

MunchkinPP

Generated by Doxygen 1.9.1



<b>1 Namespace Index</b>	<b>1</b>
1.1 Namespace List	1
<b>2 Class Index</b>	<b>3</b>
2.1 Class List	3
<b>3 File Index</b>	<b>5</b>
3.1 File List	5
<b>4 Namespace Documentation</b>	<b>7</b>
4.1 badthings Namespace Reference	7
4.1.1 Function Documentation	7
4.1.1.1 emptyBadThings()	7
4.1.1.2 looseClass()	7
4.1.1.3 looseHand()	8
4.1.1.4 looseLevel()	8
4.1.1.5 maleDeadFemaleLevelDown()	8
4.1.1.6 playerDies()	8
4.2 cardtypeaction Namespace Reference	8
4.2.1 Function Documentation	8
4.2.1.1 curse()	9
4.2.1.2 item()	9
4.2.1.3 itemBuff()	9
4.2.1.4 joker()	9
4.2.1.5 lvlUp()	9
4.2.1.6 monster()	9
4.2.1.7 munchClass()	10
4.2.1.8 race()	10
4.3 extras Namespace Reference	10
4.4 munch_tut Namespace Reference	10
4.4.1 Typedef Documentation	10
4.4.1.1 Color	10
<b>5 Class Documentation</b>	<b>11</b>
5.1 BadThingsRetVal Struct Reference	11
5.2 Button Class Reference	11
5.2.1 Constructor & Destructor Documentation	12
5.2.1.1 Button() [1/3]	12
5.2.1.2 Button() [2/3]	12
5.2.1.3 Button() [3/3]	12
5.2.2 Member Function Documentation	12
5.2.2.1 draw()	12
5.2.2.2 operator=() [1/2]	12
5.2.2.3 operator=() [2/2]	13

5.2.2.4 poll_click()	13
5.2.3 Member Data Documentation	13
5.2.3.1 callback	13
5.2.3.2 color	13
5.2.3.3 id	13
5.2.3.4 origin	13
5.2.3.5 rect	13
5.2.3.6 visible	14
5.3 CardTypeRetVal Struct Reference	14
5.4 ExtrasRetVal Struct Reference	14
5.5 GameState Struct Reference	14
5.5.1 Member Data Documentation	14
5.5.1.1 buttons	15
5.5.1.2 canvas_size	15
5.5.1.3 end_turn	15
5.5.1.4 mouseparams	15
5.5.1.5 player01	15
5.5.1.6 remove_card	15
5.5.1.7 run_away	15
5.5.1.8 should_continue	15
5.5.1.9 should_exit	16
5.6 InputEvent Struct Reference	16
5.6.1 Member Data Documentation	16
5.6.1.1 type	16
5.7 MouseParams Struct Reference	16
5.7.1 Member Data Documentation	16
5.7.1.1 clickP	17
5.7.1.2 event	17
5.7.1.3 markerCorner	17
5.7.1.4 markerId	17
5.7.1.5 poly	17
5.7.1.6 tutText	17
5.8 MunchkinCard Class Reference	17
5.8.1 Constructor & Destructor Documentation	18
5.8.1.1 MunchkinCard() [1/4]	18
5.8.1.2 MunchkinCard() [2/4]	19
5.8.1.3 MunchkinCard() [3/4]	19
5.8.1.4 MunchkinCard() [4/4]	19
5.8.2 Member Function Documentation	19
5.8.2.1 cardsConstr()	19
5.8.2.2 operator=() [1/2]	19
5.8.2.3 operator=() [2/2]	20

5.8.3 Member Data Documentation	20
5.8.3.1 badThings	20
5.8.3.2 badThingsFunc	20
5.8.3.3 bonis	20
5.8.3.4 cardName	20
5.8.3.5 debuff	20
5.8.3.6 effect	20
5.8.3.7 handsNeeded	21
5.8.3.8 itemEffect	21
5.8.3.9 itemLarge	21
5.8.3.10 itemNeeds	21
5.8.3.11 itemType	21
5.8.3.12 itemValue	21
5.8.3.13 lvUp	21
5.8.3.14 markerID	21
5.8.3.15 monsStrength	22
5.8.3.16 parentCardType	22
5.8.3.17 strengthBoni	22
5.8.3.18 treasures	22
5.8.3.19 type	22
5.9 PlayerStats Struct Reference	22
5.9.1 Member Data Documentation	23
5.9.1.1 availableClasses	23
5.9.1.2 availableRaces	23
5.9.1.3 bonis	23
5.9.1.4 carriesLargeItems	23
5.9.1.5 curses	24
5.9.1.6 hands	24
5.9.1.7 hasArmor	24
5.9.1.8 hasHat	24
5.9.1.9 hasShoes	24
5.9.1.10 itemEffects	24
5.9.1.11 lvl	24
5.9.1.12 munchClasses	24
5.9.1.13 munchRaces	25
5.9.1.14 runStrength	25
5.9.1.15 sex	25
5.9.1.16 strength	25
<b>6 File Documentation</b>	<b>27</b>
6.1 BadThings.cpp File Reference	27
6.2 BadThings.h File Reference	27

6.2.1 Typedef Documentation	28
6.2.1.1 BadThingsFunc	28
6.3 Button.cpp File Reference	28
6.4 Button.h File Reference	28
6.4.1 Typedef Documentation	28
6.4.1.1 button_callback	28
6.4.2 Enumeration Type Documentation	28
6.4.2.1 ButtonOrigin	28
6.5 CardTypeActions.cpp File Reference	29
6.5.1 Function Documentation	29
6.5.1.1 runAway()	29
6.6 CardTypeActions.h File Reference	29
6.6.1 Typedef Documentation	30
6.6.1.1 CardTypeFunc	30
6.7 common.h File Reference	30
6.8 Extras.cpp File Reference	30
6.9 Extras.h File Reference	31
6.9.1 Typedef Documentation	31
6.9.1.1 ExtrasFunc	31
6.10 GameState.h File Reference	31
6.10.1 Enumeration Type Documentation	31
6.10.1.1 MouseEvent	31
6.11 InputEvent.h File Reference	32
6.11.1 Enumeration Type Documentation	32
6.11.1.1 EventType	32
6.12 main.cpp File Reference	32
6.12.1 Function Documentation	33
6.12.1.1 button_continue()	34
6.12.1.2 button_end_turn()	34
6.12.1.3 button_exit()	34
6.12.1.4 callBackFunction()	34
6.12.1.5 cameraCalibration()	34
6.12.1.6 cameraCalibrationProcess()	34
6.12.1.7 chooseCardColor()	35
6.12.1.8 createArucoMarkers()	35
6.12.1.9 createKnownBoardPositions()	35
6.12.1.10 getChessboardCorners()	35
6.12.1.11 loadCameraCalibration()	35
6.12.1.12 logic()	35
6.12.1.13 main()	36
6.12.1.14 saveCameraCalibration()	36
6.12.1.15 startWebcamMonitoring()	36

