wxBlokus Reference Manual

Generated by Doxygen 1.5.1

Tue Dec 11 22:47:16 2007

Contents

Chapter 1

wxBlokus Hierarchical Index

1.1   wxBlokus Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 2

wxBlokus Class Index

2.1 wxBlokus Class List

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

Chapter 3

wxBlokus Class Documentation

3.1 BlokusApp Class Reference

Application entry point.

`#include <blokusApp.h>`

3.1.1 Detailed Description

Application entry point.

Sets up a **wxBlokus** (p. 30) frame as a top-level window when the application starts.

Definition at line 10 of file blokusApp.h.

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

- blokusApp.h
- blokusApp.cpp

3.2 BlokusTile Class Reference

Blokus tile.

#include <BlokusTile.h>

Public Member Functions

- u8 **getAmount** ()

  Get amount.

- void **setAmount** (u8 amount)

  Set amount.

- u8 **getScore** ()

  Get score.

- void **setScore** (u8 score)

  Set score.

- Tile ∗ **getTileData** ()

  Get ***Tile*** *(p. 25) data.*

- void **setTile** (**Tile** ∗tile)

  Set ***Tile*** *(p. 25) data.*

Protected Attributes

- **Tile** ∗ **TileData**
- u8 **Amount**
- u8 **Score**

3.2.1 Detailed Description

Blokus tile.

Wraps tile data and adds amount and score to it.

Definition at line 8 of file BlokusTile.h.

3.2.2 Member Function Documentation

3.2.2.1 u8 BlokusTile::getAmount () [inline]

Get amount.

Returns amount of tiles.

Definition at line 13 of file BlokusTile.h.

References Amount.

Referenced by Player::getScore(), TilePanel::OnPaint(), DefaultRules::PlacedTile(), Game::Save-Game(), Player::setNextValidTile(), and DefaultRules::ValidPlaceTile().

3.2.2.2 void BlokusTile::setAmount (u8 *amount*) [inline]

Set amount.

Sets amount of tiles.

Definition at line 17 of file BlokusTile.h.

References Amount.

Referenced by Game::LoadGame(), and DefaultRules::PlacedTile().

3.2.2.3 u8 BlokusTile::getScore () [inline]

Get score.

Gets score per tiles.

Definition at line 21 of file BlokusTile.h.

References Score.

Referenced by Player::getScore().

3.2.2.4 void BlokusTile::setScore (u8 *score*) [inline]

Set score.

Sets score per tiles.

Definition at line 25 of file BlokusTile.h.

References Score.

3.2.2.5 Tile∗ BlokusTile::getTileData () [inline]

Get **Tile** (p. 25) data.

Returns pointer to actual tile data.

Definition at line 29 of file BlokusTile.h.

References TileData.

Referenced by wxBlokus::KeyDown(), TilePanel::OnPaint(), FieldPanel::OnPaint(), Game::Place-Tile(), and DefaultRules::ValidPlaceTile().

3.2.2.6 void BlokusTile::setTile (Tile ∗ *tile*) [inline]

Set **Tile** (p. 25) data.

Associates **BlokusTile** (p. 6) with actual data.

Definition at line 33 of file BlokusTile.h.

References TileData.

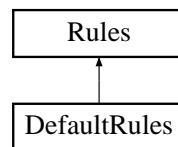The documentation for this class was generated from the following file:

- BlokusTile.h

3.3 DefaultRules Class Reference

Default rules.

`#include <DefaultRules.h>`

Inheritance diagram for DefaultRules::

Public Member Functions

- **DefaultRules** ()

  DefaultRules (p. 9) constructor.

- virtual ~**DefaultRules** ()

  DefaultRules (p. 9) destructor.

- **Player** ∗ **CreatePlayers** ()

  Create players.

- bool **ValidPlaceTile** (**Field** ∗field, **Player** ∗player, **BlokusTile** ∗tile, u8 x, u8 y)

  Checks if a tile can be placed at a certain location.

- void **PlacedTile** (**Field** ∗field, **Player** ∗player, **BlokusTile** ∗tile, u8 x, u8 y)

  Called after a tile has been placed.

3.3.1 Detailed Description

Default rules.

The default Blokus rules

Definition at line 8 of file DefaultRules.h.

