Cgame Api

Generated by Doxygen 1.9.1

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

Chapter 3

Data Structure Documentation

3.1 _boundingBox Struct Reference

`#include <GameCore.h>`

Data Fields

- int h
- int w
- int degree
- SDL_Point coords [4]
- SDL_Point originCoords [4]
- struct SDL_Point center

3.1.1 Field Documentation

3.1.1.1 center

`struct SDL_Point _boundingBox::center`

3.1.1.2 coords

`SDL_Point _boundingBox::coords[4]`

3.1.1.3 degree

```
int _boundingBox::degree
```

3.1.1.4 h

```
int _boundingBox::h
```

3.1.1.5 originCoords

```
SDL_Point _boundingBox::originCoords[4]
```

3.1.1.6 w

```
int _boundingBox::w
```

The documentation for this struct was generated from the following file:

- GameCore.h

3.2 _CoreGameData Struct Reference

```
#include <GameCore.h>
```

Data Fields

- int map [mapsize][mapsize]
- SDL_Texture * GroundSheet
- Trigger triggerList [20]
- node * start
- Entity inventory
- int triggerCount
- MapTile Tiles [100]
- int window_h
- int window_w
- Room * currentRoom

3.2.1 Field Documentation

3.2.1.1 currentRoom

```
Room* _CoreGameData::currentRoom
```

3.2.1.2 GroundSheet

```
SDL_Texture* _CoreGameData::GroundSheet
```

3.2.1.3 inventory

```
Entity _CoreGameData::inventory
```

3.2.1.4 map

```
int _CoreGameData::map[mapsize][mapsize]
```

3.2.1.5 start

```
node* _CoreGameData::start
```

3.2.1.6 Tiles

```
MapTile _CoreGameData::Tiles[100]
```

3.2.1.7 triggerCount

```
int _CoreGameData::triggerCount
```

3.2.1.8 triggerList

```
Trigger _CoreGameData::triggerList[20]
```

3.2.1.9 window_h

```
int _CoreGameData::window_h
```

3.2.1.10 window_w

```
int _CoreGameData::window_w
```

The documentation for this struct was generated from the following file:

- GameCore.h

3.3 _Entity Struct Reference

```
#include <GameCore.h>
```

Data Fields

- int ID
- int state
- SDL_Rect sprite
- SDL_Rect cutter
- SDL_Texture * spriteSheet
- char textureName [20]
- int velx
- int vely
- int animationStates [15]
- BoundingBox box

3.3.1 Field Documentation

3.3.1.1 animationStates

```
int _Entity::animationStates[15]
```

3.3.1.2 box

```
BoundingBox _Entity::box
```

3.3.1.3 cutter

```
SDL_Rect _Entity::cutter
```

3.3.1.4 ID

```
int _Entity::ID
```

3.3.1.5 sprite

```
SDL_Rect _Entity::sprite
```

3.3.1.6 spriteSheet

```
SDL_Texture* _Entity::spriteSheet
```

3.3.1.7 state

```
int _Entity::state
```

3.3.1.8 textureName

```
char _Entity::textureName[20]
```

3.3.1.9 velx

```
int _Entity::velx
```

3.3.1.10 vely

```
int _Entity::vely
```

The documentation for this struct was generated from the following file:

- GameCore.h

3.4 _inventory Struct Reference

```
#include <InventoryUtils.h>
```

Data Fields

- Entity items [3]

3.4.1 Field Documentation

3.4.1.1 items

```
Entity _inventory::items[3]
```

The documentation for this struct was generated from the following file:

- InventoryUtils.h

3.5 _MapTile Struct Reference

```
#include <GameCore.h>
```

Data Fields

- SDL_Texture * tileTexture
- SDL_Rect tileRect

3.5.1 Field Documentation

3.5.1.1 tileRect

```
SDL_Rect _MapTile::tileRect
```

3.5.1.2 tileTexture

```
SDL_Texture* _MapTile::tileTexture
```

The documentation for this struct was generated from the following file:

- GameCore.h

3.6 _node Struct Reference

```
#include <GameCore.h>
```

Data Fields

- Entity item
- struct _node * next

3.6.1 Field Documentation

3.6.1.1 item

```
Entity _node::item
```

3.6.1.2 next

```
struct _node* _node::next
```

The documentation for this struct was generated from the following file:

- GameCore.h

3.7 _packet Struct Reference

```
#include <Net_utils.h>
```

Data Fields

- uint32_t state
- uint32_t x
- uint32_t y
- uint32_t ID

3.7.1 Field Documentation

3.7.1.1 ID

```
uint32_t _packet::ID
```

3.7.1.2 state

```
uint32_t _packet::state
```

3.7.1.3 x

```
uint32_t _packet::x
```

3.7.1.4 y

```
uint32_t _packet::y
```

The documentation for this struct was generated from the following file:

- Net_utils.h

3.8 _room Struct Reference

```
#include <GameCore.h>
```

Data Fields

- unsigned int door:4
- char * mapName
- node * items
- node * staticBlocks
- node * mobs
- struct _room * rooms [4]
- int gened:1
- int bound:1

3.8.1 Field Documentation

3.8.1.1 bound

```
int _room::bound
```

3.8.1.2 door

```
unsigned int _room::door
```

3.8.1.3 gened

```
int _room::gened
```

3.8.1.4 items

```
node* _room::items
```

3.8.1.5 mapName

```
char* _room::mapName
```

3.8.1.6 mobs

```
node* _room::mobs
```

3.8.1.7 rooms

```
struct _room* _room::rooms[4]
```

3.8.1.8 staticBlocks

```
node* _room::staticBlocks
```

The documentation for this struct was generated from the following file:

