VI50-2010A-FSOL-01

VI50 : Vision et Réalité Virtuelle
Examen Final
Semestre Automne 2010

Durée : 2h, Aucun document, ni calculatrice, ni SmartPhone autorisé
Chaque partie devra être rendue sur des feuilles séparées

PARTIE 1 : Physique (5 points)

Question 1.1 :

Pourquoi les écrans à plasma ont-il un meilleur contraste que les écran LCD?

Exercice 1.2 :

Un pendule est constitué d'un fil idéal de longueur l, fixé en O et auquel est suspendu un objet ponctuel M de masse m. Il oscille dans le plan et sa position est repérée par θ . (cf. Figure 1)

1) Que représentent e_r , e_θ , $\vec{T}$ et $\vec{P}$?

2) Quelle est l'équation différentielle vérifiée par θ ?
Indications: Il y a deux méthode possibles: 2ème loi de Newton ou Théorème du moment cinétique.

3) Donnez la période des petites oscillations en résolvant l'équation différentielle pour θ petit.
Rappel: θ petit implique $\sin\theta \approx \theta$

4) Que se passe-t-il si on considère que la masse n'est plus ponctuelle en considérant que le frottement induit par le milieu sur la masse est un frottement fluide petite vitesse de coefficient λ ?

Question bonus 1.3 :

Comment peut on chronométrer la cuisson d'un œuf à la coque avec un mètre ruban pour SEUL instrument de mesure? (Indication: La cuisson d'un œuf à la coque est de trois minutes. Il y a au moins trois méthodes possibles)

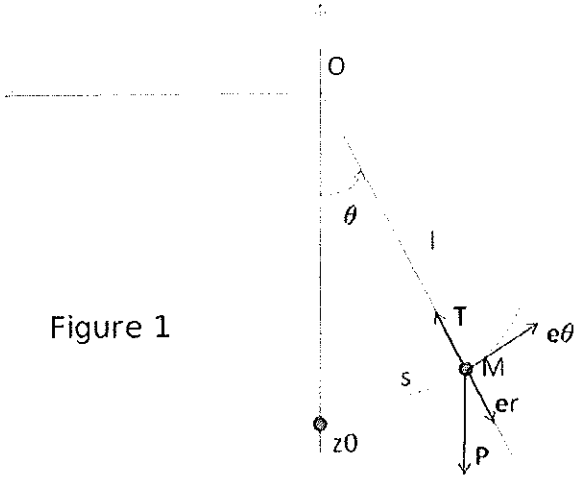

Figure 1

PARTIE 2 : Stéréo (5 points)

Question 2.1 :

Quels sont le principe et l'intérêt de la rectification épipolaire ?

Question 2.2 :

On considère un système de stéréovision calibré (la focale f des caméras et la baseline B entre les caméras sont donc connues) et rectifié. Comment obtenir la carte de profondeur (Z) à partir de la carte de disparité (d) ?

Question 2.3 :

Décrire le principe de la mise en correspondance par programmation dynamique dans le cadre de la stéréovision.

PARTIE 3 : LOD (5 points)

Question 3.1 :

Proposez une structure de donnée pour représenter un maillage 3D.

Question 3.2 :

Proposez un algorithme d'opérateur de fermeture de segment en utilisant la réponse à la question 3.1.

Question 3.2 :

Proposez un algorithme pour simplifier un maillage 3D en utilisant les réponses aux deux questions précédentes.

PARTIE 4 : Réalité Virtuelle (5 points)

Question 4.1 :

1) Donnez une définition de la réalité virtuelle en trois phrases.

2) Quelle est la différence entre la réalité documentée et la réalitée augmentée ?

3) Qu'est-ce qu'un péripherique haptique ?

Exercice 4.2 :

Réaliser un script « Camfly.js » permettant de gérer le déplacement d'une caméra libre, utilisant le clavier et la souris.
Le choix des touches ainsi que le type d'implémentation est libre.
Le pseudo langage est autorisé.
Une partie de la documentation unity 3D est disponible en annexe.

Exercice bonus 4.3 :

Proposez une solution permettant de mettre en place un système de stéréoscopie active ou passive sous Unity3D.

ANNEXE 1 : Camera Unity3D

Camera Inherits from Behaviour
A Camera is a device through which the player views the world.
A screen space point is defined in pixels. The bottom-left of the screen is (0,0); the right-top is (pixelWidth,pixelHeight). The z position is in world units from the camera.