3.3.2 Constructor & Destructor Documentation

3.3.2.1 DefaultRules::DefaultRules ()

DefaultRules (p. 9) constructor.

Sets field width, height, **Player** (p. 18) count and initialize standard tiles.

Definition at line 5 of file DefaultRules.cpp.

References Rules::FieldHeight, Rules::FieldWidth, and Rules::PlayerCount.

3.3.2.2 DefaultRules::∼DefaultRules () [virtual]

DefaultRules (p. 9) destructor.

Destroy the standard tiles.

Definition at line 15 of file DefaultRules.cpp.

3.3.3 Member Function Documentation

3.3.3.1 Player * DefaultRules::CreatePlayers () [virtual]

Create players.

Allocates a list of players and gives them a **TileSet** (p. 28).

Implements **Rules** (p. 24).

Definition at line 204 of file DefaultRules.cpp.

References Rules::PlayerCount.

3.3.3.2 bool DefaultRules::ValidPlaceTile (Field * *field*, Player * *player*, BlokusTile * *tile*, u8 *x*, u8 *y*) [virtual]

Checks if a tile can be placed at a certain location.

Checks if a tile can be placed at a certain location, according to the official Blokus rules.

Implements **Rules** (p. 24).

Definition at line 20 of file DefaultRules.cpp.

References Tile::Blocks, BlokusTile::getAmount(), Player::getTeam(), BlokusTile::getTileData(), Tile::Height, Player::PlacedFirstTile(), and Tile::Width.

3.3.3.3 void DefaultRules::PlacedTile (Field * *field*, Player * *player*, BlokusTile * *tile*, u8 *x*, u8 *y*) [virtual]

Called after a tile has been placed.

Removes a tile from the count and sets that player has used a tile.

Implements **Rules** (p. 24).

Definition at line 198 of file DefaultRules.cpp.

References BlokusTile::getAmount(), BlokusTile::setAmount(), and Player::UsedTile().

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

- DefaultRules.h
- DefaultRules.cpp

3.4 FieldPanel Class Reference

Field panel.

#include <FieldPanel.h>

Public Member Functions

- **FieldPanel (Game** ∗game, wxWindow ∗parent, wxWindowID id=wxID_ANY, const wx-Point &pos=wxDefaultPosition, const wxSize &size=wxDefaultSize)

  FieldPanel (p. 11) constructor.

- void **OnPaint** (wxPaintEvent &event)

  Paint event.

- u8 **getTileX** ()

  Gets X coordinate for the tile.

- u8 **getTileY** ()

  Gets Y coordinate for the tile.

- void **setTileX** (u8 value)

  Sets X coordinate for the tile.

- void **setTileY** (u8 value)

  Sets Y coordinate for the tile.

- void **setGame** (**Game** ∗game)

  Sets pointer to game.

3.4.1 Detailed Description

Field panel.

Displays a field.

Definition at line 9 of file FieldPanel.h.

3.4.2 Constructor & Destructor Documentation

3.4.2.1 FieldPanel::FieldPanel (Game ∗ *game*, wxWindow ∗ *parent*, wxWindowID *id* = wxID_ANY, const wxPoint & *pos* = wxDefaultPosition, const wxSize & *size* = wxDefaultSize)

FieldPanel (p. 11) constructor.

Shows itself, initialises and sets values.

Definition at line 10 of file FieldPanel.cpp.

3.4.3   Member Function Documentation

3.4.3.1   void FieldPanel::OnPaint (wxPaintEvent & *event*)

Paint event.

Paints out the field.

Definition at line 22 of file FieldPanel.cpp.

References Tile::Blocks, Game::getCurrentPlayer(), Player::getCurrentTile(), Game::get-
Field(), Player::getTeam(), BlokusTile::getTileData(), Tile::Height, Game::isValidPlace(), and
Tile::Width.

3.4.3.2   u8 FieldPanel::getTileX () `[inline]`

Gets X coordinate for the tile.

Gets the X coordinate for the displayed tile.

Definition at line 23 of file FieldPanel.h.

Referenced by wxBlokus::KeyDown().

3.4.3.3   u8 FieldPanel::getTileY () `[inline]`

Gets Y coordinate for the tile.

Gets the Y coordinate for the displayed tile.

Definition at line 27 of file FieldPanel.h.

Referenced by wxBlokus::KeyDown().

