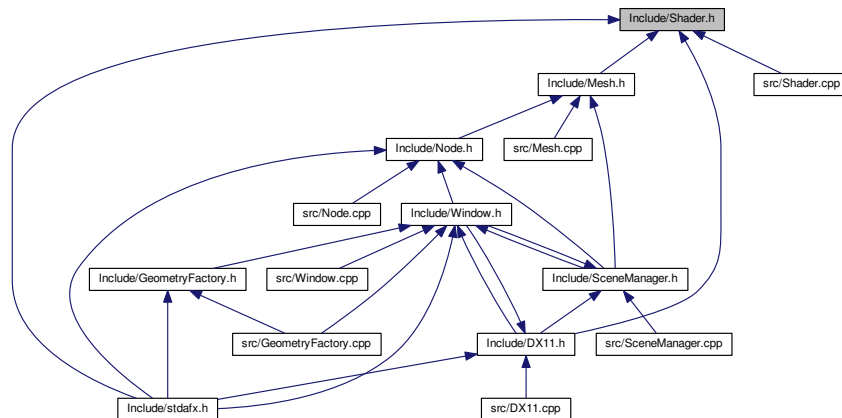
```
#include "Dementia.h"
```

```
#include "Light.h"
```

Inclou el graf de dependències per a Shader.h:



Aquest gràfic mostra quins fitxers inclouen, de forma directa o indirecta, aquest fitxer:



## Classes

- class [Dementia::Shader](#)

*Clase que conte tots elements necessaris per treballar amb els shaders.*

## Espais de Noms

- [Dementia](#)

*El espai de nom principal del framework. Totes les classes basiques o importants es troben aqui dins.*

## Constant Groups

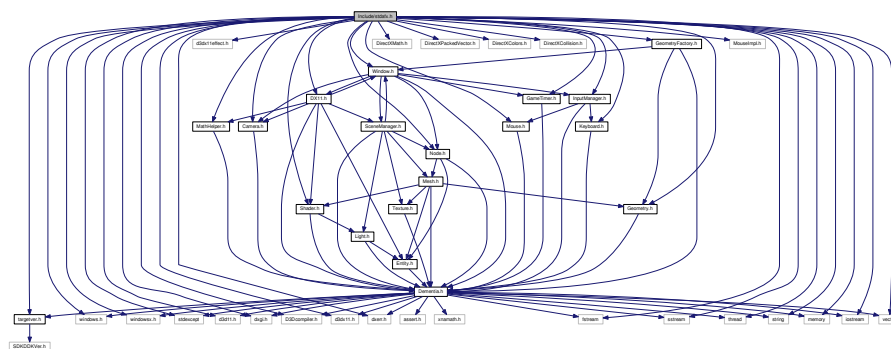
- [Dementia](#)

*El espai de nom principal del framework. Totes les classes basiques o importants es troben aqui dins.*

## 7.17 Referència del Fitxer Include/stdafx.h

```
#include "targetver.h"
#include <windows.h>
#include <windowsx.h>
#include <stdexcept>
#include <d3d11.h>
#include <dxgi.h>
#include <D3Dcompiler.h>
#include <d3dx11.h>
#include <dxerr.h>
#include "d3dx11effect.h"
#include <fstream>
#include <sstream>
#include <thread>
#include <string>
#include <memory>
#include <iostream>
#include <vector>
#include <DirectXMath.h>
#include <DirectXPackedVector.h>
#include <DirectXColors.h>
#include <DirectXCollision.h>
#include "GameTimer.h"
#include "Mouse.h"
#include "Keyboard.h"
#include "Node.h"
#include "InputManager.h"
#include "MouseImpl.h"
#include "Geometry.h"
#include "Camera.h"
#include "GeometryFactory.h"
#include "MathHelper.h"
#include "Shader.h"
#include "Window.h"
#include "DX11.h"
```

Inclou el graf de dependències per a stdafx.h:



### Definicions

- `#define UTILS_EXPORT`
- `#define WIN32_LEAN_AND_MEAN`

### 7.17.1 Documentació de les Definicions

#### 7.17.1.1 #define UTILS\_EXPORT

Definició a la línia 8 del fitxer stdafx.h.

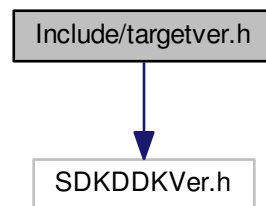
#### 7.17.1.2 #define WIN32\_LEAN\_AND\_MEAN

Definició a la línia 12 del fitxer stdafx.h.