A viewport space point is normalized and relative to the camera. The bottom-left of the camera is (0,0); the top-right is (1,1). The z position is in world units from the camera.

A world space point is defined in global coordinates (eg. Transform.position).

See Also: camera component.

Variables

fieldOfView	The field of view of the camera in degrees.
nearClipPlane	The near clipping plane distance.
farClipPlane	The far clipping plane distance.
renderingPath	Rendering path.
actualRenderingPath	Actually used rendering path (Read Only).
orthographicSize	Camera's half-size when in orthographic mode.
orthographic	Is the camera orthographic (*true*) or perspective (*false*)?
depth	Camera's depth in the camera rendering order.
aspect	The aspect ratio (width divided by height).
cullingMask	This is used to render parts of the scene selectively.
backgroundColor	The color with which the screen will be cleared.
rect	Where on the screen is the camera rendered in normalized coordinates.
pixelRect	Where on the screen is the camera rendered in pixel coordinates.
targetTexture	Destination render texture **(Unity Pro only)**.
pixelWidth	How wide is the camera in pixels (Read Only).
pixelHeight	How tall is the camera in pixels (Read Only).
cameraToWorldMatrix	Matrix that transforms from camera space to world space (Read Only).
worldToCameraMatrix	Matrix that transforms from world to camera space.
projectionMatrix	Set a custom projection matrix.
velocity	Get the world-space speed of the camera (Read Only).
clearFlags	How the camera clears the background.
layerCullDistances	Per-layer culling distances.
depthTextureMode	How and if camera generates a depth texture.

Functions

ResetWorldToCameraMatrix	Make the rendering position reflect the camera's position in the scene.
ResetProjectionMatrix	Make the projection reflect normal camera's parameters.
ResetAspect	Revert the aspect ratio to the screen's aspect ratio.
WorldToScreenPoint	Transforms position from world space into screen space.
WorldToViewportPoint	Transforms position from world space into viewport space.
ViewportToWorldPoint	Transforms position from viewport space into world space.
ScreenToWorldPoint	Transforms position from screen space into world space.
ScreenToViewportPoint	Transforms position from screen space into viewport space.
ViewportToScreenPoint	Transforms position from viewport space into screen space.

ViewportPointToRay	Returns a ray going from camera through a viewport point.
ScreenPointToRay	Returns a ray going from camera through a screen point.
Render	Render the camera manually.
RenderWithShader	Render the camera with shader replacement.
SetReplacementShader	Make the camera render with shader replacement.
ResetReplacementShader	Remove shader replacement from camera.
RenderToCubemap	Render into a cubemap from this camera.
CopyFrom	Makes this camera's settings match other camera.

Class Variables

main	The first enabled camera tagged "MainCamera" (Read Only).
current	The camera we are currently rendering with, for low-level render control only (Read Only).
allCameras	Returns all enabled cameras in the scene.

Inherited Variables

enabled	Enabled Behaviours are Updated, disabled Behaviours are not.
transform	The Transform attached to this GameObject (null if there is none attached).
rigidbody	The Rigidbody attached to this GameObject (null if there is none attached).
camera	The Camera attached to this GameObject (null if there is none attached).
light	The Light attached to this GameObject (null if there is none attached).
animation	The Animation attached to this GameObject (null if there is none attached).
constantForce	The ConstantForce attached to this GameObject (null if there is none attached).
renderer	The Renderer attached to this GameObject (null if there is none attached).
audio	The AudioSource attached to this GameObject (null if there is none attached).
guiText	The GUIText attached to this GameObject (null if there is none attached).
networkView	The NetworkView attached to this GameObject (Read Only). (null if there is none attached)
guiTexture	The GUITexture attached to this GameObject (Read Only). (null if there is none attached)
collider	The Collider attached to this GameObject (null if there is none attached).
hingeJoint	The HingeJoint attached to this GameObject (null if there is none attached).
particleEmitter	The ParticleEmitter attached to this GameObject (null if there is none attached).
gameObject	The game object this component is attached to. A component is always attached to a game object.
tag	The tag of this game object.
name	The name of the object.
hideFlags	Should the object be hidden, saved with the scene or modifiable by the user?

