

오픈소스로 여는 뉴노멀

2020 공개SW 페스티벌



과학기술정보통신부
Ministry of Science and ICT



정보통신산업진흥원
National IT Industry Promotion Agency

오픈소스 프로젝트 잘 관리하기

(개인 프로젝트라는 씨앗에서 커뮤니티라는 꽃을 피우기까지)

넥슨코리아
옥찬호

utilForever / Lightbot

Unwatch ▾

2

★ Star

1

🍴 Fork

0

<> Code

! Issues

🔗 Pull requests

▶ Actions

📁 Projects

📖 Wiki

🛡 Security

📈 Insights

...

🔑 master ▾

Go to file

Add file ▾

📄 Code ▾

About



utilForever Initial commit ...

on Aug 18 🕒 1



.gitignore

Initial commit

2 months ago



LICENSE

Initial commit

2 months ago



README.md

Initial commit

2 months ago

README.md



Lightbot

Lightbot simulator using C++ with some reinforcement learning

Lightbot simulator using C++ with some reinforcement learning

📖 Readme

📄 MIT License

Releases

No releases published

[Create a new release](#)

Packages

No packages published

[Publish your first package](#)



여러분이 꿈꾸는
오픈소스 프로젝트의 모습은?

 tensorflow / tensorflow Watch ▾

8.3k

 Star

150k

 Fork

83.1k

 Code Issues 3.8k Pull requests 183 Actions Projects 1 Security 29

...

 flutter / flutter Watch ▾

3.2k

 Star

105k

 Fork

14.7k

 Code Issues 5k+ Pull requests 156 Actions Projects 134 Wiki Security

...

 facebook / react Watch ▾

6.8k

 Star

158k

 Fork

31.3k

 Code Issues 478 Pull requests 121 Actions Projects Wiki Security

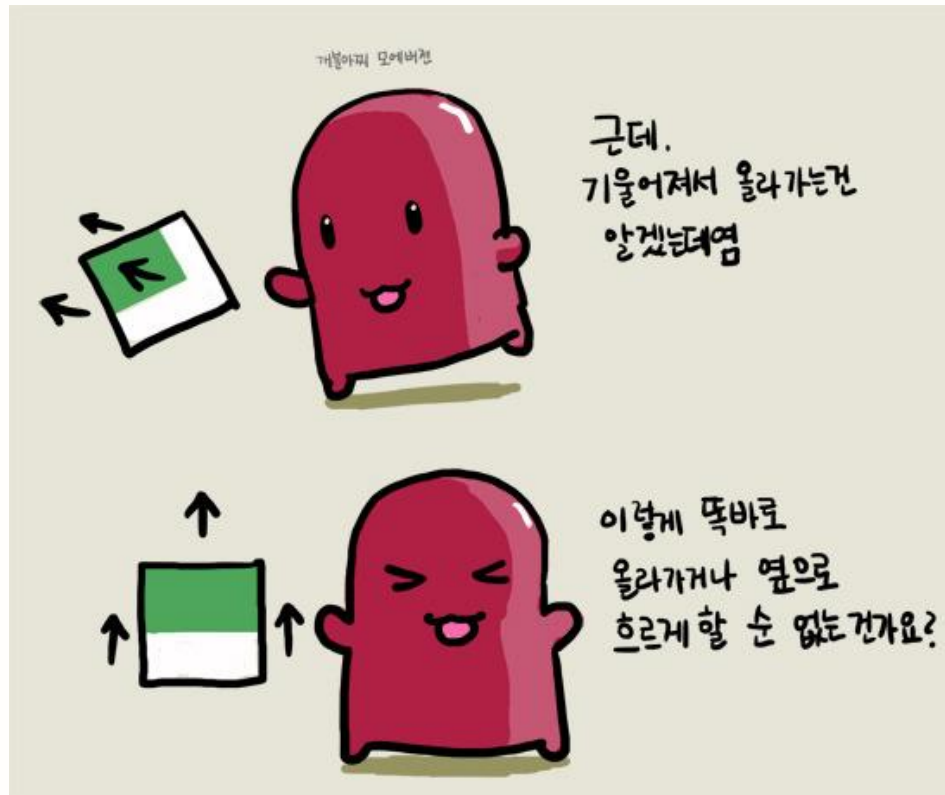
...



▶ 생각해 봅시다.

지나가던 개발자가 여러분이 만든 프로젝트를 우연히 보게 되었습니다.

이 개발자는 프로젝트에 기여를 하고 싶어합니다. 그런데 어떻게 기여하죠?



▶ 생각해 봅시다.

이 개발자가 기여를 할 수 있게끔 도와줘야 합니다! 어떻게요?

- 프로젝트 설치 및 사용 방법 알려주기
- 새로 구현할 기능이나 발생한 버그 정리하기
- 이슈 템플릿 작성하기
- 프로젝트의 구조를 문서화해서 제공하기
- 프로젝트 기여 방법을 문서로 제공하기
- 질문/답변을 위한 창구 만들기
- PR이 올라왔을 때 코드 리뷰하기

▶ 프로젝트 설치 및 사용 방법 알려주기

열심히 만든 프로젝트를 다른 사람이 사용할 수 없다면 의미가 없겠죠?
프로젝트를 어떻게 하면 사용할 수 있는지 알려주세요.
설치 방법이나 사용 방법을 문서로 정리해두면 좋습니다.

Quick Start

You will need CMake to build the code. If you're using Windows, you need Visual Studio 2017 in addition to CMake.

First, clone the code:

```
git clone https://github.com/utilForever/RosettaStone.git --recursive  
cd RosettaStone
```

Python API

Build and install the package by running

```
pip install -U .
```

▶ 프로젝트 설치 및 사용 방법 알려주기

중요한 팁! 운영체제마다 설치 및 사용 방법이 다른 경우가 있습니다.