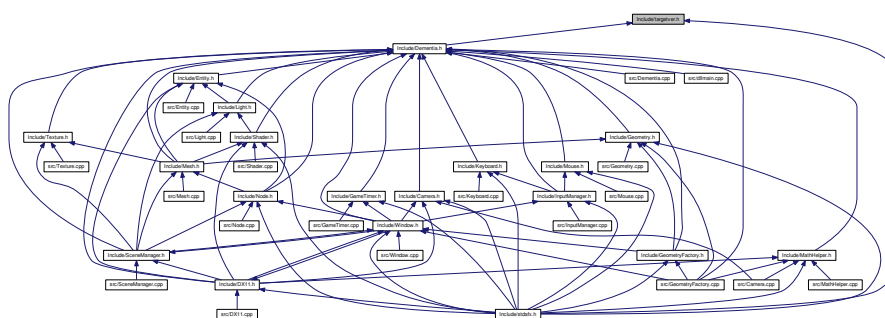
## 7.18 Referència del Fitxer Include/targetver.h

```
#include <SDKDDKVer.h>
```

Inclou el graf de dependències per a targetver.h:



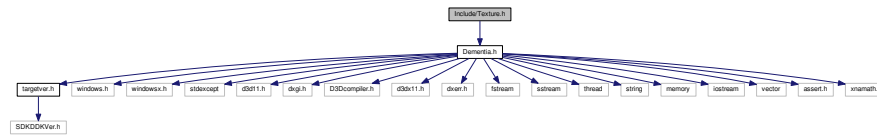
Aquest gràfic mostra quins fitxers inclouen, de forma directa o indirecta, aquest fitxer:



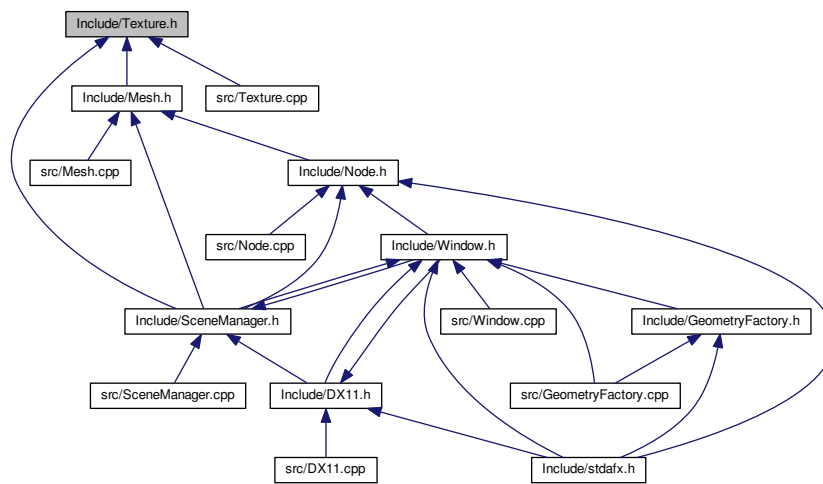
## 7.19 Referència del Fitxer Include/Texture.h

```
#include "Dementia.h"
```

Inclou el graf de dependències per a Texture.h:



Aquest gràfic mostra quins fitxers inclouen, de forma directa o indirecta, aquest fitxer:



## Classes

- class [Dementia::Texture](#)

*Clase que conte tot el necessari per dibuixar les textures per pantalla.*

## Espais de Noms

- [Dementia](#)

*El espai de nom principal del framework. Totes les classes basiques o importants es troben aqui dins.*

## Constant Groups

- [Dementia](#)

*El espai de nom principal del framework. Totes les classes basiques o importants es troben aqui dins.*

## 7.20 Referència del Fitxer Include/Window.h

```
#include "Dementia.h"
```





## Paràmetres

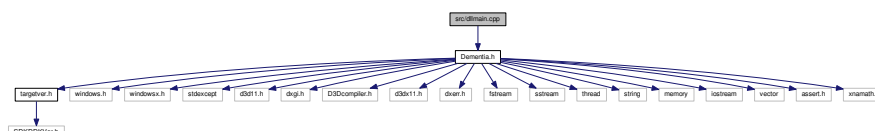
<i>errorCode</i>	The error code.
------------------	-----------------

Definició a la línia 6 del fitxer Dementia.cpp.

