

[Main](#) [3D Mee](#) [2D Mee](#) [IM Framework](#) [Desktop 3D](#) [VeryCD](#) [Others](#)

Welcome to **vrcats.com** Projects



What's in the box

In this projects site you will find the latest information of projects planned, developed and managed by vrcats.com. You can also download the most recent packages and distributions of freeware/shareware here.

Following is a list of projects:

- [3D Mee](#): a 3D Figure Editor program and a bone animation library
- [2D Mee](#): a 2D Face/Pose Editor program for creating your own 2D cartoon figures
- [Desktop 3D](#): a collection of small desktop gadgets, most in 3D amazing looks
- [IM Framework](#): an evolutionary communication library, networking is never so easy
- [VeryCD Grabber](#): a time-saving tool for heavy VeryCD.com downloaders
- [CuteQt RSS Reader](#): a generic RSS client written in Qt
- [Kaixin001 Trailer](#): a web spider style trailer for kaixin001.com
- [Kids Compiler](#): an educational computer language for kids to learn programming
- [RTO3S](#): RealTime Object Oriented Operating System for ATMEGA 16/32

All listed above are GPL open source projects. You can download the source code from:

<http://code.google.com/u/vrcats>

Please check out GPL license guideline if you have any questions about the licenses.



3D Mee



Introduction

3D Mee Editor is an enhanced version of the 2D Mee Editor. It provides a set of cross-platform tool to create 3D figures from models, and comes with a complete set of classes and libraries to use them in Qt/C++ projects.

Motivations

It is not an easy task to use 3D models in an application. To do this the developers are forced to do a lot of massive work. First they must be familiar with OpenGL/DirectX, then they must be able to design a 3D figure and related animations in an expensive tool (3DMax/Maya, or something else), and then write a complex loader to load and display the models. They are also required to deal with the complex animations, conflict detection, physics calculation etc. In simple phase, to make a 3D car, developers have to start from the wheel and axis.

In fact this shouldn't be so complicated. My idea is to provide a set of tools for people to design 3D figures (a set of conjuncted models) and related actions, even a set of well designed figures to the developers. The developer doesn't have to have the knowledge of 3D engines, nor how to load/render the models, they don't even have to design their figures. All they have to do is to put the figure file into the resource directory, and create an instance of the Figure Class.

The word "figure" here refers to a set of models. The models (I call them components) in the figure are connected to each other in a tree structure. The connections are made with "conjunctions". When you design a figure, the components will bend/rotate/move around the conjunctions, like real-world objects.

The figure class provide a very simple interface for the developers to manipulate the figures. They can replace a texture of the figure, or change color/size/position of a part in the figure. Of course they are also entitled to call

actions, which will make the figure to start moving, creating an amazing animation with very low cost or coding. For example, you can load a figure called "Boy", change his cloths, hair color, and wear him a pair of glasses. In your application, you can simply call:

```
boy.action("sneeze");
```

To make him sneeze, given there is a embedded action in the figure called "sneeze"). You can also set how fast the sneeze last, and get notified when the action is finished.



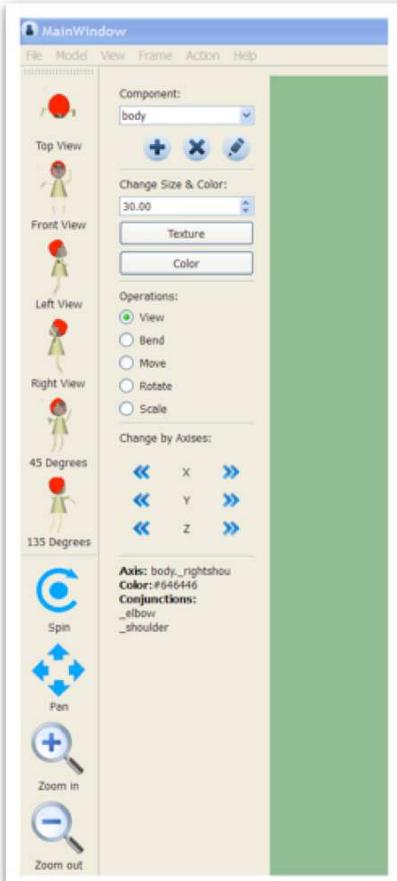
Not Only An Editor

The project starts with a figure editor as shown in the screen shot, but is not limited to an editor.

At the beginning the projects was planned for design human figures, as you can see it is a "Mee Editor". However, virtually you can design a figure for anything. Human beings, yourself, your friends and relatives; Animal, your pets, wild animals, ocean creatures; Building, vehicles, objects, more and more. In the tools we provide a real-world measurement, so you can create them in a unified size and use them without bothering to calibration. Also, with one figure well defined, you can reuse it. For example you have a well designed human figure, you can replace its face texture, cloth, and resize the components to make it a complete new figure. Furthermore, all you have done here will not interfere the actions, that mean you don't need to re-design the animations; they will automatically be applied to the new figure.

This project targets on those applications who wants to use 3D in their interfaces. Apparently simple games will be the first to benefit from this project. However, it is not limited to games. Any application who wants to use 3D in their interface will enjoy the facilities. With a universal figure store set up, application don't even need to store figures to local. They can access the figures online at any time anywhere, ideally for free. Hence, it would be great fun to design a figure for each of your friends and to use them in one's application. This will add strong flavors to those social network enabled applications.

The final scenario is to provide an easy to use library to allow people use 3D figures and bone animations. The figure designer is just a beginning.



Toolbar for view and modify figures

Use cases:

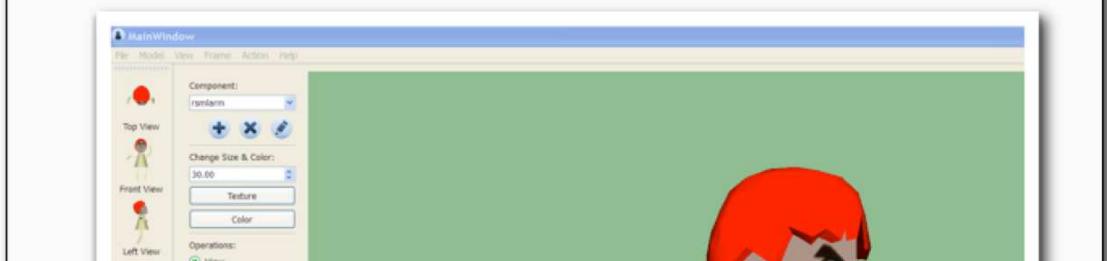
1. Small 3D games: no 3D engine required, light weight, easy to use, all you need to do is to develop the game logics, don't worry about rendering 3D and animations any more, we do that for you.
2. Social network applications: imagine you can use the 3D figures of your friends in your applications, you can see them working, snoozing, you can feel their emotions, not from their twitter message, but by looking at them.
3. Desktop gadgets: it is so easy to add 3D objects to your desktop, you can have a 3D clock, 3D dashboard, 3D weather window on your desktop. There is no reason for 2D gadgets to survive longer.
4. Desktop applications: Add some 3D animation/transitions to your programs, express your intelligence in a fancy way.

Technologies used

- Qt: makes this solution cross platform, and easy to use in any C++ projects
- Lib3DS: this library is used to load .3ds models
- OpenGL: we render all objects using OpenGL, and optimize towards better OpenGL performance. OpenGL also enable the solution to work on all platforms easily

Products

1. 3D Figure assemble tool: to be able to assemble 3DS models into a figure by connecting the conjunctions, also able to import figure textures(particularly faces) from 2D mee editor
2. 3D Figure action editor: to be able to create actions for a figure, and save the actions for future use
3. Interface classes: with these classes user will be able to load and manipulate figures easily



