Widgets

1.0

Készítette Doxygen 1.8.13

Tartalomjegyzék

1. fejezet

Hierarchikus mutató

1.1. Osztályhierarchia

Majdnem (de nem teljesen) betűrendbe szedett leszármazási lista:

2. fejezet

Osztálymutató

2.1. Osztálylista

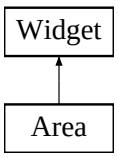
Az összes osztály, struktúra, unió és interfész listája rövid leírásokkal:

3. fejezet

Osztályok dokumentációja

3.1. Area osztályreferencia

Az Area osztály származási diagramja:



Publikus tagfüggvények

- Area (int x, int y)

  Lehoz egy j terletet a X vagy O kijelzre.
- void SetEventVoid (std::function< void(Area *)> event)

  Bea az esemt, ami akkor kvetkezik be, ha rttintanak.
- void SetValue (int value)

  Bea az t
- void SetMarkerColour (int r, int g, int b)

  A kijelzzt adja meg.
- int GetValue ()

  Megadja az aktus t
- int GetXPostion ()

  Megadja az x koordinjDoxyCompactList
  int GetYPosition ()
  Megadja az y koordinjDoxyCompactList
  virtual void Draw ()
  Kirajzolja a widgetet.
  virtual void Handle (genv::event ev)
  Kezeli a widgetet az inputnak megfelelen.
  virtual bool IsInLine (int x, int y)
  Megadja, hogy egy koordin rajta van-e a widgeten.

Privát attribútumok

- *int XVal*
- *int YVal*
- *int Value*
- *Colour * MarkColour*
- *std::function< void(Area *)> OnEvent*

Additional Inherited Members

3.1.1. Konstruktorok és destruktorok dokumentációja

3.1.1.1. Area()

```
Area::Area (
            int x,
            int y )
```

Lehoz egy j terletet a X vagy O kijelzre.

Paraméterek

x	*int Az x koordin*
y	*int Az y koordin*

3.1.2. Tagfüggvények dokumentációja

3.1.2.1. Draw()

```
void Area::Draw ( )  [virtual]
```

Kirajzolja a widgetet.

Visszatérési érték

virtual void

Megvalósítja a következőket: Widget.

3.1.2.2. GetValue()

```
int Area::GetValue ( )
```

Megadja az aktus t

Visszatérési érték

int Az aktus DoxyReturn

3.1.2.3. GetXPostion()

```
int Area::GetXPostion ( )
```

Megadja az x koordinj

Visszatérési érték

int Az x koordin

3.1.2.4. GetYPosition()

```
int Area::GetYPosition ( )
```

Megadja az y koordinj

Visszatérési érték

int Az y koordin

3.1.2.5. Handle()

```
void Area::Handle (
            genv::event ev )  [virtual]
```

Kezeli a widgetet az inputnak megfelelen.

Paraméterek

ev	*genv::event Az input eventje*

Visszatérési érték

virtual void

Megvalósítja a következőket: Widget.

3.1.2.6. IsInLine()

```
bool Area::IsInLine (
            int x,
            int y )  [virtual]
```

Megadja, hogy egy koordin rajta van-e a widgeten.

Paraméterek

x	*int Az x koordin*
y	*int Az y koordin*

Visszatérési érték

virtual bool

Megvalósítja a következőket: Widget.

3.1.2.7. SetEventVoid()

```
void Area::SetEventVoid (
            std::function< void(Area *)> event )
```

Bea az esemt, ami akkor kvetkezik be, ha rttintanak.

Paraméterek

Area∗	*Sajagdja vissza*

Visszatérési érték

void

3.1.2.8. SetMarkerColour()

```
void Area::SetMarkerColour (
            int r,
            int g,
            int b )
```

A kijelzzt adja meg.

Paraméterek

r	*int A piros*
g	*int A zld*
b	*int A krt*

Visszatérési érték

void

3.1.2.9. SetValue()

```
void Area::SetValue (
            int value )
```

Bea az t

Paraméterek

value	*int A bend*

Visszatérési érték

void

3.1.3. Adattagok dokumentációja

3.1.3.1. MarkColour

```
Colour* Area::MarkColour  [private]
```

A kijelzz

3.1.3.2. OnEvent

```
std::function<void(Area*)> Area::OnEvent  [private]
```

Az event hatra meghdm

3.1.3.3. Value

```
int Area::Value  [private]
```

Az aktus

3.1.3.4. XVal

```
int Area::XVal  [private]
```

Az x koordin

3.1.3.5. YVal

```
int Area::YVal  [private]
```

Az y koordin

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

- *Area.h*
- *Area.cpp*

3.2. asd osztályreferencia

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

- *asd.h*
- *asd.cpp*

3.3. genv::box struktúrareferencia

Publikus tagfüggvények

- ***box*** *(int x, int y)*
- *void* ***operator()*** *(canvas &out)*

Publikus attribútumok

- *int* ***vec_x***
- *int* ***vec_y***

Ez a dokumentáció a struktúráról a következő fájl alapján készült:

- *graphics.hpp*

3.4. genv::box_to struktúrareferencia

Publikus tagfüggvények

- ***box_to*** *(unsigned x, unsigned y)*
- *void* ***operator()*** *(canvas &out)*

Publikus attribútumok

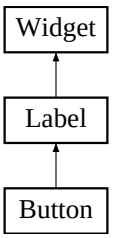
- *int* ***pos_x***
- *int* ***pos_y***

Ez a dokumentáció a struktúráról a következő fájl alapján készült:

- *graphics.hpp*

3.5. Button osztályreferencia

A Button osztály származási diagramja:



Publikus tagfüggvények

- *Button (int x, int y, int xSize, int ySize, std::string text, std::function< void()> fv)*
  Létrehoz egy új gombot.
- *virtual void Draw ()*
  Kirajzolja a widgetet.
- *virtual void Handle (genv::event ev)*
  Kezeli a widgetet az inputnak megfelelõen.
- *virtual bool IsInLine (int x, int y)*
  Megadja, hogy egy koordináta rajta van-e a widgeten.
- *void SetOnClickColor (int r, int g, int b)*
  A kattintáskor látható színt változtatja meg.

Privát attribútumok

- *std::function< void()> Function*
- *Colour * OnClickColor*
- *bool isClicking*

Additional Inherited Members

3.5.1. Konstruktorok és destruktorok dokumentációja

3.5.1.1. Button()

```
Button::Button (
            int x,
            int y,
```

```
            int xSize,
            int ySize,
            std::string text,
            std::function< void()> fv )
```

Létrehoz egy új gombot.

Visszatérési érték

Button(int x, int y, int xSize, int ySize, std::string text,

3.5.2. Tagfüggvények dokumentációja

3.5.2.1. Draw()

```
void Button::Draw ( )  [virtual]
```

Kirajzolja a widgetet.

Visszatérési érték

virtual void

Újraimplementált ősök: Label.

3.5.2.2. Handle()

```
void Button::Handle (
            genv::event ev )  [virtual]
```

Kezeli a widgetet az inputnak megfelelõen.

Paraméterek

ev	*genv::event Az input eventje*

Visszatérési érték

virtual void

Újraimplementált ősök: Label.

3.5.2.3. IsInLine()

```
bool Button::IsInLine (
            int x,
            int y )  [virtual]
```

Megadja, hogy egy koordináta rajta van-e a widgeten.

Paraméterek

x	*int Az x koordináta*
y	*int Az y koordináta*

Visszatérési érték

virtual bool

Újraimplementált ősök: Label.

3.5.2.4. SetOnClickColor()

```
void Button::SetOnClickColor (
            int r,
            int g,
            int b )
```

A kattintáskor látható színt változtatja meg.

