API-MicroGameAtZero

Version 0.1.0 Alpha

Core Class:

AUDIO Class:

CAMERA2D Class:

OBJECT Class:

KINECTBODY Class:

RIGIDBODY Class:

STATICBODY Class:

AREA Class:

SCENE Class:

UI Class:

UI BUTTON Class:

UI CHECKBOX Class:

UI IMAGE Class:

UI NUMBER Class:

UI RECT Class:

UI TEXT Class:

Vector2:

Structures:

Enumeration:

Others:

Core Class

Defines the game loop and scene add/load functions.

MICROGAMEATZERO::getInstance()

```
static MICROGAMEATZERO* MICROGAMEATZERO::getInstance()
```

This function returns the instance of the MICROGAMEATZERO. If no instance exists, the function creates an instance of the MICROGAMEATZERO.

Parameters

None

Returns

MICROGAMEATZERO * pointer to the MICROGAMEATZERO instance

Example

```
#include “microGameAtZero/microGameAtZero.h”


MICROGAMEATZERO *game = MICROGAMEATZERO::getInstance();
```

initMicroGameAtZero

microGameAtZero_err MICROGAMEATZERO::initMicroGameAtZero (settingsEngine *settings*)

This function initialization the microGameAtZero engine.

Parameters

settings engine settings (x size, y size of the display, and max fps)

Returns

MICRO_GAME_AT_ZERO_OK everything is OK
MICRO_GAME_AT_ZERO_INVALID_PARAM invalid parameter

Example

```
#include “microGameAtZero/microGameAtZero.h”

settingsEngine settings;

settings.maxFps = 20;
settings.screenX = 240;
settings.screenY = 240;
MICROGAMEATZERO *game = MICROGAMEATZERO::getInstance();
game->initMicroGameAtZero(settings);
```

addNewScene

microGameAtZero_err MICROGAMEATZERO::addNewScene (SCENE * *pScene*)

This function adds a new scene.

Parameters

pScene pointer to the new scene to be add

Returns

MICRO_GAME_AT_ZERO_OK everything is ok
MICRO_GAME_AT_ZERO_INVALID_PARAM invalid parameter

Example

See example folder

loadScene

microGameAtZero_err MICROGAMEATZERO::loadScene (uint16_t *position*)

This function loads the selected scene.

Parameters

position index of the selected scene.

Returns

MICRO_GAME_AT_ZERO_OK everything is OK
MICRO_GAME_AT_ZERO_INVALID_PARAM PARAM there is no scene at this position

Example

```
#include “microGameAtZero/microGameAtZero.h”


MICROGAMEATZERO *game = MICROGAMEATZERO::getInstance();

game->loadScene(0);
```

saveFile

```
microGameAtZero_err MICROGAMEATZERO::saveFile ( char *  pFileName,
                                                char *  pData
                                              )
```

This function saves the passed data to the selected file.

Parameters

pFileName	name of the save file
pData	pointer to the data to be saved

Returns

MICRO_GAME_AT_ZERO_OK everything is OK
MICRO_GAME_AT_ZERO_INIT_ERROR in the event of a hardware initialization error

Example

```
#include “microGameAtZero/microGameAtZero.h”


MICROGAMEATZERO *game = MICROGAMEATZERO::getInstance();

char data[] = {“test file”};

Game->saveFile(FILE_NAME,data);
```

loadFile

char * MICROGAMEATZERO::loadFile (char * *pFileName*)

This function loads the selected file.

Parameters

pFileName	name of the file

Returns

char* loaded data

Example

```
#include “microGameAtZero/microGameAtZero.h”


char *data;

MICROGAMEATZERO *game = MICROGAMEATZERO::getInstance();


data = game->loadFile(FILE_NAME);
```

sendSerial

microGameAtZero_err MICROGAMEATZERO::sendSerial (char * *pData*)

This function writes data to the UART interface.

Parameters

pData pointer to the data to be write

Returns

the number of transferred bytes

Example

```
#include “microGameAtZero/microGameAtZero.h”


char data[] = {“Hello world”};

MICROGAMEATZERO *game = MICROGAMEATZERO::getInstance();


game->sendSerial(data);
```

startGame

void startGame ()

This function starts the game loop before calling this function, everything must be initialized.

Parameters

None

Returns

None

Example

```
#include “microGameAtZero/microGameAtZero.h”



MICROGAMEATZERO *game = MICROGAMEATZERO::getInstance();

.

.

.

//after setup and initialization the game

game->startGame();
```

getCollision

vector2 MICROGAMEATZERO::getCollision (Object * *pObj1*,
Object * *pObj2*
)

This function checks if the two passed objects are colliding.

Parameters

pObj1	pointer to the first object
pObj2	pointer to the second object

Returns

specifies in x and y direction how large the overlap of the two objects. If one or both value are 0, there is no collision.

Example

See example folder

getButton

bool MICROGAMEATZERO::getButton (uint8_t *selectButton*)

This function returns the status of the selected button (example A or B)

Parameters

selectButton	selceted button

Returns

True	the button is pressed
False	the button is not pressed

Example

```
#include “microGameAtZero/microGameAtZero.h”



MICROGAMEATZERO *game = MICROGAMEATZERO::getInstance();

bool pressed = game->getButton(B_BUTTON);
```

getJoyPad

bool MICROGAMEATZERO::getJoyPad (uint8_t *direction*)

This function returns the status of the selected joypad direction.

Parameters

direction	selected direction

Returns

True	the direction is pressed
False	the direction is not pressed

Example

```
#include “microGameAtZero/microGameAtZero.h”



MICROGAMEATZERO *game = MICROGAMEATZERO::getInstance();

bool pressed = game-> getJoyPad (LEFT);
```

AUDIO Class

Defines the audio controlling class.

AUDIOENGINE::getInstance

static AUDIOENGINE* AUDIOENGINE::getInstance()

This function returns the instance of the AUDIOENGINE. If no instance exists, the function creates an instance of the AUDIOENGINE.

Parameters

None

Returns

pointer to the AUDIOENGINE instance

Example

```
#include “microGameAtZero/microGameAtZero.h”


AUDIOENGINE*audio = AUDIOENGINE::getInstance();
```

stopPlaying

void AUDIOENGINE::stopPlaying (audioChannel_t *channel*) static

This function turns off the selected channel for audio output.

Parameters

channelselected channel (0 to 7)

Returns

None

Example

See example folder

startPlaying

```
microGameAtZero_err AUDIOENGINE::startPlaying ( audioChannel_t channel,
                                                uint16_t       positionStart,
                                                bool           oneShot
                                              )
```

This function starts the audio playback of the selected channel at the selected position, in the desired mode (one-shot or loop)

Parameters

channel	the channel to be switched on
positionStart	at which position the sound should be the start
oneShot	if true the sound is playing only once otherwise playing in a loop

Returns

MICRO_GAME_AT_ZERO_OK	is everything is ok
MICRO_GAME_AT_ZERO_INVALID_PARAM	if the selected channel is not in the range 0 to 7 or no sound has been loaded into the channel

Example

See example folder

contiuePlay

microGameAtZero_err AUDIOENGINE::continuePlay (audioChannel_t *channel*)

This function starts the selected channel at the last position where it was stopped.

Parameters

channel	selected channel (0 to 7)

Returns

MICRO_GAME_AT_ZERO_OK	is everything is ok
MICRO_GAME_AT_ZERO_INVALID_PARAM	PARAM if the selected channel is not in the range 0 to 7
CHANNEL_IS_PLAYING	if the channel is already playing

Example

See example folder

setSampleRate

microGameAtZero_err AUDIOENGINE::setSampleRate (audioSampleRate_t *sample*)

This function sets the sample rate of the audio output.