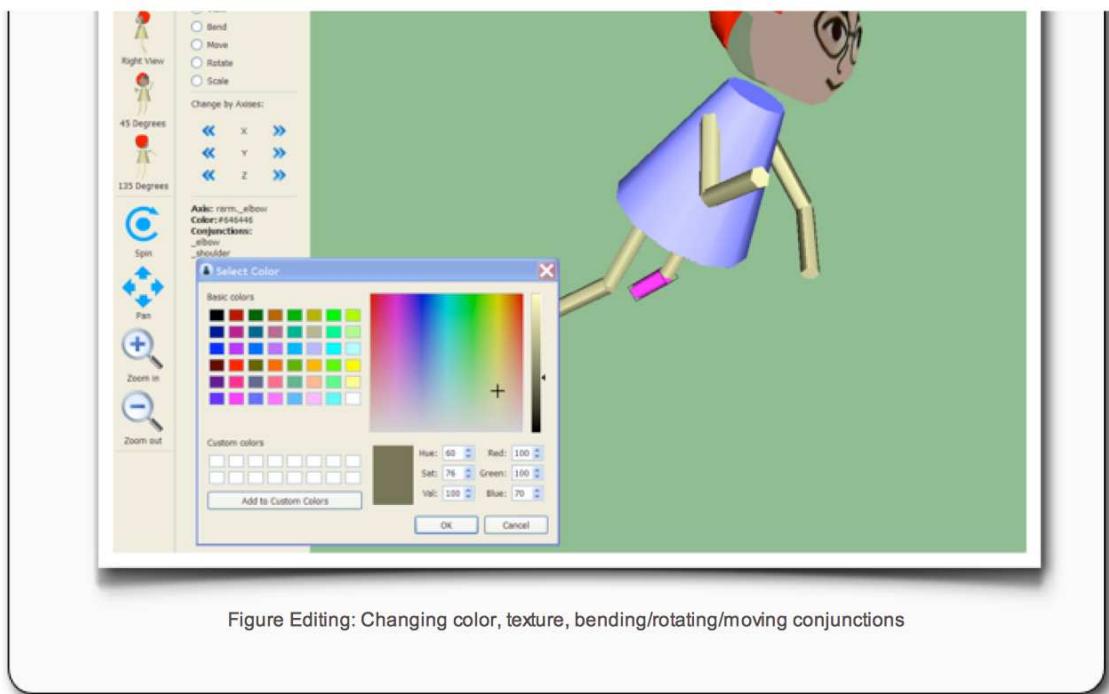


Figure Editing: Changing color, texture, bending/rotating/moving conjunctions

Status

05/01/2011:

- 3D assemble tool coding finished, testing in progress
- Action editor mostly finished, testing coming up
- Interface classes pending

Known issues

- Loading pose on Linux platform will result in segmentation fault

Next steps

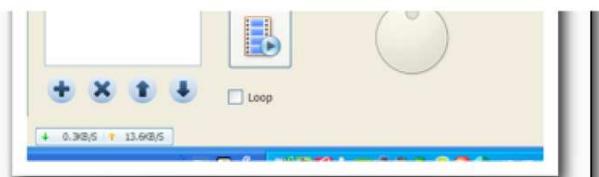
- Finish Action editor
- Perform more system testing on assemble tool
- System testing plan for action editor
- Design a packing protocol for 3D Mee Figures
- Research on best interface to use the figures
- Implement interface classes
- Performance tuning, protocol refining
- Test with more models and applications
- Optimize the assemble UI, provide more reference guidelines
- Optimize the action design workflow
- Write a users manual for editor
- Generate documentation for interface classes

More concepts

- Make a 3D figure standard for applications
- Support ANSI C
- Implement Java/C# interfaces
- Develop figure sharing protocols
- Build up an online figure store
- Provide more tools eg. World designer/loader, embedded physics engine

The screenshot shows the Action editor interface. It has two main panes. The left pane lists 'Poses' from Pose0 to Pose17. The right pane shows a 'Key Frames' section with a dropdown menu set to 'Stand'. Below the dropdown are buttons for '+' and 'X'. At the bottom, there are controls for 'Delay' (set to 0.75), 'Curve' (set to 'Raise'), and a 'Preview' button.

- Make one stop solution for 3D applications



Action Editing Area



vrcats.com All Rights Reserved. 2005-2011

2D Mee



Introduction

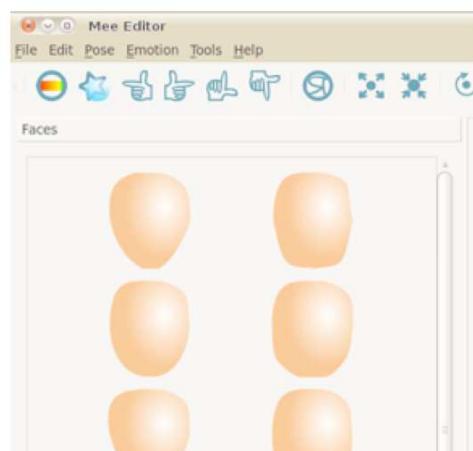
2D Mee Editor is a PC port of Nitendo Mii Editor. It is compatible to the Nitendo Wii Figure format, and produces reusable picture formats for all computer systems, eg. SVG and PNG.

Motivations

One of the most interesting parts of Nitendo Wii is the Mii Editor. With it you can design a virtual 2D figure for you and your friends/relatives, and use them in the games and applications.

Unfortunately on PC we don't yet have a similar tool. So this project aims at creating a tool to enable PC users to create/design 2D human figures with assembling components. The design can be exported as vector or raster image, to meet the requirements of programs.

On PC, 2D figures can be widely used. A simplest case is to use the figure in VCards, for email contacts etc. 2D



figures can also be used in communication programs, games, and whatever applications that can display a human figure or human face.

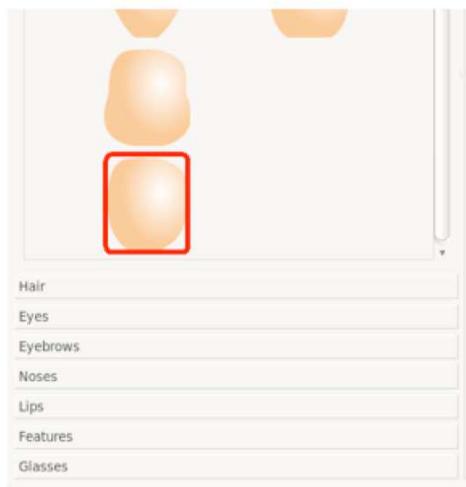
2D figures are not static. You can design different emotions, poses for them, and assemble them into animations. Therefore you will be able to use not only pictures, but also animated pictures in the application, which makes it much more fun.

We also plan to build up a online 2D Figure Store, where you can upload the figures you designed, and make them usable by others, with charge or for free. You can also limit the use to your buddies by connecting it to the social network framework, so that only your friends can make use of your figures.

- Connect the designer to 3D Mee Editor
- Provide more design assistance
- Build a online figure store
- Connect the figure system to IMFramework

More concepts

- Build a component store
- Build a figure based online live website
- Integrate video recognition technology to emotion/pose design



Face Shape Selection Tool

Use cases:

1. Messaging applications: replace your friends icon with their comics picture, watch their emotions and actions while you chat to them
2. VCards: use the comics picture instead of low quality photos, people can remember you easily
3. 2D games: invite your friends to the games, no more playing with NPC, play with your friends, or NPC who look like your friends, it's much more fun

Technologies used

- Qt: makes this tool cross platform
- SVG: high quality vector based pictures

Products

1. 2D Face Editing Tool
2. 2D Emotion/Pose Editing Tool
3. Wrapper class for easy use of emotions and actions

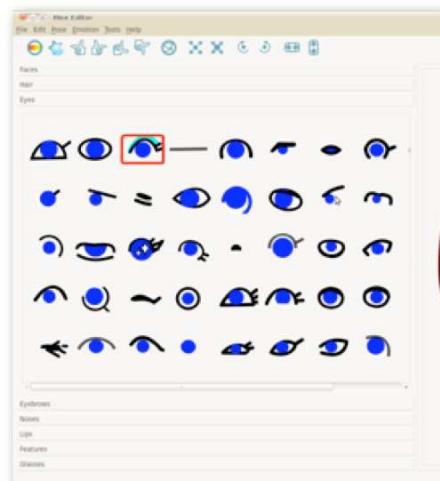
Status

05/01/2011:

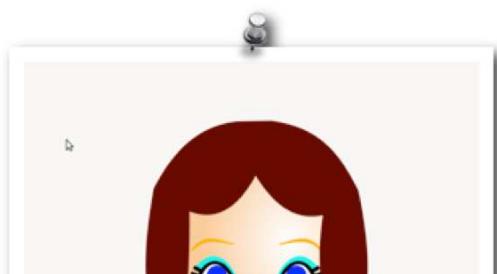
- 2D face editing tool done
- Face editing testing done
- All resources design done
- Emotion/pose tool in progress
- Emotion/pose tool testing pending
- Wrapper class pending

Known issues

- Exported SVG not rendering properly in Firefox and Opera, but usable in SVG editors