Paraméterek

r	*int Az új szín piros értéke*
g	*int Az új szín zöld értéke*
b	*int Az új szín kék értéke*

Visszatérési érték

void

3.5.3. Adattagok dokumentációja

3.5.3.1. Function

`std::function<void()> Button::Function [private]`
A kattintáskor meghívandó funció

3.5.3.2. isClicking

`bool Button::isClicking [private]`
Le van-e nyomva a gomb

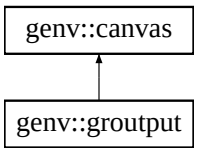
3.5.3.3. OnClickColor

`Colour* Button::OnClickColor [private]`
A kattintáskor látható szín
Ez a dokumentáció az osztályról a következő fájlok alapján készült:

- *Button.h*
- *Button.cpp*

3.6. genv::canvas osztályreferencia

A genv::canvas osztály származási diagramja:



Publikus tagfüggvények

- ***canvas*** *(int w, int h)*
- ***canvas*** *(const canvas &c)*
- *genv::canvas* & ***operator=*** *(const genv::canvas &c)*
- *bool* ***open*** *(unsigned width, unsigned height)*
- *bool* ***save*** *(const std::string &file) const*
- *void* ***transparent*** *(bool t)*
- *void* ***set_color*** *(int r, int g, int b)*
- *bool* ***move_point*** *(int x, int y)*
- *void* ***draw_dot*** *()*
- *void* ***draw_line*** *(int x, int y)*
- *void* ***draw_box*** *(int x, int y)*
- *void* ***draw_text*** *(const std::string &str)*
- *void* ***blitfrom*** *(const canvas &c, short x1, short y1, unsigned short x2, unsigned short y2, short x3, short y3)*
- *bool* ***load_font*** *(const std::string &fname, int fontsize=16, bool antialias=true)*
- *void* ***set_antialias*** *(bool antialias)*
- *int* ***x*** *() const*
- *int* ***y*** *() const*
- *int* ***cascent*** *() const*
- *int* ***cdescent*** *() const*
- *int* ***twidth*** *(const std::string &s) const*
- *virtual void* ***refresh*** *()*
- *template<typename T>*
  void ***call_with_rel*** *(T meth, int vec_x, int vec_y)*

Védett tagfüggvények

- *template<typename T>*
  int ***sgn*** *(const T &a)*

Védett attribútumok

- *short* ***pt_x***
- *short* ***pt_y***
- *SDL_Surface* ∗ ***buf***
- *int* ***draw_clr***
- *bool* ***transp***
- *_TTF_Font* ∗ ***font***
- *bool* ***antialiastext***
- *std::string* ***loaded_font_file_name***
- *int* ***font_size***

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- *graphics.hpp*

3.7. genv::color struktúrareferencia

Publikus tagfüggvények

- ***color*** *(int r, int g, int b)*
- *void* ***operator()*** *(canvas &out)*

Publikus attribútumok

- *int* ***red***
- *int* ***green***
- *int* ***blue***

Ez a dokumentáció a struktúráról a következő fájl alapján készült:

- *graphics.hpp*

3.8. Colour osztályreferencia

Publikus tagfüggvények

- *Colour ()*
- *Colour (int r, int g, int b)*
- *void AdjustColour (int r, int g, int b)*
- *void SetColour (int r, int g, int b)*
- *void SetThisColour ()*
- *int GetR ()*
- *int GetG ()*
- *int GetB ()*

Privát tagfüggvények

- *void CheckRGB ()*

Privát attribútumok

- *int R*
- *int G*
- *int B*

3.8.1. Konstruktorok és destruktorok dokumentációja

3.8.1.1. Colour() [1/2]

`Colour::Colour ( )`

Egy szGB kt tlja, alapelmezetten fekete

3.8.1.2. Colour() [2/2]

```
Colour::Colour (
            int r,
            int g,
            int b )
```

Egy szGB kt tlja

Paraméterek

r	*a sziros k*
g	*a szld k*
b	*a sz*

3.8.2. Tagfüggvények dokumentációja

3.8.2.1. AdjustColour()

```
void Colour::AdjustColour (
            int r,
            int g,
            int b )
```

A sz ma a megadott kkel

Paraméterek

r	*a piros ms m*
g	*a zld ms m*
b	*a ks m*

3.8.2.2. CheckRGB()

```
void Colour::CheckRGB ( )  [private]
```

Ellenrzi, hogy a szk nem-e lk ki a 0-255 tartomb

3.8.2.3. GetB()

```
int Colour::GetB ( )
```

Megadja a tlt szrtt

Visszatérési érték

A tlt krtDoxyReturn

3.8.2.4. GetG()

```
int Colour::GetG ( )
```

Megadja a tlt szld t

Visszatérési érték

A tlt zld DoxyReturn

3.8.2.5. GetR()

```
int Colour::GetR ( )
```

Megadja a tlt sziros t

Visszatérési érték

A tlt piros DoxyReturn

3.8.2.6. SetColour()

```
void Colour::SetColour (
            int r,
            int g,
            int b )
```

Megvoztatja a tlt sz az j sze

Paraméterek

r	*az j piros*
g	*az j zld*
b	*az j krt*

3.8.2.7. SetThisColour()

```
void Colour::SetThisColour ( )
```

A gout szt la az ala tlt sze

3.8.3. Adattagok dokumentációja

3.8.3.1. B

```
int Colour::B  [private]
```

A tlt krt

3.8.3.2. G

```
int Colour::G  [private]
```

A tlt zld

3.8.3.3. R

```
int Colour::R  [private]
```

A tlt piros

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

- *Colour.h*
- *Colour.cpp*

3.9. genv::event struktúrareferencia

Publikus attribútumok

- *int* ***keycode***
- *int* ***pos_x***
- *int* ***pos_y***
- *int* ***button***
- *int* ***time***
- *int* ***type***

Ez a dokumentáció a struktúráról a következő fájl alapján készült:

- *graphics.hpp*

3.10. genv::font struktúrareferencia

Publikus tagfüggvények

- ***font*** *(const std::string &s, int fs, bool a=true)*
- *void* ***operator()*** *(canvas &out)*

Publikus attribútumok

- *std::string* ***font_name***
- *int* ***font_size***
- *bool* ***antialias***

Ez a dokumentáció a struktúráról a következő fájl alapján készült:

- *graphics.hpp*

3.11. GameHandler osztályreferencia

Publikus tagfüggvények

- *GameHandler (int x, int y)*
  Lehozza a jkkezelt

Privát tagfüggvények

- *void LoadMainMenu ()*
  Betlti a fment
- *void LoadGame ()*
  Betlti a jkot.
- *void PlaceAt (int x, int y)*
  Az adott kr alapj megfelel koordinkra elhelyez egy t.
- *void DeleteAreas ()*
  Tlri az sszes "kis nzetet".
- *void DisableAreas ()*
  Deaktiva az sszes "kis nzetet", nem lehet rk kattintani.
- *void EnableAreas ()*
  Aktiva az sszes "kis nzetet", lehet rk kattintani.
- *void DoAIStep ()*
  A sztes jkos lgyet.
- *void ShowWinWindow (std::string text)*
  A gyzelmi ablakot jelen meg.

Privát attribútumok

- *bool IsXTurn*
- *bool IsPlayerX*
- *GUIHandler * handler*
- *Level * level*
- *int GameMode*
- *int NeedToWin*
- *int LevelSize*
- *Area *** *Areas*
- *Label * turnDisplay*
- *MinMax * ai*

3.11.1. Konstruktorok és destruktorok dokumentációja

3.11.1.1. GameHandler()

```
GameHandler::GameHandler (
            int x,
            int y )
```

Lehozza a jkkezelt

Paraméterek

x	*int A krny szss*

Paraméterek

y	*int A krny magass*

3.11.2. Tagfüggvények dokumentációja

3.11.2.1. DeleteAreas()

`void GameHandler::DeleteAreas ( ) [private]`
Tlri az sszes "kis nzetet".
Visszatérési érték

void

3.11.2.2. DisableAreas()

`void GameHandler::DisableAreas ( ) [private]`
Deaktiva az sszes "kis nzetet", nem lehet rk kattintani.
Visszatérési érték

void

3.11.2.3. DoAIStep()

`void GameHandler::DoAIStep ( ) [private]`
A sztes jkos lgyet.
Visszatérési érték

void

3.11.2.4. EnableAreas()

`void GameHandler::EnableAreas ( ) [private]`
Aktiva az sszes "kis nzetet", lehet rk kattintani.
Visszatérési érték

void

3.11.2.5. LoadGame()

`void GameHandler::LoadGame ( ) [private]`
Betlti a jkot.
Visszatérési érték

void

3.11.2.6. LoadMainMenu()

`void GameHandler::LoadMainMenu ( ) [private]`
Betlti a fment
Visszatérési érték

void

3.11.2.7. PlaceAt()

```
void GameHandler::PlaceAt (
            int x,
            int y )  [private]
```

Az adott kr alapj megfelel koordinkra elhelyez egy t.