- GameCore.h

3.9 _SystemData Struct Reference

```
#include <GameCore.h>
```

Data Fields

- GameData * gameData
- SDL_Renderer ** render
- void * rendering
- void * systemIO
- void * mainSystem
- void * lockEntityies
- void * LockGameData

3.9.1 Field Documentation

3.9.1.1 gameData

```
GameData* _SystemData::gameData
```

3.9.1.2 lockEntityies

```
void* _SystemData::lockEntityies
```

3.9.1.3 LockGameData

```
void* _SystemData::LockGameData
```

3.9.1.4 mainSystem

```
void* _SystemData::mainSystem
```

3.9.1.5 render

```
SDL_Renderer** _SystemData::render
```

3.9.1.6 rendering

```
void* _SystemData::rendering
```

3.9.1.7 systemIO

```
void* _SystemData::systemIO
```

The documentation for this struct was generated from the following file:

- GameCore.h

3.10 _TriggerAreas Struct Reference

```
#include <GameCore.h>
```

Data Fields

- SDL_Rect Rect
- int doornum

3.10.1 Field Documentation

3.10.1.1 doornum

```
int _TriggerAreas::doornum
```

3.10.1.2 Rect

```
SDL_Rect _TriggerAreas::Rect
```

The documentation for this struct was generated from the following file:

- GameCore.h

Chapter 4

File Documentation

4.1 Base_code.c File Reference

```
#include "Base_code.h"
#include "MapRenderer.h"
#include "InputEvents.h"
#include "Rendering.h"
#include "LinkedList.h"
#include "FileIO.h"
#include "CollisionEvents.h"
#include "BoundingboxUtil.h"
#include "Room.h"
#include "CollisionDetection.h"
```

Functions

- DWORD WINAPI mainSystem (void *vararg)

4.1.1 Function Documentation

4.1.1.1 mainSystem()

```
DWORD WINAPI mainSystem (
            void * vararg )
```

this is used the main calculation thread

Parameters

vararg	this is a SystemData var

Returns

the thread returns

4.2 Base_code.h File Reference

```
#include "GameCore.h"
```

Functions

- DWORD WINAPI mainSystem (void *vararg)

4.2.1 Function Documentation

4.2.1.1 mainSystem()

```
DWORD WINAPI mainSystem (
            void * vararg )
```

this is used the main calculation thread

Parameters

vararg	this is a SystemData var

Returns

the thread returns

4.3 BoundingboxUtil.c File Reference

```
#include "BoundingboxUtil.h"
#include <math.h>
```

Functions

- SDL_Point pointRotation (SDL_Point center, SDL_Point point, int degree)
- void rotateBoundingBox (BoundingBox *box, int degree)
- BoundingBox initBoundingBox (int x, int y, int h, int w)
- void moveBoundingBox (BoundingBox *outBox, int dx, int dy)
- void setBoundingBox (BoundingBox *outBox, int x, int y)

4.3.1 Function Documentation

4.3.1.1 initBoundingBox()

```
BoundingBox initBoundingBox (
            int x,
            int y,
            int h,
            int w )
```

creates bounding box

Parameters

x	x coordinate
y	y coordinate
h	height
w	width

Returns

a bounding box with specified settings

4.3.1.2 moveBoundingBox()

```
void moveBoundingBox (
            BoundingBox * outBox,
            int dx,
            int dy )
```

moves a boundingbox in a directions

Parameters

outBox	the box to move
dx	the x velocity
dy	the y velocity

4.3.1.3 pointRotation()

```
SDL_Point pointRotation (
            SDL_Point center,
```

```
            SDL_Point point,
            int degree )
```

this function is used to rotate the a point around a center by a numbe of degree

Parameters

center	the point to rotate around
point	the point that you want to rotate
degree	the degree amount that you want to rotate

Returns

the rotated point

4.3.1.4 rotateBoundingBox()

```
void rotateBoundingBox (
            BoundingBox * box,
            int degree )
```

This function rotates a boinding box about a point

Parameters

box	the bounding boc that you want to rotate around
degree	the amount that you want to rotate the box around

4.3.1.5 setBoundingBox()

```
void setBoundingBox (
            BoundingBox * outBox,
            int x,
            int y )
```

sets the bounding box loactions to a x,y location

Parameters

outBox	the box to set
x	the x coordinate
y	the y corrdinate

4.4 BoundingboxUtil.h File Reference

```
#include "GameCore.h"
```

Functions

- void rotateBoundingBox (BoundingBox *box, int degree)
- BoundingBox initBoundingBox (int x, int y, int h, int w)
- void moveBoundingBox (BoundingBox *outBox, int dx, int dy)
- void setBoundingBox (BoundingBox *outBox, int x, int y)

4.4.1 Function Documentation

4.4.1.1 initBoundingBox()

```
BoundingBox initBoundingBox (
            int x,
            int y,
            int h,
            int w )
```

creates bounding box

Parameters

x	x coordinate
y	y coordinate
h	height
w	width

Returns

a bounding box with specified settings

4.4.1.2 moveBoundingBox()

```
void moveBoundingBox (
            BoundingBox * outBox,
            int dx,
            int dy )
```

moves a boundingbox in a directions

Parameters

outBox	the box to move
dx	the x velocity
dy	the y velocity

4.4.1.3 rotateBoundingBox()

```
void rotateBoundingBox (
            BoundingBox * box,
            int degree )
```

This function rotates a boinding box about a point

Parameters

box	the bounding boc that you want to rotate around
degree	the amount that you want to rotate the box around

4.4.1.4 setBoundingBox()

```
void setBoundingBox (
            BoundingBox * outBox,
            int x,
            int y )
```

sets the bounding box loactions to a x,y location

Parameters

outBox	the box to set
x	the x coordinate
y	the y corrdinate

4.5 CollisionActions.c File Reference

```
#include "CollisionActions.h"
#include "math.h"
#include "BoundingboxUtil.h"
```