Eye Selection Page

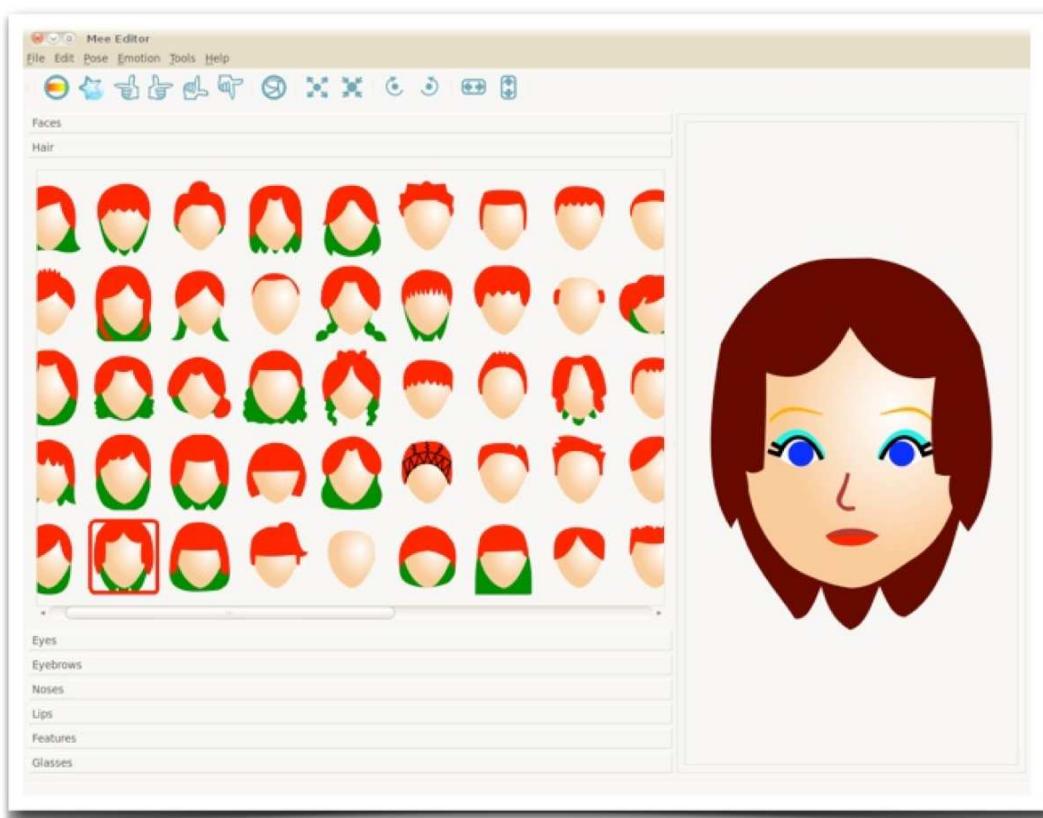




A Girl Face Designed by 2D Editor

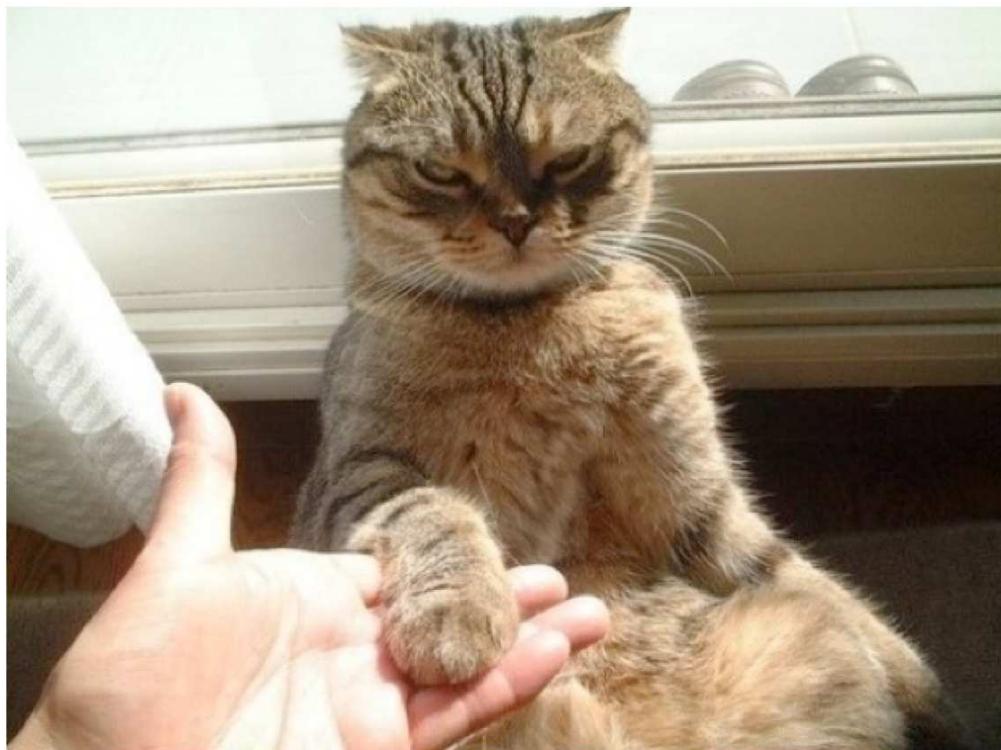
Next steps

- Finish the emotion/pose editor
- Finish the wrapper classes
- Write users' manual and documentation
- Design more components



Run Time Screen Shot

IM Framework



Introduction

In this project we plan to provide a unified end to end communication interface. With the interface the developer is able to find the other party, and communicate to it with minimal complexity.

Motivations

You may have developed many TCP/IP network applications. Where did you spend most of your time? To me the following things have bothered me for long:

- To find the other party's IP address and Port?
- To avoid Port conflicts?
- To process the nasty socket data streams, wait, loop, wait...
- To interpret the data into comprehensible messages

We should really have a library that can do above for us. Ideally the library should be able to get the presence of your desired destination, build up a communication tunnel between you and the destination, and shoot messages/data two ways.

Moreover, there are some even better ways to communicate. In most applications, we don't really want send/receive messages, all we want is just to keep tuned with a remote object. Of course you can synchronize by communications, wouldn't it be nice if the library already provide the synchronization?

Another example is to send a SQL query and get results. To interpret this process into communications, or say messages, is not a trivial task. There should be a library to do this repeating boring task.

In this IM Framework project we provide a library/server with the following facilities to the developers:

1. Presence APIs: able to query who is online, what is the best way to contact him/her
2. Messaging APIs: able to send/receive text messages or binary blobs to a remote application
3. Synchronizable base class: inherit this class, then your class will have the ability to synchronize with a remote instance of the compatible class types.
4. Query APIs: send a query to the other end, and get notified when the results are ready. You can even send the query to multiple servers, and get one combined result. To operate the result, you don't have to get it in whole; you can select the most interested part to retrieve.

The reason that this project is called IM Framework is that we used Instant Messaging protocols as our low level backends at the beginning. However as it evolves, the backend scope has expanded. We can use ftp, http, facebook, other IM protocols, and whatever open APIs to do the presence and communication. That means with this framework you can talk to any open system if there is a plugin available.



Use cases:

IM Framework can be used anywhere that needs network and communication. Almost any network application can benefit from the super easy use of IM Framework

1. Instant Messaging: IMF provides a unified interface for communication, it is never so easy to make a messenger that supports multiple protocols
2. Realtime network applications: For example realtime slide show, with IMF the slide can be played in synchronization among multiple users, and users can make comments at realtime.
3. Web Services: IMF brings the possibility to describe web service in a complete new way. You don't need a server at all, just show your contacts that you have the ability to provide this service, and they will be able to use your service instantly. You can even freely define your own ways to provide the service, for example in natural languages, but not complicated SOAP with considerable programming and performance overhead.
4. P2P: the nature of IMF is peer to peer communication. Therefore it has great potential to do P2P communication/sharing. The advantage is that we don't need a server at all, IMF will take care of the negotiation and most mass, so that the developers can concentrate on the data management logics and interfaces.