Parameters

sample the sample rate of the output (SAMPLE_16_KHZ, SAMPLE_22_KHZ or SAMPLE_44_KHZ)

Returns

MICRO_GAME_AT_ZERO_OK if everything's is okay and otherwise the error code

Example

See example folder

getSampleRate

audioSampleRate_t AUDIOENGINE::getSampleRate ( )

This function returns the set sample rate.

Parameters

None

Returns

SAMPLE_16_KHZ, SAMPLE_22_KHZ or SAMPLE_44_KHZ

Example

See example folder

setChannel

```
microGameAtZero_err AUDIOENGINE::setChannel ( audioChannel_t  channel,
                                              const uint8_t * pSound,
                                              uint8_t         volume,
                                              uint32_t        sizeSound
                                            )
```

This function sets the sound for the selected channel and volume.

Parameters

channel	selected channel (0 to 7)
pSound	pointer to the sound array
volume	volume level to be set range 0 to 100
sizeSound	length of the sound

Returns

MICRO_GAME_AT_ZERO_OK	is everything is ok
MICRO_GAME_AT_ZERO_INVALID_PARAM	if the selected channel is not in the range 0 to 7 or/and the volume is not in the range 0 to 100.

Example

See example folder

setVolumeChannel

```
microGameAtZero_err AUDIOENGINE::setVolumeChannel ( audioChannel_t channel,
                                                    uint8_t        volume
                                                  )
```

This function sets on the selected channel the desired volume.

Parameters

channel selected channel (0 to 7)

volume volume level to be set range 0 to 100

Returns

MICRO_GAME_AT_ZERO_OK	everything is ok
MICRO_GAME_AT_ZERO_INVALID_PARAM	if the selected channel is not in the range 0 to 7 or/and the volume is not in the range 0 to 100.

Example

See example folder

setMainVolume

microGameAtZero_err AUDIOENGINE::setMainVolume (uint8_t *volume*)

This function sets the main volume.

Parameters

volume volume level to be set range 0 to 100

Returns

MICRO_GAME_AT_ZERO_OK	is everything is ok
MICRO_GAME_AT_ZERO_INVALID_PARAM	if the selected volume is not in the range 0 to 100

Example

See example folder

getMainVolume

microGameAtZero_err AUDIOENGINE::getMainVolume ( )

This function returns the main volume level.

Parameters

None

Returns

main volume level

Example

See example folder

isPlaying

bool AUDIOENGINE::isPlaying (audioChannel_t *channel*)

This function returns whether the selected channel is switched on or not.

Parameters

channel selected channel (0 to 7)

Returns

True	if switched on
False	if switched off

Example

See example folder

playBeep

static void playBeep ()

This function starts playing a beep sound in the one-shot mode.

Parameters

None

Returns

None

Example

See example folder

setMute

void AUDIOENGINE::setMute (bool *mute*)

With this function, you can mute the audio output.

Parameters

mute if true is muted otherwise not

Returns

None

Example

See example folder

getMute

bool AUDIOENGINE::getMute ( )

This function returns the mute status.

Parameters

None

Returns

True	is muted
False	is not muted

Example

See example folder

playSound

static void playSound ()

This function sets the audio output with the values of the channels with the adjusted volume.

Parameters

None

Returns

None

Example

See example folder

CAMERA2D Class

Defines a camera that acts as the viewing area of the scene.

CAMERA2D

CAMERA2D ()

Construct a new CAMERA2D::CAMERA2D object.

Parameters

None

Returns

CAMERA2D object.

Example

See example folder

setCamera

void CAMERA2D::setCamera (camera_s *settings*)

This function passes the camera settings to the camera.

Parameters

settings camera settings

Returns

None

Example

See example folder

OBJECT Class

Defines an object with the basic functions.

Do not use this class, but the derived classes. (KinectBody, RigidBody, StaticBody and Area)

Object()

Object()

Construct a new Object:: Object object.

Parameters

None

Returns

Object object

Example

The Object should not be used directly, but the derived classes.

setupCollisionWindow

```
void Object::setupCollisionWindow ( vector2       position,
                                    vector2       size,
                                    collisionType setCollisionType = GROUND_AND_WALL
                                  )
```

This function sets the collision window of the object.

Parameters

Position	This value specifies the position of the collision window in relation to the object center. If x = 0 and y = 0 the collision window is centerd. Otherwise, it will be moved away from the center by the entered value.
size	The size of the collision window
setCollisionType	collision type

Returns

None

Example

See example folder

getCollisionWindow

collisionSettings Object::getCollisionWindow ( )

This function returns the collision window settings.

Parameters

None

Returns

collisionSettings	collision window settings

Example

See example folder

getTexture

uint8_t * Object::getTexture ( ) virtual

This function returns the texture (is a virtual function, see the derived classes)

Parameters

None

Returns

NULL

Reimplemented in KinectBody, StaticBody, and RigidBody.

Example

See example folder

getTransparentColor

int32_t Object::getTransparentColor ( ) virtual

This function returns the transparent color (is a virtual function, see the derived classes)

Parameters

None

Returns

-1

Reimplemented in KinectBody, StaticBody, and RigidBody.

Example

See example folder

getValues

objectSettings Object::getValues ( )

This function returns the object settings (position, mass, and size)

Parameters

None

Returns

objectSettings object settings

Example

See example folder

setHidden

void Object::setHidden (bool *set*)

This function can set the visibility of the object.

Parameters

set if ture the object is hidden, otherwise not.

Returns

None

Example

See example folder

getHidden

bool Object::getHidden ( )

This function returns the hidden status of the object.

Parameters

None

Returns

True	the object is hidden
False	the object is not hidden

Example

See example folder

setPhysic

void Object::setPhysic (physicParam *settings*)

This function sets the physic of the object.

Parameters

settings physic settings

Returns

None

Example

See example folder

getPhysic

physicParam Object::getPhysic ( )

This function returns the physic settings.

Parameters

None

Returns

physicParam physic settings

Example

See example folder

setVelocity

void Object::setVelocity (vector2 *vel*)

This function sets the velocity of the object.

Parameters

vel velocity to be set

Returns

None

Example

See example folder

setGravity

void Object::setGravity (int8_t *grav*)

This function sets the gravity acting on the object.

Parameters

grav	gravity to the object

Returns

None

Example

See example folder

setShowTexture

void Object::setShowTexture (uint8_t *set*)

This function set which texture should be displayed.

Parameters

set index of the texture that should be shown

Returns

None

Example

See example folder

getObjectNumb

uint16_t Object::getObjectNumb (   )

This function returns the object number of the object (identification number).

Parameters

None

Returns

object number

Example

See example folder

setObjectNumb

void Object::setObjectNumb (uint16_t *numb*)

This function sets the object number of the object (identification number).

Parameters

numb object number

Returns

None

Example

See example folder

setPosition

void Object::setPosition (vector2 *position*)

This function sets the object to the new position.

Parameters

position	position to be set

Returns

None

Example

See example folder

KINECTBODY Class

Defines a kinectBody control. Is a derivate of the object class.

KinectBody

KinectBody class is inherited from Object.

KinectBody()

Construct a new KinectBody::KinectBody object.

Parameters

None

Returns

KinectBody object

Example

See example folder

setTexture

```
void KinectBody::setTexture ( vector2     size,
                              uint8_t **  ppTexture,
                              vector2     position,
                              int8_t      mass,
                              int32_t     transparentColor
                            )
```

This function sets the settings of the object with texture, size and position.

Parameters

size	size of the object (texture size == object size)
ppTexture	pointer to the texture array (all textures in the array must have the same size)
position	start position of the object
mass	mass of the object
transparentColor	color that should not be rendered (the transparent color of all texture in the array must be the same)

Returns

None

Example

See example folder

move

vector2 KinectBody::move (vector2 *update*)

This function shifts the position with the given value.