Inherited Functions

GetComponent	Returns the component of Type type if the game object has one attached, null if it doesn't.
GetComponentInChildren	Returns the component of Type type in the GameObject or any of its children using depth first search.
GetComponentsInChildren	Returns all components of Type type in the GameObject or any of its children.
GetComponents	Returns all components of Type type in the GameObject.
CompareTag	Is this game object tagged tag?
SendMessageUpwards	Calls the method named methodName on every MonoBehaviour in this game object and on every ancestor of the be
SendMessage	Calls the method named methodName on every MonoBehaviour in this game object.
BroadcastMessage	Calls the method named methodName on every MonoBehaviour in this game object or any of its children.
GetInstanceID	Returns the instance id of the object.

ANNEXE 2 : GameObject Unity3D

GameObject Inherits from Object
Base class for all entities in Unity scenes.

See Also: Component

Variables

isStatic	Editor only API that specifies if a game object is static.
transform	The Transform attached to this GameObject. (null if there is none attached)
rigidbody	The Rigidbody attached to this GameObject (Read Only). (null if there is none attached)
camera	The Camera attached to this GameObject (Read Only). (null if there is none attached)
light	The Light attached to this GameObject (Read Only). (null if there is none attached)
animation	The Animation attached to this GameObject (Read Only). (null if there is none attached)
constantForce	The ConstantForce attached to this GameObject (Read Only). (null if there is none attached)
renderer	The Renderer attached to this GameObject (Read Only). (null if there is none attached)
audio	The AudioSource attached to this GameObject (Read Only). (null if there is none attached)
guiText	The GUIText attached to this GameObject (Read Only). (null if there is none attached)
networkView	The NetworkView attached to this GameObject (Read Only). (null if there is none attached)
guiTexture	The GUITexture attached to this GameObject (Read Only). (null if there is none attached)
collider	The Collider attached to this GameObject (Read Only). (null if there is none attached)
hingeJoint	The HingeJoint attached to this GameObject (Read Only). (null if there is none attached)
particleEmitter	The ParticleEmitter attached to this GameObject (Read Only). (null if there is none attached)
layer	The layer the game object is in. A layer is in the range [0...32]
active	Is the GameObject active? Activates/Deactivates the GameObject.
tag	The tag of this game object.

Constructors

GameObject	Creates a new game object, named **name**.

Functions

GetComponent	Returns the component of Type type if the game object has one attached, null if it doesn't. You can access both b this function.
GetComponentInChildren	Returns the component of Type type in the GameObject or any of its children using depth first search.
GetComponents	Returns all components of Type type in the GameObject.
GetComponentsInChildren	Returns all components of Type type in the GameObject or any of its children.
SetActiveRecursively	Sets the active state of this and all the game objects children to state.
CompareTag	Is this game object tagged with tag?
SendMessageUpwards	Calls the method named methodName on every MonoBehaviour in this game object and on every ancestor of the be
SendMessage	Calls the method named methodName on every MonoBehaviour in this game object.
BroadcastMessage	Calls the method named methodName on every MonoBehaviour in this game object or any of its children.
AddComponent	Adds a component class named className to the game object.
SampleAnimation	Samples an animation at a given time for any animated properties.

Class Functions

CreatePrimitive	Creates a game object with a primitive mesh renderer and appropriate collider.
FindWithTag	Returns one active GameObject tagged tag. Returns null if no GameObject was found.
FindGameObjectsWithTag	Returns a list of active GameObjects tagged tag. Returns null if no GameObject was found.
Find	Finds a game object by name and returns it.

Inherited members

Inherited Variables

name	The name of the object.
hideFlags	Should the object be hidden, saved with the scene or modifiable by the user?

Inherited Functions

GetInstanceID	Returns the instance id of the object.

Inherited Class Functions

operator bool	Does the object exist?
Instantiate	Clones the object original and returns the clone.
Destroy	Removes a gameobject, component or asset.
DestroyImmediate	Destroys the object obj immediately. It is strongly recommended to use Destroy instead.
FindObjectsOfType	Returns a list of all active loaded objects of Type type.
FindObjectOfType	Returns the first active loaded object of Type type.
operator ==	Compares if two objects refer to the same
operator !=	Compares if two objects refer to a different object
DontDestroyOnLoad	Makes the object target not be destroyed automatically when loading a new scene.