3.4.3.4   void FieldPanel::setTileX (u8 *value*) `[inline]`

Sets X coordinate for the tile.

Sets the X coordinate for the displayed tile.

Definition at line 31 of file FieldPanel.h.

Referenced by wxBlokus::KeyDown(), and wxBlokus::OnNew().

3.4.3.5   void FieldPanel::setTileY (u8 *value*) `[inline]`

Sets Y coordinate for the tile.

Sets the Y coordinate for the displayed tile.

Definition at line 35 of file FieldPanel.h.

Referenced by wxBlokus::KeyDown(), and wxBlokus::OnNew().

3.4.3.6   void FieldPanel::setGame (Game ∗ *game*) `[inline]`

Sets pointer to game.

Sets the **Game** (p. 14) pointer from which the class reads.

Definition at line 39 of file FieldPanel.h.

Referenced by wxBlokus::OnNew().

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

- FieldPanel.h
- FieldPanel.cpp

3.5 Game Class Reference

Game (p. 14) class.

`#include <Game.h>`

Public Member Functions

- **Game** (**Rules** ∗rules, void(∗endGameCallback)())

    Game (p. 14) *constructor.*

- ∼**Game** ()

    Game (p. 14) *destructor.*

- bool **PlaceTile** (**BlokusTile** ∗tile, u8 x, u8 y)

    Place tile.

- bool **isValidPlace** (**BlokusTile** ∗tile, u8 x, u8 y)

    Checks if tile placement is valid.

- **Field** ∗ **getField** ()

    Get Field.

- **Rules** ∗ **getRules** ()

    Get Rield.

- u8 **getPlayerCount** ()

    Get **Player** (p. 18) *count.*

- **Player** ∗ **getPlayerList** ()

    Get **Player** (p. 18) *list.*

- **Player** ∗ **getCurrentPlayer** ()

    Get current **Player** (p. 18).

- **Player** ∗ **getPlayer** (u8 index)

    Get **Player** (p. 18).

- void **CyclePlayers** ()

    Cycle players.

- void **EndGame** ()

    End game.

- bool **SaveGame** (const char ∗path)

    Save game.

- bool **LoadGame** (const char ∗path)

    Load game.

3.5.1 Detailed Description

Game (p. 14) class.

Most game-related stuff happens here.

Definition at line 10 of file Game.h.

3.5.2 Constructor & Destructor Documentation

3.5.2.1 Game::Game (Rules * *rules*, void(*)() *endGameCallback*)

Game (p. 14) constructor.

Has all game components allocated by **Rules** (p. 23).

Definition at line 5 of file Game.cpp.

References Rules::CreateField(), Rules::CreatePlayers(), and Rules::getPlayerCount().

3.5.2.2 Game::∼Game ()

Game (p. 14) destructor.

Destroys field and players.

Definition at line 20 of file Game.cpp.

3.5.3 Member Function Documentation

3.5.3.1 bool Game::PlaceTile (BlokusTile * *tile*, u8 *x*, u8 *y*)

Place tile.

Places a tile if possible.

Definition at line 29 of file Game.cpp.

References Tile::Blocks, CyclePlayers(), EndGame(), getCurrentPlayer(), Player::getTeam(), BlokusTile::getTileData(), Tile::Height, Rules::PlacedTile(), Rules::ValidPlaceTile(), and Tile::Width.

Referenced by wxBlokus::KeyDown().

3.5.3.2 bool Game::isValidPlace (BlokusTile * *tile*, u8 *x*, u8 *y*) [inline]

Checks if tile placement is valid.

Calls GameRules::ValidPlaceTile, returns true if tile placement is valid.

Definition at line 28 of file Game.h.

References getCurrentPlayer(), and Rules::ValidPlaceTile().

Referenced by FieldPanel::OnPaint().

3.5.3.3   Field∗ Game::getField () [inline]

Get Field.

Returns field.

Definition at line 32 of file Game.h.

Referenced by wxBlokus::KeyDown(), and FieldPanel::OnPaint().

3.5.3.4   Rules∗ Game::getRules () [inline]

Get Rield.

Returns current **Rules** (p. 23).

Definition at line 36 of file Game.h.

3.5.3.5   u8 Game::getPlayerCount () [inline]

Get **Player** (p. 18) count.