6.12.1.16 update()	36
6.12.2 Variable Documentation	36
6.12.2.1 arucoSquareDimension	36
6.12.2.2 calbirationSquareDimension	36
6.12.2.3 cards	37
6.12.2.4 chessboardDimensions	37
6.12.2.5 functMap	37
6.12.2.6 gamestate	37
6.13 MunchkinCards.cpp File Reference	37
6.14 MunchkinCards.h File Reference	37
6.14.1 Enumeration Type Documentation	38
6.14.1.1 CardType	38
6.14.1.2 ItemType	38
6.14.1.3 ParentCardType	39
6.15 TTMunchkinTut.cpp File Reference	39
<b>Index</b>	<b>41</b>





# Chapter 1

## Namespace Index

### 1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">badthings</a>	7
<a href="#">cardtypeaction</a>	8
<a href="#">extras</a>	10
<a href="#">munch_tut</a>	10



## Chapter 2

# Class Index

### 2.1 Class List

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

<a href="#">BadThingsRetVal</a>	11
<a href="#">Button</a>	11
<a href="#">CardTypeRetVal</a>	14
<a href="#">ExtrasRetVal</a>	14
<a href="#">GameState</a>	14
<a href="#">InputEvent</a>	16
<a href="#">MouseParams</a>	16
<a href="#">MunchkinCard</a>	17
<a href="#">PlayerStats</a>	22



## Chapter 3

# File Index

### 3.1 File List

Here is a list of all files with brief descriptions:

<a href="#">BadThings.cpp</a>	27
<a href="#">BadThings.h</a>	27
<a href="#">Button.cpp</a>	28
<a href="#">Button.h</a>	28
<a href="#">CardTypeActions.cpp</a>	29
<a href="#">CardTypeActions.h</a>	29
<a href="#">common.h</a>	30
<a href="#">Extras.cpp</a>	30
<a href="#">Extras.h</a>	31
<a href="#">GameState.h</a>	31
<a href="#">InputEvent.h</a>	32
<a href="#">main.cpp</a>	32
<a href="#">MunchkinCards.cpp</a>	37
<a href="#">MunchkinCards.h</a>	37
<a href="#">TTMunchkinTut.cpp</a>	39



## Chapter 4

# Namespace Documentation

### 4.1 badthings Namespace Reference

#### Functions

- [BadThingsRetVal emptyBadThings](#) ([GameState](#) &[gamestate](#), const [MunchkinCard](#) &card)
- [BadThingsRetVal looseHand](#) ([GameState](#) &[gamestate](#), const [MunchkinCard](#) &card)
- [BadThingsRetVal looseClass](#) ([GameState](#) &[gamestate](#), const [MunchkinCard](#) &card)
- [BadThingsRetVal playerDies](#) ([GameState](#) &[gamestate](#), const [MunchkinCard](#) &card)
- [BadThingsRetVal looseLevel](#) ([GameState](#) &[gamestate](#), const [MunchkinCard](#) &card)
- [BadThingsRetVal maleDeadFemaleLevelDown](#) ([GameState](#) &[gamestate](#), const [MunchkinCard](#) &card)

#### 4.1.1 Function Documentation

##### 4.1.1.1 emptyBadThings()

```
BadThingsRetVal badthings::emptyBadThings (  
    GameState & gamestate,  
    const MunchkinCard & card )
```

##### 4.1.1.2 looseClass()

```
BadThingsRetVal badthings::looseClass (  
    GameState & gamestate,  
    const MunchkinCard & card )
```

#### 4.1.1.3 looseHand()

```
BadThingsRetVal badthings::looseHand (
    GameState & gamestate,
    const MunchkinCard & card )
```

#### 4.1.1.4 looseLevel()

```
BadThingsRetVal badthings::looseLevel (
    GameState & gamestate,
    const MunchkinCard & card )
```

#### 4.1.1.5 maleDeadFemaleLevelDown()

```
BadThingsRetVal badthings::maleDeadFemaleLevelDown (
    GameState & gamestate,
    const MunchkinCard & card )
```

#### 4.1.1.6 playerDies()

```
BadThingsRetVal badthings::playerDies (
    GameState & gamestate,
    const MunchkinCard & card )
```

## 4.2 cardtypeaction Namespace Reference

### Functions

- [CardTypeRetVal curse](#) ([GameState](#) &[gamestate](#), const [MunchkinCard](#) &[card](#))
- [CardTypeRetVal joker](#) ([GameState](#) &[gamestate](#), const [MunchkinCard](#) &[card](#))
- [CardTypeRetVal monster](#) ([GameState](#) &[gamestate](#), const [MunchkinCard](#) &[card](#))
- [CardTypeRetVal munchClass](#) ([GameState](#) &[gamestate](#), const [MunchkinCard](#) &[card](#))
- [CardTypeRetVal race](#) ([GameState](#) &[gamestate](#), const [MunchkinCard](#) &[card](#))
- [CardTypeRetVal item](#) ([GameState](#) &[gamestate](#), const [MunchkinCard](#) &[card](#))
- [CardTypeRetVal itemBuff](#) ([GameState](#) &[gamestate](#), const [MunchkinCard](#) &[card](#))
- [CardTypeRetVal lvlUp](#) ([GameState](#) &[gamestate](#), const [MunchkinCard](#) &[card](#))

#### 4.2.1 Function Documentation



#### 4.2.1.1 curse()

```
CardTypeRetVal cardtypeaction::curse (
    GameState & gamestate,
    const MunchkinCard & card )
```

#### 4.2.1.2 item()

```
CardTypeRetVal cardtypeaction::item (
    GameState & gamestate,
    const MunchkinCard & card )
```

#### 4.2.1.3 itemBuff()

```
CardTypeRetVal cardtypeaction::itemBuff (
    GameState & gamestate,
    const MunchkinCard & card )
```