Functions

- void stopEntityMovement (Entity *mover, Entity *collision)

4.5.1 Function Documentation

4.5.1.1 stopEntityMovement()

```
void stopEntityMovement (
            Entity * mover,
            Entity * collision )
```

stops a moving entity that is colliding with a static entity

Parameters

mover	the moving entity that is colliding
collision	the static entity that is colliding

4.6 CollisionActions.h File Reference

```
#include "GameCore.h"
```

Functions

- void stopEntityMovement (Entity ∗mover, Entity ∗collision)

4.6.1 Function Documentation

4.6.1.1 stopEntityMovement()

```
void stopEntityMovement (
            Entity * mover,
            Entity * collision )
```

stops a moving entity that is colliding with a static entity

Parameters

mover	the moving entity that is colliding
collision	the static entity that is colliding

4.7 CollisionDetection.c File Reference

```
#include "CollisionDetection.h"
#include "GameCore.h"
#include <math.h>
```

Functions

- bool onSegment (SDL_Point p, SDL_Point q, SDL_Point r)
- int orientation (SDL_Point p, SDL_Point q, SDL_Point r)
- bool doIntersect (SDL_Point p1, SDL_Point q1, SDL_Point p2, SDL_Point q2)
- int distance (SDL_Point p1, SDL_Point p2)
- int optCheckCollisions (BoundingBox *box1, BoundingBox *box2, int dist)
- bool checkCollision (SDL_Rect a, SDL_Rect b)

4.7.1 Function Documentation

4.7.1.1 checkCollision()

```
bool checkCollision (
            SDL_Rect a,
            SDL_Rect b )
```

checks collisions between 2 SDl_Rects

Parameters

a	the first rect
b	the second rect

Returns

if thare is a collision for not

4.7.1.2 distance()

```
int distance (
            SDL_Point p1,
            SDL_Point p2 )
```

4.7.1.3 doIntersect()

```
bool doIntersect (
             SDL_Point p1,
             SDL_Point q1,
             SDL_Point p2,
             SDL_Point q2 )
```

4.7.1.4 onSegment()

```
bool onSegment (
             SDL_Point p,
             SDL_Point q,
             SDL_Point r )
```

4.7.1.5 optCheckCollisions()

```
int optCheckCollisions (
             BoundingBox * box1,
             BoundingBox * box2,
             int dist )
```

a distance optimized collision detection

Parameters

box1	the bounding box that is is colliding with the second entity(box2)
box2	the entity that is being collided with.
dist	the distance that the boxes have to have between the to do check for collision

Returns

the collition side on box2

4.7.1.6 orientation()

```
int orientation (
             SDL_Point p,
             SDL_Point q,
             SDL_Point r )
```

4.8 CollisionDetection.h File Reference

```
#include <math.h>
#include <SDL2/SDL.h>
#include <stdbool.h>
#include "GameCore.h"
```

Functions

- bool doIntersect (SDL_Point p1, SDL_Point q1, SDL_Point p2, SDL_Point q2)
- bool checkCollision (SDL_Rect a, SDL_Rect b)
- int optCheckCollisions (BoundingBox *box1, BoundingBox *box2, int dist)

4.8.1 Function Documentation

4.8.1.1 checkCollision()

```
bool checkCollision (
            SDL_Rect a,
            SDL_Rect b )
```

checks collisions between 2 SDl_Rects

Parameters

a	the first rect
b	the second rect

Returns

if thare is a collision for not

4.8.1.2 doIntersect()

```
bool doIntersect (
            SDL_Point p1,
            SDL_Point q1,
            SDL_Point p2,
            SDL_Point q2 )
```

4.8.1.3 optCheckCollisions()

```
int optCheckCollisions (
            BoundingBox * box1,
            BoundingBox * box2,
            int dist )
```

a distance optimized collision detection

Parameters

box1	the bounding box that is is colliding with the second entity(box2)
box2	the entity that is being collided with.
dist	the distance that the boxes have to have between the to do check for collision

Returns

the collition side on box2

4.9 CollisionEvents.c File Reference

```
#include "CollisionEvents.h"
#include "LinkedList.h"
#include "CollisionDetection.h"
#include "CollisionActions.h"
#include "InputEvents.h"
#include "Room.h"
```

Macros

- #define MAX(a, b) (((a) > (b)) ? (a) : (b))

Functions

- void doorTiggerCollision (GameData *data)
- void staticObjectCollision (Entity *entity, node *list)

4.9.1 Macro Definition Documentation

4.9.1.1 MAX

```
#define MAX(
            a,
            b ) (((a) > (b)) ?  (a) :  (b))
```

4.9.2 Function Documentation

4.9.2.1 doorTiggerCollision()

```
void doorTiggerCollision (
            GameData * data )
```

see if thare is a collision of one of the trigger areas

Parameters

data	a instens of the game data

4.9.2.2 staticObjectCollision()

```
void staticObjectCollision (
            Entity * entity,
            node * list )
```

this is function is used to whether an entity is colliding with any entity in a list of entitys

Parameters

entity	the entitys that you want to check collisions with
list	the list of entitys that you want to check against

4.10 CollisionEvents.h File Reference

```
#include "GameCore.h"
```

Functions

- void doorTiggerCollision (GameData *data)
- void staticObjectCollision (Entity *entity, node *list)

4.10.1 Function Documentation

4.10.1.1 doorTiggerCollision()

```
void doorTiggerCollision (
            GameData * data )
```

see if thare is a collision of one of the trigger areas

Parameters

data	a instens of the game data

4.10.1.2 staticObjectCollision()

```
void staticObjectCollision (
            Entity * entity,
            node * list )
```

this is function is used to whether an entity is colliding with any entity in a list of entitys

Parameters

entity	the entitys that you want to check collisions with
list	the list of entitys that you want to check against

4.11 config.h File Reference

Macros

- #define DISCONNECT 0
  MESSAGE CODES FOR NET CODE.
- #define REG 1
- #define SUPPLY_UPDATE 2
- #define REQUEST_UPDATE 3
- #define REQUEST_PCOUNT 4
- #define MSG_TYPE_PACKAGE 10
- #define MSG_TYPE_REG_INFO 11

4.11.1 Macro Definition Documentation

4.11.1.1 DISCONNECT

```
#define DISCONNECT 0
```

MESSAGE CODES FOR NET CODE.