Technologies used

- Qt: makes this solution cross platform, and easy to use in any C++ projects
- Some opensource protocol implementations, including QXmpp, etc.
- OpenSSL and other crypto algorithm libraries

Products

1. A IM Framework server executable for each platform
2. Library with classe facilities to use IM Framework
3. Some backend plugins
4. Useful account management tool
5. Detailed documentation for using IMF and developing backend plugins
6. A set of demo applications that show how easy it is to use IMF

Status

05/01/2011:

- IM Framework mostly finished, missing inter-process communication part
- IMF testing in progress
- Interface classes defined, mostly finished
- MSN, XMPP, IRC backends implemented, testing in progress
- Demo application implemented, tests pending

Known issues

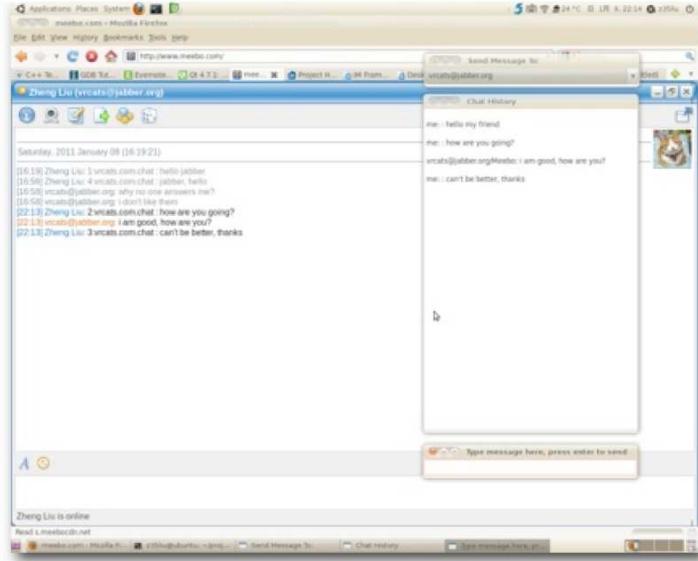
- Due to MSN protocol upgrade, MSN backend is not working and needs upgrade

Next steps

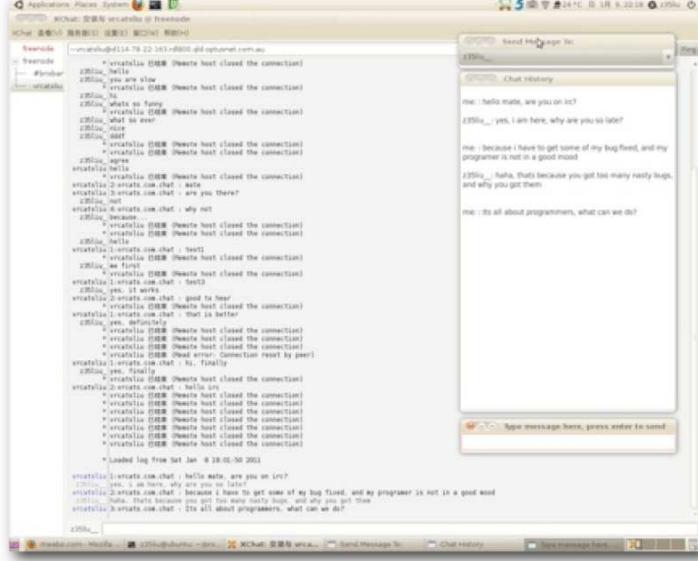
- Finish and polish the server
- Test on multiple platforms
- Review the interface classes and finalize
- Design the account management tool and encryption methods
- Test demo applications
- Propose more unit tests and system tests
- Performance tuning, protocol refining

Example applications

We plan to make some example applications to demonstrate how easy and powerful IMF is. So far one of them is completed called messenger. It is a multi-protocol messenger using IMF. Here are the screenshots:



messenger talking to a jabber client via google talk account



messenger talking to an irc client via irc.freenode.net server

Can you Believe...

How many lines of code do you need to achieve such a multi-protocol messenger? 1000? 3000? No, carefully zoom in the following image and count the lines, it is all listed here, the answer is 82.

Although we wasted a lot of lines in include and interface arrangement, it is still amazing, isn't it? Imagine you can use the best communication channels with a couple of lines of code (not only Instant Messaging channels, but also include other protocols we plan to involve), you can spend your efforts on polishing the interface and implementing the brilliant ideas, leave all the dirty works to us!

There are another two examples being developed. One of them is a small round game to demonstrate how to synchronize between two remote objects, or among remote objects. The other one demonstrates how to do a SQL query through IM Framework, and get exactly what you want by refining the results on the server side.

IMF is bringing the most exciting revolutions to network programming, get yourself involved now! If you are interested of how IMF is designed, please read: [IMF Design Document](#)

```

File Edit View Terminal Help
z3Silu@ubuntu: ~/projects/imframeworkservice/adapter/examples/msg
1 #include "messenger.h"
2
3 int main(int argc, char** argv)
4 {
5     QCoreApplication app(argc, argv);
6     VRIM vr;
7     vr.run();
8     return app.exec();
9 }
10
11 // return app.exec(); //vrcatslu CIMR (Reete here)
12 // vrcatslu what is ever
13 // vrcatslu hello
14 // vrcatslu what
15 // vrcatslu what is funny
16 // vrcatslu what ever
17 // vrcatslu CIMR (Reete here)
18 // vrcatslu what
19 // vrcatslu what is funny
20 // vrcatslu what ever
21 // vrcatslu hello
22 // vrcatslu what
23 // vrcatslu what is funny
24 // vrcatslu what ever
25 // vrcatslu hello
26 // vrcatslu what
27 // vrcatslu what is funny
28 // vrcatslu what ever
29 // vrcatslu hello
30 // vrcatslu what
31 // vrcatslu what is funny
32 // vrcatslu what ever
33 // vrcatslu hello
34 // vrcatslu what
35 // vrcatslu what is funny
36 // vrcatslu what ever
37 // vrcatslu hello
38 // vrcatslu what
39 // vrcatslu what is funny
40 // vrcatslu what ever
41 // vrcatslu hello
42 // vrcatslu what
43 // vrcatslu what is funny
44 // vrcatslu what ever
45 // vrcatslu hello
46 // vrcatslu what
47 // vrcatslu what is funny
48 // vrcatslu what ever
49 // vrcatslu hello
50 // vrcatslu what
51 // vrcatslu what is funny
52 // vrcatslu what ever
53 // vrcatslu hello
54 // vrcatslu what
55 // vrcatslu what is funny
56 // vrcatslu what ever
57 // vrcatslu hello
58 // vrcatslu what
59 // vrcatslu what is funny
60 // vrcatslu what ever
61 // vrcatslu hello
62 // vrcatslu what
63 // vrcatslu what is funny
64 // vrcatslu what ever
65 // vrcatslu hello
66 // vrcatslu what
67 // vrcatslu what is funny
68 // vrcatslu what ever
69 // vrcatslu hello
70 // vrcatslu what
71 // vrcatslu what is funny
72 // vrcatslu what ever
73 // vrcatslu hello
74 // vrcatslu what
75 // vrcatslu what is funny
76 // vrcatslu what ever
77 // vrcatslu hello
78 // vrcatslu what
79 // vrcatslu what is funny
80 // vrcatslu what ever
81 // vrcatslu hello
82 // vrcatslu what
83 // vrcatslu what is funny
84 // vrcatslu what ever
85 // vrcatslu hello
86 // vrcatslu what
87 // vrcatslu what is funny
88 // vrcatslu what ever
89 // vrcatslu hello
90 // vrcatslu what
91 // vrcatslu what is funny
92 // vrcatslu what ever
93 // vrcatslu hello
94 // vrcatslu what
95 // vrcatslu what is funny
96 // vrcatslu what ever
97 // vrcatslu hello
98 // vrcatslu what
99 // vrcatslu what is funny
100 // vrcatslu what ever
101 // vrcatslu hello
102 // vrcatslu what
103 // vrcatslu what is funny
104 // vrcatslu what ever
105 // vrcatslu hello
106 // vrcatslu what
107 // vrcatslu what is funny
108 // vrcatslu what ever
109 // vrcatslu hello
110 // vrcatslu what
111 // vrcatslu what is funny
112 // vrcatslu what ever
113 // vrcatslu hello
114 // vrcatslu what
115 // vrcatslu what is funny
116 // vrcatslu what ever
117 // vrcatslu hello
118 // vrcatslu what
119 // vrcatslu what is funny
120 // vrcatslu what ever
121 // vrcatslu hello
122 // vrcatslu what
123 // vrcatslu what is funny
124 // vrcatslu what ever
125 // vrcatslu hello
126 // vrcatslu what
127 // vrcatslu what is funny
128 // vrcatslu what ever
129 // vrcatslu hello
130 // vrcatslu what
131 // vrcatslu what is funny
132 // vrcatslu what ever
133 // vrcatslu hello
134 // vrcatslu what
135 // vrcatslu what is funny
136 // vrcatslu what ever
137 // vrcatslu hello
138 // vrcatslu what
139 // vrcatslu what is funny
140 // vrcatslu what ever
141 // vrcatslu hello
142 // vrcatslu what
143 // vrcatslu what is funny
144 // vrcatslu what ever
145 // vrcatslu hello
146 // vrcatslu what
147 // vrcatslu what is funny
148 // vrcatslu what ever
149 // vrcatslu hello
150 // vrcatslu what
151 // vrcatslu what is funny
152 // vrcatslu what ever
153 // vrcatslu hello
154 // vrcatslu what
155 // vrcatslu what is funny
156 // vrcatslu what ever
157 // vrcatslu hello
158 // vrcatslu what
159 // vrcatslu what is funny
160 // vrcatslu what ever
161 // vrcatslu hello
162 // vrcatslu what
163 // vrcatslu what is funny
164 // vrcatslu what ever
165 // vrcatslu hello
166 // vrcatslu what
167 // vrcatslu what is funny
168 // vrcatslu what ever
169 // vrcatslu hello
170 // vrcatslu what
171 // vrcatslu what is funny
172 // vrcatslu what ever
173 // vrcatslu hello
174 // vrcatslu what
175 // vrcatslu what is funny
176 // vrcatslu what ever
177 // vrcatslu hello
178 // vrcatslu what
179 // vrcatslu what is funny
180 // vrcatslu what ever
181 // vrcatslu hello
182 // vrcatslu what
183 // vrcatslu what is funny
184 // vrcatslu what ever
185 // vrcatslu hello
186 // vrcatslu what
187 // vrcatslu what is funny
188 // vrcatslu what ever
189 // vrcatslu hello
190 // vrcatslu what
191 // vrcatslu what is funny
192 // vrcatslu what ever
193 // vrcatslu hello
194 // vrcatslu what
195 // vrcatslu what is funny
196 // vrcatslu what ever
197 // vrcatslu hello
198 // vrcatslu what
199 // vrcatslu what is funny
200 // vrcatslu what ever
201 // vrcatslu hello
202 // vrcatslu what
203 // vrcatslu what is funny
204 // vrcatslu what ever
205 // vrcatslu hello
206 // vrcatslu what
207 // vrcatslu what is funny
208 // vrcatslu what ever
209 // vrcatslu hello
210 // vrcatslu what
211 // vrcatslu what is funny
212 // vrcatslu what ever
213 // vrcatslu hello
214 // vrcatslu what
215 // vrcatslu what is funny
216 // vrcatslu what ever
217 // vrcatslu hello
218 // vrcatslu what
219 // vrcatslu what is funny
220 // vrcatslu what ever
221 // vrcatslu hello
222 // vrcatslu what
223 // vrcatslu what is funny
224 // vrcatslu what ever
225 // vrcatslu hello
226 // vrcatslu what
227 // vrcatslu what is funny
228 // vrcatslu what ever
229 // vrcatslu hello
230 // vrcatslu what
231 // vrcatslu what is funny
232 // vrcatslu what ever
233 // vrcatslu hello
234 // vrcatslu what
235 // vrcatslu what is funny
236 // vrcatslu what ever
237 // vrcatslu hello
238 // vrcatslu what
239 // vrcatslu what is funny
240 // vrcatslu what ever
241 // vrcatslu hello
242 // vrcatslu what
243 // vrcatslu what is funny
244 // vrcatslu what ever
245 // vrcatslu hello
246 // vrcatslu what
247 // vrcatslu what is funny
248 // vrcatslu what ever
249 // vrcatslu hello
250 // vrcatslu what
251 // vrcatslu what is funny
252 // vrcatslu what ever
253 // vrcatslu hello
254 // vrcatslu what
255 // vrcatslu what is funny
256 // vrcatslu what ever
257 // vrcatslu hello
258 // vrcatslu what
259 // vrcatslu what is funny
260 // vrcatslu what ever
261 // vrcatslu hello
262 // vrcatslu what
263 // vrcatslu what is funny
264 // vrcatslu what ever
265 // vrcatslu hello
266 // vrcatslu what
267 // vrcatslu what is funny
268 // vrcatslu what ever
269 // vrcatslu hello
270 // vrcatslu what
271 // vrcatslu what is funny
272 // vrcatslu what ever
273 // vrcatslu hello
274 // vrcatslu what
275 // vrcatslu what is funny
276 // vrcatslu what ever
277 // vrcatslu hello
278 // vrcatslu what
279 // vrcatslu what is funny
280 // vrcatslu what ever
281 // vrcatslu hello
282 // vrcatslu what
283 // vrcatslu what is funny
284 // vrcatslu what ever
285 // vrcatslu hello
286 // vrcatslu what
287 // vrcatslu what is funny
288 // vrcatslu what ever
289 // vrcatslu hello
290 // vrcatslu what
291 // vrcatslu what is funny
292 // vrcatslu what ever
293 // vrcatslu hello
294 // vrcatslu what
295 // vrcatslu what is funny
296 // vrcatslu what ever
297 // vrcatslu hello
298 // vrcatslu what
299 // vrcatslu what is funny
300 // vrcatslu what ever
301
main.cpp 33.3 messenger.h 30.1 messenger.cpp (*) 30.0.1