Returns player count.

Definition at line 40 of file Game.h.

Referenced by wxBlokus::ShowScore().

3.5.3.6   Player∗ Game::getPlayerList () [inline]

Get **Player** (p. 18) list.

Returns pointer to player list.

Definition at line 44 of file Game.h.

3.5.3.7   Player∗ Game::getCurrentPlayer () [inline]

Get current **Player** (p. 18).

Returns pointer to current player.

Definition at line 48 of file Game.h.

Referenced by isValidPlace(), wxBlokus::KeyDown(), FieldPanel::OnPaint(), PlaceTile(), wx-Blokus::RefreshTilePanels(), and wxBlokus::wxBlokus().

3.5.3.8   Player∗ Game::getPlayer (u8 *index*) [inline]

Get **Player** (p. 18).

Returns pointer to requested player.

Definition at line 52 of file Game.h.

Referenced by LoadGame(), SaveGame(), and wxBlokus::ShowScore().

3.5.3.9   void Game::CyclePlayers () [inline]

Cycle players.

Sets next player as current.

Definition at line 56 of file Game.h.

Referenced by wxBlokus::OnSkip(), and PlaceTile().

3.5.3.10   void Game::EndGame () [inline]

End game.

Calls endgame callback.

Definition at line 66 of file Game.h.

Referenced by PlaceTile().

3.5.3.11   bool Game::SaveGame (const char ∗ *path*)

Save game.

Saves current state to path.

Definition at line 74 of file Game.cpp.

References Tile::Blocks, BlokusTile::getAmount(), TileSet::getNumTiles(), getPlayer(), Tile-Set::getTile(), Tile::Height, Player::PlacedFirstTile(), and Tile::Width.

Referenced by wxBlokus::OnSave().

3.5.3.12   bool Game::LoadGame (const char ∗ *path*)

Load game.

Loads previous state from path.

Definition at line 108 of file Game.cpp.

References Tile::Blocks, TileSet::getNumTiles(), getPlayer(), TileSet::getTile(), Tile::Height, BlokusTile::setAmount(), and Tile::Width.

Referenced by wxBlokus::OnLoad().

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

- Game.h
- Game.cpp

3.6 Player Class Reference

Player (p. 18).

`#include <Player.h>`

Public Member Functions

- **Player** ()

  Player (p. 18) constructor.

- u8 **getTeam** ()

  Get team.

- void **setTeam** (u8 team)

  Set team.

- void **setNextTile** ()

  Set next tile.

- void **setPreviousTile** ()

  Set previous tile.

- **BlokusTile** ∗ **getPreviousTile** ()

  Get previous tile.

- **BlokusTile** ∗ **getNextTile** ()

  Get next tile.

- **BlokusTile** ∗ **getCurrentTile** ()

  Get current tile.

- bool **setNextValidTile** ()

  Set next valid tile.

- **TileSet** ∗ **getTileSet** ()

  Get TileSet (p. 28).

- bool **PlacedFirstTile** ()

  Gets player's placed first tile status.

- void **UsedTile** ()

  Used tile.

- u16 **getScore** ()

  Get score.

Protected Member Functions

- u8 **NextIndex** ()

    Next index.

- u8 **PreviousIndex** ()

    Previous index.

Protected Attributes

- u8 **Team**
- u8 **TileIndex**
- **TileSet Tiles**
- bool **HasPlacedTile**

3.6.1 Detailed Description

Player (p. 18).

Contains information on a certain player.

Definition at line 11 of file Player.h.

3.6.2 Constructor & Destructor Documentation

3.6.2.1 Player::Player () [inline]

Player (p. 18) constructor.

Sets default properties.

Definition at line 16 of file Player.h.

References HasPlacedTile, setTeam(), and TileIndex.

3.6.3 Member Function Documentation

3.6.3.1 u8 Player::getTeam () [inline]

Get team.

Returns player's team.

Definition at line 21 of file Player.h.

References Team.

Referenced by FieldPanel::OnPaint(), Game::PlaceTile(), wxBlokus::RefreshTilePanels(), Default-Rules::ValidPlaceTile(), and wxBlokus::wxBlokus().

3.6.3.2 void Player::setTeam (u8 *team*) [inline]

Set team.

Sets player's team.

Definition at line 25 of file Player.h.

References Team.

