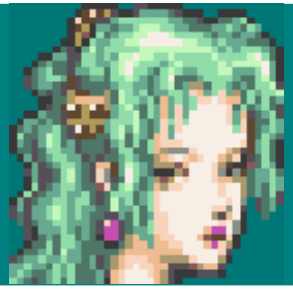
The Racket Package System

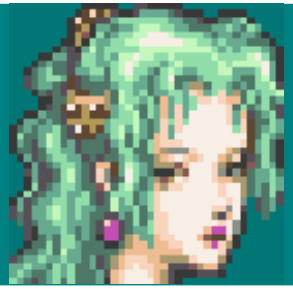
Planet 5.0 and Beyond!

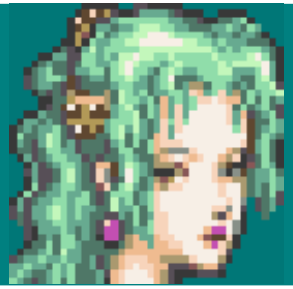
.oO I want to write a program to simulate my experience in Narshe.



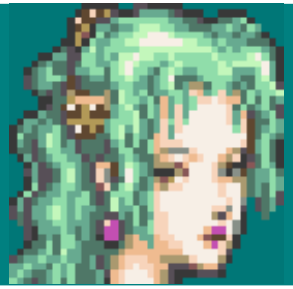
.oO I want to write a program to simulate my experience in Narshe.

```
% emacs magitek.rkt
```



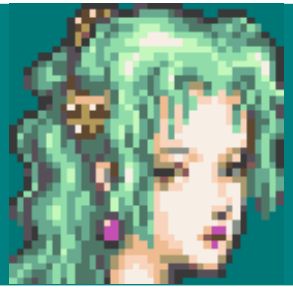


.oO Now I need to share my package...



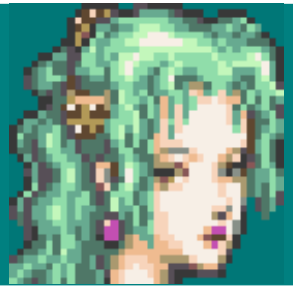
.oO Now I need to share my package...

```
% scp magitek.rkt server:public-html/
```

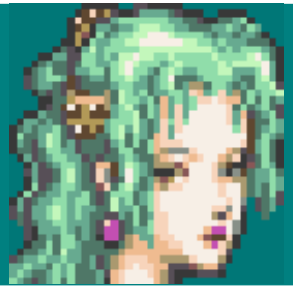


Terra wrote: Please try my program at:
http://terra.com/magitek.rkt

```
% scp magitek.rkt server:public-html/
```



.oO I should expand my simulation to include the treasure house.

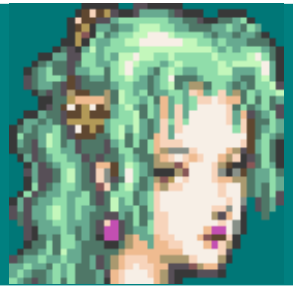


.oO I should expand my simulation to include the treasure house.

```
% mkdir narshe
```

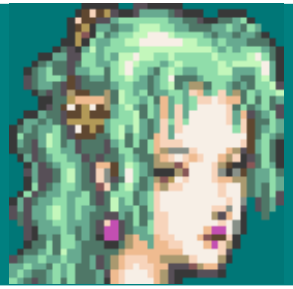
.oO I should expand my simulation to include the treasure house.

```
% mv magitek.rkt narshe/magitek.rkt
% mkdir narshe
```



.oO I should expand my simulation to include the treasure house.

```
% emacs narshe/lone-wolf.rkt
% mv magitek.rkt narshe/magitek.rkt
% mkdir narshe
```

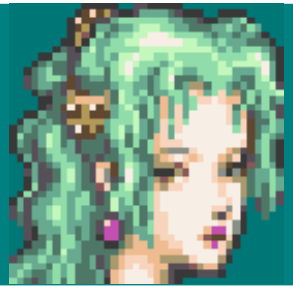


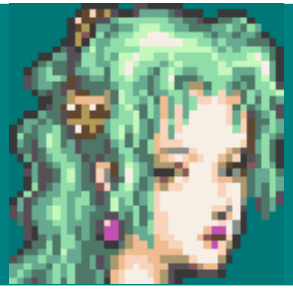
.oO I should expand my simulation to include the treasure house.

```
% scp -r narshe server:public-html/
% emacs narshe/lone-wolf.rkt
% mv magitek.rkt narshe/magitek.rkt
% mkdir narshe
```

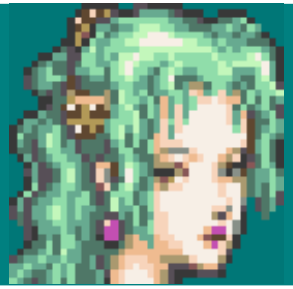
Terra wrote: Please try my program at:
http://terra.com/narshe

```
% scp -r narshe server:public-html/
% emacs narshe/lone-wolf.rkt
% mv magitek.rkt narshe/magitek.rkt
% mkdir narshe
```



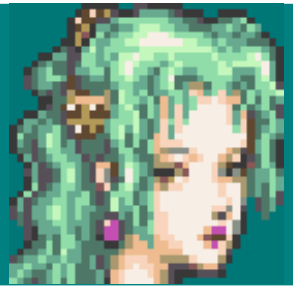


.oO Ah, I messed up the first time... let me fix that



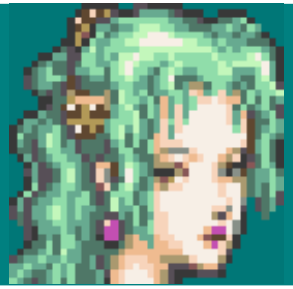
.oO Ah, I messed up the first time... let me fix that

```
% emacs narshe/lone-wolf
```



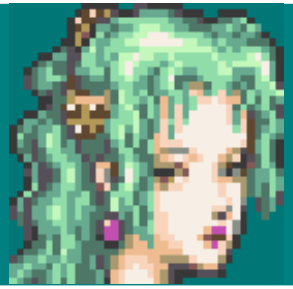
.oO Ah, I messed up the first time... let me fix that

```
% scp -r narshe server:public-html/
% emacs narshe/lone-wolf
```



Terra wrote: Please re-download my program at: http://terra.com/narshe

```
% scp -r narshe server:public-html/
% emacs narshe/lone-wolf
```



Banon: Did you know you can install as a package to get updates without having to remember what to download?

