
Subject: How to create a GridCtrl with fixed cell size
Posted by [cbporter](#) on Mon, 19 May 2008 10:07:18 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi!

I need something like a vertical multirow list. So I thought about using a GridCtrl.

I need a fixed cell size and the column count must adjust itself depending on the size of the control. I have attached a test package that shows how I tried to do this. The problem is that after I resize the window, the width of the columns gets lost somehow. Is there a better way to do this?

Thanks,
Raul

File Attachments

1) [GridTest.rar](#), downloaded 430 times

Subject: Re: How to create a GridCtrl with fixed cell size
Posted by [mrjt](#) on Mon, 19 May 2008 10:23:27 GMT
[View Forum Message](#) <> [Reply to Message](#)

Do you mean something like a spreadsheet?

I don't know about GridCtrl, but you can do this in ArrayCtrl. In constructor:

```
a.HeaderObject().Absolute();
for (int i = 0; i < 20; i++)
    a.AddColumn("Test", 100);
```

Technically this doesn't add more columns as it grows, just adds a maximum and expanding the ctrl reveals them.

Subject: Re: How to create a GridCtrl with fixed cell size
Posted by [cbporter](#) on Mon, 19 May 2008 10:32:53 GMT
[View Forum Message](#) <> [Reply to Message](#)

mrjt wrote on Mon, 19 May 2008 13:23 Do you mean something like a spreadsheet?

No, not like a spreadsheet. For example, if at first I have 5 columns, and after resize 6 could fit, then I need the control to add a column and the first row to display the first 6 items, not just the first 5 as before the resize (and the rest of the rows to pick up the change, and also shift their items). I call this a vertical list. Something like the way Toolbar expands and aligns items as you change the size of the window.

Anyway, I need to look into the difference between GridCtrl and ArrayCtrl. Believe it or not, I never needed anything else except ColumnList up until now.

Subject: Re: How to create a GridCtrl with fixed cell size

Posted by [unodgs](#) on Mon, 19 May 2008 10:34:04 GMT

[View Forum Message](#) <> [Reply to Message](#)

I've never adjusted column count in Layout method, maybe there are some problems with proper column width calculation in this special case. I'll check it today evening.

Subject: Re: How to create a GridCtrl with fixed cell size

Posted by [mrjt](#) on Mon, 19 May 2008 10:48:15 GMT

[View Forum Message](#) <> [Reply to Message](#)

Quote:

No, not like a spreadsheet. For example, if at first I have 5 columns, and after resize 6 could fit, then I need the control to add a column and the first row to display the first 6 items, not just the first 5 as before the resize (and the rest of the rows to pick up the change, and also shift their items). I call this a vertical list Smile. Something like the way Toolbar expands and aligns items as you change the size of the window.

Well, I was picturing something like this:

If you don't want to have to add all of column data at the start you could just detect when an unfilled column is being displayed and then fill it. Trying to add columns as the ctrl expands sounds a bit insane to me , and you may get performance issues from all the layout code that gets executed.

File Attachments

1) [arrayctrl.png](#), downloaded 1469 times

Subject: Re: How to create a GridCtrl with fixed cell size

Posted by [cbporter](#) on Mon, 19 May 2008 12:18:07 GMT

[View Forum Message](#) <> [Reply to Message](#)

Thank you very much for the suggestion! I tried something similar and came pretty close to something useful. But there were problems, because columns which were only partially visible needed to be invisible and trailing items also had to be void, without any cell decoration.

So I tried something related to what I said earlier:

Quote:Something like the way Toolbar expands and aligns items as you change the size of the window.

So I went ahead and used a ToolBar! I used a custom style which still requires tweaking, but it looks and behaves like I wanted it to:

I know that some time ago I started using ToolButton as a Button replacement, and now I'm using ToolBar as a grid. I swear, I'm not spending all my time trying to find new and perverse uses for widgets !