ANNEXE 3 : Transform Unity3D

Transform Inherits from Component, IEnumerable
Position, rotation and scale of an object.

Every object in a scene has a Transform. It's used to store and manipulate the position, rotation and scale of the object. Every Transform can have a parent, which allows you to apply position, rotation and scale hierarchically. This is the hierarchy seen in the Hierarchy pane. They also support enumerators so you can loop through children using:

JavaScript

```
// Moves all transform children 10 units upwards!

for (var child : Transform in transform) {
   child.position += Vector3.up * 10.0;
}
```

See Also: The component reference, Physics class.

Variables

position	The position of the transform in world space.
localPosition	Position of the transform relative to the parent transform.
eulerAngles	The rotation as Euler angles in degrees.
localEulerAngles	The rotation as Euler angles in degrees relative to the parent transform's rotation.
right	The red axis of the transform in world space.
up	The green axis of the transform in world space.
forward	The blue axis of the transform in world space.
rotation	The rotation of the transform in world space stored as a Quaternion.

localRotation	The rotation of the transform relative to the parent transform's rotation.
localScale	The scale of the transform relative to the parent.
parent	The parent of the transform.
worldToLocalMatrix	Matrix that transforms a point from world space into local space (Read Only).
localToWorldMatrix	Matrix that transforms a point from local space into world space (Read Only).
root	Returns the topmost transform in the hierarchy.
childCount	The number of children the Transform has.
lossyScale	The global scale of the object (Read Only).

Functions

Translate	Moves the transform in the direction and distance of translation.
Rotate	Applies a rotation of eulerAngles.z degrees around the z axis, eulerAngles.x degrees around the x axis, and eulerAn (in that order).
RotateAround	Rotates the transform about axis passing through point in world coordinates by angle degrees.
LookAt	Rotates the transform so the forward vector points at /target/'s current position.
TransformDirection	Transforms direction from local space to world space.
InverseTransformDirection	Transforms a direction from world space to local space. The opposite of Transform.TransformDirection.
TransformPoint	Transforms position from local space to world space.
InverseTransformPoint	Transforms position from world space to local space. The opposite of Transform.TransformPoint.
DetachChildren	Unparents all children.
Find	Finds a child by name and returns it.
IsChildOf	Is this transform a child of parent?

Inherited Variables

transform	The Transform attached to this GameObject (null if there is none attached).
rigidbody	The Rigidbody attached to this GameObject (null if there is none attached).
camera	The Camera attached to this GameObject (null if there is none attached).
light	The Light attached to this GameObject (null if there is none attached).
constantForce	The ConstantForce attached to this GameObject (null if there is none attached).
renderer	The Renderer attached to this GameObject (null if there is none attached).
audio	The AudioSource attached to this GameObject (null if there is none attached).
guiText	The GUIText attached to this GameObject (null if there is none attached).
networkView	The NetworkView attached to this GameObject (Read Only). (null if there is none attached)
guiTexture	The GUITexture attached to this GameObject (Read Only). (null if there is none attached)
collider	The Collider attached to this GameObject (null if there is none attached).
hingeJoint	The HingeJoint attached to this GameObject (null if there is none attached).
particleEmitter	The ParticleEmitter attached to this GameObject (null if there is none attached).
gameObject	The game object this component is attached to. A component is always attached to a game object.
name	The name of the object.
hideFlags	Should the object be hidden, saved with the scene or modifiable by the user?

ANNEXE 4 : Collision Unity3D

Collision
Describes collision.

Collision information is passed to Collider.OnCollisionEnter, Collider.OnCollisionStay and Collider.OnCollisionExit events. See Also: ContactPoint.

Variables

relativeVelocity	The relative linear velocity of the two colling objects (Read Only).
rigidbody	The Rigidbody we hit (Read Only). This is null if the object we hit is a collider with no rigidbody attached.
collider	The Collider we hit (Read Only).
transform	The Transform of the object we hit (Read Only).
gameObject	/gameObject/ is the object we are colliding with. (Read Only).
contacts	The contact points generated by the physics engine.

RaycastHit Struct
Structure used to get information back from a raycast.

See Also: Physics.Raycast, Physics.Linecast, Physics.RaycastAll.