Paraméterek

x	*int Az x koordin*
y	*int Az y koordin*

Visszatérési érték

void

3.11.2.8. ShowWinWindow()

```
void GameHandler::ShowWinWindow (
            std::string text )  [private]
```

A gyzelmi ablakot jelen meg.

Paraméterek

text	*std::string A szveg arrogy ki nyert*

Visszatérési érték

void

3.11.3. Adattagok dokumentációja

3.11.3.1. ai

```
MinMax* GameHandler::ai  [private]
```

A sztes jkos

3.11.3.2. Areas

```
Area*** GameHandler::Areas  [private]
```

A jk kzben latyzeteket itt tlom

3.11.3.3. GameMode

```
int GameHandler::GameMode  [private]
```

A jkm

3.11.3.4. handler

```
GUIHandler* GameHandler::handler  [private]
```

A grafikai widget kezel

3.11.3.5. IsPlayerX

```
bool GameHandler::IsPlayerX  [private]
```

Megadja, hogy az els jkos x-el jzik-e

3.11.3.6. IsXTurn

```
bool GameHandler::IsXTurn  [private]
```

Megadja, hogy X kvetkezik-e

3.11.3.7. level

```
Level* GameHandler::level  [private]
```

A pa

3.11.3.8. LevelSize

```
int GameHandler::LevelSize  [private]
```

Pa mte

3.11.3.9. NeedToWin

```
int GameHandler::NeedToWin  [private]
```

Pontsz gyzelemhez

3.11.3.10. turnDisplay

```
Label* GameHandler::turnDisplay  [private]
```

A jelenelgi ls kijelzje

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

- *GameHandler.h*
- *GameHandler.cpp*

3.12. genv::grinput osztályreferencia

Publikus tagfüggvények

- *grinput & **wait_event** (event &)*
- *void **timer** (int wait)*
- ***operator const void** * () const*

Statikus publikus tagfüggvények

- *static grinput & **instance** ()*

Privát attribútumok

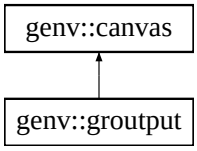
- *bool **quit***

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- *graphics.hpp*

3.13. genv::groutput osztályreferencia

A genv::groutput osztály származási diagramja:



Publikus tagfüggvények

- *void **showmouse** (bool toggle)*
- *void **movemouse** (int x, int y)*
- *bool **open** (unsigned width, unsigned height, bool fullscreen=false)*
- *virtual void **refresh** ()*
- *void **set_title** (const std::string &title)*

Statikus publikus tagfüggvények

- *static groutput & **instance** ()*

Additional Inherited Members

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- *graphics.hpp*

3.14. GUIHandler osztályreferencia

Publikus tagfüggvények

- *GUIHandler (int &xx, int &yy)*
- *void AddWidget (Widget ∗w)*
- *void RemoveWidget (int num)*
- *void RemoveWidget (Widget ∗w)*
- *void **DeleteAllWidget** ()*
- *void Start (bool exitOnEscape, int timer)*
- *void **Exit** ()*
- *bool **GetIsRunning** ()*
- *int **GetWidgetNumber** (Widget ∗a)*

Privát tagfüggvények

- *void Draw ()*
- *void Handle (genv::event ev)*

Privát attribútumok

- *std::vector< Widget ∗ > Widgets*
- *int SelectedWidget*
- *Colour ∗ bgColour*
- *int WindowX*
- *int WindowY*
- *bool **Exiting***
- *bool **IsRunning***
- *bool isEscapeExit*

3.14.1. Konstruktorok és destruktorok dokumentációja

3.14.1.1. GUIHandler()

```
GUIHandler::GUIHandler (
            int & xx,
            int & yy )
```

Ez az oszt kezeli az sszes widgetet

Paraméterek

xx	*az ablak szss*
yy	*az ablak magass*

3.14.2. Tagfüggvények dokumentációja

3.14.2.1. AddWidget()

```
void GUIHandler::AddWidget (
            Widget * w )
```

Ezzel a fggvvyel lehet j widgetet hozzni az oszthoz

Paraméterek

w	*egy mutatzzni knt widgetre*

3.14.2.2. Draw()

```
void GUIHandler::Draw ( )  [private]
```

Ez a fggv felel a krny trl widgetek kirajzol

3.14.2.3. Handle()

```
void GUIHandler::Handle (
            genv::event ev )  [private]
```

Ez a fggv kezeli az inputokat

Paraméterek

ev	*a megkapott eventet tlja*

3.14.2.4. RemoveWidget() [1/2]

```
void GUIHandler::RemoveWidget (
            int num )
```

Ez a fggv az adott pozon lev widgetet tla el az osztbs trli a krnyrl

Paraméterek

num	*adja meg a vektorban lev sorszt a trlni knt vektornak*

3.14.2.5. RemoveWidget() [2/2]

```
void GUIHandler::RemoveWidget (
            Widget * w )
```

Ez a fggv trli a megadott widgetet az osztba lzik

Paraméterek

w	*a trlni knt widget*

3.14.2.6. Start()

```
void GUIHandler::Start (
            bool exitOnEscape,
            int timer )
```

Ez a fggv inda el a f ciklust ami kezeli az inputot widgeteket

Paraméterek

exitOnEscape	*ha igazra van itva a grafikus felletbl ki lehet li az Esc billenty lenyomval*
timer	*megadja, hogy h milliszekundomonk rajzolja jra a krnyt*

3.14.3. Adattagok dokumentációja

3.14.3.1. bgColour

`Colour* GUIHandler::bgColour [private]`
Ez az ablak hzt adja meg

3.14.3.2. isEscapeExit

`bool GUIHandler::isEscapeExit [private]`
Ezzel ellenrzm, hogy ki kell-e li az Esc billenty lenyomra

3.14.3.3. SelectedWidget

`int GUIHandler::SelectedWidget [private]`
Ez az en kivsztott widget sorszt tlja

3.14.3.4. Widgets

`std::vector<Widget*> GUIHandler::Widgets [private]`
Ez a vektor tlja a widgeteket

3.14.3.5. WindowX

`int GUIHandler::WindowX [private]`
Ebben tlom az ablak szsst

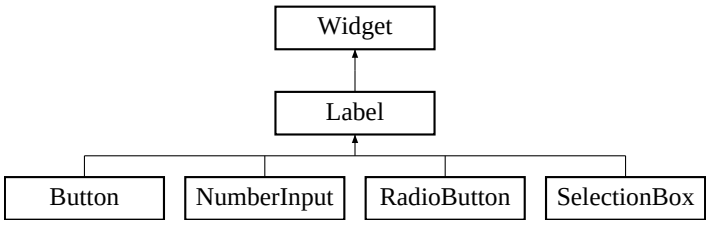
3.14.3.6. WindowY

`int GUIHandler::WindowY [private]`
Ebben tlom az ablak magasst
Ez a dokumentáció az osztályról a következő fájlok alapján készült:

- *GUIHandler.h*
- *GUIHandler.cpp*

3.15. Label osztályreferencia

A Label osztály származási diagramja:



Publikus tagfüggvények

- *Label (int x, int y, int xSize, int ySize, std::string text)*
- *virtual void Draw ()*
- *virtual void Handle (genv::event ev)*
- *virtual bool IsInLine (int x, int y)*
- *bool IsTextFits ()*
- *void SetFontColour (int r, int g, int b)*
- *void SetText (std::string newText)*

Védett attribútumok

- *Colour * fontColor*
- *std::string Text*

3.15.1. Konstruktorok és destruktorok dokumentációja

3.15.1.1. Label()

```
Label::Label (
            int x,
            int y,
            int xSize,
            int ySize,
            std::string text )
```

Egy Label widgetet hoz létre ami egyszerűen egy szöveget tárol és rajzol ki

Paraméterek

x	*Az X pozíciót tárolja*
y	*Az Y pozíciót tátolja*
xSize	*Az X tengelyen való hosszt tárolja*
ySize	*Az Y tengelyen való magasságot tárolja*
text	*A kirajzolni kívánt szöveg*

3.15.2. Tagfüggvények dokumentációja

3.15.2.1. Draw()

```
void Label::Draw ( )  [virtual]
```

Ez a függvény felel a widget kirajzolásáért
Megvalósítja a következőket: Widget.
Újraimplementáló leszármazottak: RadioButton, NumberInput, SelectionBox és Button.