```

vrcats.com All Rights Reserved. 2005-2011

Desktop 3D



Introduction

This project is initialized to create a set of desktop gadgets. In the long term, we aim to develop a complete set of useful 3D applications, together with a windowing system designed for 3D windows, to replace the existing 2D desktop system.

Motivations

Using 2D applications is tedious. The development of 3D hardware and software libraries support has enabled desktop to run complex 3D applications without unaffordable resource consumption. It is time to start developing applications with 3D interfaces.

3D applications can provide user significant different interface. At this time, using a complete 3D interface is not practical; therefore in the desktop gadgets we use a mixture of 2D and 3D. Wherever possible, fancy 3D interfaces will be used. For those UI that displays a large number of text and pictures, traditional 2D interfaces are kept.

The project targets on desktop users on Window, OSX, and Linux. However it wouldn't be difficult to port them to mobile devices given the device has OpenGL ES support and a good CPU.

Technologies used

- Qt: makes this application set available on all popular platforms
- 3D Mee Editor: to design nice 3D figures and objects that used in the applications
- OpenGL: to achieve best performance while remaining cross platform feature
- IM Framework: add network features and enable social network functions
- Qt Mobility: some function support like PIM etc.

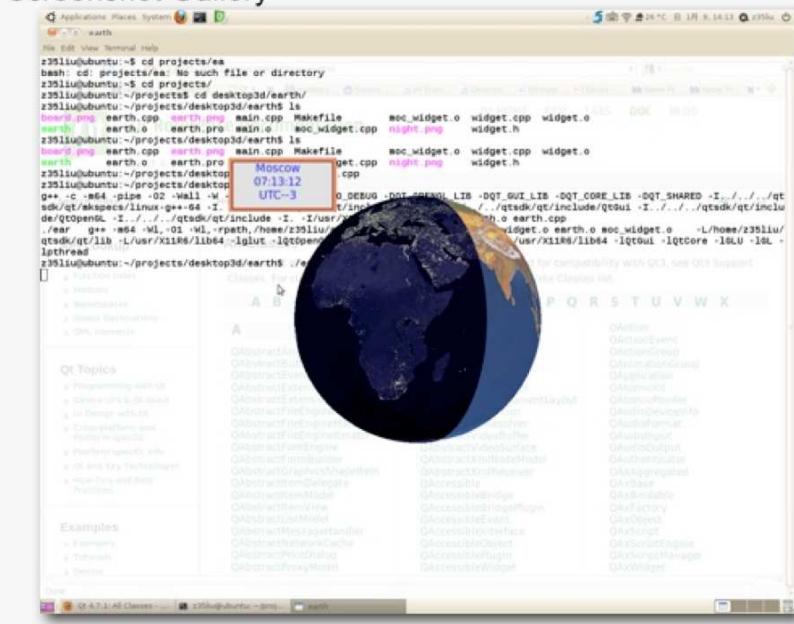
Products

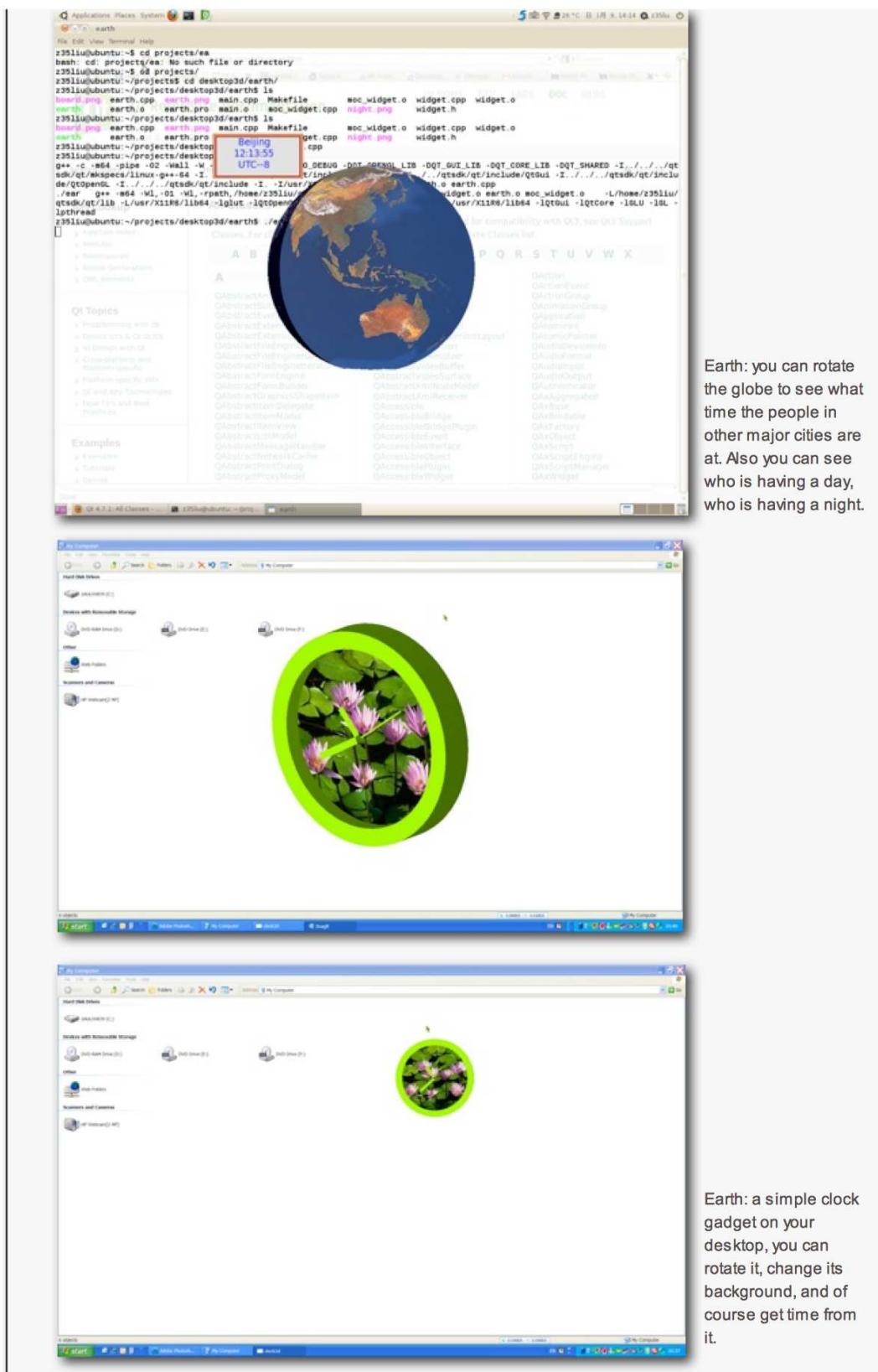
1. A set of 3D desktop gadgets to conduct daily tasks
2. Documentation and user manuals
3. A unified development guideline and compatibility guide

Sub-projects

1. 3D Clock: a simple round analog clock. You can rotate it and place it anywhere on your desktop, a good tool to get time everyday
2. 3D Weather Window: a real wood window where you can find weather report and forecast on the glass. Requires http connectivity to retrieve real-time weather information.
3. 3D Fishing Pond: a port of Nitendo Wii Play Fishing game to PC, the only difference is that it is in real 3D, not a flash game. Enjoy pond fishing when you are tired with massive work.
4. Earth: a rotating globe that can show you the time and time zones of major cities in the world
5. More applications have been proposed, see Appendix.

Screenshot Gallery







Weather Window: a small gadget that you can get the world weather information, what can you see in the window?



Fishing Pond: a fishing game ported from the Wii console. You can catch every fish you see here.

Status

05/01/2011:

- 3D clock finished, test done
- 3D weather window finished, test done
- 3D Fishing pond 75% finished, fishing part to be implemented, test pending
- More applications planned, see appendix
- Documentations and manuals pending

Known issues

- Transparent window border doesn't work well on Gnome Windowing System.

Next steps

- Finish Fishing pond and test it
- Work on more applications
- Integrate IM Framework when ready
- Write documentations and manuals
- Create a motion input device using OpenCV

More concepts

- Implement a windowing system to suite 3D windows
- Fully integrate IM Framework
- Port to mobiles where possible
- Expand motion input device family
- Port to intelligence appliance like TV, DVD, PVR.
- Build up a 3D application store/farm, encourage more developers to work on 3D applications

Appendix: 88 upcoming gadget ideas:

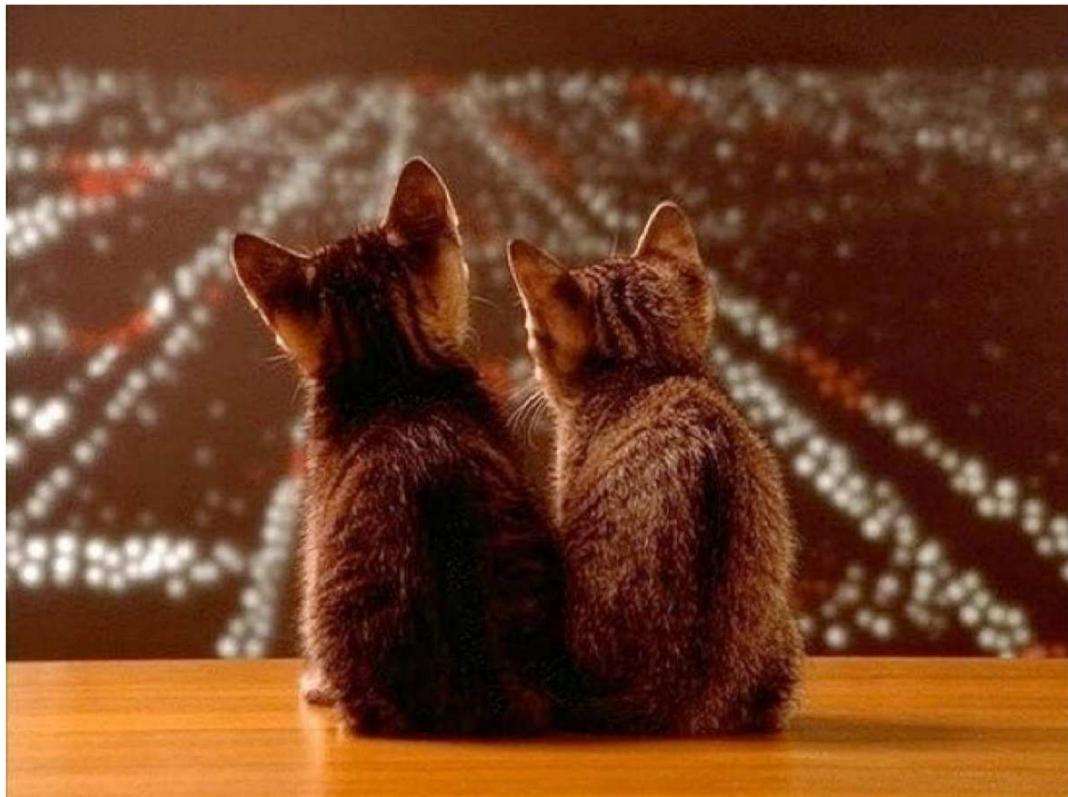
1. Movie Camera: show top movies and ratings