Variables

point	The impact point in world space where the ray hit the collider.
normal	The normal of the surface the ray hit.
barycentricCoordinate	The barycentric coordinate of the triangle we hit.
distance	The distance from the ray's origin to the impact point.
triangleIndex	The index of the triangle that was hit.
textureCoord	The uv texture coordinate at the impact point.

textureCoord2	The secondary uv texture coordinate at the impact point.
lightmapCoord	The uv lightmap coordinate at the impact point.
collider	The Collider that was hit.
rigidbody	The Rigidbody of the collider that was hit. If the collider is not attached to a rigidbody then it is null.
transform	The Transform of the rigidbody or collider that was hit.

ANNEXE 5 : Input Unity3D

Input
Interface into the Input system.

Use this class to read the axes set up in the Input Manager, and to access multi-touch/accelerometer data on mobile devices.

To read an axis use Input.GetAxis with one of the following default axes: "Horizontal" and "Vertical" are mapped to joystick, A, W, S, D and the arrow keys. "Mouse X" and "Mouse Y" are mapped to the mouse delta. "Fire1", "Fire2" "Fire3" are mapped to Ctrl, Alt, Cmd keys and three mouse or joystick buttons. New input axes can be added in the Input Manager.

If you are using input for any kind of movement behaviour use Input.GetAxis. It gives you smoothed and configurable input that can be mapped to keyboard, joystick or mouse. Use Input.GetButton for action like events only. Don't use it for movement, Input.GetAxis will make the script code smaller and simpler.

Note also that the Input flags are not reset until "Update()", so its suggested you make all the Input Calls in the Update Loop.

Class Variables

mousePosition	The current mouse position in pixel coordinates. (Read Only)
anyKey	Is any key or mouse button currently held down? (Read Only)
anyKeyDown	Returns true the first frame the user hits any key or mouse button (Read Only).
inputString	Returns the keyboard input entered this frame (Read Only).
acceleration	Last measured linear acceleration of a device in three-dimensional space (Read Only).
accelerationEvents	Returns list of acceleration measurements which occurred during the last frame (Read Only) (Allocates temporary variat
accelerationEventCount	Number of acceleration measurements which occurred during last frame.
touches	Returns list of objects representing status of all touches during last frame (Read Only) (Allocates temporary variables).
touchCount	Number of touches (Read Only).
multiTouchEnabled	Property indicating whether the system handles multiple touches.
deviceOrientation	Device physical orientation as reported by OS (Read Only).

Class Functions

GetAxis	Returns the value of the virtual axis identified by axisName.
GetAxisRaw	Returns the value of the virtual axis identified by axisName with no smoothing filtering applied.
GetButton	Returns true while the virtual button identified by buttonName is held down.
GetButtonDown	Returns true during the frame the user pressed down the virtual button identified by buttonName.
GetButtonUp	Returns true the first frame the user releases the virtual button identified by buttonName.
GetKey	Returns true while the user holds down the key identified by name. Think auto fire.
GetKeyDown	Returns true during the frame the user starts pressing down the key identified by name.
GetKeyUp	Returns true during the frame the user releases the key identified by name.
GetJoystickNames	Returns an array of stings describing the connected joysticks.
GetMouseButton	Returns whether the given mouse button is held down.
GetMouseButtonDown	Returns true during the frame the user pressed the given mouse button.
GetMouseButtonUp	Returns true during the frame the user releases the given mouse button.
ResetInputAxes	Resets all input. After ResetInputAxes all axes return to 0 and all buttons return to 0 for one frame.
GetAccelerationEvent	Returns specific acceleration measurement which occurred during last frame (Does not allocate temporary variables).
GetTouch	Returns object representing status of a specific touch (Does not allocate temporary variables).

ANNEXE 6 : MonoBehaviour Unity3D

MonoBehaviour Inherits from Behaviour
MonoBehaviour is the base class every script derives from.

Using Javascript every script automatically derives from MonoBehaviour. When using C# or Boo you have to explicitly derive from MonoBehaviour.

Note: The checkbox for disabling a MonoBehavior (on the editor) will only prevent Start(), Awake(), Update(), FixedUpdate(), and OnGUI() from executing. If none of these functions are present, the checkbox is not displayed.

See Also: The chapter on scripting in the manual.