Parameters

update	the position is moved by this value

Returns

Current position

Example

See example folder

setAnimation

```
microGameAtZero_err KinectBody::setAnimation ( uint8_t **  ppTexture,
                                               uint8_t     numbTextures,
                                               uint8_t     changeRate,
                                               int32_t     transparentColor
                                             )
```

This function adds a new animation to the object.

Parameters

ppTexture	list of animation textures
numbTextures	how many textures the animation has
changeRate	how fast the animation should be played
transparentColor	color that should not be rendered (8-bit ture color)

Returns

>= 0	index of the animation
MICRO_GAME_AT_ZERO_INVALID_PARAM	invalid parameter
MICRO_GAME_AT_ZERO_FULL_ERROR	no more room for a new animation

Example

See example folder

removeAnimation

microGameAtZero_err KinectBody::removeAnimation (uint8_t *animationNumber*)

This function removes the selected animation.

Parameters

animationNumber	index of the animation which should be removed

Returns

MICRO_GAME_AT_ZERO_OK	everything is OK
MICRO_GAME_AT_ZERO_INVALID_PARAM	is not existing

Example

See example folder

startAnimation

```
microGameAtZero_err KinectBody::startAnimation ( uint8_t  animationNumber,
                                                 bool     oneShot
                                               )
```

This function starts the selected animation.

Parameters

animationNumber	index of the animation to be starts
oneShot	if true the animation is playing only once otherwise playing in a loop

Returns

MICRO_GAME_AT_ZERO_OK	everything is OK
MICRO_GAME_AT_ZERO_INVALID_PARAM	PARAM invalid parameter

Example

See example folder

stopAnimation

microGameAtZero_err KinectBody::stopAnimation ( )

This function stops the current animation.

Parameters

None

Returns

MICRO_GAME_AT_ZERO_OK	everything is OK
MICRO_GAME_AT_ZERO_INVALID_PARAM	no animation is running

Example

See example folder

animationStatus

int8_t KinectBody::animationStatus ( )

This function returns the index of the currently running animation.

Parameters

None

Returns

-1 no animation is started otherwise the index of the animation

Example

See example folder

getTexture

uint8_t * KinectBody::getTexture ( ) virtual

This function returns the texture to be displayed.

Parameters

None

Returns

uint8_t* pointer to the texture

Reimplemented from Object.

Example

See example folder

getTransparentColor

int32_t KinectBody::getTransparentColor ( ) virtual

This function returns the transparent color of the current texture.

Parameters

None

Returns

transparent color value

Reimplemented from Object.

Example

See example folder

setupCollisionWindow

Public Member Function inherited from Object

getCollisionWindow

Public Member Function inherited from Object

getValues

Public Member Function inherited from Object

setHidden

Public Member Function inherited from Object

getHidden

Public Member Function inherited from Object

setShowTexture

Public Member Function inherited from Object

setPosition

Public Member Function inherited from Object

RIGIDBODY Class

Defines a rigidBody control. Is a derivate of the object class.

RigidBody

Rigid Body class is inherited from Object.

RigidBody()

Construct a new RigidBody::RigidBody object.

Parameters

None

Returns

RigidBodyobject

Example

See example folder

setTexture

```
void RigidBody::setTexture ( vector2     size,
                             uint8_t **  image,
                             vector2     position,
                             int8_t      mass,
                             int32_t     transparentColor
                           )
```

This function sets the settings of the object with texture, size, and position.

Parameters

size	size of the object (texture size == object size)
image	pointer to the texture array (all textures in the array must have the same size)
position	start position of the object
mass	mass of the object
transparentColor	color that should not be rendered (the transparent color of all texture in the array must have the same)

Returns

None

Example

See example folder

getTexture

uint8_t * RigidBody::getTexture ( )

This function returns the currently used texture.

Parameters

None

Returns

uint8_t* used texture

Reimplemented from Object.

Example

See example folder

update

void RigidBody::update (uint32_t *deltaTime*)

This function calculates the new position of the object with the physicparam. The function is called automatically and should not be called by the user.

Parameters

deltaTime The time that has passed since last call

Returns

None

Example

See example folder

setBreak

void RigidBody::setBreak (bool *set*)

This function sets the break. If the break is set on, the update function call is switched off and the position doesn't change.

Parameters

set true break on, false break off

Returns

None

Example

See example folder

getTransparentColor

int32_t RigidBody::getTransparentColor ( )

This function returns the transparent color.

Parameters

None

Returns

int32_t transparent color

Reimplemented from Object.

Example

See example folder

setPhysic

Public Member Function inherited from Object

getPhysic

Public Member Function inherited from Object

setVelocity

Public Member Function inherited from Object

setGravity

Public Member Function inherited from Object

setupCollisionWindow

Public Member Function inherited from Object

getCollisionWindow

Public Member Function inherited from Object

getValues

Public Member Function inherited from Object

setHidden

Public Member Function inherited from Object

getHidden

Public Member Function inherited from Object

setShowTexture

Public Member Function inherited from Object

setPosition

Public Member Function inherited from Object

STATICBODY Class

Defines a staticBody control. Is a derivate of the object class.

StaticBody()

StaticBody class is inherited from Object.

StaticBody ()

Construct a new StaticBody:: StaticBody object.

Parameters

None

Returns

StaticBody object

Example

See example folder

setTexture

```
void StaticBody::setTexture ( vector2    size,
                              uint8_t ** image,
                              vector2    position,
                              int8_t     mass,
                              int32_t    transparentColor
                            )
```

This function sets the settings of the object with texture, size, and position.

Parameters

size	size of the object (texture size == object size)
image	pointer to the texture array (all textures in the array must have the same size)
position	start position of the object
mass	mass of the object
transparentColor	color that should not be rendered (the transparent color of all texture in the array must be the same)

Returns

None

Example

See example folder

getTransparentColor

int32_t StaticBody::getTransparentColor ( ) virtual

This function returns the transparent color (8-bit ture color).

Parameters

None

Returns

int32_t	trapnsparent color

Reimplemented from Object

Example

See example folder

setupCollisionWindow

Public Member Function inherited from Object

getCollisionWindow

Public Member Function inherited from Object

getValues

Public Member Function inherited from Object

setHidden

Public Member Function inherited from Object

getHidden

Public Member Function inherited from Object

setShowTexture

Public Member Function inherited from Object

setPosition

Public Member Function inherited from Object

AREA Class

Defines a area control. Is a derivate of the object class.

AREA()

AREA class is inherited from Object.

AREA ()

Construct a new AREA:: AREA object.

Parameters

None

Returns

AREA object

Example

See example folder

setArea

```
void AREA::setArea ( vector2        size,
                     vector2        position,
                     collisionType  collisionLevel
                   )
```

This function sets the settings of the area.

Parameters

size	the size of the area
position	the position of the area
collisionLevel	what collision level the area has

Returns

None

Example

See example folder

setupCollisionWindow

Public Member Function inherited from Object

getCollisionWindow

Public Member Function inherited from Object

getValues

Public Member Function inherited from Object

setHidden

Public Member Function inherited from Object

getHidden

Public Member Function inherited from Object

setShowTexture

Public Member Function inherited from Object

setPosition

Public Member Function inherited from Object

SCENE Class

Defines scene control.

SCENE()

SCENE()

Construct a new SCENE::SCENE object.

Parameters

None

Returns

SCENE object

Example

See example folder

setBackgroundColor

void SCENE::setBackgroundColor (uint8_t *color*)

This function sets the background color of the scene.

Parameters

color background color to be set (8-bit ture color)

Returns

None

Example

See example folder

getBackgroundColor

uint8_t SCENE::getBackgroundColor ( )

This function returns the current background color.

Parameters

None

Returns

uint8_t	current background color (8-bit ture color)

Example

See example folder

setTileMape

```
microGameAtZero_err SCENE::setTileMap ( tileMap        map,
                                        sceneLayer_t  index
                                       )
```

This function sets the tilemap for the selected layer of the scene.