File Attachments

1) [ToolBar.png](#), downloaded 1412 times

Subject: Re: How to create a GridCtrl with fixed cell size

Posted by [cbporter](#) on Mon, 19 May 2008 14:07:59 GMT

[View Forum Message](#) <> [Reply to Message](#)

This was a fun little experiment with ToolBar. It seems that I have pushed this widget to its limits. Besides the performance problems, there seem to be problems with the maximum number of items. I guess nobody tested ToolBar with over 13000 buttons in it.

So it's back to serious business now. Does anybody think that either ArrayCtrl or GridCtrl could be made to look like the picture in my previous post (without any change in their implementation)? If yes, I think the effort to reuse these classes is worth it. If not, then I need to implement a custom control. This is going to take some time because I don't like half-baked custom controls, so the result should be quite customizable, being able to handle tight layouts like in the picture and more loose and nested ones like in this example of the gnome Control Center:

Subject: Re: How to create a GridCtrl with fixed cell size

Posted by [unodgs](#) on Mon, 19 May 2008 15:23:24 GMT

[View Forum Message](#) <> [Reply to Message](#)

Please change your code to:

```
grd.Ready(false);
grd.Clear(true);
for (int i = 0; i < w; i++)
    grd.AddColumn("Test", 105);
grd.Ready(true);
```

Ready(false) tells grid to not refresh until ready flag will be set again. Normally for each addcolumn grid recalculates widths of columns.

Subject: Re: How to create a GridCtrl with fixed cell size

Posted by [unodgs](#) on Mon, 19 May 2008 15:25:37 GMT

[View Forum Message](#) <> [Reply to Message](#)

Gnome configuration panel if possible with GridCtrl. For group header you can join columns in row:

```
grid.AddRow("Look and Feel").JoinRow();
```

Subject: Re: How to create a GridCtrl with fixed cell size
Posted by [cbporter](#) on Mon, 19 May 2008 16:04:52 GMT

[View Forum Message](#) <> [Reply to Message](#)

Thanks for the help! GridCtrl is a complex and feature rich control (even if a little daunting with it's shear amount of members), and after some experimenting I think I'll manage with it. Anyway congratulations for the nice design of the class. Using this I'm sure to save a few days work of producing a quality list control.

I have some questions still:

1. How do you hide the header?
2. How can I make it select only one cell on click and use left & right arrow keys for navigation?

PS:

Quote:grid.AddRow("Look and Feel").JoinRow();

That doesn't seem to compile. Never mind though, I used JoinRow directly on the grid and it worked.

Subject: Re: How to create a GridCtrl with fixed cell size
Posted by [mrjt](#) on Mon, 19 May 2008 16:09:33 GMT

[View Forum Message](#) <> [Reply to Message](#)

It seems to me that what you need is very similar to ColumnList, with only 2 (main) differences:
- items drawn left to right (not top to bottom)
- number of columns determined dynamically based on item width
so my suggestion would to modify that.

I've attached my quick and dirty attempt (it works quite well though). Use like a ColumnList, but you'll have to set ItemWidth to get sensible results when using a Display (default width is 50).

File Attachments

1) [RowList.zip](#), downloaded 382 times

Subject: Re: How to create a GridCtrl with fixed cell size
Posted by [unodgs](#) on Mon, 19 May 2008 17:56:32 GMT

[View Forum Message](#) <> [Reply to Message](#)

cbporter wrote on Mon, 19 May 2008 12:04Thanks for the help! GridCtrl is a complex and feature rich control (even if a little daunting with it's shear amount of members), and after some experimenting I think I'll manage with it. Anyway congratulations for the nice design of the class. Using this I'm sure to save a few days work of producing a quality list control.

Thanks. Of course partially class design is borrowed from ArrayCtrl. I think the main reason it's not widely used is lack of documentation. Fortunately this is gonna change soon.