Overridable Functions

Update	Update is called every frame, if the MonoBehaviour is enabled.
LateUpdate	LateUpdate is called every frame, if the Behaviour is enabled.
FixedUpdate	This function is called every fixed framerate frame, if the MonoBehaviour is enabled.
Awake	Awake is called when the script instance is being loaded.
Start	Start is called just before any of the Update methods is called the first time.
Reset	Reset to default values.
OnMouseEnter	OnMouseEnter is called when the mouse entered the GUIElement or Collider.
OnMouseOver	OnMouseOver is called every frame while the mouse is over the GUIElement or Collider.
OnMouseExit	OnMouseExit is called when the mouse is not any longer over the GUIElement or Collider.
OnMouseDown	OnMouseDown is called when the user has pressed the mouse button while over the GUIElement or C
OnMouseUp	OnMouseUp is called when the user has released the mouse button.
OnMouseDrag	OnMouseDrag is called when the user has clicked on a GUIElement or Collider and is still holding dow
OnTriggerEnter	OnTriggerEnter is called when the Collider other enters the trigger.
OnTriggerExit	OnTriggerExit is called when the Collider other has stopped touching the trigger.
OnTriggerStay	OnTriggerStay is called once per frame for every Collider other that is touching the trigger.
OnCollisionEnter	OnCollisionEnter is called when this collider/rigidbody has begun touching another rigidbody/collider.
OnCollisionExit	OnCollisionExit is called when this collider/rigidbody has stopped touching another rigidbody/collider.
OnCollisionStay	OnCollisionStay is called once per frame for every collider/rigidbody that is touching rigidbody/collider.
OnControllerColliderHit	OnControllerColliderHit is called when the controller hits a collider while performing a Move.
OnJointBreak	Called when a joint attached to the same game object broke.
OnParticleCollision	OnParticleCollision is called when a particle hits a collider.
OnBecameVisible	OnBecameVisible is called when the renderer became visible by any camera.
OnBecameInvisible	OnBecameInvisible is called when the renderer is no longer visible by any camera.
OnLevelWasLoaded	This function is called after a new level was loaded.
OnEnable	This function is called when the object becomes enabled and active.
OnDisable	This function is called when the behaviour becomes disabled () or inactive.
OnPreCull	OnPreCull is called before a camera culls the scene.
OnPreRender	OnPreRender is called before a camera starts rendering the scene.
OnPostRender	OnPostRender is called after a camera finished rendering the scene.
OnRenderObject	OnRenderObject is called after camera has rendered the scene.
OnWillRenderObject	OnWillRenderObject is called once for each camera if the object is visible.
OnGUI	OnGUI is called for rendering and handling GUI events.
OnRenderImage	OnRenderImage is called after all rendering is complete to render image
OnDrawGizmosSelected	Implement this OnDrawGizmosSelected if you want to draw gizmos only if the object is selected.
OnDrawGizmos	Implement this OnDrawGizmos if you want to draw gizmos that are also pickable and always drawn.
OnApplicationPause	Sent to all game objects when the player pauses.
OnApplicationFocus	Sent to all game objects when the player gets or looses focus.
OnApplicationQuit	Sent to all game objects before the application is quit.
OnPlayerConnected	Called on the server whenever a new player has successfully connected.
OnServerInitialized	Called on the server whenever a Network.InitializeServer was invoked and has completed.
OnConnectedToServer	Called on the client when you have successfully connected to a server
OnPlayerDisconnected	Called on the server whenever a player disconnected from the server.
OnDisconnectedFromServer	Called on the client when the connection was lost or you disconnected from the server.
OnFailedToConnect	Called on the client when a connection attempt fails for some reason.
OnFailedToConnectToMasterServer	Called on clients or servers when there is a problem connecting to the MasterServer.
OnMasterServerEvent	Called on clients or servers when reporting events from the MasterServer.
OnNetworkInstantiate	Called on objects which have been network instantiated with Network.Instantiate
OnSerializeNetworkView	Used to customize synchronization of variables in a script watched by a network view.

Class Functions

print	Logs message to the Unity Console. This function is identical to Debug.Log.

Inherited members

Inherited Variables

enabled	Enabled Behaviours are Updated, disabled Behaviours are not.
transform	The Transform attached to this GameObject (null if there is none attached).
rigidbody	The Rigidbody attached to this GameObject (null if there is none attached).
camera	The Camera attached to this GameObject (null if there is none attached).
light	The Light attached to this GameObject (null if there is none attached).