Parameters

map	tilemap to be set
index	the selected layer (in the moment just one layer available OBJECT_LAYER)

Returns

MICRO_GAME_AT_ZERO_OK	is everything is okey
MICRO_GAME_AT_ZERO_INVALID_PARAM	invalid parameter

Example

See example folder

getTileMape

tileMap ATTR_RAM SCENE::getTileMap (sceneLayer_t *index*)

This function returns the tilemap of the selected layer.

Parameters

index selcted layer

Returns

tileMap	tilemap of the selected layer

Example

See example folder

sceneLogic

virtual function

addStatic

microGameAtZero_err SCENE::addStatic (StaticBody* *pObj*)

This function adds a static object to the scene.

Parameters

pObj	static object to be add

Returns

index of the static object (identification number).

MICRO_GAME_AT_ZERO_INVALID_PARAM	invalid parameter or there is no more room for this object

Example

See example folder

removeStatic

microGameAtZero_err SCENE::removeArea (AREA* *pObj*)

This function remove the passed area from the scene.

Parameters

pObj pointer to the area to be remove

Returns

MICRO_GAME_AT_ZERO_OK	everything is OK
MICRO_GAME_AT_ZERO_INVALID_PARAM	no pObj is passed

Example

See example folder

getStatic

StaticBody* SCENE::getStatic (uint16_t *numbObjects*)

This function returns the static body of the passed index.

Parameters

numbObjects	index of the static body

Returns

if NULL no object on this possition, otherwise the object.

Example

See example folder

getStaticCount

uint16_t SCENE::getStaticCount (   )

This function returns the current number of static body objects included in the scene.

Paramters

None

Returns

current number of static body object in the scene.

Example

See example folder

addTexture

```
microGameAtZero_err SCENE::addTexture ( uint8_t *  pTexture,
                                        int32_t    transparentColor,
                                        bool       wallOrGround
                                      )
```

This function adds a texture tile to the scene.

Parameters

pTexture	pointer to the texture to be add
transparentColor	color that should not be rendered (8-bit ture color)
wallOrGround	if true then the collision type GROUND_AND_WALL is set for this texture (collision window size == texture size)

Returns

position of the texture in the array (identification number).

MICRO_GAME_AT_ZERO_INVALID_PARAM	invalid parameter

Example

See example folder

getTexture

textureTile *ATTR_RAM SCENE::getTexture (uint16_t *numbTexture*)

This function returns the texture tile on the passed index.

Parameters

numbTexture	position of the texture in the array (identification number)

Returns

textureTile*	structure of the texture tile on the passed position is no texture tile on this position a NULL is returned.

Example

See example folder

addKinect

microGameAtZero_err SCENE::addKinect (KinectBody* *pObj*)

This function adds a Kinect body to the scene.

Parameters

pObj pointer to the Kinect body to be add.

Returns

index of the kinect body (identification number).

MICRO_GAME_AT_ZERO_INVALID_PARAM invalid parameter or there is no more space

Example

See example folder

removeKinect

microGameAtZero_err SCENE::removeArea (AREA* *pObj*)

This function remove the passed area from the scene.

Parameters

pObj	pointer to the area to be remove

Returns

MICRO_GAME_AT_ZERO_OK	everything is OK
MICRO_GAME_AT_ZERO_INVALID_PARAM	no pObj is passed

Example

See example folder

getKinect

KinectBody* SCENE::getKinect (uint16_t *numbObjects*)

This function returns the Kinect body on the passed index.

Parameters

numbTexture position of the Kinect body in the array (identification number).

Returns

Kinect body on the passed position, if no Kinect body on this position a NULL is returned.

Example

See example folder

getKinectCount

uint16_t SCENE::getKinectCount ( )

This function returns the current number of Kinect body objects included in the scene.

Parameters

None

Returns

the current number of Kinect body object in the scene

Example

See example folder

addRigid

microGameAtZero_err SCENE::addRigid (RigidBody* *pObj*)

This function adds a rigid body to the scene.

Parameters

pObj pointer to the rigid body to be add

Returns

index of the rigid body (identification number)

MICRO_GAME_AT_ZERO_INVALID_PARAM invalid parameter or there is no more space

Example

See example folder

removeRigid

microGameAtZero_err SCENE::removeRigid (RigidBody* *pObj*)

This function removes the passed rigid body from the scene.

Parameters

pObj pointer to the rigid body to be remove

Returns

MICRO_GAME_AT_ZERO_OK	everything is OK
MICRO_GAME_AT_ZERO_INVALID_PARAM	no pObj is passed

Example

See example folder

getRigid

RigidBody* SCENE::getRigid (uint16_t *numbObjects*)

This function returns the rigid body on the passed index.

Parameters

numbTexture position of the rigid body in the array (identification number).

Returns

rigid body on the passed position, if no rigid body on this position a NULL is returned.

Example

See example folder

getRigidCount

uint16_t SCENE::getRigidCount ( )

This function returns the current number of rigid body objects included in the scene.

Parameters

None

Returns

current number of rigid body object in the scene.

Example

See example folder

addArea

microGameAtZero_err SCENE::addArea (AREA* *pObj*)

This function adds an area to the scene.

Parameters

pObj	pointer to the area to be add.

Returns

index of the area (identification number).

MICRO_GAME_AT_ZERO_INVALID_PARAM	invalid parameter or there is no more space

Example

See example folder

removeArea

microGameAtZero_err SCENE::removeArea (AREA* *pObj*)

This function remove the passed area from the scene.

Parameters

pObj pointer to the area to be remove

Returns

MICRO_GAME_AT_ZERO_OK	everything is OK
MICRO_GAME_AT_ZERO_INVALID_PARAM	no pObj is passed

Example

See example folder

getArea

AREA* SCENE::getArea (uint16_t *numbObjects*)

This function returns the area on the passed index.

Parameters

numbTexture position of the area in the array (identification number).

Returns

area on the passed position, if no area on this position a NULL is returned.

Example

See example folder

getAreaCount

uint16_t SCENE::getAreaCount ( )

This function returns the current number of area objects included in the scene.

Parameters

None

Returns

current number of area object in the scene.

Example

See example folder

getSceneParam

vector2 SCENE::getSceneParam ( )

This function returns the scene size.

Parameters

None

Returns

scene size.

Example

See example folder

addCamera

microGameAtZero_err SCENE::addCamera (CAMERA2D* *pCam*)

This function adds a 2D camera to the scene.

Parameters

pCam pointer to the camera to be add.

Returns

MICRO_GAME_AT_ZERO_OK	everything is ok
MICRO_GAME_AT_ZERO_INVALID_PARAM	invalid parameter

Example

See example folder

getCamera

CAMERA2D* SCENE::getCamera ( )

This function returns the connected camera.

Parameters

None

Returns

CAMERA2D*

Example

See example folder

addUI

microGameAtZero_err SCENE::addUI (UI * *pUi*)

This function adds a UI interface to the scene.

Parameters

pUi pointer to the UI interface to be add.

Returns

MICRO_GAME_AT_ZERO_OK	everything is ok
MICRO_GAME_AT_ZERO_INVALID_PARAM	invalid parameter

Example

See example folder

getUI

UI ATTR_RAM * SCENE::getUI ( )

This function returns the UI interface of the scene.

Parameters

None

Returns

ui interface of the scene

Example

See example folder

moveCollisionWallGround

vector2 SCENE::moveCollisionWallGround (KinectBody* *pObject*,
vector2 *positionUpdate*
)

This function moves the body by the given value and checks if it a collision with tiles where the collision type GROUND_AND_WALL is enabled. If a collision is detected, the Kinect body will be position on the ground or/and near the wall.