3.15.2.2. Handle()

```
virtual void Label::Handle (
            genv::event ev )  [inline], [virtual]
```

Ez a függvény kezeli az eventeket (ebben az oszályban üres, mivel a Label csak szöveg kijelzésére használt)

Paraméterek

ev	*Az aktuális event objektum*

Megvalósítja a következőket: Widget.
Újraimplementáló leszármazottak: RadioButton, Button, NumberInput és SelectionBox.

3.15.2.3. IsInLine()

```
bool Label::IsInLine (
            int x,
            int y )  [virtual]
```

Megadja, hogy az egér a saját keretein belül van-e

Paraméterek

x	*Az egér X pozíciója*
y	*Az egér Y pozíciója*

Megvalósítja a következőket: Widget.
Újraimplementáló leszármazottak: RadioButton, Button, NumberInput és SelectionBox.

3.15.2.4. IsTextFits()

```
bool Label::IsTextFits ( )
```

Megadja, hogy az éppen tárolt szöveg elfér-e a widgeten

Visszatérési érték

Elfér-e a "Text" a widgeten

3.15.2.5. SetFontColour()

```
void Label::SetFontColour (
            int r,
            int g,
            int b )
```

Átálítja az betű színét

Paraméterek

r	*A betű piros értéke*
g	*A betű zöld értéke*
b	*A betű kék értéke*

3.15.2.6. SetText()

```
void Label::SetText (
            std::string newText )
```

Átálítja az aktuális szöveget

Paraméterek

newText	*Az új szöveg*

3.15.3. Adattagok dokumentációja

3.15.3.1. fontColor

```
Colour* Label::fontColor  [protected]
```

A szöveg színét tárolja

3.15.3.2. Text

```
std::string Label::Text  [protected]
```

Az aktuális szöveget tárolja
Ez a dokumentáció az osztályról a következő fájlok alapján készült:

- *Label.h*
- *Label.cpp*

3.16. Level osztályreferencia

Publikus tagfüggvények

- *Level (int levelSize, int win)*

Lehoz egy j pDoxyCompactList
int GetValue (int x, int y)
Megadja a pa egy adott koordinjev mez t
int ValueToWin ()
Visszatsk megadja, hogy h pont kell a gyzelemhez.
int GetSize ()
A pa mtdja meg.
void Place (int x, int y, bool xVal)
A megadott koordinkra elhelyez egy t.
int IsGameWon ()
Megadja, hogy megnyerte-e valaki a jkot.
void RemoveValue (int x, int y)
Egy adott koordinrlri az t.
bool IsAreaEmpty (int x, int y)
Megadja, hogy egy adott koordin krl resek-e a mezk
bool IsLevelEmpty ()
Megadja, hogy res-e a pa.
void Recount ()
Megszlja, hogy h X van a pDoxyCompactList
void Reset ()
Visszaa a plapelmezetten resbe.
int GetLastX ()
Az utolj mtt mez x koordinjdja vissza.
int GetLastY ()
Az utolj mtt mez y koordinjdja vissza.
void WriteCurrentPosition (std::ostream &writer)
A megadott m kia a pa aktus tartalmDoxyCompactList

Privát tagfüggvények

* *int CheckArea (int x, int y, int LookFor)*
  Egy terleten ellenrzi, hogy elontja van-e a jkosnak a gyzelemhez.

Privát attribútumok

* *int ** LevelData*
* *int Size*
* *int NeedToWin*
* *int XCount*
* *int OCount*
* *int LastX*
* *int LastY*

3.16.1. Konstruktorok és destruktorok dokumentációja

3.16.1.1. Level()

```
Level::Level (
            int levelSize,
            int win )
```

Lehoz egy j p

Paraméterek

levelSize	*int A pa mte (X*X)*
win	*int A szkss pontsz jk megnyerhez*

3.16.2. Tagfüggvények dokumentációja

3.16.2.1. CheckArea()

```
int Level::CheckArea (
            int x,
            int y,
            int LookFor )  [private]
```

Egy terleten ellenrzi, hogy elontja van-e a jkosnak a gyzelemhez.

Paraméterek

x	*int A terlet kzepk x koordinja*
y	*int A terlet kzepk y koordinja*
LookFor	*int A jkos, akit figyelni kell*

Visszatérési érték

int Az adott jkos , ha nyert, 0 ha nem

3.16.2.2. GetLastX()

```
int Level::GetLastX ( )
```

Az utolj mtt mez x koordinjdja vissza.
Visszatérési érték

int Az utolj mtt mez x koordinja

3.16.2.3. GetLastY()

```
int Level::GetLastY ( )
```

Az utolj mtt mez y koordinjdja vissza.
Visszatérési érték

int Az utolj mtt mez y koordinja

3.16.2.4. GetSize()

```
int Level::GetSize ( )
```

A pa mtdja meg.
Visszatérési érték

int A pa mte

3.16.2.5. GetValue()

```
int Level::GetValue (
            int x,
            int y )
```

Megadja a pa egy adott koordinjev mez t

Paraméterek

x	*int Az x koordin*
y	*int Az y koordin*

Visszatérési érték

int Az x,y koordinn lev DoxyReturn

3.16.2.6. IsAreaEmpty()

```
bool Level::IsAreaEmpty (
            int x,
            int y )
```

Megadja, hogy egy adott koordin krl resek-e a mezk

Paraméterek

x	*int Az x koordin*
y	*int Az y koordin*

Visszatérési érték

bool esek-e a mezk egy inervalumon bell

3.16.2.7. IsGameWon()

```
int Level::IsGameWon ( )
```

Megadja, hogy megnyerte-e valaki a jkot.
Visszatérési érték

int A jk aktus apota sz (-1 nincs v, 0 dntetlen, 1 x nyert, 2 O nyert)

3.16.2.8. IsLevelEmpty()

```
bool Level::IsLevelEmpty ( )
```

Megadja, hogy res-e a pa.
Visszatérési érték

bool es-e a pa

3.16.2.9. Place()

```
void Level::Place (
            int x,
            int y,
            bool xVal )
```

A megadott koordinkra elhelyez egy t.

Paraméterek

x	*int Az x koordin*
y	*int Az y koordin*
xVal	*bool Az elhelyezni knt -e*

Visszatérési érték

void

3.16.2.10. Recount()

```
void Level::Recount ( )
```

Megszlja, hogy h X van a p

Visszatérési érték

void

3.16.2.11. RemoveValue()

```
void Level::RemoveValue (
            int x,
            int y )
```

Egy adott koordinrlri az t.