2. Musical Note: music gallery and player
3. Book: personal book collections, online library

4. Alarm Clock: alarm clock, connects to online calendar service
5. Brick Wall: instant news and push news, news comments
6. Album: photo gallery and slide player, connects to major online photo storage
7. Postbox: email client
8. Desklight: share your articles with people interested
9. Fish Tank: electrical fishtank, can talk to your friends fish tank
10. Forest: display carbon footprint and world environment status
11. Globe: world time and date, moon phase
12. Butterfly: a visual emotion twitter, use the butterfly pose to display your emotions
13. Flags: put on a flag when you support someone's opinion, and share your support to friends
14. House: type your friends name, you can see what he is doing in the house
15. Farm: plant something here, you can harvest later
16. Train: a collection of travel photos/blogs of yours and others
17. Projector: a networked slide show application
18. Lego: using lego to arrange your daily tasks, and finish them
19. PostIt: note book, connects to your online notebook
20. Calculator: calculator that you can do simple programmings on
21. TV: top videos and watches, selected video feeds
22. CPU: display system usage and temperatures, suggest computer usage tips
23. Water Tap: show how much download/upload you are doing, now and total
24. Junkbox: something will jump out to tell a joke
25. OSX Appbar: type a program name, find it instantly
26. Hanging Sock: birthday/anniversary reminder, and you can send a gift to friend's sock
27. Messenger Boy: unified chatting client
28. Twinkle Star: entertainment news and picture feeds
29. Map Roll: online map and positions of interest
30. Rock Scissor Paper: a tool that enable you to make decisions by doing rock scissor paper to a friend
31. Doreamon: a knowledge base robot that can answer any questions you have
32. Tape: to record and send a short voice message to someone, either local or online
33. Mickey: tells a bedtime story every day
34. Net: a collection of most used web sites
35. Antic Telephone: PIM contacts, connects to your online contact books
36. Comic Books: comics feed and comics download
37. A Bag of Instant Noodle: when you are angry, crash it and listen to the sound
38. Frypan with Egg: tasty menus and recepies
39. Flying Pig: some incredible news and stories
40. Fruit and Vegis: help you to plan your diet and health food tips
41. Smiling Face: pulling most embarrassing stories from everyone in the space
42. Tumbler: make a tumbler of your favourite person on your desktop
43. Air Pump: if you don't like someone, ask his Mee to apologize to you
44. Beating Heart: send a heartbeat to your friends, or send a heart bomb to someone you love
45. Double Happiness: tell others about your lucks and happiness, share the best
46. Redcross: health tips and services, self health evaluation
47. Reversed Fook: write down your best experiences and feelings, and share them to yourself when you are not feeling so well
48. Laughing Buddha: psychological doctor
49. Fan: display a poem each day
50. Rainbow: type in your idea at the moment, see who have the same idea, and where they are
51. Balloon: inflate it until it explodes
52. Bubbles: use your microphone to make bubbles and let them fly on your desktop
53. Feather: write a message and let it fly to someone randomly
54. Firework: record an important moment, play some fireworks on your desktop now
55. Porcelain: share your antic collections to friends
56. Lighthouse: one piece of proverbs everyday