Parameters

pObject	Kinect body to be move
positionUpdate	the position is move by this value

Returns

new position of the object

Example

See example folder

UI Class

Defines the UI control.

UI()

UI class is inherited from Object.

UI ()

Construct a new UI:: UI object.

Parameters

None

Returns

UI object

Example

See example folder

addButton

microGameAtZero_err UI::addButton (BUTTON* *pButton*)

This function adds a new button to the UI interface.

Parameters

pButton	pointer to the button to be add

Returns

position of the button in the array (identification number).

MICRO_GAME_AT_ZERO_FULL_ERROR	no more space or pButton is NULL

Example

See example folder

setHidenButton

```
microGameAtZero_err UI::setHiddenButton ( uint8_t  numberButton,
                                          bool     hidden
                                        )
```

This function sets the hidden flag of the selected button.

Parameters

numberButton	index of the selected button object
hiden	if ture the button is hidden otherwise it will be shown.

Returns

MICRO_GAME_AT_ZERO_OK	everything is OK
MICRO_GAME_AT_ZERO_INVALID_PARAM	invalide parameter

Example

See example folder

modifyButtonText

```
microGameAtZero_err UI::modifyButtonText ( uint8_t  numberButton,
                                           char *   pText
                                         )
```

This function changes the text on the selected button.

Parameters

numberButton	index of the selected button object
pText	new text

Returns

MICRO_GAME_AT_ZERO_OK	everything is OK
MICRO_GAME_AT_ZERO_INVALID_PARAM	invalid parameter

Example

See example folder

removeButton

microGameAtZero_err UI::removeButton (uint8_t *numberButton*)

This function removes the selected button from the UI interface.

Parameters

numberButton index of the selected button object

Returns

MICRO_GAME_AT_ZERO_OK	everything is OK
MICRO_GAME_AT_ZERO_INVALID_PARAM	invalid parameter

Example

See example folder

setHighLightButton

```
microGameAtZero_err UI::setHighLightButton ( uint8_t  numberButton,
                                             bool     high
                                           )
```

This function sets the highlight status of the selected button.

Parameters

numberButton	index of the selected button object
high	if true the highlight is on otherwise not

Returns

MICRO_GAME_AT_ZERO_OK	everything is OK
MICRO_GAME_AT_ZERO_INVALID_PARAM	PARAM invalid parameter

Example

See example folder

getButton

BUTTON * UI::getButton (uint8_t *numberButton*)

This function returns the button object on the selected position.

Parameters

numberButton	index of the selected button object

Returns

button object on the selected position or NULL

Example

See example folder

getButtonAmount

uint8_t UI::getButtonAmount ( )

This function returns the number of buttons in the UI interface.

Parameters

None

Returns

uint8_t number of buttons

Example

See example folder

addCursor

microGameAtZero_err UI::addCursor (cursor_ui *settings*)

This function adds a new cursor to the UI interface.

Parameters

settings	cursor settings

Returns

MICRO_GAME_AT_ZERO_OK	everything is ok

Example

See example folder

moveCursorTo

void UI::moveCursorTo (vector2 *position*)

This function moves the cursor to the passed position.

Parameters

position	new cursor position

Returns

None

Example

See example folder

setHidenCursor

void UI::setHiddenCursor (bool *hidde*)

This function sets the hidden flag of the cursor.

Parameters

hiden if ture the cursor is hidden otherwise it will be shown.

Returns

None

Example

See example folder

getCursor

cursor_ui UI::getCursor ( )

This function returns the cursor settings.

Parameters

None

Returns

cursor_ui cursor settings.

Example

See example folder

addImage

microGameAtZero_err UI::addImage (IMAGE * *pNewImage*)

This function adds a new image to the UI interface.

Parameters

pNewImage	pointer to the new image to be add

Returns

position of the image in the array (identification number).

MICRO_GAME_AT_ZERO_FULL_ERROR	no more space or pNewImage is NULL

Example

See example folder

setHidenImage

```
microGameAtZero_err UI::setHiddenImage ( uint8_t  numberImgae,
                                         bool     hidden
                                       )
```

This function sets the hidden flag of the selected image.

Parameters

numberImgae	index of the selected image object
hidden	if true the image is hidden otherwise it will be shown.

Returns

MICRO_GAME_AT_ZERO_OK	everything is OK
MICRO_GAME_AT_ZERO_INVALID_PARAM	invalide parameter

Example

See example folder

modifyImage

```
microGameAtZero_err UI::modifyImage ( uint8_t    numbImage,
                                      uint8_t *  pTexture
                                    )
```

This function sets a new texture to the selected image.

Parameters

numbImage	index of the selected image object
pTexture	pointer to the new texture

Returns

MICRO_GAME_AT_ZERO_OK	everything is ok
MICRO_GAME_AT_ZERO_INVALID_PARAM	PARAM no more space or invalide parameter

Example

See example folder

removeImage

microGameAtZero_err UI::removeImage (uint8_t *imageNumber*)

This function removes the selected image from the UI interface.

Parameters

imageNumber index of the selected image object

Returns

MICRO_GAME_AT_ZERO_OK	everything is ok
MICRO_GAME_AT_ZERO_INVALID_PARAM	invalide parameter

Example

See example folder

getImage

IMAGE* UI::getImage (uint8_t *imageNumber*)

This function returns the selected image object.

Parameters

imageNumber	index of the selected image object

Returns

image object or NULL

Example

See example folder

getImageAmount

uint8_t UI::getImageAmount ( )

This function returns the number of images in the UI interface.

Parameters

None

Returns

number of images

Example

See example folder

addText

microGameAtZero_err UI::addText (TEXT * *pNewText*)

This function adds a new text object to the UI interface.

Parameters

pNewText	pointer to the new text object to be add

Returns

position of the text object in the array (identification number).

MICRO_GAME_AT_ZERO_FULL_ERROR	no more space or pNewText is NULL

Example

See example folder

setHidenText

```
microGameAtZero_err UI::setHiddenText ( uint8_t  numberText,
                                        bool     hidden
                                      )
```

This function sets the hidden flag of the selected text object.

Parameters

numberText	index of the selected text object
hidden	if ture the text object is hidden otherwise it will be shown.

Returns

MICRO_GAME_AT_ZERO_OK	everything is OK
MICRO_GAME_AT_ZERO_INVALID_PARAM	invalide parameter

Example

See example folder

modifyText

microGameAtZero_err UI::modifyText (uint8_t *textNumber*,
char * *pText*
)

This function changes the text of the selected text object.

Parameters

textNumber	index of the selected text object
pText	pointer to the new text to be set

Returns

MICRO_GAME_AT_ZERO_OK	everything is OK
MICRO_GAME_AT_ZERO_INVALID_PARAM	PARAM invalide parameter

Example

See example folder

removeText

microGameAtZero_err UI::removeText (uint8_t *textNumber*)

This function removes the selected text object from the UI interface.

Parameters

textNumber index of the selected text object

Returns

MICRO_GAME_AT_ZERO_OK	everything is OK
MICRO_GAME_AT_ZERO_INVALID_PARAM	PARAM invalide parameter

Example

See example folder

getText

TEXT* UI::getText (uint8_t *textNumber*)

This function returns the selected text object.

Parameters

textNumber index of the selected text object

Returns

text object

Example

See example folder

getTextAmount

uint8_t UI::getTextAmount ( )

This function returns the amount of the text object in the UI interface.

Parameters

None

Returns

amount of the text objects

Example

See example folder

addNumber

microGameAtZero_err UI::addNumber (NUMBER * *pNewNumber*)

This function adds a new number object to the UI interface.

Parameters

pNewNumber	pointer to the new number object to be add

Returns

position of the number object in the array (identification number).

