64 Bit Graphical World

Generated by Doxygen 1.8.17

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 2

Class Documentation

2.1 Atom Struct Reference

```
#include <Humanoid.h>
```

Public Attributes

- GLuint **VAO**
- GLuint **VBO**
- GLfloat **vertices** [288]
- glm::vec3 **center**

2.1.1 Detailed Description

Basic building block of graphical human

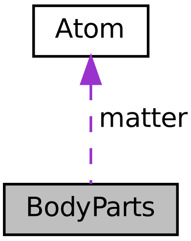
The documentation for this struct was generated from the following file:

- include/Humanoid.h

2.2 BodyParts Class Reference

```
#include <Humanoid.h>
```

Collaboration diagram for BodyParts:



Public Member Functions

- **BodyParts** (GLfloat a, GLfloat b, GLfloat c, glm::vec3 Center, GLint d, BodyPart h, GLuint k)
- void **drawBodyPart** ()
- void **addBodyPart** (BodyParts ∗part, glm::vec3 Offset)
- void **swingHand** (GLint a)

Public Attributes

- Atom ∗ **matter**
- BodyPart **type**
- GLuint **texture**
- glm::vec3 **pCentroid** = glm::vec3(0.0f)
- glm::vec3 **shift** = glm::vec3(0.0f)
- glm::vec3 **point_Inflne1** = glm::vec3(0.0f)
- glm::vec3 **point_Inflne2** = glm::vec3(0.0f)
- glm::vec3 **point_Inflne3** = glm::vec3(0.0f)
- glm::vec3 **point_Inflne4** = glm::vec3(0.0f)
- GLfloat **angle0** = 0.0f
- GLfloat **angle1** = 0.0f
- GLfloat **angle2** = 0.0f
- GLfloat **cen** = 0.0f
- GLfloat **length**
- GLfloat **breadth**
- GLfloat **height**
- glm::vec3 **center**
- glm::vec3 **center1**
- std::vector< std::pair< BodyParts ∗, glm::vec3 > > **children**
- GLint **ShaderProgram**

2.2.1 Detailed Description

Graphical Body part

The documentation for this class was generated from the following files:

- include/Humanoid.h
- src/Humanoid.cpp

2.3 Camera Class Reference

```
#include <Camera.h>
```

Public Member Functions

- void updateCameraVectors ()
- Camera (glm::vec3 position=glm::vec3(0.0f, 0.0f, 0.0f), glm::vec3 up=glm::vec3(0.0f, 1.0f, 0.0f), GLfloat yaw=YAW, GLfloat pitch=PITCH)
- Camera (GLfloat posX, GLfloat posY, GLfloat posZ, GLfloat upX, GLfloat upY, GLfloat upZ, GLfloat yaw, GLfloat pitch)
- glm::mat4 **GetViewMatrix** ()
- void ProcessKeyboard (Camera_Movement direction, GLfloat deltaTime, GLboolean constrainPitch=true)
- void ProcessMouseMovement (GLfloat xoffset, GLfloat yoffset, GLboolean constrainPitch=true)
- void ProcessMouseScroll (GLfloat yoffset)
- GLfloat **getZoom** ()
- glm::vec3 **getPosition** ()
- void **setPosition** (glm::vec3 Position)
- glm::vec3 **getFront** ()
- void **setFront** (glm::vec3 front)

2.3.1 Detailed Description

An abstract camera class that processes input and calculates the corresponding Eular Angles, Vectors and Matrices for use in OpenGL

2.3.2 Constructor & Destructor Documentation

2.3.2.1 Camera() [1/2]

```
Camera::Camera (
            glm::vec3 position = glm::vec3(0.0f, 0.0f, 0.0f),
            glm::vec3 up = glm::vec3(0.0f, 1.0f, 0.0f),
            GLfloat yaw = YAW,
            GLfloat pitch = PITCH )
```

Constructor with vectors

2.3.2.2 Camera() [2/2]

```
Camera::Camera (
            GLfloat posX,
            GLfloat posY,
            GLfloat posZ,
            GLfloat upX,
            GLfloat upY,
            GLfloat upZ,
            GLfloat yaw,
            GLfloat pitch )
```

Constructor with scalar values

2.3.3 Member Function Documentation

2.3.3.1 ProcessKeyboard()

```
void Camera::ProcessKeyboard (
             Camera_Movement direction,
             GLfloat deltaTime,
             GLboolean constrainPitch = true )
```

Processes input received from any keyboard-like input system. Accepts input parameter in the form of camera defined ENUM (to abstract it from windowing systems)

2.3.3.2 ProcessMouseMovement()

```
void Camera::ProcessMouseMovement (
             GLfloat xoffset,
             GLfloat yoffset,
             GLboolean constrainPitch = true )
```

Processes input received from a mouse input system. Expects the offset value in both the x and y direction.