4.11.1.2 MSG_TYPE_PACKAGE

```
#define MSG_TYPE_PACKAGE 10
```

4.11.1.3 MSG_TYPE_REG_INFO

```
#define MSG_TYPE_REG_INFO 11
```

4.11.1.4 REG

```
#define REG 1
```

4.11.1.5 REQUEST_PCOUNT

```
#define REQUEST_PCOUNT 4
```

4.11.1.6 REQUEST_UPDATE

```
#define REQUEST_UPDATE 3
```

4.11.1.7 SUPPLY_UPDATE

```
#define SUPPLY_UPDATE 2
```

4.12 Console.c File Reference

```
#include "Console.h"
#include "GameCore.h"
```

Functions

- DWORD WINAPI Consolse ()

4.12.1 Function Documentation

4.12.1.1 Consolse()

```
DWORD WINAPI Consolse ( )
```

4.13 Console.h File Reference

4.14 FileIO.c File Reference

```
#include "FileIO.h"
```

Functions

- void writeEntityToFile (char *name, Entity *entity)
- Entity readEntityFromFile (char *name, SDL_Renderer *rend)
- void LoadMapFile (char *name, GameData *data)

  this function has depricated
- void LoadBigMapFile (char *name, GameData *data)

  reads in a .map file map size 22
- void LoadTileData (GameData *data)

4.14.1 Function Documentation

4.14.1.1 LoadBigMapFile()

```
void LoadBigMapFile (
            char * name,
            GameData * data )
```

reads in a .map file map size 22

this loads a map into the system

Parameters

name	the file name of the map you want to load
data	pointer to the game data to load map into

4.14.1.2 LoadMapFile()

```
void LoadMapFile (
            char * name,
            GameData * data )
```

this function has depricated

this loads a map into the system

Parameters

name	the file name of the map you want to load
data	pointer to the game data to load map into

4.14.1.3 LoadTileData()

```
void LoadTileData (
            GameData * data )
```

this loads in all this tile data for the map

Parameters

data	the pointer to game data to load into

4.14.1.4 readEntityFromFile()

```
Entity readEntityFromFile (
            char * name,
            SDL_Renderer * rend )
```

this function read a function into the system

Parameters

name	the file name of the entity file
rend	an SDl_render to make the entity texture data

Returns

the entity that is read from the file

4.14.1.5 writeEntityToFile()

```
void writeEntityToFile (
            char * name,
            Entity * entity )
```

this functions that save a entitys to a file

Parameters

name	file name to save to
entity	the entity that you want to save

4.15 FileIO.h File Reference

```
#include <stdio.h>
#include "GameCore.h"
```

Functions

- void writeEntityToFile (char ∗name, Entity ∗entity)
- Entity readEntityFromFile (char ∗name, SDL_Renderer ∗rend)
- void LoadMapFile (char ∗name, GameData ∗data)

  this function has depricated
- void LoadTileData (GameData ∗data)

4.15.1 Function Documentation

4.15.1.1 LoadMapFile()

```
void LoadMapFile (
            char * name,
            GameData * data )
```

this function has depricated

this loads a map into the system

Parameters

name	the file name of the map you want to load
data	pointer to the game data to load map into

4.15.1.2 LoadTileData()

```
void LoadTileData (
            GameData * data )
```

this loads in all this tile data for the map

Parameters

data	the pointer to game data to load into

4.15.1.3 readEntityFromFile()

```
Entity readEntityFromFile (
            char * name,
            SDL_Renderer * rend )
```

this function read a function into the system

Parameters

name	the file name of the entity file
rend	an SDl_render to make the entity texture data

Returns

the entity that is read from the file

4.15.1.4 writeEntityToFile()

```
void writeEntityToFile (
            char * name,
            Entity * entity )
```

this functions that save a entitys to a file

Parameters

name	file name to save to
entity	the entity that you want to save

4.16 GameCore.h File Reference

```
#include <SDL2/SDL.h>
#include <SDL2/SDL_image.h>
#include <SDL2/SDL_ttf.h>
#include <windows.h>
```

Data Structures

- struct _boundingBox
- struct _Entity
- struct _TriggerAreas
- struct _node
- struct _room
- struct _MapTile
- struct _CoreGameData
- struct _SystemData

Macros

- #define mapsize 22

Typedefs

- typedef struct _boundingBox BoundingBox
- typedef struct _Entity Entity
- typedef struct _TriggerAreas Trigger
- typedef struct _node node
- typedef struct _room Room
- typedef struct _MapTile MapTile
- typedef struct _CoreGameData GameData
- typedef struct _SystemData SystemData