Quote:

I have some questions still:

1. How do you hide the header?
2. How can I make it select only one cell on click and use left & right arrow keys for navigation?

1. Use HideRow(0) or Header(false) which I've just added In both cases please use the newest version of gridctrl as I did some fixes needed to make your case work properly.

2. SelectRow(false);

Quote:

That doesn't seem to compile. Never mind though, I used JoinRow directly on the grid and it worked.

Yes, you're right. It should be grd.Add(..).JoinRow. AddRow returns reference to ItemRect so you are able to write something like this: grid.AddRow(..).Bg(Yellow).SetFont(Arial(7)) etc.

File Attachments

1) [GridCtrl.7z](#), downloaded 424 times

Subject: Re: How to create a GridCtrl with fixed cell size

Posted by [cbporter](#) on Tue, 20 May 2008 00:23:04 GMT

[View Forum Message](#) <> [Reply to Message](#)

mrjt wrote on Mon, 19 May 2008 19:09It seems to me that what you need is very similar to ColumnList, with only 2 (main) differences:

- items drawn left to right (not top to bottom)
- number of columns determined dynamically based on item width

so my suggestion would to modify that.

I've attached my quick and dirty attempt (it works quite well though). Use like a ColumnList, but you'll have to set ItemWidth to get sensible results when using a Display (default width is 50).

Thank you very much for using your free time to try this. You are right: I do need something like a ColumnList: actually I was using that before, but I need another draw order. It seems that the way such a list with multiple columns is drawn is confusing for a lot of people, so I had to disable the multiple column option, leaving it to a default of one. But because of the waste of space, I find myself in this situation...

I'll definitely try out you code and probably merge it into ColumnList if possible, seeing as U++ lacks a traditional list control with an optional "icons" layout (like CListCtrl from MFC).

But before that, since I already started with GridCtrl, I have to clear up some issues first.

I used this pretty ugly code to get the control to look and behave as I want it. By using Add variants I could only get a static number of items on a row, so I had to use Set:

```
void KanjiFlash::PopulateGrid()
{
    if (!curList)
```

```

return;

grdKanji.Clear();
Vector<Kanji *> &v = *curList;
int cx = 0, cy = 0;

int w = grdKanji.GetSize().cx / CellHeight;
int h = v.GetCount() / w;
if (v.GetCount() * 1.0 / w != h)
    h++;

//section 1
//for (int i = 0 ; i < h; i++)
//    grdKanji.AddRow("").AlignBottom().Height(CellHeight);
for (int i = 0; i < v.GetCount(); i++)
{
    const Kanji& kanji = *v[i];
    grdKanji.Set(cy, cx, kanji.Literal());
    cx++;
    if (cx >= w)
    {
        cx = 0;
        cy++;
    }
}

//section 2
//for (int i = 0 ; i < h; i++)
//    grdKanji.GetRow(i).AlignBottom().Height(CellHeight);
}

void KanjiFlash::Layout()
{
    int w = grdKanji.GetSize().cx / CellHeight;

    grdKanji.Ready(false);
    grdKanji.Clear(true);
    for (int i = 0; i < w; i++)
        grdKanji.AddColumn("", CellHeight);
    grdKanji.Ready(true);
    PopulateGrid();
}

```

This pretty much works, except some problems.

There are two sections of commented out code. I need to uncomment the first section if I want to adjust the look of each individual row. Here I get two problems:

1. The control doesn't seem to show the scrollbar correctly, and even if it shows it, sometimes the

last row is scrolled only about to the middle. It seems that the Height statement is the culprit (CellHeight is 40 under my tests).

2. AlignBottom or any other align doesn't seem to work.

If I uncomment the second section (it doesn't matter if the first is commented or not), rows are drawn at incorrect positions, but react to mouse at correct coordinates.

Also on a side note, I couldn't find a GetRowCount, event though a GetColumnCount is present.