#### 4.2.1.4 joker()

```
CardTypeRetVal cardtypeaction::joker (
    GameState & gamestate,
    const MunchkinCard & card )
```

#### 4.2.1.5 lvlUp()

```
CardTypeRetVal cardtypeaction::lvlUp (
    GameState & gamestate,
    const MunchkinCard & card )
```

#### 4.2.1.6 monster()

```
CardTypeRetVal cardtypeaction::monster (
    GameState & gamestate,
    const MunchkinCard & card )
```

#### 4.2.1.7 munchClass()

```
CardTypeRetVal cardtypeaction::munchClass (
    GameState & gamestate,
    const MunchkinCard & card )
```

#### 4.2.1.8 race()

```
CardTypeRetVal cardtypeaction::race (
    GameState & gamestate,
    const MunchkinCard & card )
```

### 4.3 extras Namespace Reference

### 4.4 munch\_tut Namespace Reference

#### Typedefs

- using `Color` = `std::array< int, 3 >`

#### 4.4.1 Typedef Documentation

##### 4.4.1.1 Color

```
using munch_tut::Color = typedef std::array<int, 3>
```

## Chapter 5

# Class Documentation

### 5.1 BadThingsRetVal Struct Reference

```
#include <BadThings.h>
```

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

- [BadThings.h](#)

### 5.2 Button Class Reference

```
#include <Button.h>
```

#### Public Member Functions

- [Button](#) (int \_id, const cv::Rect &\_rect, const cv::Scalar &\_color, bool \_visible=true, [ButtonOrigin](#) \_↔  
origin=[ButtonOrigin::topleft](#), const [button\\_callback](#) &\_callback={})
- [Button](#) (const [Button](#) &)=default
- [Button](#) ([Button](#) &&)=default
- [Button](#) & [operator=](#) (const [Button](#) &)=default
- [Button](#) & [operator=](#) ([Button](#) &&)=default
- bool [poll\\_click](#) (const cv::Point &point, const cv::Size &canvas\_size) const
- void [draw](#) (cv::Mat &canvas) const

#### Public Attributes

- int [id](#)
- cv::Rect [rect](#)
- [button\\_callback](#) [callback](#)
- cv::Scalar [color](#)
- bool [visible](#)
- [ButtonOrigin](#) [origin](#)

## 5.2.1 Constructor & Destructor Documentation

### 5.2.1.1 Button() [1/3]

```
Button::Button (
    int _id,
    const cv::Rect & _rect,
    const cv::Scalar & _color,
    bool _visible = true,
    ButtonOrigin _origin = ButtonOrigin::topleft,
    const button_callback & _callback = {} )
```

### 5.2.1.2 Button() [2/3]

```
Button::Button (
    const Button & ) [default]
```

### 5.2.1.3 Button() [3/3]

```
Button::Button (
    Button && ) [default]
```

## 5.2.2 Member Function Documentation

### 5.2.2.1 draw()

```
void Button::draw (
    cv::Mat & canvas ) const
```

### 5.2.2.2 operator=() [1/2]

```
Button& Button::operator= (
    Button && ) [default]
```

### 5.2.2.3 operator=() [2/2]

```
Button& Button::operator= (
    const Button & ) [default]
```

### 5.2.2.4 poll\_click()

```
bool Button::poll_click (
    const cv::Point & point,
    const cv::Size & canvas_size ) const
```

## 5.2.3 Member Data Documentation

### 5.2.3.1 callback

```
button_callback Button::callback
```

### 5.2.3.2 color

```
cv::Scalar Button::color
```

### 5.2.3.3 id

```
int Button::id
```

### 5.2.3.4 origin

```
ButtonOrigin Button::origin
```

### 5.2.3.5 rect

```
cv::Rect Button::rect
```

#### 5.2.3.6 visible

```
bool Button::visible
```

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

- [Button.h](#)
- [Button.cpp](#)

### 5.3 CardTypeRetVal Struct Reference

```
#include <CardTypeActions.h>
```

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

- [CardTypeActions.h](#)

### 5.4 ExtrasRetVal Struct Reference

```
#include <Extras.h>
```

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

- [Extras.h](#)

### 5.5 GameState Struct Reference

```
#include <GameState.h>
```

#### Public Attributes

- [PlayerStats](#) player01
- [MouseParams](#) mouseparams
- `std::vector< Button >` buttons
- `bool` should\_exit
- `bool` should\_continue
- `bool` end\_turn
- `bool` run\_away
- `bool` remove\_card
- `cv::Size` canvas\_size

#### 5.5.1 Member Data Documentation

#### 5.5.1.1 buttons

```
std::vector<Button> GameState::buttons
```

#### 5.5.1.2 canvas\_size

```
cv::Size GameState::canvas_size
```

#### 5.5.1.3 end\_turn

```
bool GameState::end_turn
```

#### 5.5.1.4 mouseparams

```
MouseParams GameState::mouseparams
```

#### 5.5.1.5 player01

```
PlayerStats GameState::player01
```

#### 5.5.1.6 remove\_card

```
bool GameState::remove_card
```

#### 5.5.1.7 run\_away

```
bool GameState::run_away
```

#### 5.5.1.8 should\_continue

```
bool GameState::should_continue
```

#### 5.5.1.9 should\_exit

```
bool GameState::should_exit
```

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

- [GameState.h](#)

## 5.6 InputEvent Struct Reference

```
#include <InputEvent.h>
```

### Public Attributes

- [EventType](#) type

#### 5.6.1 Member Data Documentation

##### 5.6.1.1 type

```
EventType InputEvent::type
```

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

- [InputEvent.h](#)

## 5.7 MouseParams Struct Reference

```
#include <GameState.h>
```

### Public Attributes

- `vector< cv::Point >` [poly](#)
- `int` [markerId](#)
- `vector< cv::Point2f >` [markerCorner](#)
- `vector< string >` [tutText](#)
- `cv::Point` [clickP](#)
- [MouseEvent](#) event

#### 5.7.1 Member Data Documentation



#### 5.7.1.1 clickP

```
cv::Point MouseParams::clickP
```

#### 5.7.1.2 event

```
MouseEvent MouseParams::event
```

#### 5.7.1.3 markerCorner

```
vector<cv::Point2f> MouseParams::markerCorner
```

#### 5.7.1.4 markerId

```
int MouseParams::markerId
```

#### 5.7.1.5 poly

```
vector<cv::Point> MouseParams::poly
```

#### 5.7.1.6 tutText

```
vector<string> MouseParams::tutText
```

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

- [GameState.h](#)