4.16.1 Macro Definition Documentation

4.16.1.1 mapsize

```
#define mapsize 22
```

4.16.2 Typedef Documentation

4.16.2.1 BoundingBox

```
typedef struct _boundingBox BoundingBox
```

4.16.2.2 Entity

```
typedef struct _Entity Entity
```

4.16.2.3 GameData

```
typedef struct _CoreGameData GameData
```

4.16.2.4 MapTile

```
typedef struct _MapTile MapTile
```

4.16.2.5 node

```
typedef struct _node node
```

4.16.2.6 Room

```
typedef struct _room Room
```

4.16.2.7 SystemData

```
typedef struct _SystemData SystemData
```

4.16.2.8 Trigger

```
typedef struct _TriggerAreas Trigger
```

4.17 InputEvents.c File Reference

```
#include "InputEvents.h"
```

Functions

- int linkEntityToUserInput (Entity *entity, GameData *gameData)
- void bindEntityToBoard (Entity *entity, GameData gameData)
- void bindEntityToRect (Entity *entity, SDL_Rect rect)
- void bindEntitysToBoard (GameData data)

  this function has deprecated
- void bindEntitysToRect (GameData *data, SDL_Rect rect)

4.17.1 Function Documentation

4.17.1.1 bindEntitysToBoard()

```
void bindEntitysToBoard (
            GameData data )
```

this function has deprecated

4.17.1.2 bindEntitysToRect()

```
void bindEntitysToRect (
            GameData * data,
            SDL_Rect rect )
```

4.17.1.3 bindEntityToBoard()

```
void bindEntityToBoard (
            Entity * entity,
            GameData gameData )
```

set the bound of the movement area to the window screen

Parameters

entity	entity that you want to bind to the window
gameData	current stat of global data

4.17.1.4 bindEntityToRect()

```
void bindEntityToRect (
            Entity * entity,
            SDL_Rect rect )
```

set the bound of the movement area to a rect

Parameters

entity	entity that you want to bind to the window
rect	the rectangle that you want the entity to be bound in

4.17.1.5 linkEntityToUserInput()

```
int linkEntityToUserInput (
            Entity * entity,
            GameData * gameData )
```

links an entity to user input

Parameters

entity	the entity that you want to link
gameData	current states of global data

Returns

returns the input state that what found

4.18 InputEvents.h File Reference

```
#include "GameCore.h"
```

Functions

- int linkEntityToUserInput (Entity ∗entity, GameData ∗gameData)
- void bindEntityToBoard (Entity ∗entity, GameData gameData)
- void bindEntityToRect (Entity ∗entity, SDL_Rect rect)
- void bindEntitysToRect (GameData ∗data, SDL_Rect rect)
- void bindEntitysToBoard (GameData data)

  this function has deprecated

4.18.1 Function Documentation

4.18.1.1 bindEntitysToBoard()

```
void bindEntitysToBoard (
            GameData data )
```

this function has deprecated

4.18.1.2 bindEntitysToRect()

```
void bindEntitysToRect (
            GameData * data,
            SDL_Rect rect )
```

4.18.1.3 bindEntityToBoard()

```
void bindEntityToBoard (
            Entity * entity,
            GameData gameData )
```

set the bound of the movement area to the window screen

Parameters

entity	entity that you want to bind to the window
gameData	current stat of global data

4.18.1.4 bindEntityToRect()

```
void bindEntityToRect (
            Entity * entity,
            SDL_Rect rect )
```

set the bound of the movement area to a rect

Parameters

entity	entity that you want to bind to the window
rect	the rectangle that you want the entity to be bound in

4.18.1.5 linkEntityToUserInput()

```
int linkEntityToUserInput (
            Entity * entity,
            GameData * gameData )
```

links an entity to user input

Parameters

entity	the entity that you want to link
gameData	current states of global data

Returns

returns the input state that what found

4.19 InventoryUtils.c File Reference

```
#include "InventoryUtils.h"
```

4.20 InventoryUtils.h File Reference

```
#include "GameCore.h"
```

Data Structures

- struct _inventory

Typedefs

- typedef struct _inventory Inventory

4.20.1 Typedef Documentation

4.20.1.1 Inventory

```
typedef struct _inventory Inventory
```

4.21 ItemUtils.c File Reference

```
#include "ItemUtils.h"
#include "Room.h"
```

Functions

- void spawnItems (Room room)

4.21.1 Function Documentation

4.21.1.1 spawnItems()

```
void spawnItems (
            Room room )
```

4.22 ItemUtils.h File Reference

4.23 LinkedList.c File Reference

```
#include <stdio.h>
#include "LinkedList.h"
```

Macros

- #define TRUE 1
- #define FALSE 0

Typedefs

- typedef int BOOL

Functions

- node * NewElement (Entity entity)
- void Insertnode (node **head, node *newp)
- void Removenode (node **head, int ID)
- Entity * Findnode (node **head, int ID)
- void PrintList (node **head)

4.23.1 Macro Definition Documentation

4.23.1.1 FALSE

```
#define FALSE 0
```

4.23.1.2 TRUE

```
#define TRUE 1
```

4.23.2 Typedef Documentation

4.23.2.1 BOOL

```
typedef int BOOL
```

4.23.3 Function Documentation

4.23.3.1 Findnode()

```
Entity* Findnode (
            node ** head,
            int ID )
```

finds a node based on id

Parameters

head	list to search through
ID	the id that you want to fine

Returns

pointer to the found node

4.23.3.2 Insertnode()

```
void Insertnode (
            node ** head,
            node * newp )
```

adds a new node to the linked list

Parameters

head	a pointer to the start of the linked list
newp	the new node that you want to add

4.23.3.3 NewElement()

```
node* NewElement (
            Entity entity )
```

creates new node for the list

Parameters

entity	The entity that you want to add to the list

Returns

4.23.3.4 PrintList()

```
void PrintList (
            node ** head )
```

prints linked list nodes id (should do this SDL doesnt print to stdout)