Banon: Did you know you can install as a package to get updates without having to remember what to download?

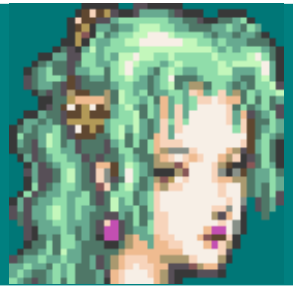
```
% raco pkg install http://terra.com/narshe
```

Banon: Did you know you can install as a package to get updates without having to remember what to download?

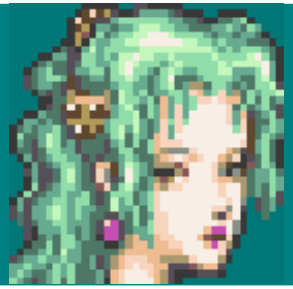
```
% ....
% raco pkg install http://terra.com/narshe
```

Banon: Did you know you can install as a package to get updates without having to remember what to download?

```
% raco pkg update
% ....
% raco pkg install http://terra.com/narshe
```

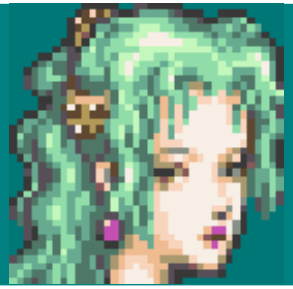


.oO I'd like to analyze the sound from Narshe.



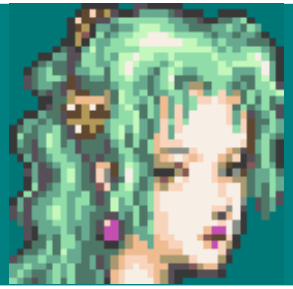
.oO I'd like to analyze the sound from Narshe.

```
% dd if=/dev/dsp of=narshe/soundtrack.wav
```



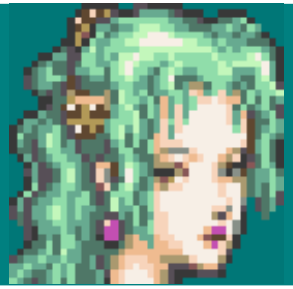
.oO I'd like to analyze the sound from Narshe.

```
% emacs narshe/music.rkt
% dd if=/dev/dsp of=narshe/soundtrack.wav
```



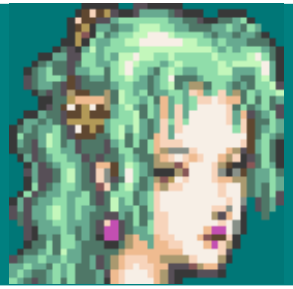
.oO I'd like to analyze the sound from Narshe.

```
% zip -r narshe.zip narshe
% emacs narshe/music.rkt
% dd if=/dev/dsp of=narshe/soundtrack.wav
```



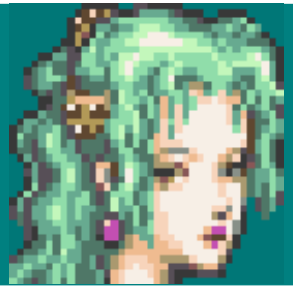
.oO I'd like to analyze the sound from Narshe.

```
% scp narshe.zip server:public-html/
% zip -r narshe.zip narshe
% emacs narshe/music.rkt
% dd if=/dev/dsp of=narshe/soundtrack.wav
```

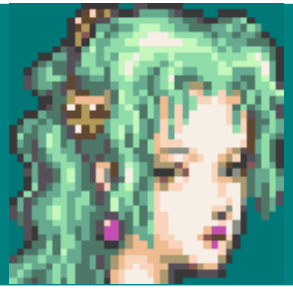


Terra wrote: I just updated my package! The new source is...

```
% raco pkg update http://terra.com/narshe.zip
```



.oO I found a way to improve the Lone Wolf scenario.



.oO I found a way to improve the Lone Wolf scenario.

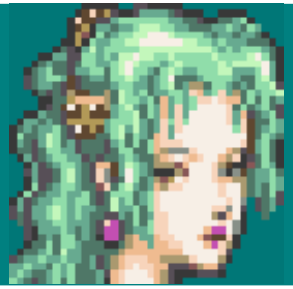
```
% emacs narshe/lone-wolf.rkt
```

.oO I found a way to improve the Lone Wolf scenario.

```
% zip -r narshe.zip narshe
% emacs narshe/lone-wolf.rkt
```

.oO I found a way to improve the Lone Wolf scenario.

```
% scp narshe.zip server:public-html/
% zip -r narshe.zip narshe
% emacs narshe/lone-wolf.rkt
```



Terra wrote: I updated the package, you may want to update!

Celes: Um, I try to check for updates regularly and it’s annoying that it always downloads the whole ZIP file and it is never different.

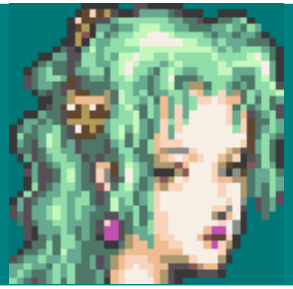
Racket Developer: You should be using a checksum file for that.

Racket Developer: You should be using a checksum file for that.