- 57.Floating Bottle: start a topic in the bottle, everyone who see the bottle can comment on it
- 58.Chess: chess game online
- 59.Game of Go: the game of go online
- 60.Snowing Crystal Ball: fortunetelling, constellations
- 61.Toilet: a game to aim your pees to the center of toilet
- 62.School: a place to discuss and copy homework, and talk to your classmates
- 63.Soccer: soccer news, time table, world cup time count down
- 64.Bascketball: basketball news, time table, live, profiles
- 65.Olympic Five Rings: sport news and predictions
- 66.Photo Camera: news and discussions for backpackers, scenenary photos from best travel destiny
- 67.Vote Box: start a vote or vote for an interesting topic
- 68.Speaker: news comments and debates
- 69.Game Console: video game emulators
- 70.Wallet: suggest best bargains around you, forward suggestions to your friends
- 71.Tai Chi: tell you how to make your life longer
- 72.Feeding Bottle: tips on how to take care of babies and kids
- 73.Pet Bowl and Bone: pet tips and photo/blog shareing
- 74.An Atom: scenetific news and articles
- 75.Monoplay Man: monoplay game online
- 76.A Wave Cylinder: record and analysis your voice, recognize people by voice, voice command
- 77.Nipple Dummy: an expert system that can help you to find out why your baby is crying
- 78.TV Station: tv programs, tv video on demand
- 79.Saving Pot: plan your finance here, and record expenses
- 80.Eyeball: this program tracks how you use your computer, and suggest the resting time for you
- 81.Magic Mirror: surprise you with a handsome/beautiful person reflected in the mirror
- 82.Colorful Cube: move in front of your web cam, the cube will change its color accordingly
- 83.Fortune Cat: share your friends of opportunities and chances
- 84.Russian Matryoshka: this is a endless matryoshka with fortune cards
- 85.Color Changing Tassel: an advertisement wall
- 86.Nautilus: featured designs show
- 87.Red Lantern: holiday and festival count down, chinese calendar
- 88.Rose: send a bunch of rose online to your love



VeryCD Grabber



Introduction

Mining for emule links in [verycd.com](#) is painful since the website page is complex and slow. We developer this web spider program to download and classify information, facilitating users to select best emule links and download them in batch.

Motivations

VeryCD.com is a leading Chinese Emule resource sharing website. Since it is heavily loaded, it is a pain to load a page and copy download links on the page manually. It will be nice if there is a tool that summarizes the download pages and open those pages to retrieve download links automatically.

VeryCD Grabber is such a tool.

Screenshots

The screenshot shows a window titled "MainWindow" with a toolbar at the top containing "Page", "Selection", and "Help" buttons, along with standard window controls. Below the toolbar are five navigation buttons: "Next", "Previous", "Page Range", "Show Cart", and "Clear". A vertical sidebar on the left lists file paths for each item in the main table. The main content is a table with the following columns: Title, Files, Size, Category, Link, Language, Year, Created, Modified, and Publisher. The table contains approximately 200 rows of file entries, mostly PDFs, with various titles such as "《黄飞鸿传作工字诀武学》扫描版[PDF]", "《金庸作品集·三联标准版·明报连载版+新修版，更新《神雕侠侣》《射雕英雄...》，《鹿鼎记》等》扫描版[PDF]", and "《赵门秘武(张家太极拳)》扫描版[PDF]". Most files are in Chinese (Simplified or Traditional), have a size between 1 MB and 20 MB, and were created between 2010 and 2011.

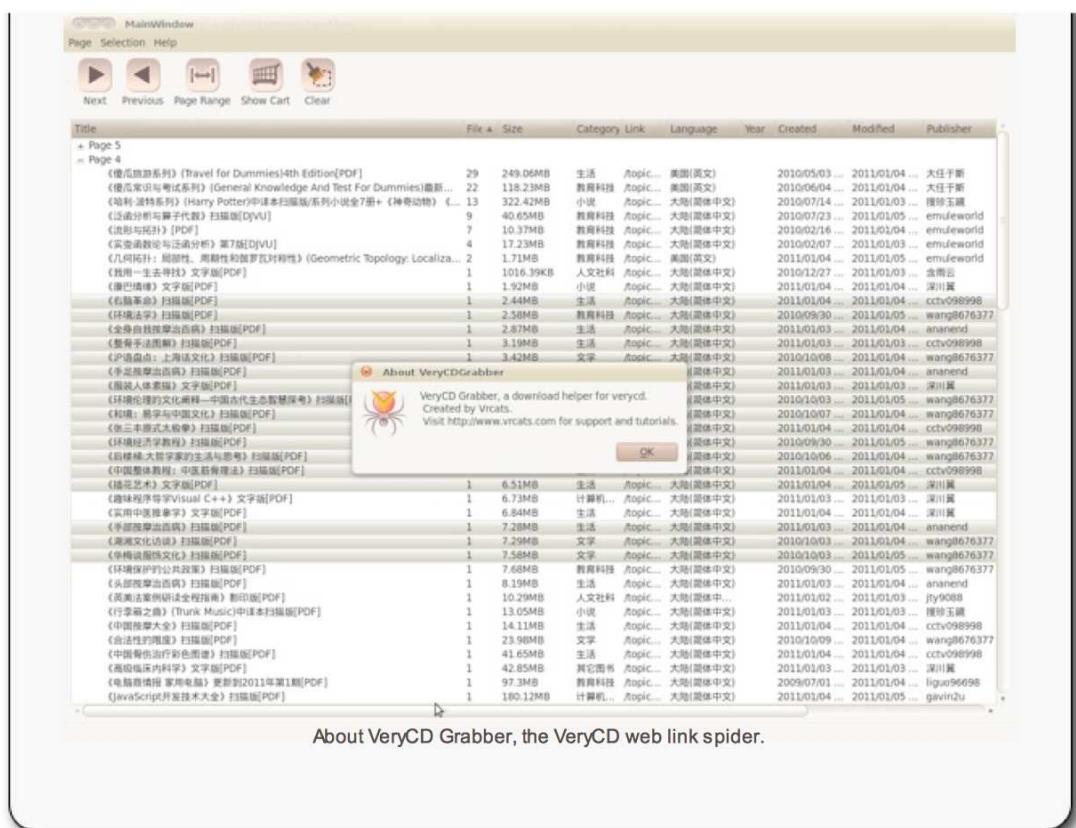
It gets feeds from vercd.com automatically. You can also indicate the source if required.

This screenshot shows a similar interface to the first one, but with a different set of files. The table columns are identical: Title, Files, Size, Category, Link, Language, Year, Created, Modified, and Publisher. It displays approximately 180 items, many of which are EPUB files (e.g., "《乔布斯的魔法演讲》(The Presentation Secrets of Steve Jobs)文字版[EPUB]"), while others are PDFs like "《皮特教授治癌：滴滴与走失》文字版[PDF]" and "《现代诗歌论：诗人与批评》文字版[PDF]". The list includes a mix of English and Chinese titles, with sizes ranging from 1.19MB to 27.46MB.

Sort by size, and select the file you want to download.

Now sort by files, to double check the files are correctly selected

Add selected files to the cart, it will open each page and retrieve all the links. Now you have a second chance to audit the files to download. Previous downloaded files are remembered and displayed in gray, so they won't be downloaded again.



About VeryCD Grabber, the VeryCD web link spider.