MICRO_GAME_AT_ZERO_FULL_ERROR	no more space or pNewNumber is NULL

Example

See example folder

setHidenNumber

```
microGameAtZero_err UI::setHiddenNumber ( uint8_t  numberNumber,
                                          bool     hidden
                                        )
```

This function sets the hidden flag of the selected number object.

Parameters

numberNumber	index of the selected number object
hidden	if ture the number object is hidden otherwise it will be shown.

Returns

MICRO_GAME_AT_ZERO_OK	everything is OK
MICRO_GAME_AT_ZERO_INVALID_PARAM	invalide parameter

Example

See example folder

modifyNumber

```
microGameAtZero_err UI::modifyNumber ( uint8_t   numberNumber,
                                       uint32_t  number
                                     )
```

This function changes the registered number of the selected number object.

Parameters

numberNumber	index of the selected number object
number	new number

Returns

MICRO_GAME_AT_ZERO_OK	everything is ok
MICRO_GAME_AT_ZERO_INVALID_PARAM	invalid parameter

Example

See example folder

removeNumber

microGameAtZero_err UI::removeNumber (uint8_t *numberNumber*)

This function removes the selected number object from the UI interface.

Parameters

numberNumber	index of the selected number object

Returns

MICRO_GAME_AT_ZERO_OK	everything is ok
MICRO_GAME_AT_ZERO_INVALID_PARAM	invalid parameter

Example

See example folder

getNumber

NUMBER * UI::getNumber (uint8_t *numberNumber*)

This function returns the selected number object.

Parameters

numberNumber	index of the selceted number object

Returns

number object or NULL

Example

See example folder

getNumberAmount

uint8_t UI::getNumberAmount ( )

This function returns the amount of number object in the UI interface.

Parameters

None

Returns

amount of number objects

Example

See example folder

addRect

microGameAtZero_err UI::addRect (RECT * *pNewRect*)

This function adds a new rect object to the UI interface.

Parameters

pNewRect	pointer to the new rect object to be add

Returns

position of the rect object in the array (identification number).

MICRO_GAME_AT_ZERO_FULL_ERROR	no more space or pNewRect is NULL

Example

See example folder

setHidenRect

microGameAtZero_err UI::setHiddenRect (uint8_t *numberRect*,
bool *hidden*
)

This function sets the hidden flag of the selected rect object.

Parameters

numberRect	index of the selected rect object
hidden	if ture the rect object is hidden otherwise it will be shown.

Returns

MICRO_GAME_AT_ZERO_OK	everything is OK
MICRO_GAME_AT_ZERO_INVALID_PARAM	invalide parameter

Example

See example folder

modifyFillArea

```
microGameAtZero_err UI::modifyFillArea ( uint8_t  numberRect,
                                         uint8_t  percent
                                       )
```

This function changes the fill level of the selected rect object.

Parameters

numberRect	index of the selected rect object
percent	new fill level in percent

Returns

MICRO_GAME_AT_ZERO_OK	everything is OK
MICRO_GAME_AT_ZERO_INVALID_PARAM	invalid parameter

Example

See example folder

modifyFillColor

```
microGameAtZero_err UI::modifyFillColor ( uint8_t  numberRect,
                                          uint8_t  color
                                        )
```

This function changes the fill color of the selected rect object.

Parameters

numberRect	index of the selected rect object
color	new fill color

Returns

MICRO_GAME_AT_ZERO_OK	everything is OK
MICRO_GAME_AT_ZERO_INVALID_PARAM	invalid parameter

Example

See example folder

modifyLineColor

```
microGameAtZero_err UI::modifyLineColor ( uint8_t  numberRect,
                                           uint8_t  color
                                         )
```

This function changes the line color of the selected rect object.

Parameters

numberRect	index of the selected rect object
color	new line color

Returns

MICRO_GAME_AT_ZERO_OK	everything is OK
MICRO_GAME_AT_ZERO_INVALID_PARAM	invalid parameter

Example

See example folder

removeRect

microGameAtZero_err UI::removeRect (uint8_t *numberRect*)

This function removes the selected rect object from the UI interface.

Parameters

numberRect	index of the selected rect object

Returns

MICRO_GAME_AT_ZERO_OK	everything is OK
MICRO_GAME_AT_ZERO_INVALID_PARAM	invalid parameter

Example

See example folder

getRect

RECT * UI::getRect (uint8_t *numberRect*)

This function returns the selected rect object.

Parameters

numberRect	index of the selceted rect object

Returns

rect object

Example

See example folder

getRectAmount

uint8_t UI::getRectAmount ( )

This function returns the amount of the rect object in the UI interface.

Parameters

None

Returns

amount of rect objects

Example

See example folder

addCheckBox

microGameAtZero_err UI::addCheckBox (CHECKBOX * *pNewCheck*)

This function adds a new checkbox object to the UI interface.

Parameters

pNewCheck	pointer to the new checkbox object to be add

Returns

position of the checkbox object in the array (identification number).

MICRO_GAME_AT_ZERO_FULL_ERROR	no more space or pNewCheck is NULL

Example

See example folder

setHidenCheck

```
microGameAtZero_err UI::setHiddenCheck ( uint8_t  numberCheck,
                                         bool     hidden
                                       )
```

This function sets the hidden flag of the selected checkbox object.

Parameters

numberCheck	index of the selected checkbox object
hiden	if ture the checkbox object is hidden otherwise it will be shown.

Returns

MICRO_GAME_AT_ZERO_OK	everything is OK
MICRO_GAME_AT_ZERO_INVALID_PARAM	invalid parameter

Example

See example folder

setCheck

```
microGameAtZero_err UI::setCheck ( uint8_t  numberCheck,
                                   bool     check
                                 )
```

This function sets the checked status of the selected checkbox object.

Parameters

numberCheck	index of the selected checkbox object
check	if true the checkbox is checked otherwise not

Returns

MICRO_GAME_AT_ZERO_OK	everything is OK
MICRO_GAME_AT_ZERO_INVALID_PARAM	invalid parameter

Example

See example folder

removeCheck

microGameAtZero_err UI::removeCheck (uint8_t *numberCheck*)

This function removes the selected checkbox from the UI interface.

Parameters

numberCheck incdex of the selected check box object

Returns

MICRO_GAME_AT_ZERO_OK	everything is OK
MICRO_GAME_AT_ZERO_INVALID_PARAM	invalid parameter

Example

See example folder

getCheckBox

CHECKBOX * UI::getCheckBox (uint8_t *numberCheck*)

This function returns the selected checkbox object.

Parameters

numberCheck	index of the selected checkbox object.

Returns

checkbox object

Example

See example folder

getCheckBoxAmount

uint8_t UI::getCheckBoxAmount ( )

This function returns the amount of checkbox object in the UI interface.

Parameters

None

Returns

amount of checkbox objects

Example

See example folder

getKeyBoardShow

bool UI::getKeyBoardShow ( )

This function returns the fade status of the keyboard.

Parameters

None

Returns

True	the keyboard is fade out
False	the keyboard is fade in

Example

See example folder

keyBoardUp

```
void UI::keyBoardUp ( char *    pTitle,
                      char *    pInput,
                      uint8_t   maxInput
                    )
```

This function fades out the keyboard with the passed title and input text.

Parameters

pTitle	pointer to the title text to be set
pInput	pointer to the input text to be set
maxInput	max amount of input chars

Returns

None

Example

See example folder

inputTextKeyBoard

bool UI::inputTextKeyBoard (char * *pInputText*)

This function returns the input text if the enter key is press.

Parameters

pInputText	pointer to the input text

Returns

True	the enter key is press
False	the enter key is not press

Example

See example folder

UI BUTTON Class

Defines a button UI control.

BUTTON(…)

Construct a new BUTTON::BUTTON object.

Parameters

settings button settings

Returns

None

Example

See example folder

setText

microGameAtZero_err BUTTON::setText (char *text*[MAX_BUTTON_TEXT])

This function sets the button text.