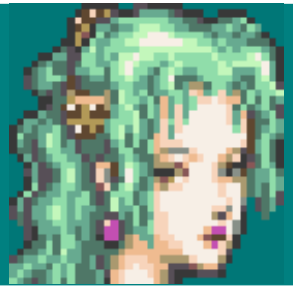
```
% md5sum narshe.zip > narshe.zip.CHECKSUM
```

Racket Developer: You should be using a checksum file for that.

```
% scp narshe.zip.CHECKSUM server:public-html/
% md5sum narshe.zip > narshe.zip.CHECKSUM
```

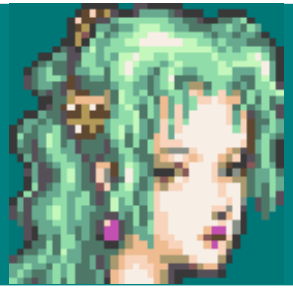


.oO I found ANOTHER way to improve everything.



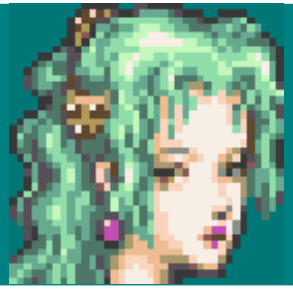
.oO I found ANOTHER way to improve everything.

```
% emacs narshe/lone-wolf.rkt
```



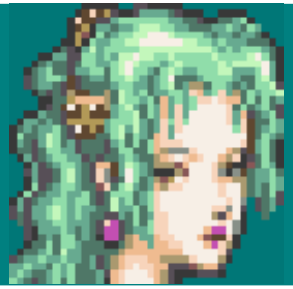
.oO I found ANOTHER way to improve everything.

```
% md5sum narshe.zip > narshe.zip.CHECKSUM
% zip -r narshe.zip narshe
% emacs narshe/lone-wolf.rkt
```



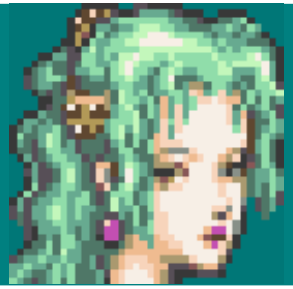
.oO I found ANOTHER way to improve everything.

```
% scp narshe.zip narshe.zip.CHECKSUM
    server:public-html/
% md5sum narshe.zip > narshe.zip.CHECKSUM
% zip -r narshe.zip narshe
% emacs narshe/lone-wolf.rkt
```



Terra wrote: Please run ’raco pkg update’ if you want the new version.

```
% scp narshe.zip narshe.zip.CHECKSUM
    server:public-html/
% md5sum narshe.zip > narshe.zip.CHECKSUM
% zip -r narshe.zip narshe
% emacs narshe/lone-wolf.rkt
```

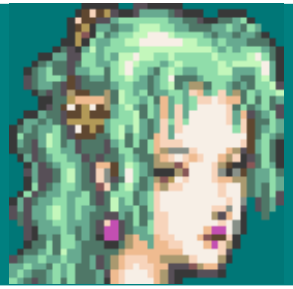


Terra: Why is it so inconvenient to update my package?

Racket Developer: It’s because you’re not using Github.

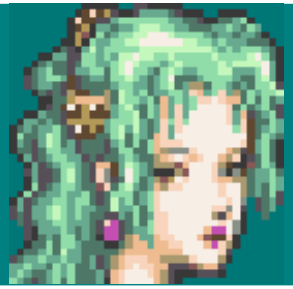
Racket Developer: It’s because you’re not using Github.

```
% git push
% ....
% git init
% cd narshe
```

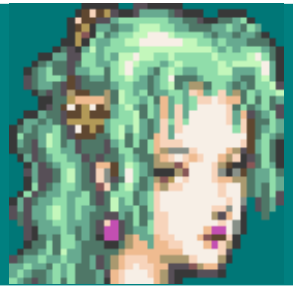


Terra wrote: Thanks, now you should use a new source:

```
% raco pkg update git://github.com/terra/narshe
% git push
% ....
% git init
% cd narshe
```

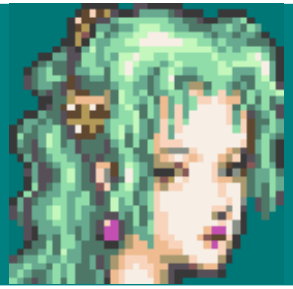


.oO Is it much easier to update now?



.oO Is it much easier to update now?

```
% git push
% emacs narshe/music.rkt
```

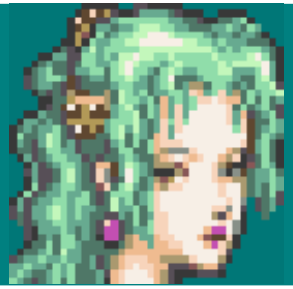


.oO That was awesome!

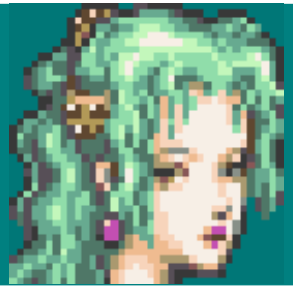
```
% git push
% emacs narshe/music.rkt
```

Terra wrote: I just did an update everyone!

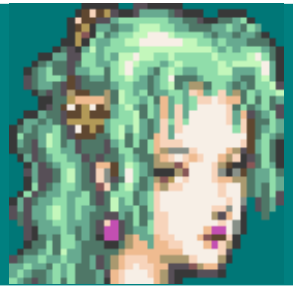
```
% git push
% emacs narshe/music.rkt
```



Racket Developer: Please stop spamming the mailing list when your package changes, just put it on the catalog and people can read the RSS feed.

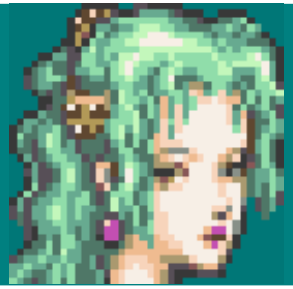


.oO That's great!



