

helper_areas.hps

Area_MoveLinearTo

```
void Area_MoveLinearTo(const tString &in asName,
                      const tString &in asTargetEntity,
                      float afAcceleration,
                      float afMaxSpeed,
                      float afSlowDownDist=0.0f,
                      bool abResetSpeed=true,
                      const tString &in asCallback="")
```

Moves the specified area to the current position of the target entity. !Does not work on Liquid Areas, instead use LiquidArea_MoveLinearTo!

- **asName**: name of the area to move, wildcard(s) * are supported.
- **asTargetEntity**: name of the entity to move the area to.
- **afAcceleration**: acceleration .in m/s²
- **afMaxSpeed**: the maximum speed of movement.
- **afSlowDownDist**: the distance at which movement slows down.
- **abResetSpeed**: if all static movement for the area should be stopped before starting this movement.
- **asCallback**: callback when movement is done: void asCallback(const tString& in asAreaName)

Area_PlaceAtEntity

```
void Area_PlaceAtEntity(const tString &in asAreaName,
                       const tString &in asTargetEntity,
                       bool abAlignRotation=false)
```

Places the specified area at another entity, optionally aligning its rotation with the target entity.

- **asAreaName**: name of the area.
- **asTargetEntity**: entity to place at
- **abAlignRotation**: if true the entity will be given the same rotation as the target entity.

Area_StopMovement

```
void Area_StopMovement(const tString &in asName)
```

Stops an area that is already moving. !This is not for liquid areas!

- **asName**: name of the area to stop moving, wildcard(s) * are supported.

Doorway_PlayerIsInGroup

```
bool Doorway_PlayerIsInGroup(const tString &in asGroupName)
```

Returns true if the player has passed through one of the doorway areas belonging to this group and hasn't gone back out.

- **asGroupName**: name of the group.

Doorway_SetPlayerIsInGroup

```
void Doorway_SetPlayerIsInGroup(const tString &in asGroupName,  
                                bool abIsInGroup=true)
```

Explicitly set, for if the player starts in a group or is teleported in or out.

- **asGroupName**: name of the group.
- **abIsInGroup**: state to set.

Doorway_EntityIsInGroup

```
bool Doorway_EntityIsInGroup(const tString &in asEntityName,  
                              const tString &in asGroupName)
```

Returns true if the specified entity has passed through one of the doorway areas belonging to this group and hasn't gone back out. Note that this will only return true if the entity has physically passed through the doorway, not if they start out inside or are teleported into the area.

- **asEntityName**: name of the entity to check.
- **asGroupName**: name of the group.

Doorway_SetEntityIsInGroup

```
void Doorway_SetEntityIsInGroup(const tString &in asEntityName,  
                                const tString &in asGroupName,  
                                bool abIsInGroup=true)
```

Explicitly set, for if the specified entity starts in a group or is teleported in or out.

- **asEntityName**: name of the entity to set the state of.

- **asGroupName**: name of the group.
 - **abIsInGroup**: state to set.
-

LiquidArea_MoveLinearTo

```
void LiquidArea_MoveLinearTo(const tString &in asName,  
                             const tString &in asTargetEntity,  
                             float afAcceleration,  
                             float afMaxSpeed,  
                             float afSlowDownDist=0.0f,  
                             bool abResetSpeed=false,  
                             const tString &in asCallback="")
```

Moves the specified liquid area to the current position of the target entity.

- **asName**: name of the liquid area to move, wildcard(s) * are supported.
 - **asTargetEntity**: name of the entity to move the liquid area to.
 - **afAcceleration**: acceleration .in m/s²
 - **afMaxSpeed**: the maximum speed of movement.
 - **afSlowDownDist**: the distance at which movement slows down.
 - **abResetSpeed**: if all static movement for the liquid area should be stopped before starting this movement.
 - **asCallback**: callback when movement is done: void asCallback(const tString& in asAreaName)
-

LiquidArea_StopMovement

```
void LiquidArea_StopMovement(const tString &in asName)
```

LiquidArea_SetEffectOnExit

```
void LiquidArea_SetEffectOnExit(const tString &in asName,  
                                bool abState)
```

Sets whether screen effects should be shown when the camera leaves the area.