Paraméterek

x	*int Az x koordin*
y	*int Az y koordin*

Visszatérési érték

void

3.16.2.12. Reset()

```
void Level::Reset ( )
```

Visszaa a plapelmezetten resbe.
Visszatérési érték

void

3.16.2.13. ValueToWin()

```
int Level::ValueToWin ( )
```

Visszatsk megadja, hogy h pont kell a gyzelemhez.
Visszatérési érték

int Pontok sz a gyzelemhez

3.16.2.14. WriteCurrentPosition()

```
void Level::WriteCurrentPosition (
            std::ostream & writer )
```

A megadott m kia a pa aktus tartalm

Paraméterek

writer	*std::ostream& A kimeneti forr*

Visszatérési érték

void

3.16.3. Adattagok dokumentációja

3.16.3.1. LastX

```
int Level::LastX  [private]
```

Az utolj mtt mez X koordinja

3.16.3.2. LastY

```
int Level::LastY  [private]
```

Az utolj mtt mez Y koordinja

3.16.3.3. LevelData

`int** Level::LevelData [private]`
Maga a pa adatai

3.16.3.4. NeedToWin

`int Level::NeedToWin [private]`
A nyeret szkss pontsz

3.16.3.5. OCount

`int Level::OCount [private]`
O-k sz

3.16.3.6. Size

`int Level::Size [private]`
A pa mte

3.16.3.7. XCount

`int Level::XCount [private]`
X-ek sz
Ez a dokumentáció az osztályról a következő fájlok alapján készült:
- *Level.h*
- *Level.cpp*

3.17. genv::line struktúrareferencia

Publikus tagfüggvények

- ***line*** *(int x, int y)*
- *void* ***operator()*** *(canvas &out)*

Publikus attribútumok

- *int* ***vec_x***
- *int* ***vec_y***

Ez a dokumentáció a struktúráról a következő fájl alapján készült:
- *graphics.hpp*

3.18. genv::line_to struktúrareferencia

Publikus tagfüggvények

- ***line_to*** *(unsigned x, unsigned y)*
- *void* ***operator()*** *(canvas &out)*

Publikus attribútumok

- *int* ***pos_x***
- *int* ***pos_y***

Ez a dokumentáció a struktúráról a következő fájl alapján készült:
- *graphics.hpp*

3.19. MinMax osztályreferencia

Publikus tagfüggvények

- *MinMax (int deep, Level ∗currentLevel, bool isX)*
  Létrehoz egy új MiniMax algoritmus alapján működő számítógépes játékost.
- *void Step ()*

Ez a funkció számolja ki a következõ lépését a MiniMax algoritmusnak.

Privát tagfüggvények

* *StepData NextStep (int deep, bool isAI)*
  Ez a függvény számolja a következõ lépést, úgy hogy rekurzívan meghívja önmagát.
* *void **CopyLevels** ()*

Privát attribútumok

* *bool IsX*
* *Level * playingLevel*
* *Level * computingLevel*
* *int MaxSteps*
* *int MapSize*
* *std::ofstream **Writer***

3.19.1. Konstruktorok és destruktorok dokumentációja

3.19.1.1. MinMax()

```
MinMax::MinMax (
            int deep,
            Level * currentLevel,
            bool isX )
```

Létrehoz egy új MiniMax algoritmus alapján működő számítógépes játékost.

Paraméterek

deep	*int Ezzel lehet megadni, hogy maximum hány lépést számoljon előre*
currentLevel	*Level* Ez a aktuális pályára mutató mutató*
isX	*bool Ez adja meg, hogy a gépi játékos az X-el játszik-e*

3.19.2. Tagfüggvények dokumentációja

3.19.2.1. NextStep()

```
StepData MinMax::NextStep (
            int deep,
            bool isAI )  [private]
```

Ez a függvény számolja a következõ lépést, úgy hogy rekurzívan meghívja önmagát.

Paraméterek

deep	*int Az aktuális érték, hogy hány lépést számolt elõre*
isAI	*bool Megadja, hogy az aktuális lépés neki tesz jót, vagy a játékosnal*

Visszatérési érték

StepData Megadja a lépés helyét, és a maximálisan szerzett pontot

3.19.2.2. Step()

```
void MinMax::Step ( )
```

Ez a funckció számolja ki a következõ lépését a MiniMax algoritmusnak.
Visszatérési érték

void

3.19.3. Adattagok dokumentációja

3.19.3.1. computingLevel

`Level* MinMax::computingLevel [private]`
Ezen a pályán számolja a lépéseit, így nem kell az éles pályán megtenni a próbalépéseket

3.19.3.2. IsX

`bool MinMax::IsX [private]`
Ez a változó tárolja, hogy a gép 'X' szerint játszik-e

3.19.3.3. MapSize

`int MinMax::MapSize [private]`
Ez a változó tárolja a pálya méretét, így nem kell újra és újra lekérni

3.19.3.4. MaxSteps

`int MinMax::MaxSteps [private]`
Ez a változó adja meg, hogy maximum hány lépést számolhat elõre. Minnél nagyobb annál pontosabban lép, de a számolás hosszúsága exponenciálisan nõ

3.19.3.5. playingLevel

`Level* MinMax::playingLevel [private]`
Ez a mutató mutat a valódi pályára, amin a játék zajlik
Ez a dokumentáció az osztályról a következő fájlok alapján készült:

- *MinMax.h*
- *MinMax.cpp*

3.20. genv::move struktúrareferencia

Publikus tagfüggvények

- ***move*** *(int x, int y)*
- *void* ***operator()*** *(canvas &out)*

Publikus attribútumok

- *int* ***vec_x***
- *int* ***vec_y***

Ez a dokumentáció a struktúráról a következő fájl alapján készült:

- *graphics.hpp*

3.21. genv::move_to struktúrareferencia

Publikus tagfüggvények

- ***move_to*** *(unsigned x, unsigned y)*
- *void* ***operator()*** *(canvas &out)*

Publikus attribútumok

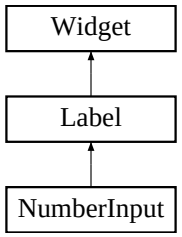
* int **pos_x**
* int **pos_y**

Ez a dokumentáció a struktúráról a következő fájl alapján készült:

* graphics.hpp

3.22. NumberInput osztályreferencia

A NumberInput osztály származási diagramja:



Publikus tagfüggvények

* NumberInput (int x, int y, int xSize, int ySize, int startingNumber, int max↩ Value, int minValue)
* void Draw ()
* void Handle (genv::event ev)
* bool IsInLine (int x, int y)
* void SetValue (int val)
* void AdjustValue (int val)
* void Step (bool up, bool big)
* void SetMaximumValue (int maximum)
* void SetMinimumValue (int minimum)
* void SetStepValue (int step)
* void SetBigStepValue (int bigstep)
* void SetButtonColour (int r, int g, int b)
* void SetButtonOnClickColour (int r, int g, int b)
* int **GetCurrentValue** ()
* void **SetEventVoid** (std::function< void(NumberInput *)> event)

Privát tagfüggvények

* void CheckValue ()

Privát attribútumok

* int CurrentNumber
* int MaximumValue
* int MinimumValue
* int StepValue
* int BigStepValue
* bool buttonUpClicking
* bool buttonDownClicking
* std::function< void(NumberInput *)> **OnEvent**
* Colour * buttonColour
* Colour * buttonClickColour

Additional Inherited Members

3.22.1. Konstruktorok és destruktorok dokumentációja

3.22.1.1. NumberInput()

```
NumberInput::NumberInput (
            int x,
            int y,
            int xSize,
            int ySize,
```

```
                int startingNumber,
                int maxValue,
                int minValue )
```

Egy sz widget aminek az kombbal, valamint a fel e nyilakkal page up, page down gombokkal szabozhat x *A widget X poz*

y *A widget Y poz*

xSize *A widget hossza (20 pixel martozik a vhez, ezek a gombok)*

ySize *A widget magass*

startingNumber *A sz kezd sz*

maxValue *A maximum a sz nem mehet ezen tl*

minValue *A minimum a sz nem lehet ennisebb*

3.22.2. Tagfüggvények dokumentációja

3.22.2.1. AdjustValue()

```
void NumberInput::AdjustValue (
            int val )
```

Hozz valamennyit a sz jelenlegi hez

Paraméterek

val	*A hpz-zand*

3.22.2.2. CheckValue()

```
void NumberInput::CheckValue ( )  [private]
```

Ellenrzi, hogy a jelenlegi em-e l a hatkon

3.22.2.3. Draw()

```
void NumberInput::Draw ( )  [virtual]
```

Ez felel a widget kirajzol
Újraimplementált ősök: Label.