Parameters

head	

4.23.3.5 Removenode()

```
void Removenode (
```

```
            node ** head,
            int ID )
```

removes a node from the linked list

Parameters

head	a pointer to the start of the linked list
ID	the ID of the entity that you want to remove

4.24 LinkedList.h File Reference

```
#include "GameCore.h"
```

Functions

- void Removenode (node ∗∗head, int ID)
- void Insertnode (node ∗∗head, node ∗newp)
- node ∗ NewElement (Entity new)
- Entity ∗ Findnode (node ∗∗head, int ID)
- void PrintList (node ∗∗head)

4.24.1 Function Documentation

4.24.1.1 Findnode()

```
Entity* Findnode (
            node ** head,
            int ID )
```

finds a node based on id

Parameters

head	list to search through
ID	the id that you want to fine

Returns

pointer to the found node

4.24.1.2 Insertnode()

```
void Insertnode (
            node ** head,
            node * newp )
```

adds a new node to the linked list

Parameters

head	a pointer to the start of the linked list
newp	the new node that you want to add

4.24.1.3 NewElement()

```
node* NewElement (
            Entity entity )
```

creates new node for the list

Parameters

entity	The entity that you want to add to the list

Returns

4.24.1.4 PrintList()

```
void PrintList (
            node ** head )
```

prints linked list nodes id (should do this SDL doesnt print to stdout)

Parameters

head	

4.24.1.5 Removenode()

```
void Removenode (
```

```
            node ** head,
            int ID )
```

removes a node from the linked list

Parameters

head	a pointer to the start of the linked list
ID	the ID of the entity that you want to remove

4.25 main.c File Reference

```
#include "GameCore.h"
#include "Rendering.h"
#include "Base_code.h"
#include "NetCode.h"
```

Functions

- int main (int argc, char **argv)

4.25.1 Function Documentation

4.25.1.1 main()

```
int main (
            int argc,
            char ** argv )
```

4.26 MapRenderer.c File Reference

```
#include "MapRenderer.h"
#include "BoundingboxUtil.h"
#include "LinkedList.h"
```

Functions

- void rendermap (SDL_Renderer *rend, GameData *gameData)
- void renderMapFromFile (SDL_Renderer *rend, GameData *gameData)

4.26.1 Function Documentation

4.26.1.1 rendermap()

```
void rendermap (
            SDL_Renderer * rend,
            GameData * gameData )
```

renders the backbround map;

Parameters

rend	renderer
gameData	current states of the gloabl game data

4.26.1.2 renderMapFromFile()

```
void renderMapFromFile (
            SDL_Renderer * rend,
            GameData * gameData )
```

this function renders the map from gamedata map data

Parameters

rend	SDl_render that you want to render to
gameData	the game data that you want to read from

4.27 MapRenderer.h File Reference

```
#include "GameCore.h"
```

Functions

- void rendermap (SDL_Renderer ∗rend, GameData ∗gameData)
- void renderMapFromFile (SDL_Renderer ∗rend, GameData ∗gameData)
- void LoadBigMapFile (char ∗name, GameData ∗data)

  reads in a .map file map size 22

4.27.1 Function Documentation

4.27.1.1 LoadBigMapFile()

```
void LoadBigMapFile (
            char * name,
            GameData * data )
```

reads in a .map file map size 22

this loads a map into the system

Parameters

name	the file name of the map you want to load
data	pointer to the game data to load map into

4.27.1.2 rendermap()

```
void rendermap (
            SDL_Renderer * rend,
            GameData * gameData )
```

renders the backbround map;

Parameters

rend	renderer
gameData	current states of the gloabl game data

4.27.1.3 renderMapFromFile()

```
void renderMapFromFile (
            SDL_Renderer * rend,
            GameData * gameData )
```

this function renders the map from gamedata map data

Parameters

rend	SDl_render that you want to render to
gameData	the game data that you want to read from

4.28 MobAI.c File Reference

```
#include "MobAI.h"
#include "GameCore.h"
```

4.29 MobAI.h File Reference

4.30 MobUtils.c File Reference

```
#include "MobUtils.h"
#include "GameCore.h"
```

Functions

- void spawnMobsInRoom (Room room)

4.30.1 Function Documentation

4.30.1.1 spawnMobsInRoom()

```
void spawnMobsInRoom (
            Room room )
```

4.31 MobUtils.h File Reference

4.32 Net_utils.c File Reference

```
#include "Net_utils.h"
```

Functions

- int extract_msg_code (char **msg)
- void sendPrepMsg (char *msg, int sock)
- void sendCode (int code, int server_sock)
- void sendPacket (int msgCode, EntityPacket *packet, int sock)
- int getRegInfo ()
- int getpcount ()
- void recevMsg (int sockdef, char *msg)
- void decodePacket (EntityPacket *packet)

4.32.1 Function Documentation

4.32.1.1 decodePacket()

```
void decodePacket (
            EntityPacket * packet )
```

this decodes the packet data

Parameters

packet	the packet entity that you want to save to

4.32.1.2 extract_msg_code()

```
int extract_msg_code (
            char ** msg )
```

gets the massage code from the netcode packet

Parameters

msg	the packet that you want to get code from

Returns

the massge code

4.32.1.3 getpcount()

```
int getpcount ( )
```

gets the player count

Returns

the player count

4.32.1.4 getRegInfo()

```
int getRegInfo ( )
```

4.32.1.5 recevMsg()

```
void recevMsg (
            int sockdef,
            char * msg )
```

receves a message from the server

Parameters

sockdef	the socket definition that you want to revive from
msg	whare you want to save the massage to

4.32.1.6 sendCode()

```
void sendCode (
            int code,
            int server_sock )
```

sends a mssage code to the a socket

Parameters

code	the code that you want to send
server_sock	the socket definition that you want to send to