Subject: Re: How to create a GridCtrl with fixed cell size

Posted by [unodgs](#) on Tue, 20 May 2008 06:19:43 GMT

[View Forum Message](#) <> [Reply to Message](#)

Quote:

1. The control doesn't seem to show the scrollbar correctly, and even if it shows it, sometimes the last row is scrolled only about to the middle. It seems that the Height statement is the culprit (CellHeight is 40 under my tests).

2. AlignBottom or any other align doesn't seem to work.

If I uncomment the second section (it doesn't matter if the first is commented or not), rows are drawn at incorrect positions, but react to mouse at correct coordinates.

Ok, I'll check it (AlignBottom should work - maybe I broke sth..)

Quote:

Also on a side note, I couldn't find a GetRowCount, event though a GetColumnCount is present.

simply use GetCount(). Maybe I'll add GetRowCount alias too.

Subject: Re: How to create a GridCtrl with fixed cell size

Posted by [unodgs](#) on Tue, 20 May 2008 07:07:26 GMT

[View Forum Message](#) <> [Reply to Message](#)

Please do following corrections to your code:

in Layout()

```
for (int i = 0; i < w; i++)
    grd.AddColumn("").AlignBottom();
```

and in PopulateGrid()

```
//section 1
for (int i = 0 ; i < h; i++)
    grd.AddRow(1, CellHeight);
```

Now everything should work. Height() is a only syntax sugar. It doesn't cause grid refresh, rows heights recalulation. However it can be confusing. I'll think about it. Align* for particular row doesn't work now. I'll fix that in the next release.

Subject: Re: How to create a GridCtrl with fixed cell size

Posted by [mrjt](#) on Tue, 20 May 2008 08:49:01 GMT

[View Forum Message](#) <> [Reply to Message](#)

Quote:I do need something like a ColumnList: actually I was using that before, but I need another draw order. It seems that the way such a list with multiple columns is drawn is confusing for a lot of people, so I had to disable the multiple column option

I have exactly the same issue, I hate the way ColumnList works. My main issues:

- Lack of key/value support (every other list-style control supports it, why not ColumnList?)
- Bizarre scrolling. A columnized list should scroll sideways to show new columns, not the vertically one item at a time. I think this is the one that really confuses people.
- Difficult selection determination

Quote:I'll definitely try out your code and probably merge it into ColumnList if possible, seeing as U++ lacks a traditional list control with an optional "icons" layout (like CListCtrl from MFC).

I would be surprised you could do this, the changes are too widespread and complete to be integrated as an addition feature without adding considerable bloat and complexity.

Subject: Re: How to create a GridCtrl with fixed cell size

Posted by [unodgs](#) on Tue, 20 May 2008 09:13:51 GMT

[View Forum Message](#) <> [Reply to Message](#)

mrjt wrote on Tue, 20 May 2008 04:49

I have exactly the same issue, I hate the way ColumnList works. My main issues:

- Lack of key/value support (every other list-style control supports it, why not ColumnList?)
- Bizarre scrolling. A columnized list should scroll sideways to show new columns, not the vertically one item at a time. I think this is the one that really confuses people.
- Difficult selection determination

I like scrolling in column list. It doesn't scroll columns but items in visible columns. This is a very good solution for long list (at least for me). You can adjust columns count by changing the width of just one column (I mean here it's easy to make more items visible without resizing the widget). Of course similar possibility could be implemented in classical approach.

What you need is arrayctrl/gridctrl which adjust columns count automatically for given list count. It can be easily done. However grid/array ctrl are heavy controls. I would suggest creating a new control (RowList e.g.).

Subject: Re: How to create a GridCtrl with fixed cell size

Posted by [cbporter](#) on Tue, 20 May 2008 12:18:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

Well I don't like the scrolling but I definitely appreciate how useful it is for list of a lot of items and I've gotten used to it. But if the user is unable to scroll and desperately tries to find an item on a different column only after a small scroll, that's a problem. Anyway, I'm OK with it in general, and in this case I'll use grid.

mrjt wrote on Tue, 20 May 2008 11:49

I would be surprised you could do this, the changes are too widespread and complete to be integrated as an addition feature without adding considerable bloat and complexity.