### 7.23 Referència del Fitxer src/dllmain.cpp

```
#include "Dementia.h"
```

Inclou el graf de dependències per a dllmain.cpp:



## Funcions

- BOOL WINAPI **DllMain** (HMODULE hModule, DWORD ul\_reason\_for\_call, LPVOID lpReserved)

### 7.23.1 Documentació de les Funcions

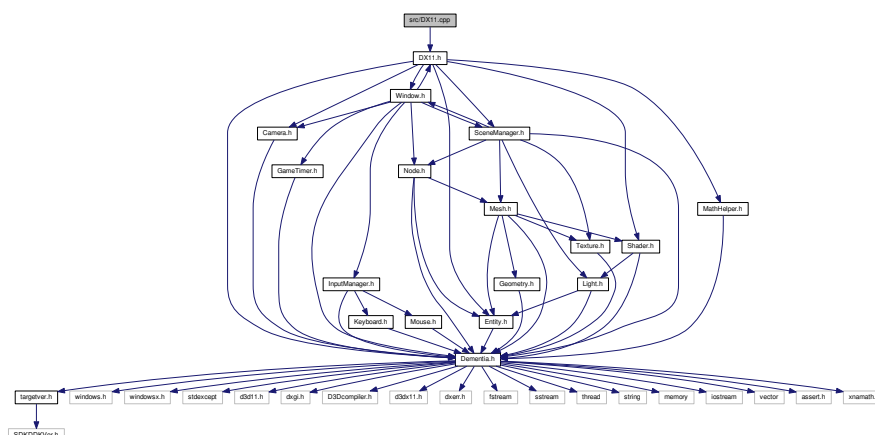
#### 7.23.1.1 BOOL WINAPI DllMain ( HMODULE *hModule*, DWORD *ul\_reason\_for\_call*, LPVOID *lpReserved* )

Definició a la línia 4 del fitxer dllmain.cpp.

## 7.24 Referència del Fitxer src/DX11.cpp

```
#include "DX11.h"
```

Inclou el graf de dependències per a DX11.cpp:



## Espais de Noms

- Dementia





## Espais de Noms

- [Dementia](#)

*El espai de nom principal del framework. Totes les classes basiques o importants es troben aqui dins.*

## Constant Groups

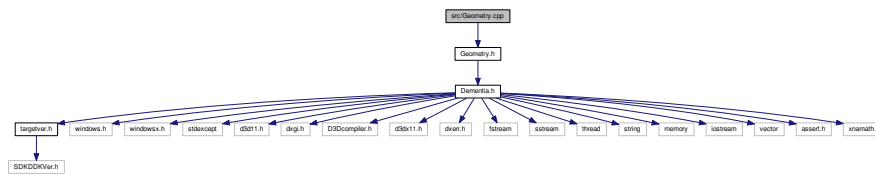
- [Dementia](#)

*El espai de nom principal del framework. Totes les classes basiques o importants es troben aqui dins.*

## 7.27 Referència del Fitxer src/Geometry.cpp

```
#include "Geometry.h"
```

Inclou el graf de dependències per a Geometry.cpp:



## Espais de Noms

- [Dementia](#)

*El espai de nom principal del framework. Totes les classes basiques o importants es troben aqui dins.*

## Constant Groups

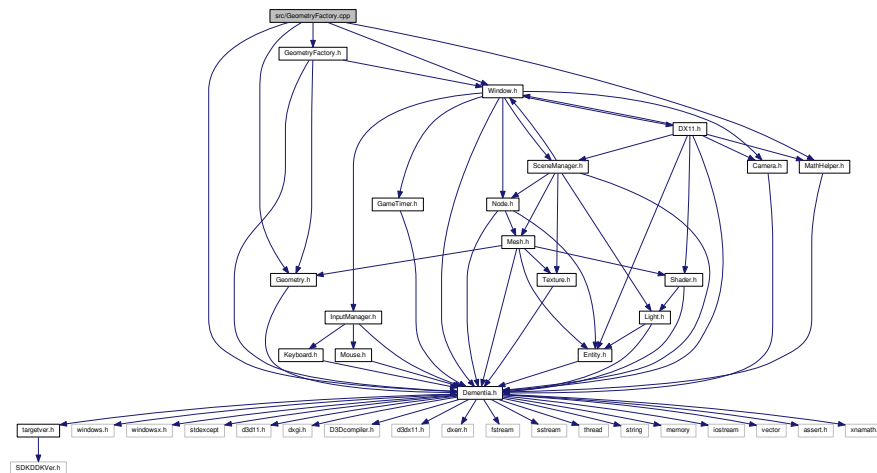
- [Dementia](#)

