

Delayed Physics Functions

General

Running physics functions like raycasts is expensive. Some scenes in the game can require up to 200 raycasts per frame. The normal way to call a physics function is to call it directly and wait for it to finish, this stops execution of the game until it is completed. This wait can be removed by using delayed physics functions. Delayed functions are run in the background and then calls a callback script function when it is done. The callback function is sent in using a string, the same way as Timers work now.

If a delayed function is called inside OnUpdate() then it is guaranteed to be completed before OnPostUpdate() is called. All callback functions will be called in the order that they were added.

Delayed functions should be used when possible, but sometimes it might be easier to call the normal stop and wait function.

Functions

The delayed functions are added to both maps and entities to make it easier to setup the callback function.

All map functions also have helpers inside helper_map.hps

Map / Entity

The normal non-delayed version of these functions can be called with cLux_*

void CheckLineOfSight(const tString& asCallbackFunc, const cVector3f& avStart, const cVector3f& avEnd, bool abCheckOnlyShadowCasters)

Checks if the line of sight between two positions is clear. Callback function syntax: void func(bool abClear)

void GetClosestEntity(const tString& asCallbackFunc, const cVector3f& avStart, const cVector3f& avDir, float afRayLength, int allInteractType, bool abCheckLineOfSight)

Returns the closest entity in direction. Callback function syntax: void func(bool abSuccessful, float afDistance, iPhysicsBody@ apBody, iLuxEntity@ apEntity)

void GetClosestBody(const tString& asCallbackFunc, const cVector3f& avStart, const cVector3f& avDir, float afRayLength)

Returns the closest body in direction. Callback function syntax: void func(bool abSuccessful, float afDistance, const cVector3f&in avNormal, iPhysicsBody@ apBody)

void GetClosestCharCollider(const tString& asCallbackFunc, const cVector3f& avStart, const cVector3f& avDir, float afRayLength, bool abCheckDynamic)

Returns the closest physics body that can collide with a character. Callback function syntax: void func(bool abSuccessful, float afDistance, const cVector3f& in avNormal, iPhysicsBody@ apBody)

void GetLightLevelAtPos(const tString& asCallbackFunc, const cVector3f& avPos, iLight@ apSkipLight, float afRadiusAdd)

Get how bright the position of the map is. Callback function syntax: void func(float afLightLevel)

AI Base

AI classes also have some additional delayed functions

void GetPlayerIsInLineOfSight(const tString& asCallbackFunc, float afFOV, const cVector3f &avForward, bool abCheckFOV)

void GetPlayerIsInLineOfSight(const tString& asCallbackFunc)

Get if the player is visible. Callback syntax: void func(bool abInSight)

void GetEntityIsInPlayerLineOfSight(const tString& asCallbackFunc, bool abCheckFOV)

Get if the player sees the AI. Callback syntax: void func(bool abInSight)

void GetDistanceToGround(const tString& asCallbackFunc, float afMaxTestDistance, bool abCheckDynamic, int aNumOfRays=1, float afRadius=0.25, bool abGetClosest=true)

Get the distance to the ground. Callback syntax: void func(float afDistance)

Pathfinder

The pathfinder and AStar handler also support delayed execution. It now runs the algorithm over 2 frames. The first frame set up everything and finds a valid start location using raycats. The next frame then performs the AStar and find a valid goal using raycasts. When the goal is found the callback is called. There are helper functions available in helper_ai.hps

void MoveTo(const cVector3f& avPos, float afUpdateFreq, bool abExactStopAtEnd, const tString& asResultCallback=, bool abCallbackInMap=false)

void MoveToNode(const tString& asNodeName, float afUpdateFreq, bool abExactStopAtEnd, const tString& asResultCallback=, bool abCallbackInMap=false)

Gets a path from the current location of the entity parent to the position. If a result callback function is specified the function gets called. Either in the entity script or inside the map script. Callback syntax: void func(bool abPathAvailable)

From:

<https://wiki.frictionalgames.com/> - **Frictional Game Wiki**

Permanent link:

https://wiki.frictionalgames.com/hpl3/game/scripting/delayed_physics_functions

Last update: **2014/06/11 09:27**