I did some diff's and the bloat wouldn't be that great, but the complexity would definately take a hard hit. ColumnList is nice and light weight, and since it is called ColumnList, maybe it's not the best idea to merge the two? Then how about some testing, debugging if necessary, and maybe adding key/value support and including it in Bazaar?

I have one last question regarding GridCtrl: what does Item::Clickable do? It does not seem to do what I imagined. And can I set a cell so that it is not clickable/selectable. I need this only for tail cells.

Subject: Re: How to create a GridCtrl with fixed cell size

Posted by [mrjt](#) on Tue, 20 May 2008 12:28:15 GMT

[View Forum Message](#) <> [Reply to Message](#)

cbporter wrote on Tue, 20 May 2008 13:18I did some diff's and the bloat wouldn't be that great, but the complexity would definately take a hard hit. ColumnList is nice and light weight, and since it is called ColumnList, maybe it's not the best idea to merge the two? Then how about some testing, debugging if necessary, and maybe adding key/value support and including it in Bazaar? I agree. I've already started work on a list that does both row-ordering and column-ordering with (IMO) better scrolling. Like 'List View' and 'Icon View' in Explorer. It might take a little while as I'm developing it in the spare time I have while my major project compiles

Subject: Re: How to create a GridCtrl with fixed cell size

Posted by [unodgs](#) on Tue, 20 May 2008 13:33:59 GMT

[View Forum Message](#) <> [Reply to Message](#)

Quote:I have one last question regarding GridCtrl: what does Item::Clickable do? It does not seem to do what I imagined. And can I set a cell so that it is not clickable/selectable. I need this only for tail cells.

Clicakble means cell is clickable or not, i.e cell can or can't be selected. Could you post here your whole package so I could test it?

Subject: Re: How to create a GridCtrl with fixed cell size

Posted by [cbporter](#) on Tue, 20 May 2008 13:46:17 GMT

[View Forum Message](#) <> [Reply to Message](#)

In this little test case, the main diagonal should not be clickable. Am I using the API wrong?

File Attachments

1) [GridTest.rar](#), downloaded 392 times

Subject: Re: How to create a GridCtrl with fixed cell size

Posted by [unodgs](#) on Tue, 20 May 2008 15:43:31 GMT

[View Forum Message](#) <> [Reply to Message](#)

No, you're doing fine. GridCtrl tested only if a row or a column is clickable not a particular cell. I have fixed this. Use attached sources. If you find some other problems, do not hesitate to write about them. I will fix them all if possible.

File Attachments

1) [GridCtrl.7z](#), downloaded 347 times

Subject: Re: How to create a GridCtrl with fixed cell size

Posted by [cbporter](#) on Wed, 21 May 2008 13:48:17 GMT

[View Forum Message](#) <> [Reply to Message](#)

Thanks for the fix!

Now I need to determine the current row & column. I found a Get.CurrentRow (but no Get.CurrentColumn), but the value which it returned seems to be off (and even constant in my case). There is also Get.Cursor, which returns (-1, -1)? And which is the event for cell change?

And is there an easier way to do key/value stuff than using a Value and GridDisplay?

Subject: Re: How to create a GridCtrl with fixed cell size

Posted by [unodgs](#) on Wed, 21 May 2008 13:58:48 GMT

[View Forum Message](#) <> [Reply to Message](#)

cbporter wrote on Wed, 21 May 2008 09:48

Now I need to determine the current row & column. I found a Get.CurrentRow (but no Get.CurrentColumn), but the value which it returned seems to be off (and even constant in my case). There is also Get.Cursor, which returns (-1, -1)? And which is the event for cell change?