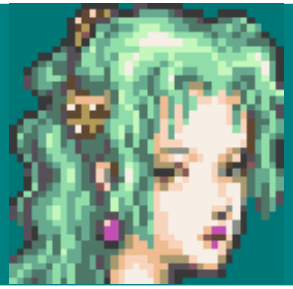
.oO That's great!

```
% raco pkg catalog-upload narshe
    git://github.com/terra/narshe
```

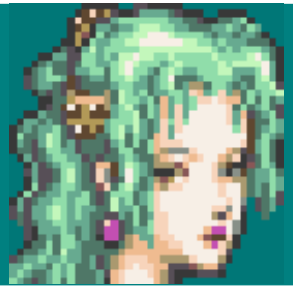


Terra wrote: Now you can use:

```
% raco pkg update narshe
% raco pkg catalog-upload narshe
    git://github.com/terra/narshe
```

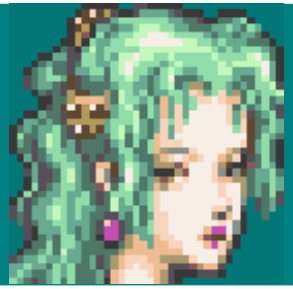


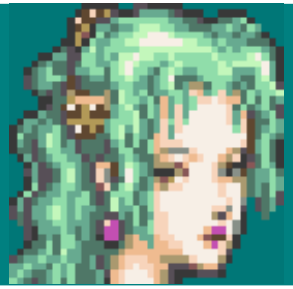
.oO Let’s make a change...



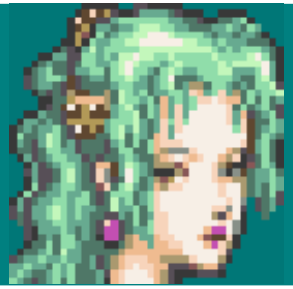
.oO Let's make a change...

```
% git push
% emacs narshe/music.rkt
```



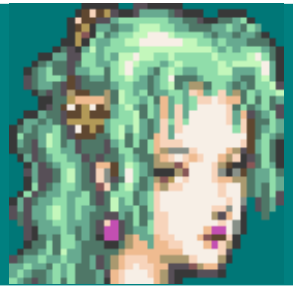


.oO I'll go to the next stage of the simulation



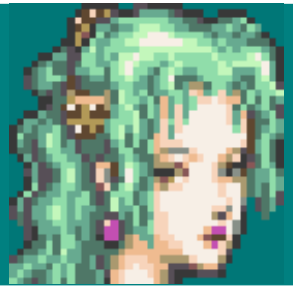
.oO I'll go to the next stage of the simulation

```
% emacs returners/banon.rkt
    (require narshe/magitek)
% mkdir returners
```



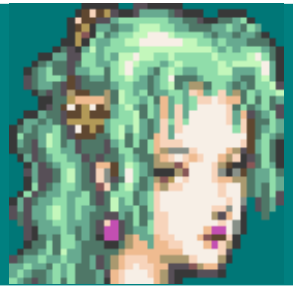
.oO I'll go to the next stage of the simulation

```
% racket returners/banon.rkt
   ERROR
% emacs returners/banon.rkt
   (require narshe/magitek)
% mkdir returners
```

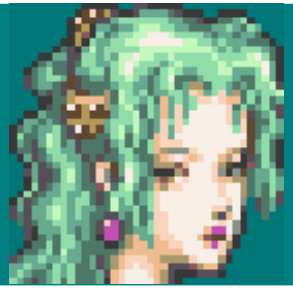


.oO I'll go to the next stage of the simulation

```
% raco pkg install --link narshe
% racket returners/banon.rkt
    ERROR
% emacs returners/banon.rkt
    (require narshe/magitek)
% mkdir returners
```

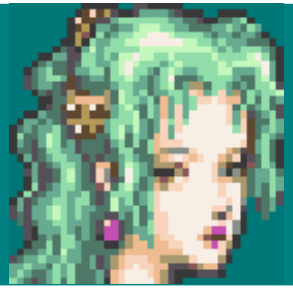


.oO Now I need to distribute Returners



.oO Now I need to distribute Returners

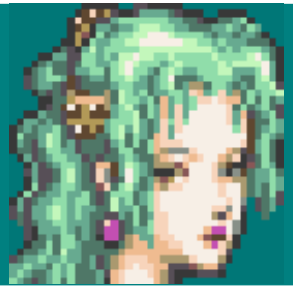
```
% raco pkg catalog-upload ....
% git push
% ....
% git init
% cd returners
```



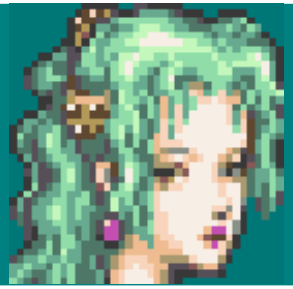
Cyan: I tried to install Returners and it died because narshe/magitek wasn’t found.

Cyan: I tried to install Returners and it died because narshe/magitek wasn't found.

```
% emacs returners/info.rkt
   #lang info
   (define deps '("narshe"))
```

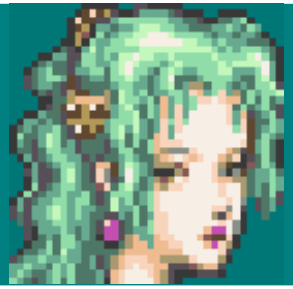


.oO I should really be using tm-halts from Racket v6



.oO I should really be using tm-halts from Racket v6

```
% emacs narshe/lone-wolf.rkt
    .... tm-halts? ....
```



Edgar: Um, I can’t use this in Racket v5.9

Edgar: Um, I can’t use this in Racket v5.9

```
% git push
% git branch narshe-for-v5.9
% git checkout master^
```

Edgar: Um, I can’t use this in Racket v5.9

```
% raco pkg catalog-version narshe 5.9
    git://github.com/terra/narshe#narshe-for-v5.9
% git push
% git branch narshe-for-v5.9
% git checkout master^
```

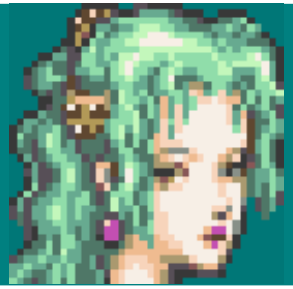
Gau: Um, when I check this out on my machine into ’narshe-for-v5.9’, then it is messed up because I can’t require ’narshe/magitek’.