3.22.2.4. Handle()

```
void NumberInput::Handle (
            genv::event ev )  [virtual]
```

Ez a fggv kezeli az eventeket

Paraméterek

ev	*Az aktus event objektum*

Újraimplementált ősök: Label.

3.22.2.5. IsInLine()

```
bool NumberInput::IsInLine (
            int x,
            int y )  [virtual]
```

Megadja, hogy az eg sajeretein bell van-e

Paraméterek

x	*Az eg poz*
y	*Az eg poz*

Újraimplementált ősök: Label.

3.22.2.6. SetBigStepValue()

```
void NumberInput::SetBigStepValue (
            int bigstep )
```

ttja a nagy egysrtt (amit a page up/down gombra l

Paraméterek

bigstep	*Az j egysrt*

3.22.2.7. SetButtonColour()

```
void NumberInput::SetButtonColour (
            int r,
            int g,
            int b )
```

ttja a gomb szt

Paraméterek

r	*Az j piros*
g	*Az j zld*
b	*Az j krt*

3.22.2.8. SetButtonOnClickColour()

```
void NumberInput::SetButtonOnClickColour (
            int r,
            int g,
            int b )
```

ttja a gomb kattintor latnDoxyParamsParaméterek r *Az j piros*
g *Az j zld*
b *Az j krt*

3.22.2.9. SetMaximumValue()

```
void NumberInput::SetMaximumValue (
            int maximum )
```

felskorlt ad a widgetnek

Paraméterek

maximum	Az j maximum

3.22.2.10. SetMinimumValue()

```
void NumberInput::SetMinimumValue (
            int minimum )
```

alslt ad a widgetnek

Paraméterek

minimum	Az j minimum

3.22.2.11. SetStepValue()

```
void NumberInput::SetStepValue (
            int step )
```

ttja a kis egysrtt (amit egy kattintal l

Paraméterek

step	Az j egysrt

3.22.2.12. SetValue()

```
void NumberInput::SetValue (
            int val )
```

ttja a sz t

Paraméterek

val	Az j

3.22.2.13. Step()

```
void NumberInput::Step (
            bool up,
            bool big )
```

A szt leti kis vagy nagy egysyivel pozitagy negatrba

Paraméterek

up	Ez adja meg, hogy pozitrba le
bug	Ez adja meg, hogy kis vagy nagy egyst l

3.22.3. Adattagok dokumentációja

3.22.3.1. BigStepValue

```
int NumberInput::BigStepValue  [private]
```

A nagy ls m

3.22.3.2. buttonClickColour

```
Colour* NumberInput::buttonClickColour  [private]
```
A gomb kattintor latnlja

3.22.3.3. buttonColour

```
Colour* NumberInput::buttonColour  [private]
```
A gomb szt tlja

3.22.3.4. buttonDownClicking

```
bool NumberInput::buttonDownClicking  [private]
```
Igaz, amennyiben a lefele gombra en rttintanak

3.22.3.5. buttonUpClicking

```
bool NumberInput::buttonUpClicking  [private]
```
Igaz, amennyiben a felfele gombra en rttintanak

3.22.3.6. CurrentNumber

```
int NumberInput::CurrentNumber  [private]
```
A jelenlegi

3.22.3.7. MaximumValue

```
int NumberInput::MaximumValue  [private]
```
A maximum

3.22.3.8. MinimumValue

```
int NumberInput::MinimumValue  [private]
```
A minimum

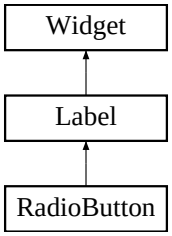
3.22.3.9. StepValue

```
int NumberInput::StepValue  [private]
```
A kis Is m

Ez a dokumentáció az osztályról a következő fájlok alapján készült:
* NumberInput.h
* NumberInput.cpp

3.23. RadioButton osztályreferencia

A RadioButton osztály származási diagramja:



Publikus tagfüggvények

* RadioButton (int x, int y, int buttonSize, std::string text, int value)
  Létrehoz egy új kiválasztó gombot.
* virtual void Draw ()
  Kirajzolja a gombot.
* virtual void Handle (genv::event ev)
  Kezeli a widgetet az inputnak megfelelően.
* virtual bool IsInLine (int x, int y)

Megadja, hogy egy koordináta rajta van-e a widgeten.
* void SetSelection (bool value)
Beállítja, hogy ki van-e jelölve.
* bool GetSelection ()
Megadja, hogy ki van-e jelölve.
* int GetValue ()
Megadja a gomb értékét.

Privát tagfüggvények

* void DrawCircle (int x, int y, int r, int thickness)
Rajzol egy kört.

Privát attribútumok

* int Value
* bool IsSelected
* Colour * onClickColour

Additional Inherited Members

3.23.1. Konstruktorok és destruktorok dokumentációja

3.23.1.1. RadioButton()

```
RadioButton::RadioButton (
            int x,
            int y,
            int buttonSize,
            std::string text,
            int value )
```

Létrehoz egy új kiválasztó gombot.

Paraméterek

x	int Az x koordináta
y	int Az y koordináta
buttonSize	int A gomb sugara
text	std::string A gomb szövege
value	int A gomb értéke

3.23.2. Tagfüggvények dokumentációja

3.23.2.1. Draw()

```
void RadioButton::Draw ( )  [virtual]
```

Kirajzolja a gombot.
Visszatérési érték

virtual void

Újraimplementált ősök: Label.

3.23.2.2. DrawCircle()

```
void RadioButton::DrawCircle (
            int x,
            int y,
```

```
                int r,
                int thickness )  [private]
```

Rajzol egy kört.

Paraméterek

x	int A kör középpontjának x koordinátája
y	int A kör középpontjának y koordinátája
r	int A kör sugara
thickness	int A kör vastagsága

Visszatérési érték

void

3.23.2.3. GetSelection()

```
bool RadioButton::GetSelection ( )
```

Megadja, hogy ki van-e jelölve.

Visszatérési érték

bool A kijelölés értéke

3.23.2.4. GetValue()

```
int RadioButton::GetValue ( )
```

Megadja a gomb értékét.

Visszatérési érték

int A gomb értéke

3.23.2.5. Handle()

```
void RadioButton::Handle (
                genv::event ev )  [virtual]
```

Kezeli a widgetet az inputnak megfelelõen.

Paraméterek

ev	genv::event Az input eventje

Visszatérési érték

virtual void

Újraimplementált ősök: Label.

3.23.2.6. IsInLine()

```
bool RadioButton::IsInLine (
                int x,
                int y )  [virtual]
```

Megadja, hogy egy koordináta rajta van-e a widgeten.

Paraméterek

x	int Az x koordináta
y	int Az y koordináta

Visszatérési érték

virtual bool

Újraimplementált ősök: Label.

3.23.2.7. SetSelection()

```
void RadioButton::SetSelection (
            bool value )
```

Beállítja, hogy ki van-e jelölve.

Paraméterek

value	bool A kijelölés értéke

Visszatérési érték

void

3.23.3. Adattagok dokumentációja

3.23.3.1. IsSelected

```
bool RadioButton::IsSelected  [private]
```

A gomb kiválasztottsága

3.23.3.2. onClickColour

```
Colour* RadioButton::onClickColour  [private]
```

A gomb színe kattintáskor

3.23.3.3. Value

```
int RadioButton::Value  [private]
```

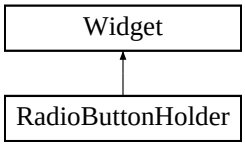
A gomb értéke

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

- RadioButton.h
- RadioButton.cpp

3.24. RadioButtonHolder osztályreferencia

A RadioButtonHolder osztály származási diagramja:



Publikus tagfüggvények

- RadioButtonHolder ()
  Létrehot egy kiválasztó gomb tartó objektumot.
- void AddRadioButton (RadioButton *button)
  A listához ad egy új gombot.
- void RemoveRadioButton (RadioButton *button)
  Eltávolít egy gombot.
- void RemoveRadioButton (int place)
  Eltávolít egy gombot.
- int GetCurrentlySelected ()
  Megadja a jelenleg kiválasztott gomb sorszámát.
- int CurrentlySelectedValue ()

Megadja a jelenleg kiválasztott gomb értékét.
* void SetEventVoid (std::function< void(RadioButtonHolder *)> event)
Egy gomra történő kattintáskor meghívandó esemény.
* virtual void Draw ()
Kirajzolja a widgeteket.
* virtual void Handle (genv::event ev)
Kezeli a widgeteket az inputnak megfelelõen.
* virtual bool IsInLine (int x, int y)
Megadja, hogy egy koordináta rajta van-e a widgeten.