Get.CursorPos return Point with currently pointed cell otherwise it returns Point(-1, -1). It should work. If not please give me some more information (calling context) or attach a test case.

Quote:

And is there an easier way to do key/value stuff than using a Value and GridDisplay?

What you mean exactly? What do you want to achieve?

Subject: Re: How to create a GridCtrl with fixed cell size
Posted by [cbporter](#) on Wed, 21 May 2008 14:13:14 GMT

[View Forum Message](#) <> [Reply to Message](#)

GetCursorPos worked. I just had to change event from WhenCursor to WhenChangeRow & WhenChangeCol. With WhenCursor it was always (-1,-1), even though this event triggered on cell change.

Quote:What you mean exactly? What do you want to achieve?

Like in most other cases, I need to insert some data in the container widget, but I need it to display some text. From what I could tell from the interface of GridCtrl, is that you insert some arbitrary type converted to a Value, and write a custom Display to display a string while the control holds you values.

Subject: Re: How to create a GridCtrl with fixed cell size
Posted by [unodgs](#) on Wed, 21 May 2008 15:45:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

Quote:

GetCursorPos worked. I just had to change event from WhenCursor to WhenChangeRow & WhenChangeCol. With WhenCursor it was always (-1,-1), even though this event triggered on cell change.

Starange, you should get correct values in WhenCursor. I'll try to check it.

Quote:

Like in most other cases, I need to insert some data in the container widget, but I need it to display some text. From what I could tell from the interface of GridCtrl, is that you insert some arbitrary type converted to a Value, and write a custom Display to display a string while the control holds you values.

Ok, now I understand. One cell holds one Value object. If you want key/value pair conected with one cell you have to use some structure as Value. As for displaying. You have to extend GridDisplay class which contain some useful methods like SetLeft/Right/CenterImage.

You could also do somehting like this:

```
grd.AddIndex(); this column will hold keys (and it will be invisible)  
grd.AddColumn(); this column will hold values  
grd.AddIndex()  
grd.AddColumn
```

In GridDisplay you can access via parent member neighbours' values. You can also read current position via col, row members.
Hope that will help you somehow

Subject: Re: How to create a GridCtrl with fixed cell size
Posted by [cbporter](#) on Thu, 22 May 2008 09:50:17 GMT

[View Forum Message](#) <> [Reply to Message](#)

[quote title=unodgs wrote on Wed, 21 May 2008 18:45]Quote:

Ok, now I understand. One cell holds one Value object. If you want key/value pair connected with one cell you have to use some structure as Value. As for displaying. You have to extend GridDisplay class which contain some useful methods like SetLeft/Right/CenterImage.

Yes, I know that. This was not a question of how, it was a question of principle. I was asking why isn't there and easier method of doing it. The scenario I described is a fairly common one IMO. It's great that you can use a Display and do really fancy stuff if you need it, but in most cases I just want to display a simple text extracted from the Value in exactly the same position and with same style as if I had inserted a string directly. In such cases, using Display is a little overkill, and writing them can be repetitive and tedious.

Let's take for example a Paint method from Dispaly:

```
virtual void Paint(Draw& w, const Rect& r, const Value& q,  
                   Color ink, Color paper, dword style) const
```

It has 6 parameters, and all have useful predefined values which you'll rarely ignore or change and go ahead and use them. A large portion of these displays consists of a simple w.DrawText and maybe some position calculation. Since U++ is in general good at detecting common idioms and offering some API, mechanism or clever trick for it, I was wondering why it doesn't offer something like that in this case. For example, something like:

```
struct MyDisplay: TextDisplay  
{  
    virtual String Text(const Value &q) const;  
}
```

Subject: Re: How to create a GridCtrl with fixed cell size
Posted by [unodgs](#) on Thu, 22 May 2008 10:10:04 GMT

[View Forum Message](#) <> [Reply to Message](#)