## 5.8 MunchkinCard Class Reference

```
#include <MunchkinCards.h>
```

## Public Member Functions

- [MunchkinCard](#) ()
- [MunchkinCard](#) (int \_markerID, const string &\_cardName, const string &\_effect, const string &\_badThings, const string &\_itemEffect, const string &\_itemNeeds, vector< string > \_bonis, [ParentCardType](#) \_parent←CardType, [CardType](#) \_type, [ItemType](#) \_itemType, int \_strengthBoni, int \_debuff, int \_monsStrength, int \_lvlUp, int \_treasures, int \_itemValue, int \_handsNeeded, bool \_itemLarge)
- [MunchkinCard](#) (const [MunchkinCard](#) &other)=default
- [MunchkinCard](#) ([MunchkinCard](#) &&other)=default
- [MunchkinCard](#) & operator= (const [MunchkinCard](#) &other)=default
- [MunchkinCard](#) & operator= ([MunchkinCard](#) &&other)=default

## Static Public Member Functions

- static vector< [MunchkinCard](#) > [cardsConstr](#) ()

## Public Attributes

- string [cardName](#)
- string [effect](#)
- vector< string > [bonis](#)
- string [badThings](#)
- string [itemEffect](#)
- string [itemNeeds](#)
- [ParentCardType](#) [parentCardType](#)
- [CardType](#) [type](#)
- [ItemType](#) [itemType](#)
- int [markerID](#)
- int [strengthBoni](#)
- int [debuff](#)
- int [monsStrength](#)
- int [lvlUp](#)
- int [treasures](#)
- int [itemValue](#)
- int [handsNeeded](#)
- bool [itemLarge](#)
- [BadThingsFunc](#) [badThingsFunc](#)

## 5.8.1 Constructor & Destructor Documentation

### 5.8.1.1 MunchkinCard() [1/4]

```
MunchkinCard::MunchkinCard ( ) [inline]
```

### 5.8.1.2 MunchkinCard() [2/4]

```

MunchkinCard::MunchkinCard (
    int _markerID,
    const string & _cardName,
    const string & _effect,
    const string & _badThings,
    const string & _itemEffect,
    const string & _itemNeeds,
    vector< string > _bonis,
    ParentCardType _parentCardType,
    CardType _type,
    ItemType _itemType,
    int _strengthBoni,
    int _debuff,
    int _monsStrength,
    int _lvlUp,
    int _treasures,
    int _itemValue,
    int _handsNeeded,
    bool _itemLarge )

```

### 5.8.1.3 MunchkinCard() [3/4]

```

MunchkinCard::MunchkinCard (
    const MunchkinCard & other ) [default]

```

### 5.8.1.4 MunchkinCard() [4/4]

```

MunchkinCard::MunchkinCard (
    MunchkinCard && other ) [default]

```

## 5.8.2 Member Function Documentation

### 5.8.2.1 cardsConstr()

```

vector< MunchkinCard > MunchkinCard::cardsConstr ( ) [static]

```

### 5.8.2.2 operator=() [1/2]

```

MunchkinCard& MunchkinCard::operator= (
    const MunchkinCard & other ) [default]

```

### 5.8.2.3 operator=() [2/2]

```
MunchkinCard& MunchkinCard::operator= (  
    MunchkinCard && other ) [default]
```

## 5.8.3 Member Data Documentation

### 5.8.3.1 badThings

```
string MunchkinCard::badThings
```

### 5.8.3.2 badThingsFunc

```
BadThingsFunc MunchkinCard::badThingsFunc
```

### 5.8.3.3 bonis

```
vector<string> MunchkinCard::bonis
```

### 5.8.3.4 cardName

```
string MunchkinCard::cardName
```

### 5.8.3.5 debuff

```
int MunchkinCard::debuff
```

### 5.8.3.6 effect

```
string MunchkinCard::effect
```

### 5.8.3.7 handsNeeded

```
int MunchkinCard::handsNeeded
```

### 5.8.3.8 itemEffect

```
string MunchkinCard::itemEffect
```

### 5.8.3.9 itemLarge

```
bool MunchkinCard::itemLarge
```

### 5.8.3.10 itemNeeds

```
string MunchkinCard::itemNeeds
```

### 5.8.3.11 itemType

```
ItemType MunchkinCard::itemType
```

### 5.8.3.12 itemValue

```
int MunchkinCard::itemValue
```

### 5.8.3.13 lvlUp

```
int MunchkinCard::lvlUp
```

### 5.8.3.14 markerID

```
int MunchkinCard::markerID
```

#### 5.8.3.15 monsStrength

```
int MunchkinCard::monsStrength
```

#### 5.8.3.16 parentCardType

```
ParentCardType MunchkinCard::parentCardType
```

#### 5.8.3.17 strengthBoni

```
int MunchkinCard::strengthBoni
```

#### 5.8.3.18 treasures

```
int MunchkinCard::treasures
```

#### 5.8.3.19 type

```
CardType MunchkinCard::type
```

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

- [MunchkinCards.h](#)
- [MunchkinCards.cpp](#)

## 5.9 PlayerStats Struct Reference

```
#include <GameState.h>
```

## Public Attributes

- string [sex](#)
- vector< string > [bonis](#)
- vector< string > [curses](#)
- vector< string > [itemEffects](#)
- vector< string > [munchClasses](#)
- vector< string > [munchRaces](#)
- int [lvl](#)
- int [strength](#)
- int [hands](#)
- int [runStrength](#)
- int [availableClasses](#)
- int [availableRaces](#)
- bool [carriesLargeItems](#)
- bool [hasArmor](#)
- bool [hasHat](#)
- bool [hasShoes](#)