*El espai de nom principal del framework. Totes les classes basiques o importants es troben aqui dins.*

## 7.28 Referència del Fitxer src/GeometryFactory.cpp

```
#include "Dementia.h"
#include "GeometryFactory.h"
#include "Window.h"
#include "Geometry.h"
#include "MathHelper.h"
```

Inclou el graf de dependències per a GeometryFactory.cpp:



## Espais de Noms

- [Dementia](#)

*El espai de nom principal del framework. Totes les classes basiques o importants es troben aqui dins.*

## Constant Groups

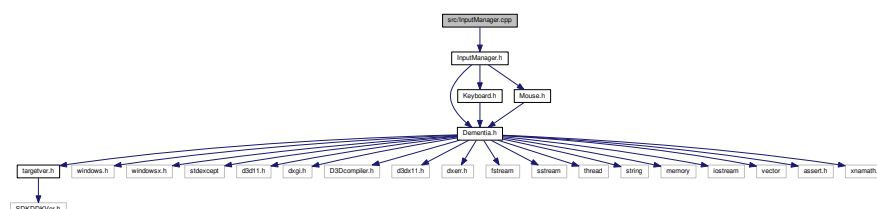
- [Dementia](#)

*El espai de nom principal del framework. Totes les classes basiques o importants es troben aqui dins.*

## 7.29 Referència del Fitxer src/InputManager.cpp

```
#include "InputManager.h"
```

Inclou el graf de dependències per a InputManager.cpp:



## Espais de Noms

- [Dementia](#)

*El espai de nom principal del framework. Totes les classes basiques o importants es troben aqui dins.*

## Constant Groups

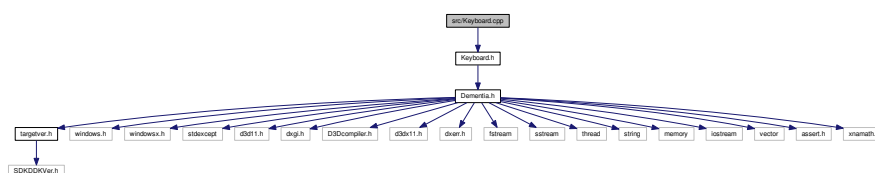
- Dementia

*El espai de nom principal del framework. Totes les classes basiques o importants es troben aqui dins.*

### 7.30 Referència del Fitxer src/Keyboard.cpp

```
#include "Keyboard.h"
```

Inclou el graf de dependències per a Keyboard.cpp:



## Esais de Noms

- Dementia

*El espai de nom principal del framework. Totes les classes basiques o importants es troben aqui dins.*

## Constant Groups

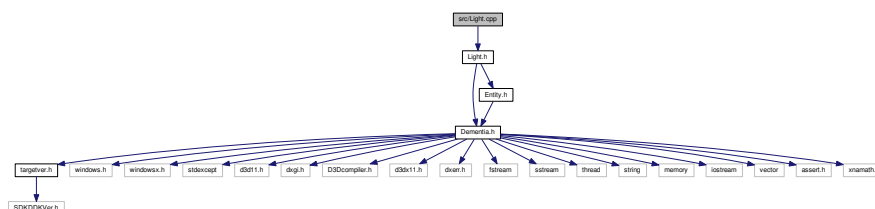
- Dementia

*El espai de nom principal del framework. Totes les classes basiques o importants es troben aqui dins.*

### 7.31 Referència del Fitxer src/Light.cpp

```
#include "Light.h"
```

Inclou el graf de dependències per a Light.cpp:



## Espais de Noms

- Dementia

*El espai de nom principal del framework. Totes les classes basiques o importants es troben aqui dins.*



## Constant Groups

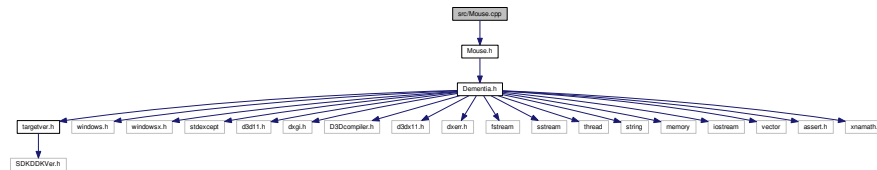
- [Dementia](#)