You can use converters. For example:

```
struct SexConv : Convert  
{  
    Value Format(const Value& q) const  
    {  
        switch(int(q))  
        {
```

```
case SEX_MALE: return "Male";
case SEX_FEMALE: return "Female";
case SEX_CHILD: return "Child";
}
return Null;
}
};

grid.AddColumn("Key column").SetConvert(Single<SexConverter>());
grid.Add(SEX_MALE).Add(SEX_FEMALE);
```

Subject: Re: How to create a GridCtrl with fixed cell size
Posted by [cbporter](#) on Thu, 22 May 2008 13:03:10 GMT

[View Forum Message](#) <> [Reply to Message](#)

Great! I never thought about using a Converter for this. And since String is ref counted (and my strings are also short), it should be fairly quick to extract a field from the value this way.

I've done some code clean-up and am very happy with the result. I do miss the ColumnList look a little, so I'm going to add an option to switch between a ColumnList output and a GridCtrl one, with grid being default .

I still have 2 questions not 100% related to GridCtrl:

1. I noticed that displaying a large number of CJK characters is pretty slow. I tried replacing all the CJK characters with English ones, and the control was blazing fast again, so I don't believe it's related to GridCtrl. I wonder if there is a native limitation in the drawing speed of these chars (they are quite a bit more complex than other ones), or is this related to U++. I remember Mirek saying once that after he fixed some bugs regarding the determination of the metrics for these chars, it became slow, so he had to speed it up.

2. Is there an Array like container, but which does not free the memory occupied by its items? I'm using Vector<Foo *> right now.

Subject: Re: How to create a GridCtrl with fixed cell size
Posted by [mrjt](#) on Thu, 22 May 2008 16:38:40 GMT

[View Forum Message](#) <> [Reply to Message](#)

2. ArrayCtrl::SetCtrl() ?

Subject: Re: How to create a GridCtrl with fixed cell size
Posted by [cbporter](#) on Thu, 22 May 2008 19:15:52 GMT

mrjt wrote on Thu, 22 May 2008 19:382. `ArrayCtrl::SetCtrl()` ?

I think you misunderstood. I should have explained more clearly. I was wondering if there is a container which does not destroy its items when it is destroyed, so basically it does not own them. I have a collection of type Foo and need to provide a number of "views" for those items, so I use `Vector<Foo *>`.

Subject: Re: How to create a GridCtrl with fixed cell size

Posted by [mrjt](#) on Fri, 23 May 2008 09:14:13 GMT

cbporter wrote on Thu, 22 May 2008 20:15mrjt wrote on Thu, 22 May 2008 19:382.

`ArrayCtrl::SetCtrl()` ?

I think you misunderstood. I should have explained more clearly. I was wondering if there is a container which does not destroy its items when it is destroyed, so basically it does not own them. I have a collection of type Foo and need to provide a number of "views" for those items, so I use `Vector<Foo *>`.

I did indeed, though the explanation was perfectly clear the first time .

`Vector<Foo *>` seems a reasonable option (although you could use `Vector< Ptr<Foo> >`), but I'm struggling to think of a situation where this would be necessary. Can the view not just use a const reference to the Vector? Or does it need a special subset of items?

Subject: Re: How to create a GridCtrl with fixed cell size

Posted by [cbporter](#) on Fri, 23 May 2008 09:54:55 GMT

mrjt wrote on Fri, 23 May 2008 12:14

`Vector<Foo *>` seems a reasonable option (although you could use `Vector< Ptr<Foo> >`), but I'm struggling to think of a situation where this would be necessary. Can the view not just use a const reference to the Vector? Or does it need a special subset of items?