4.32.1.7 sendPacket()

```
void sendPacket (
            int msgCode,
            EntityPacket * packet,
            int sock )
```

sends a packet to a socket

Parameters

msgCode	massage code that you want to send with packet
packet	the packet of entity data that you want to sent
sock	the socket definition that you want to send to

4.32.1.8 sendPrepMsg()

```
void sendPrepMsg (
            char * msg,
            int sock )
```

sends a message to a socket

Parameters

msg	the message that you want to sent
sock	the socket definition that you want to send to

4.33 Net_utils.h File Reference

```
#include <stdio.h>
#include <errno.h>
#include <time.h>
#include <string.h>
#include <stdlib.h>
#include "config.h"
```

Data Structures

- struct _packet

Macros

- #define PORT 8080
- #define MAXLINE 84

Typedefs

- typedef struct _packet EntityPacket

Functions

- int extract_msg_code (char **msg)
- void sendPrepMsg (char *msg, int sock)
- void sendCode (int code, int server_sock)
- void recevMsg (int sockdef, char *msg)
- void sendPacket (int msgCode, EntityPacket *packet, int sock)
- void decodePacket (EntityPacket *packet)
- int getRegInfo ()
- int getpcount ()

4.33.1 Macro Definition Documentation

4.33.1.1 MAXLINE

```
#define MAXLINE 84
```

4.33.1.2 PORT

```
#define PORT 8080
```

4.33.2 Typedef Documentation

4.33.2.1 EntityPacket

```
typedef struct _packet EntityPacket
```

4.33.3 Function Documentation

4.33.3.1 decodePacket()

```
void decodePacket (
            EntityPacket * packet )
```

this decodes the packet data

Parameters

packet	the packet entity that you want to save to

4.33.3.2 extract_msg_code()

```
int extract_msg_code (
            char ** msg )
```

gets the massage code from the netcode packet

Parameters

msg	the packet that you want to get code from

Returns

the massge code

4.33.3.3 getpcount()

```
int getpcount ( )
```

gets the player count

Returns

the player count

4.33.3.4 getRegInfo()

```
int getRegInfo ( )
```

4.33.3.5 recevMsg()

```
void recevMsg (
            int sockdef,
            char * msg )
```

receves a message from the server

Parameters

sockdef	the socket definition that you want to revive from
msg	whare you want to save the massage to

4.33.3.6 sendCode()

```
void sendCode (
```

```
            int code,
            int server_sock )
```

sends a mssage code to the a socket

Parameters

code	the code that you want to send
server_sock	the socket definition that you want to send to

4.33.3.7 sendPacket()

```
void sendPacket (
            int msgCode,
            EntityPacket * packet,
            int sock )
```

sends a packet to a socket

Parameters

msgCode	massage code that you want to send with packet
packet	the packet of entity data that you want to sent
sock	the socket definition that you want to send to

4.33.3.8 sendPrepMsg()

```
void sendPrepMsg (
            char * msg,
            int sock )
```

sends a message to a socket

Parameters

msg	the message that you want to sent
sock	the socket definition that you want to send to

4.34 NetCode.c File Reference

```
#include "Net_utils.h"
#include "NetCode.h"
```

```
#include "LinkedList.h"
#include "FileIO.h"
```

Functions

- DWORD runNetCode (void *data)

4.34.1 Function Documentation

4.34.1.1 runNetCode()

```
DWORD runNetCode (
            void * data )
```

this is the nat code thread function

Parameters

data	this is system data for the thread

Returns

thread return

4.35 NetCode.h File Reference

```
#include <windows.h>
#include "GameCore.h"
```

Functions

- DWORD runNetCode (void *data)

4.35.1 Function Documentation

4.35.1.1 runNetCode()

```
DWORD runNetCode (
            void * data )
```

this is the nat code thread function

Parameters

data	this is system data for the thread

Returns

thread return

4.36 Rendering.c File Reference

```
#include "Rendering.h"
#include "LinkedList.h"
#include "MapRenderer.h"
#include "BoundingboxUtil.h"
#include <stdio.h>
```

Macros

- #define SPEED 2

Functions

- void renderTriggerBox (GameData *data, SDL_Renderer *rend)
- void moveEntity (GameData *data)
- void renderEntityBoxList (GameData *data, SDL_Renderer *rend)
- void renderEntitys (GameData *data, SDL_Renderer *rend)
- void renderRoomCode (GameData *data, SDL_Renderer *rend, TTF_Font *font, SDL_Color color)
- void renderInventory (GameData *gameData, SDL_Renderer *rend)
- void renderBoundingBox (BoundingBox *box, SDL_Renderer *rend)
- void renderWallBox (GameData *data, SDL_Renderer *rend)
- void animate (Entity *entity, int state)
- void animateEntitys (SystemData *data)
- DWORD WINAPI renderingSystem (void *vararg)

4.36.1 Macro Definition Documentation

4.36.1.1 SPEED

```
#define SPEED 2
```

4.36.2 Function Documentation

4.36.2.1 animate()

```
void animate (
            Entity * entity,
            int state )
```

This function moves the cutting SDL_Rect to the next frame of the animation

Parameters

entity	what ever entity you are trying to animate.
state	the animation state that corresponds to the action you are trying to animate

4.36.2.2 animateEntitys()

```
void animateEntitys (
            SystemData * data )
```

this function renders all of the entity's animations

Parameters

data	the game data that you want to render

4.36.2.3 moveEntity()

```
void moveEntity (
            GameData * data )
```

this function moves any entity that has a velocity

Parameters

data	the gamedata that you want to run

4.36.2.4 renderBoundingBox()

```
void renderBoundingBox (
            BoundingBox * box,
            SDL_Renderer * rend )
```

renders a bounding box

Parameters

box	bounding box that you want to render
rend	the sdl renderer that you want to render to