Testimonials

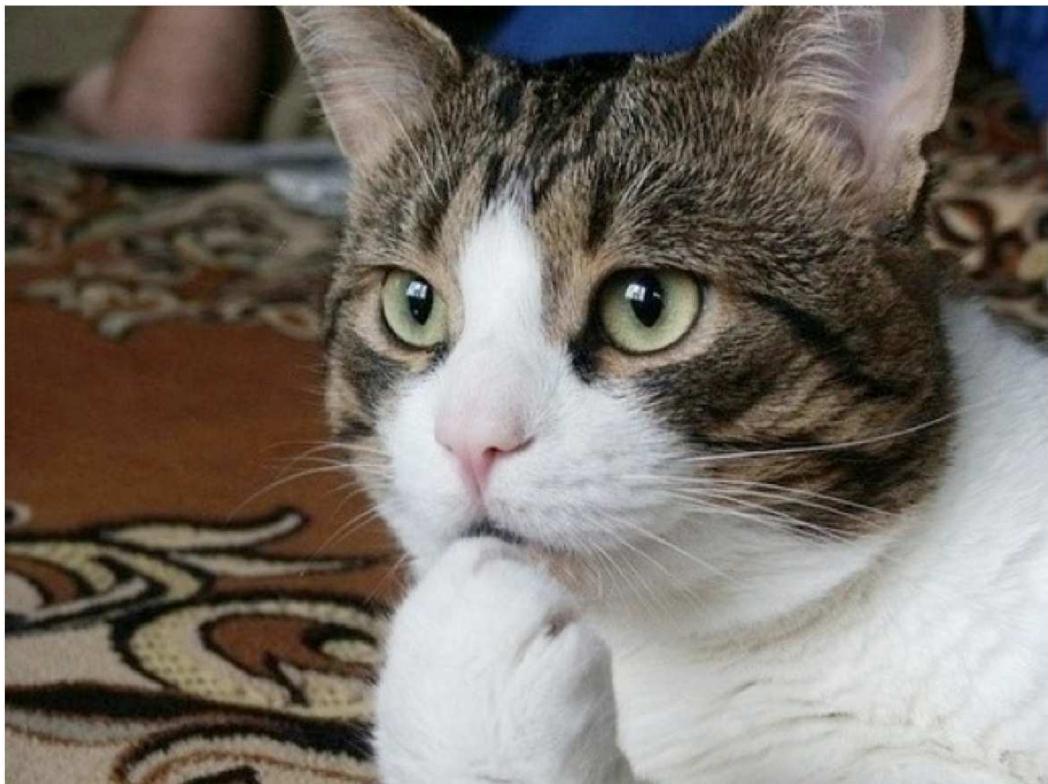
I am a heavy Emule downloader. Most of my download links are retrieved from VeryCD.com. Since I download a lot of e-books, it takes me at least 5-6 hours per week to collect the emule links manually: open the site, find the latest resources, open a page for each resource, find the link, copy them to my download tool.

With VeryCD Grabber, I only do this once a month now. Open the application, wait a few seconds for the main page to get loaded, now select the things you want to download, the previous downloaded ones are automatically filtered out. It retrieves every link for you, quickly and precisely. I only spend less than one hour per MONTH on collecting the links, which is 30-50 times faster than before.

Thanks VeryCD Grabber, it saves time, saves life.

Main 3D Mee 2D Mee IM Framework Desktop 3D VeryCD **Others**

Other Projects by vrcats.com



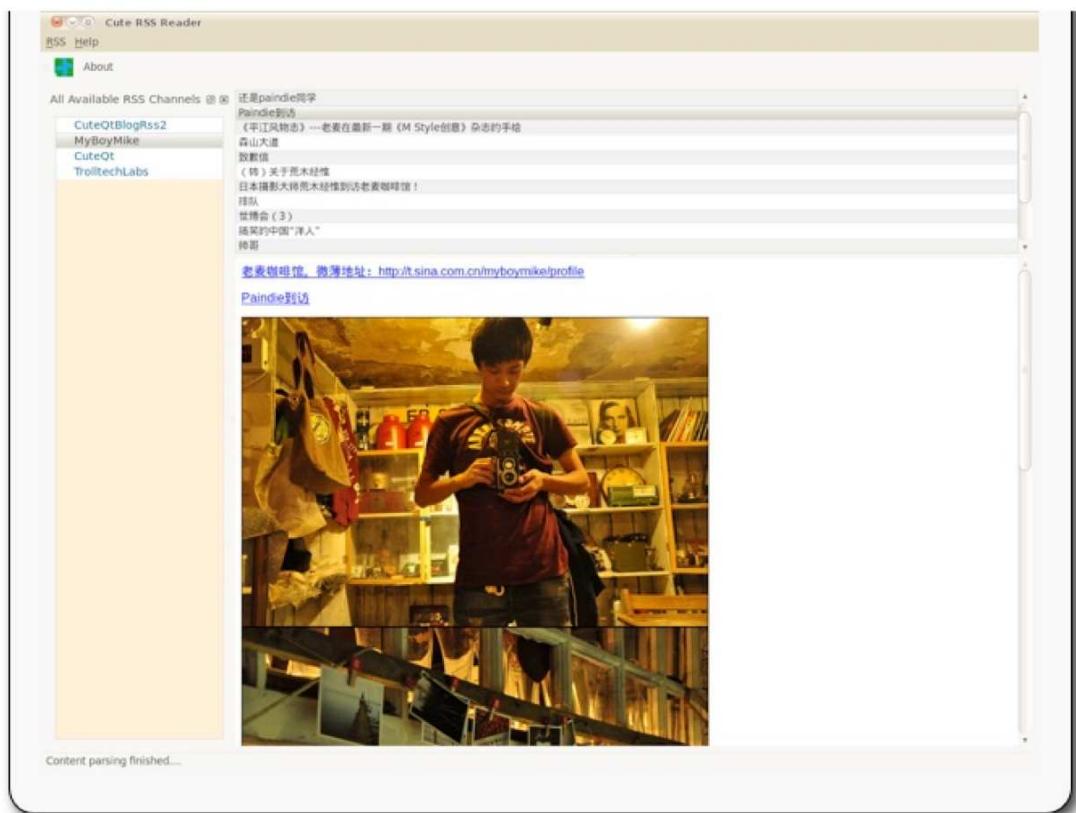
There are some many more projects done in vrcats.com, or being done by vrcats.com.

Discontinued Projects:

CuteQt RSS Reader

A generic RSS client written in Qt.

Although this project is not own by vrcats.com, vrcats.com has contributed some core parts to the project. It works nicely as yet another RSS client program.



Kaixin Trailer

Kaixin001.com is a popular SNS website. Many people spent a lot of time farming on the virtual farm, someone even got heart attacks by the hard farm work day and night.

This project produces a reliable trailer that can replace you to monitor your farm. It does everything automatically, all you need to do is to have a computer running it 24 hours. It is a cross platform console program, tested on Windows, Linux, Mac OSX. This project came online early 2008.

For copyright problems we decided not to develop and maintain this project any longer. However it is always available for educational and research purposes.

Proposed Projects:

Kids Compiler

As computer language evolves, it is getting more and more complicated. It is difficult to find a entry level language for those who wants to learn programming, particularly kids.

We plan to develop a simple compiler/interpreter with a minimal C++ like syntax. It is simplified as much as possible, while keeping the essential things. The syntax will be designed to be easily grasped, and used, and the kids can see the results on the screen and hear from the speaker immediately.

The project will be based on Qt to provide cross platform distributions. Hence, high performance graphical libraries

will be used to present fabulous effects to attract kids creating their own projects.

It consists of the compiler, an IDE, and a library back end providing simple data structures and useful stuff like canvas, 3D space, multimedia APIs, etc. The compiler produces both binary and C++ codes, so the user can build their project on any platform if they like.

Besides kids, it will become an ideal tool to create software prototypes and demos, or a tools to intensify your programming skills.

This project is in proposing stage. Some researches and background preparation are done. Further progress will be reported soon.

RTO3S

RTO3S stands for RealTime Object Oriented Operating System.

There are quite a few realtime operating systems around. However, they are all designed with the knowledge of non-embedded systems. For those very low end MPUs, like ATMEGA 16/32, 51, it is still too large and too complex. More, most of them are written in C not C++, which limited C++ programmers to get themselves more involved.

MPU embedded systems have very small RAMs (512bytes - 1K typically), and much larger Flash memory, comparing to the CPU systems. To get the best result, there shouldn't be frequent process switch or soft interrupts to the system. On the other hand, the MPUs runs quite quickly, which means it in fact can handle a large number of throughput every second. A good system should be designed base on the analysis and experiments on these special features, but not copying the CPU OS design.

We proposed to develop a minimal operating system that takes 5-8K Flash space. It should be able to abstract some most used external devices into classes, to convenient the embedded developers. Of course it must be able to handle threads and interrupts. The final scenario is to make development on MPU systems as easy as on PC systems.

We are doing some background researches and experiments on MPU systems. The project will be launched soon.