Well, since you asked, yes I have such a situation. It is rather messy. I have a collection of over 13000 items which are organized in multiple "views". These views are subsets of the original collection, and all have their particular order. Items can and often will be in a lot of views at the same time. The number of total views is not yet determined, but it is over 50. Through an interface a filter can be defined which allows and item or rejects it. This must update all the views, determine some basic statistics (i.e. items accepted vs. items rejected) and must work in "real-time" on a low-end system. Any number of views can be visible at all times (though luckily limited by screen resolution). Since applying the filter on each view is not feasible, I apply it on the main collection and use those vectors for displaying and to compute the statistics.

So there were two solutions.

1. Use a lot of configuration files (possibly XML) and a lot of instance variables and even a lot more enums. Maintenance nightmare.
2. Use 3 classes full of templates and just a small number of constants and generate the sets

programmatically by parametrized classes so I don't end up with with dozens of instance variables.

I chose the second one.

Subject: Re: How to create a GridCtrl with fixed cell size

Posted by [cbporter](#) on Sat, 24 May 2008 17:17:46 GMT

[View Forum Message](#) <> [Reply to Message](#)

I think I've found a bug. What you need to reproduce is:

1. A window with a Layout method which clear the contents of a grid and then populates it with a number of N items.
2. Two Buttons or other event dispatchers. The first will set N to a low number so that the grid doesn't require a scrollbar. The second set N do that the grid would display a scrollbar. Both call Layout after setting N.

You need to click the first button, then the second, then the first again. At the second click on the first button, when repopulating the grid that formerly had a scrollbar and now no longer needs to have one, GetSize returns a different cx for the grid. During the entire operation, the grid remains the same size, so I believe GetSize should not change.

I'm sorry for not providing a test case, but I'm dead tired right now. I'll post one tomorrow.

Subject: Re: How to create a GridCtrl with fixed cell size

Posted by [unodgs](#) on Sat, 24 May 2008 17:32:51 GMT

[View Forum Message](#) <> [Reply to Message](#)

Quote:I think I've found a bug.

Great! I'm waiting for the test case

Subject: Re: How to create a GridCtrl with fixed cell size

Posted by [cbporter](#) on Wed, 28 May 2008 18:01:57 GMT

[View Forum Message](#) <> [Reply to Message](#)

Sorry that it took so long to post the test case, but since this bug was so easy to circumvent, it was a really low priority task to get it fixed and I was busy with other work that had to be done .

File Attachments

- 1) [GridTest.rar](#), downloaded 376 times

Subject: Re: How to create a GridCtrl with fixed cell size

Posted by [cbporter](#) on Mon, 02 Jun 2008 09:06:31 GMT

[View Forum Message](#) <> [Reply to Message](#)

A secondary issue appeared in cases when after a layout change which resulted in a scrollbar the user executed another one: the top row would not be updated on scroll.

So I decided to investigate GridCtrl and I found the solution to both problems. After calling Ready(false), a Clear(true) does not update the scrollbar information, so that computations done in layout still have the scrollbar size included, because of the way GetSize investigates the frame count to return the adjusted size.

So the issues can be solved by either swapping the order of the Ready and Clear call, or by calling UpdateScrollbars after the clear.

Subject: Re: How to create a GridCtrl with fixed cell size

Posted by [unodgs](#) on Mon, 09 Jun 2008 08:08:13 GMT

[View Forum Message](#) <> [Reply to Message](#)

cbporter wrote on Mon, 02 June 2008 05:06A secondary issue appeared in cases when after a layout change which resulted in a scrollbar the user executed another one: the top row would not be updated on scroll.

So I decided to investigate GridCtrl and I found the solution to both problems. After calling Ready(false), a Clear(true) does not update the scrollbar information, so that computations done in layout still have the scrollbar size included, because of the way GetSize investigates the frame count to return the adjusted size.

So the issues can be solved by either swapping the order of the Ready and Clear call, or by calling UpdateScrollbars after the clear.

The problem is UpdateSb in Clear is called when grid is in ready state, but I think it can be called always - then you will get proper values in your Layout method. I will commit the change today to uvs.