## 5.9.1 Member Data Documentation

### 5.9.1.1 availableClasses

```
int PlayerStats::availableClasses
```

### 5.9.1.2 availableRaces

```
int PlayerStats::availableRaces
```

### 5.9.1.3 bonis

```
vector<string> PlayerStats::bonis
```

### 5.9.1.4 carriesLargeItems

```
bool PlayerStats::carriesLargeItems
```

#### 5.9.1.5 curses

```
vector<string> PlayerStats::curses
```

#### 5.9.1.6 hands

```
int PlayerStats::hands
```

#### 5.9.1.7 hasArmor

```
bool PlayerStats::hasArmor
```

#### 5.9.1.8 hasHat

```
bool PlayerStats::hasHat
```

#### 5.9.1.9 hasShoes

```
bool PlayerStats::hasShoes
```

#### 5.9.1.10 itemEffects

```
vector<string> PlayerStats::itemEffects
```

#### 5.9.1.11 lvl

```
int PlayerStats::lvl
```

#### 5.9.1.12 munchClasses

```
vector<string> PlayerStats::munchClasses
```



#### 5.9.1.13 munchRaces

```
vector<string> PlayerStats::munchRaces
```

#### 5.9.1.14 runStrength

```
int PlayerStats::runStrength
```

#### 5.9.1.15 sex

```
string PlayerStats::sex
```

#### 5.9.1.16 strength

```
int PlayerStats::strength
```

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

- [GameState.h](#)



## Chapter 6

# File Documentation

### 6.1 BadThings.cpp File Reference

```
#include "BadThings.h"  
#include "MunchkinCards.h"  
#include "GameState.h"
```

### 6.2 BadThings.h File Reference

```
#include <functional>
```

#### Classes

- struct [BadThingsRetVal](#)

#### Namespaces

- [badthings](#)

#### Typedefs

- using [BadThingsFunc](#) = std::function< [BadThingsRetVal](#)([GameState](#) &, const [MunchkinCard](#) &)>

#### Functions

- [BadThingsRetVal](#) [badthings::emptyBadThings](#) ([GameState](#) &[gamestate](#), const [MunchkinCard](#) &[card](#))
- [BadThingsRetVal](#) [badthings::looseHand](#) ([GameState](#) &[gamestate](#), const [MunchkinCard](#) &[card](#))
- [BadThingsRetVal](#) [badthings::looseClass](#) ([GameState](#) &[gamestate](#), const [MunchkinCard](#) &[card](#))
- [BadThingsRetVal](#) [badthings::playerDies](#) ([GameState](#) &[gamestate](#), const [MunchkinCard](#) &[card](#))
- [BadThingsRetVal](#) [badthings::looseLevel](#) ([GameState](#) &[gamestate](#), const [MunchkinCard](#) &[card](#))
- [BadThingsRetVal](#) [badthings::maleDeadFemaleLevelDown](#) ([GameState](#) &[gamestate](#), const [MunchkinCard](#) &[card](#))

## 6.2.1 Typedef Documentation

### 6.2.1.1 BadThingsFunc

```
using BadThingsFunc = std::function<BadThingsRetVal(GameState&, const MunchkinCard&)>
```

## 6.3 Button.cpp File Reference

```
#include "Button.h"
```

## 6.4 Button.h File Reference

```
#include "opencv2/opencv.hpp"  
#include "opencv2/imgcodecs.hpp"  
#include "opencv2/imgproc.hpp"  
#include <functional>
```

### Classes

- class [Button](#)

### Typedefs

- using [button\\_callback](#) = std::function< void(const [Button](#) &)>

### Enumerations

- enum class [ButtonOrigin](#) { [topleft](#) , [topright](#) , [bottomleft](#) , [bottomright](#) }

## 6.4.1 Typedef Documentation

### 6.4.1.1 button\_callback

```
using button_callback = std::function<void(const Button&)>
```

## 6.4.2 Enumeration Type Documentation

### 6.4.2.1 ButtonOrigin

```
enum ButtonOrigin [strong]
```

### Enumerator

toleft	
topright	
bottomleft	
bottomright	

## 6.5 CardTypeActions.cpp File Reference

```
#include "CardTypeActions.h"
#include <iostream>
#include "GameState.h"
#include "MunchkinCards.h"
```

### Functions

- bool [runAway](#) ([GameState](#) &[gamestate](#))

#### 6.5.1 Function Documentation

##### 6.5.1.1 runAway()

```
bool runAway (
    GameState & gamestate )
```

## 6.6 CardTypeActions.h File Reference

```
#include <functional>
```

### Classes

- struct [CardTypeRetVal](#)

### Namespaces

- [cardtypeaction](#)

## Typedefs

- using `CardTypeFunc` = `std::function< CardTypeRetVal(GameState &, const MunchkinCard &)>`

## Functions

- `CardTypeRetVal cardtypeaction::curse (GameState &gamestate, const MunchkinCard &card)`
- `CardTypeRetVal cardtypeaction::joker (GameState &gamestate, const MunchkinCard &card)`
- `CardTypeRetVal cardtypeaction::monster (GameState &gamestate, const MunchkinCard &card)`
- `CardTypeRetVal cardtypeaction::munchClass (GameState &gamestate, const MunchkinCard &card)`
- `CardTypeRetVal cardtypeaction::race (GameState &gamestate, const MunchkinCard &card)`
- `CardTypeRetVal cardtypeaction::item (GameState &gamestate, const MunchkinCard &card)`
- `CardTypeRetVal cardtypeaction::itemBuff (GameState &gamestate, const MunchkinCard &card)`
- `CardTypeRetVal cardtypeaction::lvlUp (GameState &gamestate, const MunchkinCard &card)`

### 6.6.1 Typedef Documentation

#### 6.6.1.1 CardTypeFunc

```
using CardTypeFunc = std::function<CardTypeRetVal(GameState&, const MunchkinCard&)>
```

## 6.7 common.h File Reference

```
#include <array>
```

## Namespaces

- `munch_tut`

## Typedefs

- using `munch_tut::Color` = `std::array< int, 3 >`

## 6.8 Extras.cpp File Reference

```
#include "Extras.h"
#include "GameState.h"
#include "MunchkinCards.h"
```

## 6.9 Extras.h File Reference

```
#include <functional>
```

### Classes

- struct [ExtrasRetVal](#)

### Namespaces

- [extras](#)

### Typedefs

- using [ExtrasFunc](#) = std::function< [ExtrasRetVal](#)([GameState](#) &, const [MunchkinCard](#) &)>

## 6.9.1 Typedef Documentation

### 6.9.1.1 ExtrasFunc

```
using ExtrasFunc = std::function<ExtrasRetVal(GameState&, const MunchkinCard&)>
```

## 6.10 GameState.h File Reference

```
#include "opencv2/opencv.hpp"  
#include "opencv2/imgcodecs.hpp"  
#include "opencv2/imgproc.hpp"  
#include <string>  
#include <vector>  
#include "Button.h"
```