Gau: Um, when I check this out on my machine into ’narshe-for-v5.9’, then it is messed up because I can’t require ’narshe/magitek’.

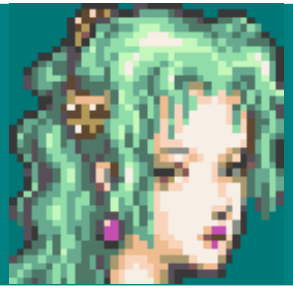
```
% emacs narshe/info.rkt
   #lang info
   (define collection "narshe")
```

Racket Developer: Internal linking is bad!

```
% emacs narshe/info.rkt
    #lang info
    (define collection "narshe")
```

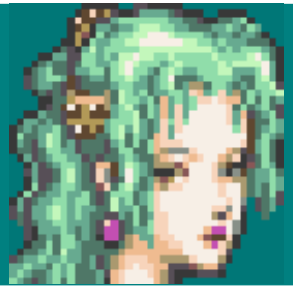


.oO I have a new feature to include about Locke



.oO I have a new feature to include about Locke

```
% emacs narshe/locke.rkt
```



.oO I have a new feature to include about Locke

```
% emacs returners/jidoor.rkt
    (require narshe/locke)
% emacs narshe/locke.rkt
```

Locke: I just installed returners on my machine that already has narshe and it was broken because narshe/lock isn't there!

```
% emacs returners/jidoor.rkt
   (require narshe/locke)
% emacs narshe/locke.rkt
```

Racket Developer: You should use version 1.0 if this is the stable interface and version 2.0 if you meant to use version 1.0 before.

```
% emacs returners/jidoor.rkt
    (require narshe/locke)
% emacs narshe/locke.rkt
```

Racket Developer: You should use version 1.0 if this is the stable interface and version 2.0 if you meant to use version 1.0 before.

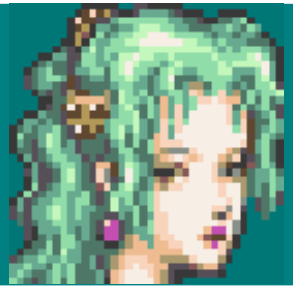
```
% emacs narshe/info.rkt
    (define version "2.0")
% emacs returners/jidoor.rkt
    (require narshe/locke)
% emacs narshe/locke.rkt
```

Racket Developer: You should use version 1.0 if this is the stable interface and version 2.0 if you meant to use version 1.0 before.

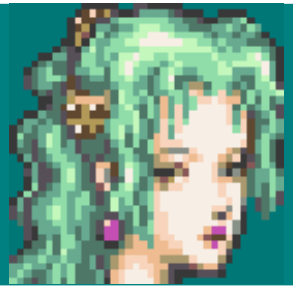
```
% emacs returners/info.rkt
    (define deps '(("narshe" #:version "2.0")))
% emacs narshe/info.rkt
    (define version "2.0")
% emacs returners/jidoor.rkt
    (require narshe/locke)
% emacs narshe/locke.rkt
```

Leo: How do I get version 0.0 of narshe after it has gone? I loved that version even though it didn't last long.

Racket Developer: This question does not make sense.

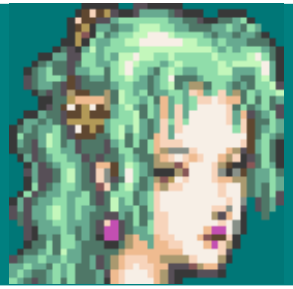


.oO I should play the music rather than analyze it.



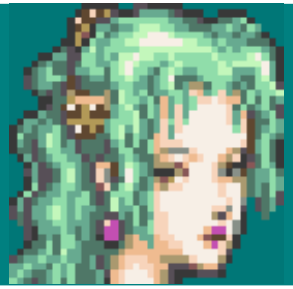
.oO I should play the music rather than analyze it.

```
% emacs narshe/music-player.rkt
    (match (system-type) .... dynamic-require ...)
```

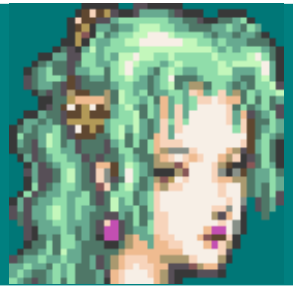


.oO I should play the music rather than analyze it.

```
% emacs narshe/info.rkt
   (define deps
    '(("openal" #:platform macosx)
      ("directaudio" #:platform windows)
      ("libsndfile" #:platform unix)))
% emacs narshe/music-player.rkt
   (match (system-type) .... dynamic-require ...)
```

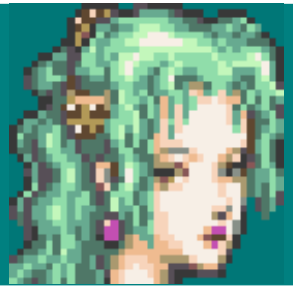


.oO Gee, maybe I should write some documentation.