4.36.2.5 renderEntityBoxList()

```
void renderEntityBoxList (
            GameData * data,
            SDL_Renderer * rend )
```

renders entitys SDl_rect

Parameters

data	the game data that you want to render
rend	the sdl render whare you want to rendre to

4.36.2.6 renderEntitys()

```
void renderEntitys (
            GameData * data,
            SDL_Renderer * rend )
```

renders game data entitys to screen

Parameters

data	the game data that you want to render
rend	the sdl renderer that you want to render to

4.36.2.7 renderingSystem()

```
DWORD WINAPI renderingSystem (
            void * vararg )
```

this is the rendering thread function

Parameters

vararg	this is a system data arg

Returns

thread return

4.36.2.8 renderInventory()

```
void renderInventory (
            GameData * gameData,
            SDL_Renderer * rend )
```

this renders that inventory entity

Parameters

gameData	the game data that you want to render
rend	the sdl renderer that you want to render to

4.36.2.9 renderRoomCode()

```
void renderRoomCode (
            GameData * data,
            SDL_Renderer * rend,
            TTF_Font * font,
            SDL_Color color )
```

renders that room code for debug

Parameters

data	the game data that you want to render
rend	the sdl renderer that you want to render to
font	the font you want to render in
color	the color that you want to render in

4.36.2.10 renderTriggerBox()

```
void renderTriggerBox (
            GameData * data,
            SDL_Renderer * rend )
```

renderd the SDL RECT that defines how an entity is renders

Parameters

data	the game state
rend	an SDL renderer

4.36.2.11 renderWallBox()

```
void renderWallBox (
            GameData * data,
            SDL_Renderer * rend )
```

rendering all the wall tiles bounding box

Parameters

gameData	the game data that you want to render
rend	the sdl renderer that you want to render to

4.37 Rendering.h File Reference

```
#include "GameCore.h"
```

Functions

- DWORD WINAPI renderingSystem (void *vararg)

4.37.1 Function Documentation

4.37.1.1 renderingSystem()

```
DWORD WINAPI renderingSystem (
            void * vararg )
```

this is the rendering thread function

Parameters

vararg	this is a system data arg

Returns

thread return

4.38 Room.c File Reference

```
#include "Room.h"
```

Functions

- Room ∗ newRoom (unsigned int indor, Room ∗prevroom)
- void freeRoomsAtDepth (Room ∗room, int depth, Room ∗last)
- int genRoomALL (Room ∗room)
- int genRoom (Room ∗room)
- Room ∗ initRooms ()
- Room ∗ enterRoom (int door, Room ∗inRoom)

4.38.1 Function Documentation

4.38.1.1 enterRoom()

```
Room* enterRoom (
            int door,
            Room * inRoom )
```

enter a door and gens that room

Parameters

door	the door that you entered from
inRoom	the room that you enterd from

Returns

the room that you ar now in

4.38.1.2 freeRoomsAtDepth()

```
void freeRoomsAtDepth (
            Room * room,
            int depth,
            Room * last )
```

this function frees rooms that are far from the current room to free memory

Parameters

room	the current room
depth	the depth to free at
last	the room that you came from

4.38.1.3 genRoom()

```
int genRoom (
            Room * room )
```

this generates whether there is a door

Parameters

room	the room that you want to gen

Returns

the number of doors gened //not used

4.38.1.4 genRoomALL()

```
int genRoomALL (
            Room * room )
```

genrates all doors and rooms in a room

Parameters

room	the room to gen doors

Returns

the door room gen not used

4.38.1.5 initRooms()

```
Room* initRooms ( )
```

initializes a room

Returns

the initalized room

4.38.1.6 newRoom()

```
Room* newRoom (
            unsigned int indor,
            Room * prevroom )
```

creates a new room

Parameters

indor	the door entered from
prevroom	a pointer to the previus room

Returns

a new room popinter

4.39 Room.h File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include "GameCore.h"
```

Macros

- #define INIT_ROOM_PTR(x) x->door=0,x->mapName="",x->rooms[0]=NULL,x->rooms[1]=NULL,x->rooms[2]=NULL,x->rooms[3]=NULL,x->gened=0;
- #define INIT_ROOM_ARRAY(x) x[0]=NULL,x[1]=NULL,x[2]=NULL,x[3]=NULL;

Functions

- Room * initRooms ()
- Room * enterRoom (int door, Room *inRoom)
- void freeRoomsAtDepth (Room *room, int depth, Room *last)

4.39.1 Macro Definition Documentation

4.39.1.1 INIT_ROOM_ARRAY

```
#define INIT_ROOM_ARRAY(
             x ) x[0]=NULL,x[1]=NULL,x[2]=NULL,x[3]=NULL;
```

4.39.1.2 INIT_ROOM_PTR

```
#define INIT_ROOM_PTR(
             x ) x->door=0,x->mapName="",x->rooms[0]=NULL,x->rooms[1]=NULL,x->rooms[2]=NULL,x->rooms[3]=NUL
```

4.39.2 Function Documentation

4.39.2.1 enterRoom()

```
Room* enterRoom (
            int door,
            Room * inRoom )
```

enter a door and gens that room

Parameters

door	the door that you entered from
inRoom	the room that you enterd from

Returns

the room that you ar now in

4.39.2.2 freeRoomsAtDepth()

```
void freeRoomsAtDepth (
            Room * room,
            int depth,
            Room * last )
```

this function frees rooms that are far from the current room to free memory

Parameters

room	the current room
depth	the depth to free at
last	the room that you came from

4.39.2.3 initRooms()

```
Room* initRooms ( )
```

initializes a room

Returns

the initalized room

Index