Parameters

text	button text to be set

Returns

MICRO_GAME_AT_ZERO_OK	everything is ok
MICRO_GAME_AT_ZERO_INVALID_PARAM	no text

Example

See example folder

getButtonSettings

button_ui BUTTON::getButtonSettings ( )

This function returns the button settings.

Parameters

None

Returns

button setting sturct

Example

See example folder

setHiden

void BUTTON::setHidden (bool *hidden*)

This function set the hidden flag of the button.

Parameters

hidden if ture the button is hidden otherwise it will be shown.

Returns

None

Example

See example folder

getHiden

bool BUTTON::getHidden ( )

This function returns the status of the hidden flag.

Parameters

None

Returns

True	button is hidden
False	button is not hidden

Example

See example folder

setHighLight

void BUTTON::setHighLight (bool *light*)

This function set the button highlight flag.

Parameters

light if true the button is highlighted otherwise not.

Returns

None

Example

See example folder

getHighLight

bool BUTTON::getHighLight (   )

This function returns the status of the highlight flag.

Parameters

None

Returns

True	button is highlighted
False	button is not highlighted

Example

See example folder

UI CHECKBOX Class

Defines a checkbox UI control.

CHECKBOX(…)

CHECKBOX::CHECKBOX (checkBox_ui *settings*)

Construct a new CHECKBOX::CHECKBOX object.

Parameters

settings check box settings

Returns

CHECKBOX*

Example

See example folder

changeCheck

void CHECKBOX::changeCheck (bool *check*)

This function set the checkbox status.

Parameters

check	if true the checkbox is checked otherwise is not checked

Returns

None

Example

See example folder

getCheckSettings

checkBox_ui CHECKBOX::getCheckSettings ( )

This function returns the checkbox settings.

Parameters

None

Returns

checkbox setting structure

Example

See example folder

setHiden

void CHECKBOX::setHidden (bool *hidden*)

This function set the hidden flag of the checkbox.

Parameters

hiden if true the checkbox is hidden otherwise it will be shown.

Returns

None

Example

See example folder

getHiden

bool CHECKBOX::getHidden ( )

This function returns the hidden status of the checkbox.

Parameters

None

Returns

True	checkbox is hidden
False	checkbox is not hidden

Example

See example folder

UI IMAGE Class

Defines an image UI control.

IMAGE(...)

IMAGE::IMAGE (image_ui *setting*)

Construct a new IMAGE::IMAGE object.

Parameters

setting image settings

Returns

IMAGE*

Example

See example folder

changeImage

microGameAtZero_err IMAGE::changeImage (uint8_t * *pTexture*)

This function changes the texture of the image object.

Parameters

texture	texture to be set

Returns

MICRO_GAME_AT_ZERO_OK	everything is ok
MICRO_GAME_AT_ZERO_INVALID_PARAM	pTexture is NULL

Example

See example folder

getImageSettings

image_ui IMAGE::getImageSettings ( )

This function returns the image values.

Parameters

None

Returns

image_ui structure of the image values

Example

See example folder

setHiden

void IMAGE::setHidden (bool *hidden*)

This function set the hidden flag of the image.

Parameters

hidden if ture the image is hidden otherwise it will be shown.

Returns

None

Example

See example folder

getHiden

bool IMAGE::getHidden ( )

This function returns the hidden status of the image.

Parameters

None

Returns

True	image is hidden
False	image is not hidden

Example

See example folder

UI NUMBER Class

Defines a number UI control.

NUMBER(…)

NUMBER::NUMBER (number_ui *settings*)

Construct a new NUMBER::NUMBER object.

Parameters

settings number settings

Returns

NUMBER*

Example

See example folder

changeNumber

microGameAtZero_err NUMBER::changeNumber (uint32_t *number*)

This function changes the registered number to the passed.

Parameters

number	the number to be set

Returns

MICRO_GAME_AT_ZERO_OK	everything is ok

Example

See example folder

getNumberSettings

number_ui NUMBER::getNumberSettings ( )

This function returns the number object settings.

Parameters

None

Returns

number object settings

Example

See example folder

getNumber

uint32_t NUMBER::getNumber (   )

This function returns the registered number.

Parameters

None

Returns

registered number

Example

See example folder

setHiden

void NUMBER::setHidden (bool *hidden*)

This function can set the hidden flag of the number object.

Parameters

hidden if ture the number object is hidden otherwise it will be shown.

Returns

None

Example

See example folder

getHiden

bool NUMBER::getHidden ( )

This function returns the hidden status of the number object.

Parameters

None

Returns

True	number object is hidden
False	number object is not hidden

Example

See example folder

UI RECT Class

Defines a rect UI control.

RECT(...)

RECT::RECT (rect_ui *setting*)

Construct a new RECT::RECT object.

Parameters

setting rectangle settings

Returns

RECT*

Example

See example folder

setFillSize

microGameAtZero_err RECT::setFillSize (uint8_t *percent*)

This function sets the fill level of the rectangle in percent (horizontal).

Parameters

percent	percent to be fill

Returns

MICRO_GAME_AT_ZERO_OK	everything is ok
MICRO_GAME_AT_ZERO_INVALID_PARAM	the input was > 100

Example

See example folder

setFillColor

void RECT::setFillColor (uint8_t *color*)

This function sets the fill color (true 8-bit color)

Parameters

color the fill color to be set (ture 8-bit color)

Returns

None

Example

See example folder

setLineColor

void RECT::setLineColor (uint8_t *color*)

This function sets the line color of the rectangle.

Parameters

color	the line color to be set (true 8-bit color)

Returns

None

Example

See example folder

getRectSettings

rect_ui RECT::getRectSettings ( )

This function returns the rectangle object settings.

Parameters

None

Returns

structur of the rectangle object settings

Example

See example folder

setHiden

void RECT::setHidden (bool *hidden*)

This function sets the hidden flag of the rectangle object.

Parameters

hidden if ture the rectangle object is hidden otherwise it will be shown.

Returns

None

Example

See example folder

getHiden

bool RECT::getHidden ( )

This function returns the hidden status of the rectangle object.

Parameters

None

Returns

True	rectangle object is hidden
False	rectangle object is not hidden

Example

See example folder

UI TEXT Class

Defines a text UI control.

TEXT(...)

TEXT::TEXT (text_ui *settings*)

Construct a new TEXT::TEXT object.

Parameters

settings

Returns

TEXT*

Example

See example folder

changeText

microGameAtZero_err TEXT::changeText (char * *pText*)

This function changes the registered text to the passed text.

Parameters

pText	pointer to the new text

Returns

MICRO_GAME_AT_ZERO_OK	everything is ok
MICRO_GAME_AT_ZERO_INVALID_PARAM	pText is NULL

Example

See example folder

getTextSettings

text_ui TEXT::getTextSettings ( )

This function returns the text object settings.

Parameters

None

Returns

structure of the text object settings

Example

See example folder

setHiden

void TEXT::setHidden (bool *hidden*)

This function sets the hidden flag of the text object.

Parameters

hidden if ture the text object is hidden otherwise it will be shown.

Returns

None

Example

See example folder

getHiden

bool TEXT::getHidden ( )

This function returns the hidden status of the text object.

Parameters

None

Returns

True	text object is hidden
False	text object is not hidden

Example

See example folder

Vector2

Defines a 2-element int16_t point vector.

operator+

void vector2::operator+ (const vector2 & *v*)

added with the passed vector2

Parameters

v vector2 to add

Returns

None

Example

See example folder

void vector2::operator+ (int16_t *s*)

This function added the passed int16_t value to x and y.

Parameters

s int16_t value to add

Returns

None

Example

See example folder

operator-

void vector2::operator- (const vector2 & *v*)

subtract with the passed vector2

Parameters

v	vector2 to subtract

Returns

None

Example

See example folder

void vector2::operator- (int16_t *s*)

This function subtract the passed int16_t value to x and y.