2.3.3.3 ProcessMouseScroll()

```
void Camera::ProcessMouseScroll (
             GLfloat yoffset )
```

Processes input received from a mouse scroll-wheel event. Only requires input on the vertical wheel-axis

2.3.3.4 updateCameraVectors()

```
void Camera::updateCameraVectors ( )
```

Calculates the front vector from the Camera's (updated) Eular Angles

The documentation for this class was generated from the following files:

- include/Camera.h
- src/Camera.cpp

2.4 ContextManager Class Reference

Public Member Functions

- void init ()
- void setOptions ()
- GLFWwindow ∗ getContext ()
- GLint getHeight ()
- GLint getWidth ()

2.4.1 Member Function Documentation

2.4.1.1 getContext()

```
GLFWwindow * ContextManager::getContext ( )
```

returns current context

2.4.1.2 getHeight()

```
GLint ContextManager::getHeight ( )
```

returns height of current context

2.4.1.3 getWidth()

```
GLint ContextManager::getWidth ( )
```

returns width of current context

2.4.1.4 init()

```
void ContextManager::init ( )
```

initialize current context

2.4.1.5 setOptions()

```
void ContextManager::setOptions ( )
```

sets param for current context

The documentation for this class was generated from the following files:

- include/Context.h
- src/Context.cpp

2.5 House Class Reference

Public Member Functions

- void **createHouse** (glm::vec3 center, GLint shaderProgram)
- void **createLight** (GLint lampProgram)
- void **createLight1** (GLint lampProgram)

Public Attributes

- GLuint **VAO**
- GLuint **VBO**
- GLboolean **isOpen** =GL_FALSE
- GLboolean **isOpen1** =GL_FALSE
- GLfloat **openWindow** =0.0f
- GLfloat **openDoor** =0.0f
- GLuint **wood**
- GLuint **sofa**
- GLuint **windowTex**
- GLuint **floorTex**
- GLuint **wallTex**
- GLuint **roofTex**
- GLfloat **length** = 55.0f
- GLfloat **breadth** = 40.0f
- GLfloat **height** = 35.0f
- GLfloat **thickness** = 0.5f

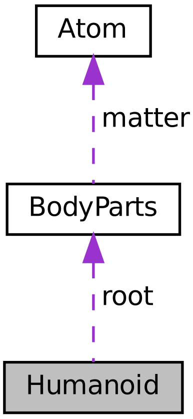
The documentation for this class was generated from the following files:

- include/House.h
- src/House.cpp

2.6 Humanoid Class Reference

```
#include <Humanoid.h>
```

Collaboration diagram for Humanoid:



Public Member Functions

- **Humanoid** (glm::vec3 center, GLint ShaderProgram, char gender)
- void addBodyPart (BodyParts ∗part, glm::vec3 Offset)
- void visible ()
- void **rotCen** ()

Public Attributes

- BodyParts ∗ **root**
- GLfloat **count** = 0.0f
- GLint **isDone** = 0
- GLboolean **isShift** = GL_FALSE

2.6.1 Detailed Description

Graphical human body

2.6.2 Member Function Documentation

2.6.2.1 addBodyPart()

```
void Humanoid::addBodyPart (
            BodyParts ∗ part,
            glm::vec3 Offset )
```

adds body parts

2.6.2.2 visible()

```
void Humanoid::visible ( )
```

Displays the human

The documentation for this class was generated from the following files:

- include/Humanoid.h
- src/Humanoid.cpp

2.7 InputDeviceManager Class Reference

Public Member Functions

- **InputDeviceManager** (ContextManager ∗cm, Camera ∗camera)
- void updateCameraPosition ()

2.7.1 Member Function Documentation

2.7.1.1 updateCameraPosition()

```
void InputDeviceManager::updateCameraPosition ( )
```

moves the camera

The documentation for this class was generated from the following files:

- include/inputdevice.h
- src/inputdevice.cpp

2.8 Mbox Class Reference

Public Member Functions

- **Mbox** (GLboolean d=true)
- void **createBox** (glm::vec3 center, GLint shaderProgram)
- void **rotcen** ()

Public Attributes

- GLuint **VAO**
- GLuint **VBO**
- GLfloat **openAngle** = 0.0f
- GLuint **texture**
- GLuint **texture1**
- GLuint **texture2**
- GLuint **innerTex**
- GLfloat **length** = 9.6f
- GLfloat **breadth** = 5.6f
- GLfloat **height** = 4.8f
- GLfloat **thickness** = 0.2f
- GLboolean **isOpen** = GL_FALSE
- GLboolean **isMusic** = GL_FALSE
- GLboolean **isPoint** = GL_TRUE

The documentation for this class was generated from the following files:

- include/Mbox.h
- src/Mbox.cpp

2.9 Shader Class Reference

Public Member Functions

- **Shader** (const GLchar ∗vertexSourcePath, const GLchar ∗fragmentSourcePath)
- void **Use** ()

Public Attributes

- GLint **Program**

The documentation for this class was generated from the following files:

- include/Shader.h
- src/Shader.cpp

Index