Referenced by Player().

3.6.3.3 void Player::setNextTile () `[inline]`

Set next tile.

Sets next tile as current.

Definition at line 29 of file Player.h.

References NextIndex(), and TileIndex.

Referenced by wxBlokus::KeyDown().

3.6.3.4 void Player::setPreviousTile () `[inline]`

Set previous tile.

Sets previous tile as current.

Definition at line 33 of file Player.h.

References PreviousIndex(), and TileIndex.

Referenced by wxBlokus::KeyDown().

3.6.3.5 BlokusTile∗ Player::getPreviousTile () `[inline]`

Get previous tile.

Gets previous tile.

Definition at line 37 of file Player.h.

References TileSet::getTile(), PreviousIndex(), and Tiles.

Referenced by wxBlokus::RefreshTilePanels(), and wxBlokus::wxBlokus().

3.6.3.6 BlokusTile∗ Player::getNextTile () `[inline]`

Get next tile.

Gets next tile.

Definition at line 41 of file Player.h.

References TileSet::getTile(), NextIndex(), and Tiles.

Referenced by wxBlokus::RefreshTilePanels(), and wxBlokus::wxBlokus().

3.6.3.7 BlokusTile∗ Player::getCurrentTile () `[inline]`

Get current tile.

Gets current tile.

Definition at line 45 of file Player.h.

References TileSet::getTile(), TileIndex, and Tiles.

Referenced by wxBlokus::KeyDown(), FieldPanel::OnPaint(), wxBlokus::RefreshTilePanels(), set-NextValidTile(), and wxBlokus::wxBlokus().

3.6.3.8 bool Player::setNextValidTile () [inline]

Set next valid tile.

Sets next valid tile as current, if there are any.

Definition at line 49 of file Player.h.

References BlokusTile::getAmount(), getCurrentTile(), TileSet::getNumTiles(), TileSet::getTile(), NextIndex(), TileIndex, and Tiles.

3.6.3.9 TileSet∗ Player::getTileSet () [inline]

Get **TileSet** (p. 28).

Gets player's **TileSet** (p. 28).

Definition at line 71 of file Player.h.

References Tiles.

3.6.3.10 bool Player::PlacedFirstTile () [inline]

Gets player's placed first tile status.

Returns true if the player has placed a tile, else false.

Definition at line 75 of file Player.h.

References HasPlacedTile.

Referenced by Game::SaveGame(), and DefaultRules::ValidPlaceTile().

3.6.3.11 void Player::UsedTile () [inline]

Used tile.

Sets player's tile as used.

Definition at line 79 of file Player.h.

References HasPlacedTile.

Referenced by DefaultRules::PlacedTile().

3.6.3.12 u16 Player::getScore () [inline]

Get score.

Calculates and returns player's score.

Definition at line 83 of file Player.h.

References BlokusTile::getAmount(), TileSet::getNumTiles(), BlokusTile::getScore(), TileSet::get-Tile(), and Tiles.

Referenced by wxBlokus::ShowScore().

3.6.3.13 u8 Player::NextIndex () `[inline, protected]`

Next index.

Returns next index.

Definition at line 98 of file Player.h.

References TileSet::getNumTiles(), TileIndex, and Tiles.

Referenced by getNextTile(), setNextTile(), and setNextValidTile().

3.6.3.14 u8 Player::PreviousIndex () `[inline, protected]`

Previous index.

Returns previous index.

Definition at line 107 of file Player.h.

References TileSet::getNumTiles(), TileIndex, and Tiles.

Referenced by getPreviousTile(), and setPreviousTile().

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

- Player.h

3.7 Rules Class Reference

Rules (p. 23).

`#include <Rules.h>`

Inheritance diagram for Rules::
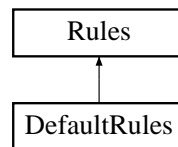
Public Member Functions

- virtual **Field** ∗ **CreateField** ()

    Create field.

- virtual **Player** ∗ **CreatePlayers** ()=0

    Create players.

- u8 **getPlayerCount** ()

    Get player count.

- virtual bool **ValidPlaceTile** (**Field** ∗field, **Player** ∗player, **BlokusTile** ∗tile, u8 x, u8 y)=0

    Tells if tile can be placed at location.