.oO Gee, maybe I should write some documentation.

```
% emacs narshe/narshe.scrbl
```



.oO Gee, maybe I should write some documentation.

```
% emacs narshe/info.rkt
    (define scribblings '(("narshe.scrbl")))
% emacs narshe/narshe.scrbl
```

Ghost: Um, raco setup dies when I install your package because scribble isn't on my EC2 instance.

```
% emacs narshe/info.rkt
    (define scribblings '(("narshe.scrbl")))
% emacs narshe/narshe.scrbl
```

Ghost: Um, raco setup dies when I install your package because scribble isn't on my EC2 instance.

```
% emacs narshe/info.rkt
    (define deps .... "scribble-lib" ....)
% emacs narshe/info.rkt
    (define scribblings '(("narshe.scrbl")))
% emacs narshe/narshe.scrbl
```

Ghost: Um, I don’t /want/ to install scribble to run your package.

```
% emacs narshe/info.rkt
    (define deps .... "scribble-lib" ....)
% emacs narshe/info.rkt
    (define scribblings '(("narshe.scrbl")))
% emacs narshe/narshe.scrbl
```

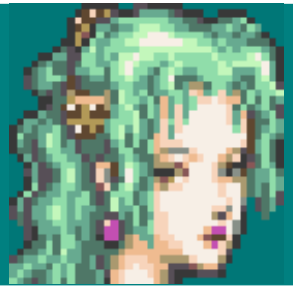
Ghost: Um, I don’t /want/ to install scribble to run your package.

```
% emacs narshe/info.rkt
    (define build-deps '("scribble-lib"))
% emacs narshe/info.rkt
    (define deps .... "scribble-lib" ....)
% emacs narshe/info.rkt
    (define scribblings '(("narshe.scrbl")))
% emacs narshe/narshe.scrbl
```

Terra: How do you install it at all without building it?

Terra: How do you install it at all without building it?

```
% raco pkg export-pkg --binary narshe
```

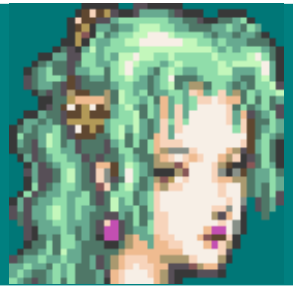


Gogo: I do the same thing for my students, but I want them to have the code too, so I use a built package and just download it from the server.

Gogo: I do the same thing for my students, but I want them to have the code too, so I use a built package and just download it from the server.

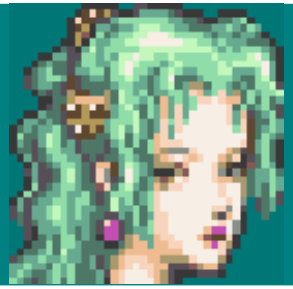
```
% wget http://..../built/v6.0/narshe.zip
```

Terra: I’d really like to run ’raco play soundtrack.wav’



Terra: I’d really like to run ’raco play soundtrack.wav’

```
% emacs narshe/raco.rkt
```



Terra: I'd really like to run 'raco play soundtrack.wav'

```
% emacs narshe/info.rkt
    (define raco-commands ....)
% emacs narshe/raco.rkt
```

Mog: I installed your package and ’raco play’ doesn’t work like you say it does.

```
% emacs narshe/info.rkt
    (define raco-commands ....)
% emacs narshe/raco.rkt
```

Mog: I installed your package and ’raco play’ doesn’t work like you say it does.

```
% emacs narshe/info.rkt
    (define setup-collects ....)
% emacs narshe/info.rkt
    (define raco-commands ....)
% emacs narshe/raco.rkt
```

Mog: This music player stuff is cool, but why do I need this Narshe stuff?

Mog: This music player stuff is cool, but why do I need this Narshe stuff?

```
% mkdir music-player
```

Mog: This music player stuff is cool, but why do I need this Narshe stuff?

```
% mkdir music-player/music
% mkdir music-player
```

Mog: This music player stuff is cool, but why do I need this Narshe stuff?

```
% mv narshe/music-player.rkt
     music-player/music/music-player.rkt
% mkdir music-player/music
% mkdir music-player
```

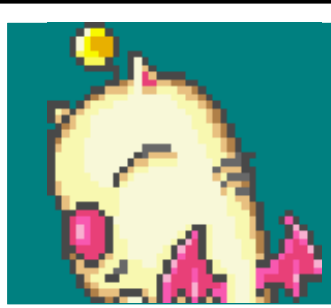
Mog: This music player stuff is cool, but why do I need this Narshe stuff?

```
% mkdir music-player/narshe
% mv narshe/music-player.rkt
     music-player/music/music-player.rkt
% mkdir music-player/music
% mkdir music-player
```

```
% emacs music-player/narshe/music-player.rkt
```

```
% emacs music-player/info.rkt
    (define collection 'multi)
    (define raco-commands ....)
% emacs music-player/narshe/music-player.rkt
```

```
% raco play ....
    ERROR
% emacs music-player/info.rkt
    (define collection 'multi)
    (define raco-commands ....)
% emacs music-player/narshe/music-player.rkt
```



```
% emacs music-player/narshe /info.rkt
    (define raco-commands ....)
% raco play ....
    ERROR
% emacs music-player/info.rkt
    (define collection 'multi)
    (define raco-commands ....)
% emacs music-player/narshe/music-player.rkt
```

```
% emacs narshe /info.rkt
    (define deps ....)
% emacs music-player/narshe /info.rkt
    (define raco-commands ....)
% raco play ....
    ERROR
% emacs music-player/info.rkt
    (define collection 'multi)
    (define raco-commands ....)
% emacs music-player/narshe/music-player.rkt
```

Relm: Um, when I compile my program that built on narshe/music-player, I get a warning from raco setup.

RelM: Um, when I compile my program that built on narshe/music-player, I get a warning from raco setup.

```
% emacs narshe /info.rkt
    (define implies ....)
```

Mash: Here’s a patch that makes it faster by using the GPU!