### Classes

- struct [MouseParams](#)
- struct [PlayerStats](#)
- struct [GameState](#)

### Enumerations

- enum class [MouseEvent](#) { [lclick](#) , [rclick](#) , [move](#) , [none](#) }

## 6.10.1 Enumeration Type Documentation

### 6.10.1.1 MouseEvent

```
enum MouseEvent [strong]
```

**Enumerator**

lclick	
rclick	
move	
none	

## 6.11 InputEvent.h File Reference

**Classes**

- struct [InputEvent](#)

**Enumerations**

- enum class [EventType](#) {  
[LmPress](#) , [LmRelease](#) , [RmPress](#) , [RmRelease](#) ,  
[KeyPress](#) , [KeyRelease](#) }

### 6.11.1 Enumeration Type Documentation

#### 6.11.1.1 EventType

```
enum EventType [strong]
```

**Enumerator**

LmPress	
LmRelease	
RmPress	
RmRelease	
KeyPress	
KeyRelease	

## 6.12 main.cpp File Reference

```
#include "opencv2/opencv.hpp"
#include "opencv2/imgcodecs.hpp"
#include "opencv2/imgproc.hpp"
#include "opencv2/highgui.hpp"
#include "opencv2/aruco.hpp"
```



```

#include "opencv2/calib3d.hpp"
#include "GameState.h"
#include "MunchkinCards.h"
#include "CardTypeActions.h"
#include "common.h"
#include "Button.h"
#include <sstream>
#include <iostream>
#include <fstream>
#include <string>
#include <stdexcept>
#include <unordered_map>
#include <functional>

```

## Functions

- void [button\\_exit](#) (const [Button](#) &b)
- void [button\\_continue](#) (const [Button](#) &b)
- void [button\\_end\\_turn](#) (const [Button](#) &b)
- void [update](#) ()
- void [logic](#) ()
- void [createArucoMarkers](#) ()
- void [createKnownBoardPositions](#) (Size boardSize, float squareEdgeLength, vector< [Point3f](#) > &corners)
- void [getChessboardCorners](#) (vector< [Mat](#) > images, vector< vector< [Point2f](#) >> &allFoundCorners, bool showResults=false)
- bool [saveCameraCalibration](#) (string name, [Mat](#) cameraMatrix, [Mat](#) distanceCoefficients)
- bool [loadCameraCalibration](#) (string name, [Mat](#) &cameraMatrix, [Mat](#) &distanceCoefficients)
- void [cameraCalibration](#) (vector< [Mat](#) > calibrationImages, Size boardSize, float squareEdgeLength, [Mat](#) &cameraMatrix, [Mat](#) &distanceCoefficients)
- void [callBackFunction](#) (int event, int x, int y, int flags, void \*userdata)
- [Color chooseCardColor](#) (int markerId)
- int [startWebcamMonitoring](#) (const [Mat](#) &cameraMatrix, const [Mat](#) &distanceCoefficients, float arucoSquareDimensions)
- void [cameraCalibrationProcess](#) ([Mat](#) &cameraMatrix, [Mat](#) &distanceCoefficients)
- int [main](#) (int argv, char \*\*argc)

## Variables

- const float [calibrationSquareDimension](#) = 0.025f
- const float [arucoSquareDimension](#) = 0.02f
- const Size [chessboardDimensions](#) = Size(6, 9)
- vector< [MunchkinCard](#) > [cards](#)
- std::unordered\_map< [CardType](#), [CardTypeFunc](#) > [functMap](#)
- [GameState](#) [gamestate](#)

### 6.12.1 Function Documentation

#### 6.12.1.1 button\_continue()

```
void button_continue (
    const Button & b )
```

#### 6.12.1.2 button\_end\_turn()

```
void button_end_turn (
    const Button & b )
```

#### 6.12.1.3 button\_exit()

```
void button_exit (
    const Button & b )
```

#### 6.12.1.4 callBackFunction()

```
void callBackFunction (
    int event,
    int x,
    int y,
    int flags,
    void * userdata )
```

#### 6.12.1.5 cameraCalibration()

```
void cameraCalibration (
    vector< Mat > calibrationImages,
    Size boardSize,
    float squareEdgeLength,
    Mat & cameraMatrix,
    Mat & distanceCoefficients )
```

#### 6.12.1.6 cameraCalibrationProcess()

```
void cameraCalibrationProcess (
    Mat & cameraMatrix,
    Mat & distanceCoefficients )
```

#### 6.12.1.7 chooseCardColor()

```
Color chooseCardColor (
    int markerId )
```

#### 6.12.1.8 createArucoMarkers()

```
void createArucoMarkers ( )
```

#### 6.12.1.9 createKnownBoardPositions()

```
void createKnownBoardPositions (
    Size boardSize,
    float squareEdgeLength,
    vector< Point3f > & corners )
```

#### 6.12.1.10 getChessboardCorners()

```
void getChessboardCorners (
    vector< Mat > images,
    vector< vector< Point2f >> & allFoundCorners,
    bool showResults = false )
```

#### 6.12.1.11 loadCameraCalibration()

```
bool loadCameraCalibration (
    string name,
    Mat & cameraMatrix,
    Mat & distanceCoefficients )
```

#### 6.12.1.12 logic()

```
void logic ( )
```

#### 6.12.1.13 main()

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

#### 6.12.1.14 saveCameraCalibration()

```
bool saveCameraCalibration (
    string name,
    Mat cameraMatrix,
    Mat distanceCoefficients )
```

#### 6.12.1.15 startWebcamMonitoring()

```
int startWebcamMonitoring (
    const Mat & cameraMatrix,
    const Mat & distanceCoefficients,
    float arucoSquareDimensions )
```

#### 6.12.1.16 update()

```
void update ( )
```

### 6.12.2 Variable Documentation

#### 6.12.2.1 arucoSquareDimension

```
const float arucoSquareDimension = 0.02f
```

#### 6.12.2.2 calibrationSquareDimension

```
const float calibrationSquareDimension = 0.025f
```

### 6.12.2.3 cards

```
vector<MunchkinCard> cards
```

### 6.12.2.4 chessboardDimensions

```
const Size chessboardDimensions = Size(6, 9)
```

### 6.12.2.5 functMap

```
std::unordered_map<CardType, CardTypeFunc> functMap
```

### 6.12.2.6 gamestate

```
GameState gamestate
```

## 6.13 MunchkinCards.cpp File Reference

```
#include "MunchkinCards.h"
```

## 6.14 MunchkinCards.h File Reference

```
#include <string>
#include <vector>
#include "opencv2/opencv.hpp"
#include "opencv2/imgcodecs.hpp"
#include "opencv2/imgproc.hpp"
#include <functional>
#include "GameState.h"
#include "BadThings.h"
```

