Client-side function

# ProcessLineOfSight

This function casts a ray between two points in the world, and tells you information about the point that was hit, if any. The two positions **must** be within the local player's draw distance as the collision data is not loaded outside this area, and the call will just fail as if the ray didn't hit.

This function is relatively expensive to call, so over use of this in scripts may have a detrimental effect on performance.

This function is useful for checking for collisions and for editor-style scripts. If you wish to find what element is positioned at a particular point on the screen, use this function combined with getWorldFromScreenPosition. If you wish to just know if something is hit, and don't care about what or where was hit, use isLineOfSightClear.

> **Note:** Due to a bug, shootThroughStuff argument does currently check for seeThroughStuff!

> **Note:** Due to a bug, seeThroughStuff argument has no effect. It mistakenly checks for "shootThrough" surfaces and will always behave as if the argument is set to FALSE (It will never hit).

## Syntax

Return values labelled for ease of reference.

```
bool                -- hit
float float float   -- hitX, hitY, hitZ
element             -- hitElement
float float float   -- normalX, normalY, normalZ
int                 -- material
float               -- lighting
int                 -- piece
int                 -- worldModelID
float float float   -- worldModelPositionX,Y,Z
float float float   -- worldModelRotationX,Y,Z
int                 -- worldLODModelID
                  processLineOfSight ( float startX, float startY, float startZ,
                                       float endX, float endY, float endZ,
                                       [ bool checkBuildings = true,
                                       bool checkVehicles = true,
                                       bool checkPlayers = true,
                                       bool checkObjects = true,
                                       bool checkDummies = true,
                                       bool seeThroughStuff = false,
                                       bool ignoreSomeObjectsForCamera = false,
                                       bool shootThroughStuff = false,
                                       element ignoredElement = nil,
                                       bool includeWorldModelInformation = false,
                                       bool bIncludeCarTyres ] )
```

### Required Arguments

- **startX:** The start *x* position
- **startY:** The start *y* position
- **startZ:** The start *z* position
- **endX:** The end *x* position
- **endY:** The end *y* position
- **endZ:** The end *z* position

### Optional Arguments

*NOTE:* When using optional arguments, you might need to supply all arguments before the one you wish to use. For more information on optional arguments, see optional arguments.

- **checkBuildings:** Allow the line of sight to be blocked by GTA's internally placed buildings, i.e. the world map.
- **checkVehicles:** Allow the line of sight to be blocked by vehicles.
- **checkPlayers:** Allow the line of sight to be blocked by players.
- **checkObjects:** Allow the line of sight to be blocked by objects.
- **checkDummies:** Allow the line of sight to be blocked by GTA's internal dummies. These are not used in the current NRP version so this argument can be set to *false*.
- **seeThroughStuff:** Allow the line of sight **pass through** collision materials that have this flag enabled (By default material IDs 52, 55 and 66 which are some fences that you can shoot throug but still walk on them).

- **ignoreSomeObjectsForCamera:** Allow the line of sight to **pass through** objects that have (K) property enabled in "object.dat" data file. (i.e. Most dynamic objects like boxes or barrels)
- **shootThroughStuff:** Allow the line of sight to **pass through** collision materials that have this flag enabled (By default material IDs 28, 29, 31, 32, 33, 74, 75, 76, 77, 78, 79, 96, 97, 98, 99, 100 which are exclusively sand / beach or underwater objects).
- **ignoredElement:** Allow the line of sight to **pass through** a certain specified element. This is usually set to the object you are tracing from so it does not interfere with the results.
- **includeWorldModelInformation :** Include the results of hitting a world model.
- **bIncludeCarTyres :** Includes car tyre hits.

## Returns

- **hit:** *true* if there is a collision, *false* otherwise

The other values are only filled if there is a collision, they contain *nil* otherwise

- **hitX, hitY, hitZ:** collision position
- **hitElement:** the NRP element hit if any, *nil* otherwise
- **normalX, normalY, normalZ:** the normal of the surface hit
- **material:** an integer representing the GTASA material ID of the surface hit when applicable (world, objects)
- **lighting:** a float between 0 (fully dark) and 1 (bright) representing the amount of light that the hit building surface will transfer to peds or vehicles that are in contact with it. The value can be affected by the game time of day, usually with a lower (darker) value being returned during the night.
- **piece:** an integer representing the part of the element hit if hitElement is a vehicle or a ped/player, *0* otherwise.
    - For a ped/player, piece represents the body part hit:

- **3:** Torso
- **4:** Ass
- **5:** Left Arm
- **6:** Right Arm
- **7:** Left Leg
- **8:** Right Leg
- **9:** Head

- 
    - For vehicles, piece represents the vehicle part hit:

- **0:** Frame
- **2:** Trunk
- **3:** Hood
- **4:** Rear
- **5:** Front left door
- **6:** Front right door
- **7:** Rear left door
- **8:** Rear right door
- **13:** Front Left tyre
- **14:** Front Right tyre
- **15:** Back Left tyre
- **16:** Back Right tyre

(Other potential IDs haven't been documented yet and might depend on vehicle model)

- **worldModelID:** If includeWorldModelInformation was set to *true* and a world model was hit, this will contain the model ID.
- **worldModelPositionX,Y,Z:** If worldModelID is set, this will contain the world model position.
- **worldModelRotationX,Y,Z:** If worldModelID is set, this will contain the world model rotation.
- **worldLODModelID:** If worldModelID is set, this will contain the LOD model ID if applicable.