- virtual void **PlacedTile** (**Field** ∗field, **Player** ∗player, **BlokusTile** ∗tile, u8 x, u8 y)=0

    *Placed **Tile** (p. 25).*

Protected Attributes

- u8 **FieldWidth**
- u8 **FieldHeight**
- u8 **PlayerCount**

3.7.1 Detailed Description

Rules (p. 23).

Virtual base class. Handles creation of field, players etc..

Definition at line 10 of file Rules.h.

3.7.2 Member Function Documentation

3.7.2.1 virtual Field* Rules::CreateField () [inline, virtual]

Create field.

Allocates and returns a field.

Definition at line 15 of file Rules.h.

References Tile::Blocks, FieldHeight, FieldWidth, Tile::Height, and Tile::Width.

Referenced by Game::Game().

3.7.2.2 virtual Player* Rules::CreatePlayers () [pure virtual]

Create players.

Returns array of players.

Implemented in **DefaultRules** (p. 10).

Referenced by Game::Game().

3.7.2.3 u8 Rules::getPlayerCount () [inline]

Get player count.

Returns player count.

Definition at line 30 of file Rules.h.

References PlayerCount.

Referenced by Game::Game().

3.7.2.4 virtual bool Rules::ValidPlaceTile (Field * *field*, Player * *player*, BlokusTile * *tile*, u8 *x*, u8 *y*) [pure virtual]

Tells if tile can be placed at location.

Returns true if the tile can be placed by player at the location.

Implemented in **DefaultRules** (p. 10).

Referenced by Game::isValidPlace(), and Game::PlaceTile().

3.7.2.5 virtual void Rules::PlacedTile (Field * *field*, Player * *player*, BlokusTile * *tile*, u8 *x*, u8 *y*) [pure virtual]

Placed **Tile** (p. 25).

Gets called by game after tile has been placed.

Implemented in **DefaultRules** (p. 10).

Referenced by Game::PlaceTile().

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

- Rules.h

3.8 Tile Struct Reference

Tile (p. 25) struct.

`#include <Tile.h>`

Public Attributes

- u8 **Width**
- u8 **Height**
- Block ∗ **Blocks**

3.8.1 Detailed Description

Tile (p. 25) struct.

Contains width, height and Block array.

Definition at line 9 of file Tile.h.

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

- Tile.h

3.9   TilePanel Class Reference

Tile (p. 25) panel.

`#include <TilePanel.h>`

Public Member Functions

- **TilePanel** (**BlokusTile** *tile, u8 team, wxWindow *parent, wxWindowID id=wxID_ANY, const wxPoint &pos=wxDefaultPosition, const wxSize &size=wxDefaultSize)

    TilePanel (p. 26) constructor.

- void **OnPaint** (wxPaintEvent &event)

    Paint event.

- void **setTeam** (u8 team)

    Set team.

- void **setTile** (**BlokusTile** *tile)

    Set Tile (p. 25).

3.9.1   Detailed Description

Tile (p. 25) panel.

Displays a tile with color related to set team.

Definition at line 9 of file TilePanel.h.

3.9.2   Constructor & Destructor Documentation

3.9.2.1   TilePanel::TilePanel (BlokusTile * *tile*, u8 *team*, wxWindow * *parent*, wxWindowID *id* = `wxID_ANY`, const wxPoint & *pos* = `wxDefaultPosition`, const wxSize & *size* = `wxDefaultSize`)

TilePanel (p. 26) constructor.

Sets tile and team.

Definition at line 11 of file TilePanel.cpp.

3.9.3   Member Function Documentation

3.9.3.1   void TilePanel::OnPaint (wxPaintEvent & *event*)

Paint event.

Draws the tile.

Definition at line 21 of file TilePanel.cpp.

References Tile::Blocks, BlokusTile::getAmount(), BlokusTile::getTileData(), Tile::Height, and Tile::Width.

3.9.3.2   void TilePanel::setTeam (u8 *team*) [inline]

Set team.

Sets display team for the tile.

Definition at line 23 of file TilePanel.h.

Referenced by wxBlokus::RefreshTilePanels().

3.9.3.3   void TilePanel::setTile (BlokusTile ∗ *tile*) [inline]

Set **Tile** (p. 25).

Sets tile to be displayed.

Definition at line 27 of file TilePanel.h.