### Classes

- class [MunchkinCard](#)

## Enumerations

- enum class `ParentCardType` { `door` , `treasure` }
- enum class `CardType` {  
`curse` , `munchClass` , `joker` , `lvlUp` ,  
`itemBuff` , `monster` , `race` , `item` ,  
`removeCard` }
- enum class `ItemType` {  
`armor` , `shoes` , `hat` , `boni` ,  
`weapon` , `joker` , `clothing` }

### 6.14.1 Enumeration Type Documentation

#### 6.14.1.1 CardType

```
enum CardType [strong]
```

##### Enumerator

curse	
munchClass	
joker	
lvlUp	
itemBuff	
monster	
race	
item	
removeCard	

#### 6.14.1.2 ItemType

```
enum ItemType [strong]
```

##### Enumerator

armor	
shoes	
hat	
boni	
weapon	
joker	
clothing	

### 6.14.1.3 ParentCardType

```
enum ParentCardType [strong]
```

Enumerator

door	
treasure	

## 6.15 TTMunchkinTut.cpp File Reference





# Index

- armor
  - MunchkinCards.h, [38](#)
- arucoSquareDimension
  - main.cpp, [36](#)
- availableClasses
  - PlayerStats, [23](#)
- availableRaces
  - PlayerStats, [23](#)
- badThings
  - MunchkinCard, [20](#)
- badthings, [7](#)
  - emptyBadThings, [7](#)
  - looseClass, [7](#)
  - looseHand, [7](#)
  - looseLevel, [8](#)
  - maleDeadFemaleLevelDown, [8](#)
  - playerDies, [8](#)
- BadThings.cpp, [27](#)
- BadThings.h, [27](#)
  - BadThingsFunc, [28](#)
- BadThingsFunc
  - BadThings.h, [28](#)
- badThingsFunc
  - MunchkinCard, [20](#)
- BadThingsRetVal, [11](#)
- boni
  - MunchkinCards.h, [38](#)
- bonis
  - MunchkinCard, [20](#)
  - PlayerStats, [23](#)
- bottomleft
  - Button.h, [29](#)
- bottomright
  - Button.h, [29](#)
- Button, [11](#)
  - Button, [12](#)
  - callback, [13](#)
  - color, [13](#)
  - draw, [12](#)
  - id, [13](#)
  - operator=, [12](#)
  - origin, [13](#)
  - poll\_click, [13](#)
  - rect, [13](#)
  - visible, [13](#)
- Button.cpp, [28](#)
- Button.h, [28](#)
  - bottomleft, [29](#)
  - bottomright, [29](#)
  - button\_callback, [28](#)
  - ButtonOrigin, [28](#)
  - toleft, [29](#)
  - topright, [29](#)
- button\_callback
  - Button.h, [28](#)
- button\_continue
  - main.cpp, [33](#)
- button\_end\_turn
  - main.cpp, [34](#)
- button\_exit
  - main.cpp, [34](#)
- ButtonOrigin
  - Button.h, [28](#)
- buttons
  - GameState, [14](#)
- calibrationSquareDimension
  - main.cpp, [36](#)
- callback
  - Button, [13](#)
- callBackFunction
  - main.cpp, [34](#)
- cameraCalibration
  - main.cpp, [34](#)
- cameraCalibrationProcess
  - main.cpp, [34](#)
- canvas\_size
  - GameState, [15](#)
- cardName
  - MunchkinCard, [20](#)
- cards
  - main.cpp, [36](#)
- cardsConstr
  - MunchkinCard, [19](#)
- CardType
  - MunchkinCards.h, [38](#)
- cardtypeaction, [8](#)
  - curse, [8](#)
  - item, [9](#)
  - itemBuff, [9](#)
  - joker, [9](#)
  - lvlUp, [9](#)
  - monster, [9](#)
  - munchClass, [9](#)
  - race, [10](#)
- CardTypeActions.cpp, [29](#)
  - runAway, [29](#)
- CardTypeActions.h, [29](#)
  - CardTypeFunc, [30](#)

- CardTypeFunc
  - CardTypeActions.h, 30
- CardTypeRetVal, 14
- carriesLargeItems
  - PlayerStats, 23
- chessboardDimensions
  - main.cpp, 37
- chooseCardColor
  - main.cpp, 34
- clickP
  - MouseParams, 16
- clothing
  - MunchkinCards.h, 38
- Color
  - munch\_tut, 10
- color
  - Button, 13
- common.h, 30
- createArucoMarkers
  - main.cpp, 35
- createKnownBoardPositions
  - main.cpp, 35
- curse
  - cardtypeaction, 8
  - MunchkinCards.h, 38
- curses
  - PlayerStats, 23
- debuff
  - MunchkinCard, 20
- door
  - MunchkinCards.h, 39
- draw
  - Button, 12
- effect
  - MunchkinCard, 20
- emptyBadThings
  - badthings, 7
- end\_turn
  - GameState, 15
- event
  - MouseParams, 17
- EventType
  - InputEvent.h, 32
- extras, 10
- Extras.cpp, 30
- Extras.h, 31
  - ExtrasFunc, 31
- ExtrasFunc
  - Extras.h, 31
- ExtrasRetVal, 14
- functMap
  - main.cpp, 37
- GameState, 14
  - buttons, 14
  - canvas\_size, 15
  - end\_turn, 15
  - mouseparams, 15
  - player01, 15
  - remove\_card, 15
  - run\_away, 15
  - should\_continue, 15
  - should\_exit, 15
- gamestate
  - main.cpp, 37
- GameState.h, 31
  - lclick, 32
  - MouseEvent, 31
  - move, 32
  - none, 32
  - rclick, 32
- getChessboardCorners
  - main.cpp, 35
- hands
  - PlayerStats, 24
- handsNeeded
  - MunchkinCard, 20
- hasArmor
  - PlayerStats, 24
- hasHat
  - PlayerStats, 24
- hasShoes
  - PlayerStats, 24
- hat
  - MunchkinCards.h, 38
- id
  - Button, 13
- InputEvent, 16
  - type, 16
- InputEvent.h, 32
  - EventType, 32
  - KeyPress, 32
  - KeyRelease, 32
  - LmPress, 32
  - LmRelease, 32
  - RmPress, 32
  - RmRelease, 32
- item
  - cardtypeaction, 9
  - MunchkinCards.h, 38
- itemBuff
  - cardtypeaction, 9
  - MunchkinCards.h, 38
- itemEffect
  - MunchkinCard, 21
- itemEffects
  - PlayerStats, 24
- itemLarge
  - MunchkinCard, 21
- itemNeeds
  - MunchkinCard, 21
- ItemType
  - MunchkinCards.h, 38