Privát tagfüggvények

* void CheckNull ()
Ellenőrzi, hogy vannak-e gombok.

Privát attribútumok

* std::vector< RadioButton * > radioButtons
* std::function< void(RadioButtonHolder *)> OnEvent
* int ButtonCount
* int CurrentlySelected

Additional Inherited Members

3.24.1. Konstruktorok és destruktorok dokumentációja

3.24.1.1. RadioButtonHolder()

```
RadioButtonHolder::RadioButtonHolder ( )
```
Létrehot egy kiválasztó gomb tartó objektumot.

3.24.2. Tagfüggvények dokumentációja

3.24.2.1. AddRadioButton()

```
void RadioButtonHolder::AddRadioButton (
            RadioButton * button )
```
A listához ad egy új gombot.

Paraméterek

button	RadioButton* Az új gomn

Visszatérési érték

void

3.24.2.2. CheckNull()

```
void RadioButtonHolder::CheckNull ( )  [private]
```
Ellenőrzi, hogy vannak-e gombok.
Visszatérési érték

void

3.24.2.3. CurrentlySelectedValue()

```
int RadioButtonHolder::CurrentlySelectedValue ( )
```
Megadja a jelenleg kiválasztott gomb értékét.

Visszatérési érték

int A jelenleg kiválasztott gomb értéke

3.24.2.4. Draw()

```
void RadioButtonHolder::Draw ( )  [virtual]
```

Kirajzolja a widgeteket.

Visszatérési érték

virtual void

Megvalósítja a következőket: Widget.

3.24.2.5. GetCurrentlySelected()

```
int RadioButtonHolder::GetCurrentlySelected ( )
```

Megadja a jelenleg kiválasztott gomb sorszámát.

Visszatérési érték

int A jelenleg kiválasztott gomb sorszáma

3.24.2.6. Handle()

```
void RadioButtonHolder::Handle (
            genv::event ev )  [virtual]
```

Kezeli a widgeteket az inputnak megfelelõen.

Paraméterek

ev	genv::event Az input eventje

Visszatérési érték

virtual void

Megvalósítja a következőket: Widget.

3.24.2.7. IsInLine()

```
bool RadioButtonHolder::IsInLine (
            int x,
            int y )  [virtual]
```

Megadja, hogy egy koordináta rajta van-e a widgeten.

Paraméterek

x	int Az x koordináta
y	int Az y koordináta

Visszatérési érték

virtual bool

Megvalósítja a következőket: Widget.

3.24.2.8. RemoveRadioButton() [1/2]

```
void RadioButtonHolder::RemoveRadioButton (
            RadioButton * button )
```

Eltávolít egy gombot.

Paraméterek

button	RadioButton* Az eltávolítandó gomb

Visszatérési érték

void

3.24.2.9. RemoveRadioButton() [2/2]

```
void RadioButtonHolder::RemoveRadioButton (
            int place )
```

Eltávolít egy gombot.

Paraméterek

place	int Az eltávolítandó gomb sorszáma

Visszatérési érték

void

3.24.2.10. SetEventVoid()

```
void RadioButtonHolder::SetEventVoid (
            std::function< void(RadioButtonHolder *)> event )
```

Egy gomra történő kattintáskor meghívandó esemény.

Paraméterek

RadioButtonHolder∗	A meghívandó esemény

Visszatérési érték

void

3.24.3. Adattagok dokumentációja

3.24.3.1. ButtonCount

```
int RadioButtonHolder::ButtonCount  [private]
```

A gombok száma

3.24.3.2. CurrentlySelected

```
int RadioButtonHolder::CurrentlySelected  [private]
```

A jeleneg kiválasztott gomb sorszáma

3.24.3.3. OnEvent

```
std::function<void(RadioButtonHolder*)> RadioButtonHolder::On↩
Event  [private]
```

A meghíndó esemény

3.24.3.4. radioButtons

```
std::vector<RadioButton*> RadioButtonHolder::radioButtons  [private]
```

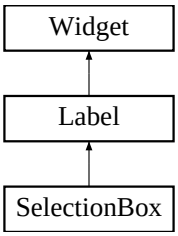
A tárolt gombok listája

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

* RadioButtonHolder.h
* RadioButtonHolder.cpp

3.25. SelectionBox osztályreferencia

A SelectionBox osztály származási diagramja:



Publikus tagfüggvények

- SelectionBox (int x, int y, int xSize, int ySize, std::vector< std::string > values)
- void Draw ()
- void Handle (genv::event ev)
- bool IsInLine (int x, int y)
- void AddItem (std::string value)
- void RemoveItem (std::string value)
- void RemoveItem (int place)
- int CurrentlySelectedValue ()
- std::string CurrentlySelectedItem ()
- void EditSelectedFontColour (int r, int g, int b)
- void EditSelectedFrontColour (int r, int g, int b)
- void SetButtonSize (int button)
- void EditMaxItemInView (int elements)

Privát attribútumok

- bool isOpened
- int maxItemsInView
- int currentlySelected
- int mouseOnItem
- int topItem
- Colour * selectedFontColour
- Colour * selectedFrontColour
- Colour * buttonColour
- Colour * buttonOnClickColour
- int buttonSize = 15
- std::vector< std::string > Values

Additional Inherited Members

3.25.1. Konstruktorok és destruktorok dokumentációja

3.25.1.1. SelectionBox()

```
SelectionBox::SelectionBox (
            int x,
            int y,
            int xSize,
            int ySize,
            std::vector< std::string > values )
```

Egy legrdl ment ad meg.

Paraméterek

x	A widget X poz
y	A widget Y poz
xSize	A widget hossza
ySize	A widget magass
values	A widget kezd it tartalmazza

3.25.2. Tagfüggvények dokumentációja

3.25.2.1. AddItem()

```
void SelectionBox::AddItem (
            std::string value )
```

elemet ad hozzistz

Paraméterek

value	Az j elem

3.25.2.2. CurrentlySelectedItem()

```
std::string SelectionBox::CurrentlySelectedItem ( )
```

Megadja a jelenleg kijellt elemet

Visszatérési érték

A kijellt elem

3.25.2.3. CurrentlySelectedValue()

```
int SelectionBox::CurrentlySelectedValue ( )
```

Megadja a jelenleg kijellt elem sorszt a listn

Visszatérési érték

A kijellt elem sorsz

3.25.2.4. Draw()

```
void SelectionBox::Draw ( )  [virtual]
```

Ez felel a widget kirajzol

Újraimplementált ősök: Label.

3.25.2.5. EditMaxItemInView()

```
void SelectionBox::EditMaxItemInView (
            int elements )
```

ttja, hogy maximum h elem lzn a leny menben. Amennyiben tbb elemet tartalmaz, mint ez a sz grgetni kell

Paraméterek

Az	egyszerre latmek sz

3.25.2.6. EditSelectedFontColour()

```
void SelectionBox::EditSelectedFontColour (
            int r,
            int g,
            int b )
```

ttja kijellt szveg szt

Paraméterek

r	Az j piros
g	Az j zld

Paraméterek

b	Az j krt

3.25.2.7. EditSelectedFrontColour()

```
void SelectionBox::EditSelectedFrontColour (
            int r,
            int g,
            int b )
```

ttja a kijellt eltzt

Paraméterek

r	Az j piros
g	Az j zld
b	Az j krt

3.25.2.8. Handle()

```
void SelectionBox::Handle (
            genv::event ev )  [virtual]
```

Ez a fggv kezeli az eventeket

Paraméterek

ev	Az aktus event objektum

Újraimplementált ősök: Label.