Referenced by wxBlokus::RefreshTilePanels().

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

- TilePanel.h
- TilePanel.cpp

3.10   TileSet Class Reference

Tile (p. 25) set.

`#include <TileSet.h>`

Public Member Functions

- **~TileSet** ()

  TileSet *(p. 28) destructor.*

- u8 **getNumTiles** ()

  Get number of tiles.

- **BlokusTile** ∗ **getTile** (u8 index)

  Get a certain tile.

- void **setTile** (u8 index, **BlokusTile** ∗tile)

  Set a certain tile.

- void **setTileAmount** (u8 tileAmount)

  Set tile amount.

- void **setTileList** (**BlokusTile** ∗tileList)

  Set tile list.

3.10.1   Detailed Description

Tile (p. 25) set.

Tile (p. 25) set for the player, contains amount of tiles and which tiles they are.

Definition at line 8 of file TileSet.h.

3.10.2   Constructor & Destructor Documentation

3.10.2.1   TileSet::~TileSet () `[inline]`

TileSet (p. 28) destructor.

Deletes tiles.

Definition at line 13 of file TileSet.h.

3.10.3   Member Function Documentation

3.10.3.1   u8 TileSet::getNumTiles () `[inline]`

Get number of tiles.

Gets the amount of tiles in the set.

Definition at line 17 of file TileSet.h.

Referenced by Player::getScore(), Game::LoadGame(), Player::NextIndex(), Player::Previous-Index(), Game::SaveGame(), and Player::setNextValidTile().

3.10.3.2 BlokusTile∗ TileSet::getTile (u8 *index*) [inline]

Get a certain tile.

Gets the requested tile.

Definition at line 21 of file TileSet.h.

Referenced by Player::getCurrentTile(), Player::getNextTile(), Player::getPreviousTile(), Player::getScore(), Game::LoadGame(), Game::SaveGame(), and Player::setNextValidTile().

3.10.3.3 void TileSet::setTile (u8 *index*, BlokusTile ∗ *tile*) [inline]

Set a certain tile.

Overwrites a tile in the set to a specified one.

Definition at line 25 of file TileSet.h.

3.10.3.4 void TileSet::setTileAmount (u8 *tileAmount*) [inline]

Set tile amount.

Sets amount of tiles, used during creation.

Definition at line 31 of file TileSet.h.

3.10.3.5 void TileSet::setTileList (BlokusTile ∗ *tileList*) [inline]

Set tile list.

Sets the pointer to a **BlokusTile** (p. 6) array.

Definition at line 35 of file TileSet.h.

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

- TileSet.h

3.11   wxBlokus Class Reference

Main frame for Blokus.

`#include <wxBlokus.h>`

Public Member Functions

- **wxBlokus** (const wxString &title, const wxPoint &pos, const wxSize &size)

    wxBlokus (p. 30) constructor

- **∼wxBlokus** ()

    wxBlokus (p. 30) destructor

- void **OnQuit** (wxCommandEvent &event)

    Quit event.

- void **OnLoad** (wxCommandEvent &event)

    Load event.

- void **OnSave** (wxCommandEvent &event)

    Save event.

- void **OnAbout** (wxCommandEvent &event)

    About event.

- void **OnSkip** (wxCommandEvent &event)

    Skip event.

- void **OnCountScore** (wxCommandEvent &event)

    Count score event.

- void **OnNew** (wxCommandEvent &event)

    New event.

- void **KeyDown** (wxKeyEvent &event)

    KeyDown event.

- void **OnGameEnd** ()

    Game (p. 14) End callback.

- void **RefreshTilePanels** ()

    Refresh tile panels.

- void **ShowScore** ()

    ShowScore.

Static Public Member Functions

- static void **GameEnd** ()

    *Static **Game** (p. 14) End callback.*

3.11.1 Detailed Description

Main frame for Blokus.

Main window. Contains **Rules** (p. 23) and **Game** (p. 14) for gameplay mechanics. A **FieldPanel** (p. 11), and a few instances of **TilePanel** (p. 26) show the previous, current and next **Tile** (p. 25). This frame also handles key events from child windows and have a couple of commands from menus that allows players save, load, skip one's turn etc.

Definition at line 20 of file wxBlokus.h.