- itemType
  - MunchkinCard, 21
- itemValue
  - MunchkinCard, 21
- joker
  - cardtypeaction, 9
  - MunchkinCards.h, 38
- KeyPress
  - InputEvent.h, 32
- KeyRelease
  - InputEvent.h, 32
- lclick
  - GameState.h, 32
- LmPress
  - InputEvent.h, 32
- LmRelease
  - InputEvent.h, 32
- loadCameraCalibration
  - main.cpp, 35
- logic
  - main.cpp, 35
- looseClass
  - badthings, 7
- looseHand
  - badthings, 7
- looseLevel
  - badthings, 8
- lvl
  - PlayerStats, 24
- lvlUp
  - cardtypeaction, 9
  - MunchkinCard, 21
  - MunchkinCards.h, 38
- main
  - main.cpp, 35
- main.cpp, 32
  - arucoSquareDimension, 36
  - button\_continue, 33
  - button\_end\_turn, 34
  - button\_exit, 34
  - calibrationSquareDimension, 36
  - callBackFunction, 34
  - cameraCalibration, 34
  - cameraCalibrationProcess, 34
  - cards, 36
  - chessboardDimensions, 37
  - chooseCardColor, 34
  - createArucoMarkers, 35
  - createKnownBoardPositions, 35
  - functMap, 37
  - gamestate, 37
  - getChessboardCorners, 35
  - loadCameraCalibration, 35
  - logic, 35
  - main, 35
  - saveCameraCalibration, 36
  - startWebcamMonitoring, 36
  - update, 36
- maleDeadFemaleLevelDown
  - badthings, 8
- markerCorner
  - MouseParams, 17
- markerID
  - MunchkinCard, 21
- markerId
  - MouseParams, 17
- monsStrength
  - MunchkinCard, 21
- monster
  - cardtypeaction, 9
  - MunchkinCards.h, 38
- MouseEvent
  - GameState.h, 31
- MouseParams, 16
  - clickP, 16
  - event, 17
  - markerCorner, 17
  - markerId, 17
  - poly, 17
  - tutText, 17
- mouseparams
  - GameState, 15
- move
  - GameState.h, 32
- munch\_tut, 10
  - Color, 10
- munchClass
  - cardtypeaction, 9
  - MunchkinCards.h, 38
- munchClasses
  - PlayerStats, 24
- MunchkinCard, 17
  - badThings, 20
  - badThingsFunc, 20
  - bonis, 20
  - cardName, 20
  - cardsConstr, 19
  - debuff, 20
  - effect, 20
  - handsNeeded, 20
  - itemEffect, 21
  - itemLarge, 21
  - itemNeeds, 21
  - itemType, 21
  - itemValue, 21
  - lvlUp, 21
  - markerID, 21
  - monsStrength, 21
  - MunchkinCard, 18, 19
  - operator=, 19
  - parentCardType, 22
  - strengthBoni, 22
  - treasures, 22

- type, 22
- MunchkinCards.cpp, 37
- MunchkinCards.h, 37
  - armor, 38
  - boni, 38
  - CardType, 38
  - clothing, 38
  - curse, 38
  - door, 39
  - hat, 38
  - item, 38
  - itemBuff, 38
  - ItemType, 38
  - joker, 38
  - lvlUp, 38
  - monster, 38
  - munchClass, 38
  - ParentCardType, 38
  - race, 38
  - removeCard, 38
  - shoes, 38
  - treasure, 39
  - weapon, 38
- munchRaces
  - PlayerStats, 24
- none
  - GameState.h, 32
- operator=
  - Button, 12
  - MunchkinCard, 19
- origin
  - Button, 13
- ParentCardType
  - MunchkinCards.h, 38
- parentCardType
  - MunchkinCard, 22
- player01
  - GameState, 15
- playerDies
  - badthings, 8
- PlayerStats, 22
  - availableClasses, 23
  - availableRaces, 23
  - bonis, 23
  - carriesLargeItems, 23
  - curses, 23
  - hands, 24
  - hasArmor, 24
  - hasHat, 24
  - hasShoes, 24
  - itemEffects, 24
  - lvl, 24
  - munchClasses, 24
  - munchRaces, 24
  - runStrength, 25
  - sex, 25
  - strength, 25
- poll\_click
  - Button, 13
- poly
  - MouseParams, 17
- race
  - cardtypeaction, 10
  - MunchkinCards.h, 38
- rclick
  - GameState.h, 32
- rect
  - Button, 13
- remove\_card
  - GameState, 15
- removeCard
  - MunchkinCards.h, 38
- RmPress
  - InputEvent.h, 32
- RmRelease
  - InputEvent.h, 32
- run\_away
  - GameState, 15
- runAway
  - CardTypeActions.cpp, 29
- runStrength
  - PlayerStats, 25
- saveCameraCalibration
  - main.cpp, 36
- sex
  - PlayerStats, 25
- shoes
  - MunchkinCards.h, 38
- should\_continue
  - GameState, 15
- should\_exit
  - GameState, 15
- startWebcamMonitoring
  - main.cpp, 36
- strength
  - PlayerStats, 25
- strengthBoni
  - MunchkinCard, 22
- toleft
  - Button.h, 29
- topright
  - Button.h, 29
- treasure
  - MunchkinCards.h, 39
- treasures
  - MunchkinCard, 22
- TTMunchkinTut.cpp, 39
- tutText
  - MouseParams, 17
- type
  - InputEvent, 16
  - MunchkinCard, 22

update

main.cpp, [36](#)

visible

Button, [13](#)

weapon

MunchkinCards.h, [38](#)