Mash: Here’s a patch that makes it faster by using the GPU!

```
% curl ... | git am
```

Mash: Here's a patch that makes it faster by using the GPU!

```
% cat music-player/data/gpu-vector.rkt
% curl ... | git am
```

Setzer: Um, I can’t install music-player and general-gpu

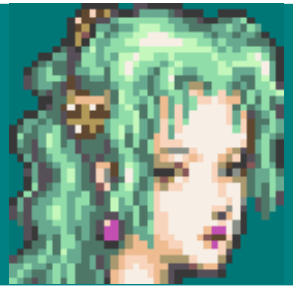
```
% cat music-player/data/gpu-vector.rkt
% curl ... | git am
```

Racket Developer: There's a conflict. I think we should talk about it.

```
% cat music-player/data/gpu-vector.rkt
% curl ... | git am
```

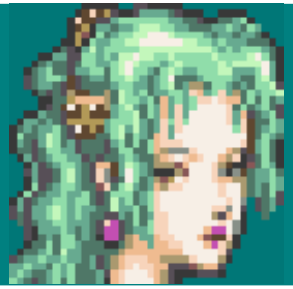
Racket Developer: There's a conflict. I think we should talk about it.

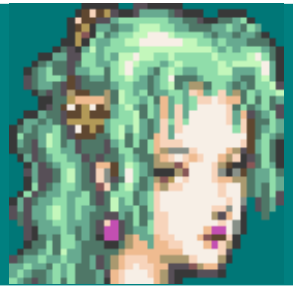
```
% rm music-player/data/gpu-vector.rkt
% cat music-player/data/gpu-vector.rkt
% curl ... | git am
```



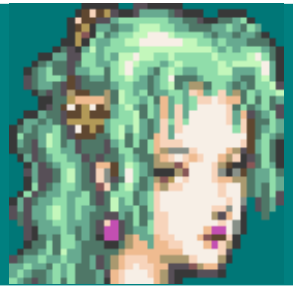
Terra: Isn’t this backwards incompatible?

```
% rm music-player/data/gpu-vector.rkt
% cat music-player/data/gpu-vector.rkt
% curl ... | git am
```





.oO I have a new implementation idea...



.oO I have a new implementation idea...

```
% emacs narshe/*
```

Shadow: Ah, everything is broken!!

```
% emacs narshe/*
```

Racket Developer: This is a different package, you should name it different.

```
% emacs narshe/*
```

Racket Developer: This is a different package, you should name it different.

```
% git branch narshe-v1
% git checkout master^
% emacs narshe/*
```

Racket Developer: This is a different package, you should name it different.

```
% raco pkg catalog-source narshe
    git://github.com/terra/narshe#narshe-v1
% git branch narshe-v1
% git checkout master^
% emacs narshe/*
```

Racket Developer: This is a different package, you should name it different.

```
% raco pkg catalog-upload narshe2
    git://github.com/terra/narshe
% raco pkg catalog-source narshe
    git://github.com/terra/narshe#narshe-v1
% git branch narshe-v1
% git checkout master^
% emacs narshe/*
```

Strago: I’d like to run all the versions at the same time for my Narshian simulation simulation environment

Racket Developer: Yes, it is possible to deprecate completely, but we generally want to allow all old versions.

Racket Developer: Yes, it is possible to deprecate completely, but we generally want to allow all old versions.

```
% emacs narshe/info.rkt
    (define collection "narshe2")
```

Umaro: I need to make sure I get everything install exactly right on my deployment.

Umaro: I need to make sure I get everything install exactly right on my deployment.

```
% raco pkg export-installed > narshe.com.pkgs
```

Umaro: I need to make sure I get everything install exactly right on my deployment.

```
% scp narshe.com.pkgs server:
% raco pkg export-installed > narshe.com.pkgs
```

Umaro: I need to make sure I get everything install exactly right on my deployment.

```
% raco pkg import-installed < narshe.com.pkgs
% scp narshe.com.pkgs server:
% raco pkg export-installed > narshe.com.pkgs
```

Umaro: It takes too long to run import-installed, because it installs everything and I’m worried that this Zip-backed package won’t have old versions.

Racket Developer: Zip-backed packages aren’t very good.

Racket Developer: Zip-backed packages aren't very good.

```
% raco pkg export-pkgs --binary narshe.com.zip
    pkg ...
```