*El espai de nom principal del framework. Totes les classes basiques o importants es troben aqui dins.*

## 7.34 Referència del Fitxer src/Mouse.cpp

```
#include "Mouse.h"
```

Inclou el graf de dependències per a Mouse.cpp:



## Espais de Noms

- [Dementia](#)

*El espai de nom principal del framework. Totes les classes basiques o importants es troben aqui dins.*

## Constant Groups

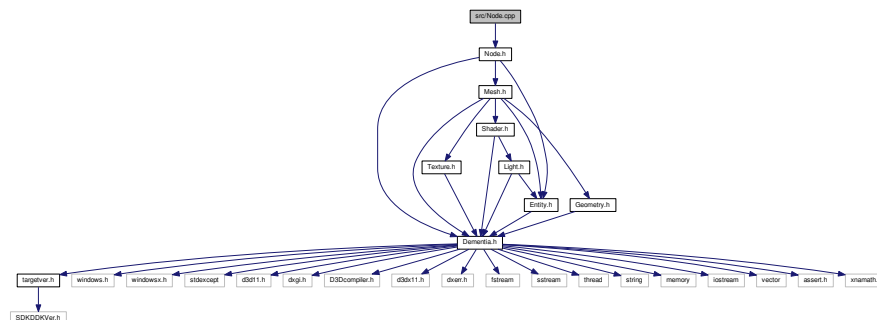
- [Dementia](#)

*El espai de nom principal del framework. Totes les classes basiques o importants es troben aqui dins.*

## 7.35 Referència del Fitxer src/Node.cpp

```
#include "Node.h"
```

Inclou el graf de dependències per a Node.cpp:



## Espais de Noms

- [Dementia](#)

*El espai de nom principal del framework. Totes les classes basiques o importants es troben aqui dins.*



## Espais de Noms

- [Dementia](#)

*El espai de nom principal del framework. Totes les classes basiques o importants es troben aqui dins.*

## Constant Groups

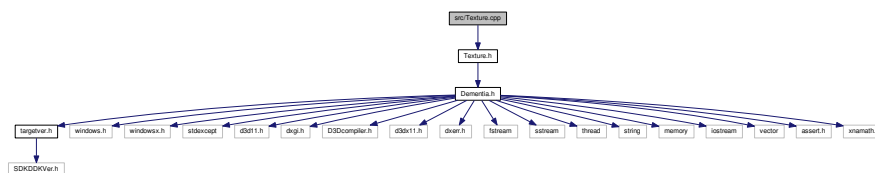
- [Dementia](#)

*El espai de nom principal del framework. Totes les classes basiques o importants es troben aqui dins.*

## 7.38 Referència del Fitxer src/Texture.cpp

```
#include "Texture.h"
```

Inclou el graf de dependències per a Texture.cpp:



## Espais de Noms

- [Dementia](#)

*El espai de nom principal del framework. Totes les classes basiques o importants es troben aqui dins.*

## Constant Groups

- [Dementia](#)

*El espai de nom principal del framework. Totes les classes basiques o importants es troben aqui dins.*





# Índex alfabètic

Button4

Dementia::Mouse, [59](#)

Button5

Dementia::Mouse, [59](#)

COUNT

Dementia::Mouse, [59](#)

DIRECTIONAL\_LIGHT

Dementia::Light, [45](#)

Dementia, [9](#)

Dementia::Entity

Light, [31](#)

Mesh, [31](#)

Dementia::Light

DIRECTIONAL\_LIGHT, [45](#)

POINT\_LIGHT, [45](#)

SPOT\_LIGHT, [45](#)

Dementia::Mouse

Button4, [59](#)

Button5, [59](#)

COUNT, [59](#)

Left, [59](#)

Middle, [59](#)

Right, [59](#)

Left

Dementia::Mouse, [59](#)

Light

Dementia::Entity, [31](#)

Mesh

Dementia::Entity, [31](#)

Middle

Dementia::Mouse, [59](#)

POINT\_LIGHT

Dementia::Light, [45](#)

Right

Dementia::Mouse, [59](#)

SPOT\_LIGHT

Dementia::Light, [45](#)