3.11.2 Constructor & Destructor Documentation

3.11.2.1 wxBlokus::wxBlokus (const wxString & *title*, const wxPoint & *pos*, const wxSize & *size*)

wxBlokus (p. 30) constructor

Allocates **Game** (p. 14), **Rules** (p. 23), and panels etc.

Definition at line 7 of file wxBlokus.cpp.

References GameEnd(), Game::getCurrentPlayer(), Player::getCurrentTile(), Player::getNext-Tile(), Player::getPreviousTile(), and Player::getTeam().

3.11.2.2 wxBlokus::∼wxBlokus ()

wxBlokus (p. 30) destructor

Deletes game and rules.

Definition at line 89 of file wxBlokus.cpp.

3.11.3 Member Function Documentation

3.11.3.1 void wxBlokus::OnQuit (wxCommandEvent & *event*)

Quit event.

Closes the frame.

Definition at line 166 of file wxBlokus.cpp.

3.11.3.2 void wxBlokus::OnLoad (wxCommandEvent & *event*)

Load event.

Loads a previously saved game from file selector.

Definition at line 181 of file wxBlokus.cpp.

References Game::LoadGame(), and RefreshTilePanels().

3.11.3.3   void wxBlokus::OnSave (wxCommandEvent & *event*)

Save event.

Saves the current game.

Definition at line 198 of file wxBlokus.cpp.

References Game::SaveGame().

3.11.3.4   void wxBlokus::OnAbout (wxCommandEvent & *event*)

About event.

Displays Blokus information.

Definition at line 171 of file wxBlokus.cpp.

3.11.3.5   void wxBlokus::OnSkip (wxCommandEvent & *event*)

Skip event.

Closes the frame.

Definition at line 110 of file wxBlokus.cpp.

References Game::CyclePlayers(), and RefreshTilePanels().

3.11.3.6   void wxBlokus::OnCountScore (wxCommandEvent & *event*)

Count score event.

Displays the score in a message box.

Definition at line 156 of file wxBlokus.cpp.

References ShowScore().

3.11.3.7   void wxBlokus::OnNew (wxCommandEvent & *event*)

New event.

Resets the game.

Definition at line 119 of file wxBlokus.cpp.

References GameEnd(), RefreshTilePanels(), FieldPanel::setGame(), FieldPanel::setTileX(), and FieldPanel::setTileY().

3.11.3.8   void wxBlokus::KeyDown (wxKeyEvent & *event*)

KeyDown event.

Moves, changes and manipulates tiles based on key presses.

Definition at line 210 of file wxBlokus.cpp.

References Game::getCurrentPlayer(), Player::getCurrentTile(), Game::getField(), Blokus-Tile::getTileData(), FieldPanel::getTileX(), FieldPanel::getTileY(), Tile::Height, Game::Place-Tile(), RefreshTilePanels(), Player::setNextTile(), Player::setPreviousTile(), FieldPanel::setTile-X(), FieldPanel::setTileY(), and Tile::Width.

3.11.3.9 void wxBlokus::OnGameEnd ()

Game (p. 14) End callback.

Displays the score in a message box.

Definition at line 161 of file wxBlokus.cpp.

References ShowScore().

Referenced by GameEnd().

3.11.3.10 void wxBlokus::RefreshTilePanels ()

Refresh tile panels.

Updates tile panels and has them redrawn.

Definition at line 95 of file wxBlokus.cpp.

References Game::getCurrentPlayer(), Player::getCurrentTile(), Player::getNextTile(), Player::getPreviousTile(), Player::getTeam(), TilePanel::setTeam(), and TilePanel::setTile().

Referenced by KeyDown(), OnLoad(), OnNew(), and OnSkip().

3.11.3.11 static void wxBlokus::GameEnd () `[inline, static]`

Static **Game** (p. 14) End callback.

Calls the current wxBlokus' GameEnd function.

Definition at line 73 of file wxBlokus.h.

References OnGameEnd().

Referenced by OnNew(), and wxBlokus().

3.11.3.12 void wxBlokus::ShowScore ()

ShowScore.

Displays the score in a message box.

Definition at line 136 of file wxBlokus.cpp.

References Game::getPlayer(), Game::getPlayerCount(), and Player::getScore().

Referenced by OnCountScore(), and OnGameEnd().

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

- wxBlokus.h
- wxBlokus.cpp

Index