Parameters

s	int16_t value to subtract

Returns

None

Example

See example folder

operatore==

bool vector2::operator== (const vector2 & *v*)

This function compares if the passed vector2 is equal.

Parameters

v	to compare

Returns

True	is equal
False	is not equal

Example

See example folder

*operator**

void vector2::operator* (int16_t *s*)

This function multiplies the passed int16_t value to x and y.

Parameters

v vector2 to multiplies

Returns

None

Example

See example folder

operator/

void vector2::operator/ (int16_t *s*)

This function divide the passed int16_t value to x and y.

Parameters

v vector2 to divide

Returns

None

Example

See example folder

set

void vector2::set (const vector2 & *v*)

This function set the values to the passed vector2.

Parameters

v	vector2 to which is set

Returns

None

Example

See example folder

void vector2::set (int16_t _*x* = 0,
int16_t _*y* = 0
)

This function set x and y to the passed x and y value.

Parameters

_x	new x value
_y	new y value

Returns

None

Example

See example folder

settingsEngine

microGameAtZeroTargetSettings.h

```
struct settingsEngine {
    uint16_t screenX = 0;
    uint16_t screenY = 0;
    uint8_t maxFps = 0;
};
```

soundChannel

audioEngine.h

```
struct soundChannel
{
    const uint8_t *sound = nullptr;
    uint32_t size = 0;
    uint8_t volumeChannel = 0;
    bool playing = false;
    uint32_t positionCount = 0;
    bool oneShot = false;
};
```

camera_s

camera2d.h

```
struct camera_s {
    bool cameraOn = false;
    KinectBody*objectToCamera = NULL;
    vector2 notFollowAreaRect;
    vector2 viewSize;
    vector2 position;
    vector2 mapeSize;
    vector2 tileSize;
};
```

cameraAreaRect

camera2d.h

```
struct cameraAreaRect {
    uint16_t min[2] = {0,0};
    uint16_t max[2] = {0,0};
};
```

animation

KinectBody.h

```
struct animation
{
    uint8_t numbTextures = 0;
    uint8_t changeRate = 0;
    int32_t transparentColor = -1;
    uint8_t **ppTexture  = NULL;
    bool inUse = false;
    bool oneShot = false;
};
```

texture

Object.h

```
struct texture
{
    uint8_t **ppTexture;
    int32_t transparentColor = -1;
};
```

objectSettings

Object.h

```
struct objectSettings
{
    vector2 position;
    vector2 size;
    int8_t mass = 0;
};
```

collisionSettings

Object.h

```
struct collisionSettings
{
    vector2 position;
    vector2 size;
    collisionType  collisionLevel = GROUND_AND_WALL;
};
```

physicParam

Object.h

```
struct physicParam{
    vector2 velocity;
    int8_t gravity = 0;
};
```

textureTile

scene.h

```
struct textureTile {
    uint8_t *texture;
    int32_t transparentColor = -1;
    collisionType collision;
};
```

tileMap

scene.h

```
struct tileMap {
    vector2 tileSize;
    vector2 amountTile;
    uint8_t  *order = NULL;
};
```

timerSettings

timerEngine.h

```
struct timerSettings {
    uint8_t timerId = 0;
    uint16_t timerValue = 0;
    timerCallback functionCall = NULL;
};
```

button_ui

button.h

```
struct button_ui {
    bool Hidden = false;
    uint8_t buttonColor = 0;
    uint8_t highLightColor = 0;
    uint8_t highLightWidth = 0;
    bool highLightOn = false;
    vector2 position ;
    vector2 size;
    uint8_t textColor = 0;
    fontType font;
    char *pText;
};
```

checkbox_ui

checkBox.h

```
struct checkBox_ui {
    bool hidden = false;
    bool check = false;
    uint8_t *checkBox;
    vector2 position ;
};
```

image_ui

image.h

```
struct image_ui {
    bool hidden = false;
    uint8_t *pTexture;
    int32_t transparentColor = -1;
    vector2 size;
    vector2 position;
};
```

number_ui

number.h

```
struct number_ui {
    bool hidden = false;
    vector2 position ;
    uint8_t textColor = 0;
    fontType font;
    uint32_t number = 0;
};
```

rect_ui

rect.h

```
struct rect_ui {
    bool hidden = false;
    uint8_t colorFill = 0;
    uint8_t colorLine = 0;
    uint8_t lineWidth = 1;
    uint16_t fillArea = 0;
    vector2 size;
    vector2 position;
};
```

text_ui

text.h

```
struct text_ui {
    bool hidden = 0;
    vector2 position ;
    uint8_t textColor = 0xFF;
    fontType font;
    char *pText;
};
```

cursor_ui

ui.h

```
struct cursor_ui {
    bool hidden = true;
    vector2 position;
    uint8_t *textureCursor = NULL;
    vector2 sizeTexture;
    int32_t transparentColor = -1;
    char textCursor = 0;
    uint8_t fontColor = 0xff;
};
```

audioChannel_t

auidoEngine.h

```
enum audioChannel_t {
    CHANNEL1 = 0,
    CHANNEL2 = 1,
    CHANNEL3 = 2,
    CHANNEL4 = 3,
    CHANNEL5 = 4,
    CHANNEL6 = 5,
    CHANNEL7 = 6,
    CHANNEL8 = 7,
    MAX_AUDIO_CHANNELS
};
```

collisionType

Object.h

```
enum collisionType
{
    COLLISION_OFF = 0,
    GROUND_AND_WALL = 1,
    PLAYER = 2,
    ENEMEY = 4,
    OBJECT = 8,
    OTHER = 16
};
```

objects_t

scene.h

```
enum objects_t {
    KINECT = 0,
    RIGID  = 1,
    STATIC = 2,
    OBJECTS_TYPES
};
```

sceneLayer_t

scene.h

```
enum sceneLayer_t {
    BACKGROUND_LAYER = 0,
    OBJECT_LAYER = 1,
    MAX_LAYER
};
```

fontType

font.h

```
enum fontType{
    FONT_10 = 0,
    FONT_20 = 1,
    FONT_MAX
};
```

audioSampleRate_t (OdroidGo)

microGameAtZeroSettings.h

```
enum audioSampleRate_t
{
    SAMPLE_16_KHZ = 0,
    SAMPLE_22_KHZ = 1,
    SAMPLE_44_KHZ = 2,
    MAX_SAMPLE_RATE
};
```

direction_t (OdroidGo)

microGameAtZeroSettings.h

```
enum direction_t
{
    UP = 1,
    DOWN,
    LEFT,
    RIGHT
};
```

button_t (OdroidGo)

microGameAtZeroSettings.h

```
enum button_t
{
    MENU = 0,
    VOLUME,
    SELECT,
    START,
    B_BUTTON,
    A_BUTTON,
    MAX_INPUT
};
```

externalButton_t (OdroidGo)

microGameAtZeroSettings.h

```
enum externalButton_t
{
    EXTERNAL_A = 0,
    EXTERNAL_B,
    EXTERNAL_C
};
```

microGameAtZero_err

typedef int8_t microGameAtZero_err

Error Code:

```
#define MICRO_GAME_AT_ZERO_FULL_ERROR        -7
#define MICRO_GAME_AT_ZERO_DIR_ERROR         -6
#define MICRO_GAME_AT_ZERO_NO_SAVE           -5
#define MICRO_GAME_AT_ZERO_READ_ERROR        -4
#define MICRO_GAME_AT_ZERO_INIT_ERROR        -3
#define MICRO_GAME_AT_ZERO_SEND_ERROR        -2
#define MICRO_GAME_AT_ZERO_INVALID_PARAM     -1
#define MICRO_GAME_AT_ZERO_OK                 0
```