- **asName**: name of the liquid area, wildcard(s) * are supported.
 - **abState**: state to set.
-

StickyArea_AttachEntity

```
void StickyArea_AttachEntity(const tString &in asArea,
```

```
const tString &in asEntity,  
const tString &in asBody)
```

Attaches an entity to a sticky area. The entity must have a physics body.

- **asArea**: name of sticky area.
 - **asEntity**: name of entity.
 - **asBody**: name of specific body on entity, if left as an empty string, the main body will be used.
-

StickyArea_AttachBody

```
void StickyArea_AttachBody(const tString &in asArea,  
const tString &in asBody)
```

Attaches a physics body to a sticky area.

- **asArea**: name of sticky area.
 - **asBody**: name of body to attach.
-

StickyArea_Detach

```
void StickyArea_Detach(const tString &in asArea)
```

Detaches anything attached to a sticky area.

- **asArea**: name of sticky area.
-

StickyArea_GetProp

```
iLuxEntity StickyArea_GetProp(const tString &in asArea)
```

Returns the entity attached to the area.

- **asArea**: name of sticky area.

Returns: iLuxEntity @, the attached entity of the area, or null if no entity is attached.

StickyArea_GetAttachedEntityName

```
tString StickyArea_GetAttachedEntityName(const tString &in asArea)
```

Returns the name of the attached entity.

- **asArea**: name of sticky area.

Returns: tString , name of the attached entity. An empty string if no entity is attached.

StickyArea_GetBody

```
iPhysicsBody StickyArea_GetBody(const tString &in asArea)
```

Returns the body attached to the area.

- **asArea**: name of sticky area.

Returns: iPhysicsBody @, the attached body of the area, or null if no body is attached.

StickyArea_GetPoseTime

```
float StickyArea_GetPoseTime(const tString &in asArea)
```

Returns the time it takes for the specified sticky area to pose an attached entity.

- **asArea**: name of sticky area.

Returns: float, pose time in seconds.

CameraAnimationNode_GetMaxYaw

```
float CameraAnimationNode_GetMaxYaw(const tString &in asArea)
```

Returns the maximum yaw for the specified node.

- **asArea**: name of animation node.

Returns: float, max yaw in radians.

CameraAnimationNode_GetMaxPitch

```
float CameraAnimationNode_GetMaxPitch(const tString &in asArea)
```

Returns the maximum pitch for the specified node.

- **asArea**: name of animation node.

Returns: float, max pitch in radians.

CameraAnimationNode_GetLookSpeedMul

```
float CameraAnimationNode_GetLookSpeedMul(const tString &in asArea)
```

Returns the look speed multiplier for the specified node.

- **asArea**: name of animation node.

Returns: float, look speed multiplier.

CameraAnimationNode_GetForwardTime

```
float CameraAnimationNode_GetForwardTime(const tString &in asArea)
```

Returns the time it takes to go to the next node in the animation.

- **asArea**: name of animation node.

Returns: float, forward time in seconds

CameraAnimationNode_GetForwardSpeed

```
float CameraAnimationNode_GetForwardSpeed(const tString &in asArea)
```

Returns the forward speed for the specified node.

- **asArea**: name of animation node.

Returns: float, forward speed measured in “nodes per second”

CameraAnimationNode_GetBackTime

```
float CameraAnimationNode_GetBackTime(const tString &in asArea)
```

Returns the time it takes to go back to this node from the next node in the animation.

- **asArea**: name of animation node.

Returns: float, back time in seconds

CameraAnimationNode_GetBackSpeed

```
float CameraAnimationNode_GetBackSpeed(const tString &in asArea)
```

Returns the back speed for the specified node.

- **asArea**: name of animation node.

Returns: float, back speed measured in “nodes per second”

CameraAnimationNode_GetAutoMove

```
float CameraAnimationNode_GetAutoMove(const tString &in asArea)
```

Returns the amount of auto move in the specified node.