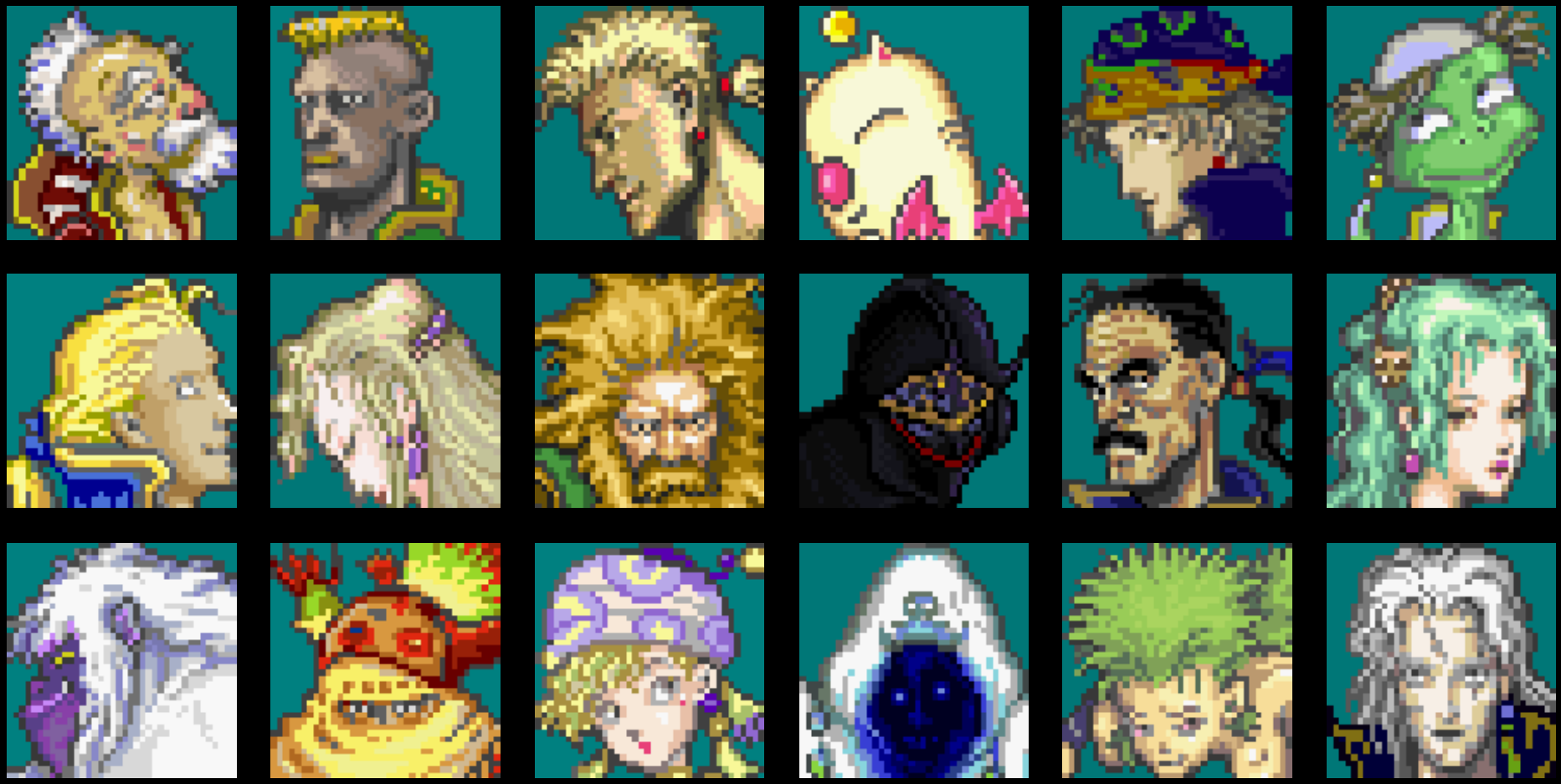
Racket Developer: Zip-backed packages aren't very good.

```
% scp narshe.com.zip server:
% raco pkg export-pkgs --binary narshe.com.zip
    pkg ...
```

Racket Developer: Zip-backed packages aren't very good.

```
% raco pkg import-pkgs narshe.com.zip
% scp narshe.com.zip server:
% raco pkg export-pkgs --binary narshe.com.zip
    pkg ...
```

Thank You!



Model of Package System

Model of Package System

- The core is the same as the leaves.

Model of Package System

- The core is the same as the leaves.
- Social processes are as valuable, if not more valuable, than technical frameworks.

Model of Package System

- The core is the same as the leaves.
- Social processes are as valuable, if not more valuable, than technical frameworks.
- Compatibility is very valuable and rarely broken (2htdp/image, #lang mzscheme, etc)

Model of Package System

- The core is the same as the leaves.
- Social processes are as valuable, if not more valuable, than technical frameworks.
- Compatibility is very valuable and rarely broken (2htdp/image, #lang mzscheme, etc)
- Incompatibility is removing features and disobeying documentation.

Model of a Package

Model of a Package

- A package is a set of modules. From any collection.

Model of a Package

- A package is a set of modules. From any collection.
- Version numbers go up to indicate new features.

Model of a Package

- A package is a set of modules. From any collection.
- Version numbers go up to indicate new features.
- Dependencies are vanilla, versioned, and platformed.
- Dependency violations are warnings, not errors.

A Good Package...

- uses a neutral name.
- is on Github*.
- has an explicit name or is ’multi.
- is listed officially and does not conflict with anything.

A Better Package...

- starts at version 0.0 and switches to 1.0 on stability.
  - If interface is same, no version change.
  - If interface is grown, version increases.
  - If interface shrinks, a fork occurs.

A Better Package...

- starts at version 0.0 and switches to 1.0 on stability.
  - If interface is same, no version change.
  - If interface is grown, version increases.
  - If interface shrinks, a fork occurs.
- updates the catalog when depencies on Racket versions are added.

A Better Package...

- starts at version 0.0 and switches to 1.0 on stability.
  - If interface is same, no version change.
  - If interface is grown, version increases.
  - If interface shrinks, a fork occurs.
- updates the catalog when depencies on Racket versions are added.
- has informative tags and description in catalog.

A Best Package...

- includes documentation and tests.
- is a ’multi package.
- specifies a license and uses a problem tracker.
- has a responsive author.

A not so good package...

- exposes the internal development of the package.
- is not for public consumption.

A not so good package...

- exposes the internal development of the package.
- is not for public consumption.
- You can communicate with different groups by sharing sources and making your own catalog.

Model of a System

- The same computer may contain many users of Racket and many installations of Racket.
- Packages can be arbitrarily shared or not shared between any combination of these through installation names and scopes.

Open Problems

- There’s work to do, but we know how to do it, it just takes time.

Open Problems

- There’s work to do, but we know how to do it, it just takes time.
- Documentation will be a challenge.

Open Problems

- There’s work to do, but we know how to do it, it just takes time.
- Documentation will be a challenge.
  - What is a structure property?

Open Problems

- There’s work to do, but we know how to do it, it just takes time.
- Documentation will be a challenge.
  - What is a structure property?
  - What functions work on lists?

Open Problems

- There’s work to do, but we know how to do it, it just takes time.
- Documentation will be a challenge.
  - What is a structure property?
  - What functions work on lists?
  - How can the Tutorial, Guide, Exegesis, Reference, data/file/net, etc documentation be extensible?

Open Problems

- There's work to do, but we know how to do it, it just takes time.
- Documentation will be a challenge.
  - What is a structure property?
  - What functions work on lists?
  - How can the Tutorial, Guide, Exegesis, Reference, data/file/net, etc documentation be extensible?
- How can we truly support multiple simultaneous versions for all aspects of the system, core and otherwise?