3.25.2.9. IsInLine()

```
bool SelectionBox::IsInLine (
            int x,
            int y )  [virtual]
```

Megadja, hogy az eg sajeretein bell van-e

Paraméterek

x	Az eg poz
y	Az eg poz

Újraimplementált ősök: Label.

3.25.2.10. RemoveItem() [1/2]

```
void SelectionBox::RemoveItem (
            std::string value )
```

Trl egy elemet a listl

Paraméterek

value	A trlni knt elem

3.25.2.11. RemoveItem() [2/2]

```
void SelectionBox::RemoveItem (
            int place )
```

Trl egy elemet a listl

Paraméterek

value	A trlni knt elem helye

3.25.2.12. SetButtonSize()

```
void SelectionBox::SetButtonSize (
            int button )
```

Megadja a kis gombok mts a men mellett laty gomb szsst

Paraméterek

button	A gombok mte

3.25.3. Adattagok dokumentációja

3.25.3.1. buttonColour

```
Colour* SelectionBox::buttonColour  [private]
```

A gomb sz

3.25.3.2. buttonOnClickColour

```
Colour* SelectionBox::buttonOnClickColour  [private]
```

A gomb sz kattintor

3.25.3.3. buttonSize

```
int SelectionBox::buttonSize = 15  [private]
```

A gombok mte

3.25.3.4. currentlySelected

```
int SelectionBox::currentlySelected  [private]
```

Megadja, hogy hanyadik elem van jelenleg kivsztva

3.25.3.5. isOpened

```
bool SelectionBox::isOpened  [private]
```

Megadja, hogy le van-e grde a lista

3.25.3.6. maxItemsInView

```
int SelectionBox::maxItemsInView  [private]
```
Megadja, hogy h elem latszerre

3.25.3.7. mouseOnItem

```
int SelectionBox::mouseOnItem  [private]
```
Megadja, hogy jelenleg hanyadik elem fltt van az eg -1 ha egyik fltt sem vagy a lista csukva van

3.25.3.8. selectedFontColour

```
Colour* SelectionBox::selectedFontColour  [private]
```
A kivsztetsz

3.25.3.9. selectedFrontColour

```
Colour* SelectionBox::selectedFrontColour  [private]
```
A kivsztltz

3.25.3.10. topItem

```
int SelectionBox::topItem  [private]
```
Megadja, hogy a legrdl listn melyik elem latfll

3.25.3.11. Values

```
std::vector<std::string> SelectionBox::Values  [private]
```
Ebben vannak tlva az k

Ez a dokumentáció az osztályról a következő fájlok alapján készült:
* SelectionBox.h
* SelectionBox.cpp

3.26. genv::stamp struktúrareferencia

Publikus tagfüggvények

* **stamp** (canvas &cc, int sx1, int sy1, int xsize, int ysize, int tx1, int ty1)
* **stamp** (canvas &cc, int tx1, int ty1)
* void **operator()** (canvas &out)

Publikus attribútumok

* canvas & **c**
* int **x1**
* int **y1**
* int **x2**
* int **y2**
* int **x3**
* int **y3**

Ez a dokumentáció a struktúráról a következő fájl alapján készült:
* graphics.hpp

3.27. StepData struktúrareferencia

Publikus tagfüggvények

* **StepData** (int a)

Publikus attribútumok

* int **x**
* int **y**
* int **point**

Ez a dokumentáció a struktúráról a következő fájl alapján készült:

- MinMax.h

3.28. genv::text struktúrareferencia

Publikus tagfüggvények

- **text** (const std::string &s)
- **text** (char c)
- void **operator()** (canvas &out)

Publikus attribútumok

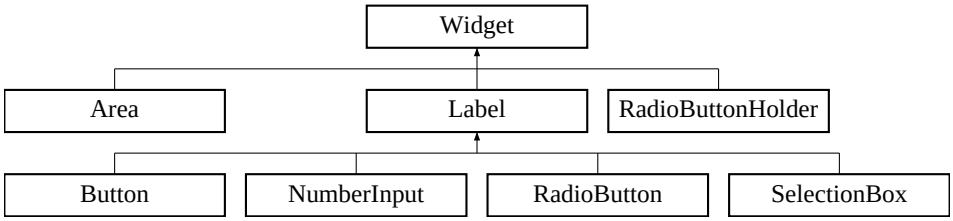
- std::string **str**

Ez a dokumentáció a struktúráról a következő fájl alapján készült:

- graphics.hpp

3.29. Widget osztályreferencia

A Widget osztály származási diagramja:



Publikus tagfüggvények

- Widget (int x, int y, int xSize, int ySize)
- virtual void Draw ()=0
- virtual void Handle (genv::event ev)=0
- virtual bool IsInLine (int x, int y)=0
- void SetBorderThickness (int thickness)
- void SetBackgroundColour (int r, int g, int b)
- void SetFrontColour (int r, int g, int b)
- void **SetEnable** (bool value)
- void **SetVisible** (bool value)

Védett attribútumok

- int X
- int Y
- int XSize
- int YSize
- Colour * bgColor
- Colour * frontColor
- int borderThickness
- bool Selected
- bool **IsEnabled**
- bool **IsVisible**

3.29.1. Konstruktorok és destruktorok dokumentációja

3.29.1.1. Widget()

```
Widget::Widget (
            int x,
            int y,
            int xSize,
            int ySize )
```

Egy j widgetet hoz le.

Paraméterek

x	A widget X poz
y	A widget Y poz
xSize	A widget hossza
ySize	A widget magass

3.29.2. Tagfüggvények dokumentációja

3.29.2.1. Draw()

```
virtual void Widget::Draw ( )  [pure virtual]
```

Ez felel a widget kirajzol
Megvalósítják a következők: Area, RadioButtonHolder, RadioButton, Label, NumberInput, SelectionBox és Button.

3.29.2.2. Handle()

```
virtual void Widget::Handle (
            genv::event ev )  [pure virtual]
```

Ez a fggv kezeli az eventeket

Paraméterek

ev	Az aktus event objektum

Megvalósítják a következők: Area, RadioButtonHolder, RadioButton, Button, Label, NumberInput és SelectionBox.

3.29.2.3. IsInLine()

```
virtual bool Widget::IsInLine (
            int x,
            int y )  [pure virtual]
```

Megadja, hogy az eg sajeretein bell van-e

Paraméterek

x	Az eg poz
y	Az eg poz

Megvalósítják a következők: Area, RadioButtonHolder, RadioButton, Button, Label, NumberInput és SelectionBox.

3.29.2.4. SetBackgroundColour()

```
void Widget::SetBackgroundColour (
            int r,
            int g,
            int b )
```

ttja a widget hzt

Paraméterek

r	Az j piros
g	Az j zld
b	Az j krt

3.29.2.5. SetBorderThickness()

```
void Widget::SetBorderThickness (
            int thickness )
```

ttja a margtagst

Paraméterek

thickness	Az j vastags

3.29.2.6. SetFrontColour()

```
void Widget::SetFrontColour (
            int r,
            int g,
            int b )
```

ttja a widget eltzt

Paraméterek

r	Az j piros
g	Az j zld
b	Az j krt

3.29.3. Adattagok dokumentációja

3.29.3.1. bgColor

```
Colour* Widget::bgColor  [protected]
```

A hz

3.29.3.2. borderThickness

```
int Widget::borderThickness  [protected]
```

A margtags

3.29.3.3. frontColor

```
Colour* Widget::frontColor  [protected]
```

Az eltz

3.29.3.4. Selected

```
bool Widget::Selected  [protected]
```

Ki van-e vsztva a widget

3.29.3.5. X

```
int Widget::X  [protected]
```

A widget X koordinja

3.29.3.6. XSize

```
int Widget::XSize  [protected]
```

A widget szss

3.29.3.7. Y

```
int Widget::Y  [protected]
```

A widget Y koordinja

3.29.3.8. YSize

```
int Widget::YSize  [protected]
```

A widget magass

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

* Widget.h
* Widget.cpp