- **asArea**: name of animation node.

Returns: float, auto move amount, 1 being full speed ahead and -1 being full speed back.

CameraAnimationNode_SetAutoMove

```
void CameraAnimationNode_SetAutoMove(const tString &in asArea,  
                                     const float afAutoMove)
```

Sets the auto move amount for the specified node.

- **asArea**: name of animation node.
 - **afAutoMove**: amount to set.
-

CameraAnimationNode_FadeAutoMove

```
void CameraAnimationNode_FadeAutoMove(const tString &in asArea,  
                                     const float afAutoMove,  
                                     const float afTime,  
                                     const eEasing  
aEasing=eEasing_QuartInOut)
```

Fades the auto move amount for the specified node.

- **asArea**: name of animation node.
- **afAutoMove**: Target auto move amount.
- **afTime**: Time that the fading will take (?).
- **aEasing**: Easing curve to be used for fading.

CameraAnimationNode_GetLookMoveDist

```
float CameraAnimationNode_GetLookMoveDist(const tString &in asArea)
```

CameraAnimationNode_GetLookMoveAngle

```
float CameraAnimationNode_GetLookMoveAngle(const tString &in asArea)
```

CameraAnimationNode_GetInteractiveMovement

```
bool CameraAnimationNode_GetInteractiveMovement(const tString &in asArea)
```

Returns if the node takes movement input from the player.

- **asArea**: name of animation node.

Returns: bool, true if the node's movement is interactive.

CameraAnimationNode_SetInteractiveMovement

```
void CameraAnimationNode_SetInteractiveMovement(const tString &in asArea,  
                                                bool abX)
```

Sets if the node should take movement input from the player.

- **asArea**: name of animation node.
- **abX**: state to set.

CameraAnimationNode_GetCrouchOnExit


```
bool CameraAnimationNode_GetCrouchOnExit(const tString &in asArea)
```

Returns true if the player should crouch if the animation stops at this node.

- **asArea**: name of animation node.

Returns: bool, true if the player should crouch on exit.

CameraAnimationNode_GetInitToCamera

```
bool CameraAnimationNode_GetInitToCamera(const tString &in asArea)
```

Returns true if the node should be placed at the camera's position and orientation when the animation starts.

- **asArea**: name of animation node.

Returns: bool, true if the node should initialize to the camera.

CameraAnimationNode_SetInitToCamera

```
void CameraAnimationNode_SetInitToCamera(const tString &in asArea,  
                                          bool abInitToCamera)
```

Sets if the node should be placed at the camera's position and orientation when the animation starts.

- **asArea**: name of animation node.
 - **abInitToCamera**: state to set.
-

CameraAnimationNode_GetUseSpeedFading

```
bool CameraAnimationNode_GetUseSpeedFading(const tString &in asArea)
```

Returns true if the node fades its speeds (back and forward) between this and the previous node in the animation.

- **asArea**: name of animation node.

Returns: bool, true if the node fades its speeds.

CameraAnimationNode_GetCallbackFunc

```
tString CameraAnimationNode_GetCallbackFunc(const tString &in asArea)
```

Returns the callback function of the specified node.

- **asArea**: name of animation node.

Returns: tString , the callback function.

CameraAnimationNode_SetCallbackFunc

```
void CameraAnimationNode_SetCallbackFunc(const tString &in asArea,  
                                          const tString &in asFunction)
```

Sets the callback function of the specified node.

- **asArea**: name of animation node.
 - **asFunction**: the callback function to set.
-

CameraAnimationNode_HasBackMovement

```
bool CameraAnimationNode_HasBackMovement(const tString &in asArea)
```

Returns true if it's possible to move from the next node in the animation to this one.

- **asArea**: name of animation node.

Returns: bool, true if it's possible to move back to this node.

From:
<https://oldwiki.frictionalgames.com/> - Frictional Game Wiki

Permanent link:
https://oldwiki.frictionalgames.com/hpl3/game/scripting/function_reference/helper_areas?rev=1446107695

Last update: 2015/10/29 08:34