이때는 운영체제마다 따로 알려주셔야 합니다. (Windows/Linux/MacOS)

Building from Ubuntu and Windows Subsystem for Linux (WSL)

RosettaStone supports Ubuntu 18.04 or higher. Using `apt-get`, install required tools and libraries by running,

```
sudo apt-get install build-essential python3-dev python3-pip cmake
```

This will install GNU compilers and CMake. Once installed, build the code by running

```
mkdir build
cd build
cmake ..
make
```

Again, use `make -j <num_threads>` flag to boost up the build performance by using multithreads.

This will build entire codebase. To run the unit test, execute

```
bin/UnitTests
```

It should show all the tests are passing.

Building from Windows

To build the code on Windows, CMake and Visual Studio 2017 (or higher) is required. Windows' version of CMake is available from [this website](#). To install Visual Studio, the community edition of the tool can be downloaded from [Visual Studio Community 2017](#). You can also use Visual Studio 2019.

Once everything is installed, run the following commands:

```
md build
cd build
cmake .. -G"Visual Studio 15 2017 Win64"
```

▶ 새로 구현할 기능이나 발생한 버그 정리하기

설치 및 사용 방법이 나와 있어서 프로젝트를 무사히 설치했습니다.
이제 개발자는 프로젝트의 어느 부분에 도움이 필요한지 궁금해합니다.
현재 프로젝트 개발 상황과 도움이 필요한 부분을 공유해주세요!

Correct the logic of some discover cards #525

 Open

utilForever opened this issue 5 days ago · 0 comments



utilForever commented 5 days ago

Owner



The card such as 'Tortollan Pilgrim' and 'Madame Lazul' discovers a card in hand or opponent's hand.
Currently, the logic of these cards is invalid because 'Tortollan Pilgrim' and 'Madame Lazul' discover a copy of a card.



utilForever added

bug

p0

labels 5 days ago



utilForever self-assigned this 5 days ago

▶ 이슈 템플릿 작성하기

1. 개발자가 프로젝트를 이리저리 살펴보던 중 버그를 발견했습니다.
2. 개발자가 프로젝트에 새로운 기능을 제안하고 싶습니다.

이 경우 어떻게 제보해야 할까요? 그리고 어떤 정보를 알려줘야 할까요?

이슈 템플릿을 작성해 어떻게 제보하는지, 어떤 정보가 필요한지 알려주세요.

이슈 템플릿 작성하기

Issue: Bug report

Create a report to help us improve. If this doesn't look right, [choose a different type](#).



Write

Preview

H B I

****Describe the bug****

A clear and concise description of what the bug is.

****To Reproduce****

Steps to reproduce the behavior:

1. Go to '...'
2. Click on '....'
3. Scroll down to '....'
4. See error

****Expected behavior****

A clear and concise description of what you expected to happen.

****Screenshots****

If applicable, add screenshots to help explain your problem.

****Desktop (please complete the following information):****

- OS: [e.g. iOS]
- Browser [e.g. chrome, safari]
- Version [e.g. 22]

****Smartphone (please complete the following information):****

- Device: [e.g. iPhone6]
- OS: [e.g. iOS8.1]
- Browser [e.g. stock browser, safari]
- Version [e.g. 22]

▶ 이슈 템플릿 작성하기

Issue: Feature request

Suggest an idea for this project. If this doesn't look right, [choose a different type](#).



Write

Preview

H B I

****Is your feature request related to a problem? Please describe.****

A clear and concise description of what the problem is. Ex. I'm always frustrated when [...]

****Describe the solution you'd like****

A clear and concise description of what you want to happen.

****Describe alternatives you've considered****

A clear and concise description of any alternative solutions or features you've considered.

****Additional context****

Add any other context or screenshots about the feature request here.

Attach files by dragging & dropping, selecting or pasting them.



Styling with Markdown is supported

Submit new issue

▶ 프로젝트 구조를 문서화해서 제공하기

개발자는 기여하기 전에 여러분의 프로젝트 구조를 살펴보고자 합니다.

이해를 돕기 위해 프로젝트 구조를 문서화해서 제공해주세요!

문서로 만들어 두면 다른 개발자에게 프로젝트를 쉽게 설명할 수 있습니다.

Step 2: Implement task interface `ITask`

In source file `Tasks.hpp`

```
class ITask
{
...
private:
    /// Processes task logic internally and returns meta data.
    /// \param player The player to run task.
    /// \return The result of task processing.
    virtual TaskStatus Impl(Player& player) = 0;

    /// Internal method of Clone().
    /// \return The cloned task.
    virtual ITask* CloneImpl() = 0;
};
```

There are two virtual methods you have to implement.

- Method `Impl` processes task logic internally and returns meta data.
- Method `CloneImpl` is internal method of `Clone()`.

▶ 프로젝트 기여 방법을 문서로 제공하기

프로젝트 구조 파악이 끝난 개발자는 코드를 작성해 기여하려고 합니다.
커밋 메시지는 어떻게 작성해야 할까요? PR은 어떻게 작성해야 할까요?
프로젝트에 기여하는 방법을 구체적으로 문서로 작성해 제공해주세요!

Pull Requests

- [Setting up your local environment](#)
 - [Step 1: Fork](#)
 - [Step 2: Build](#)
 - [Step 3: Branch](#)
- [Making Changes](#)
 - [Step 4: Code](#)
 - [How to add card](#)
 - [Step 5: Commit](#)
 - [Commit message guidelines](#)
 - [Step 6: Rebase](#)
 - [Step 7: Test](#)
 - [Step 8: Push](#)
 - [Step 9: Opening the Pull Request](#)
 - [Step 10: Discuss and Update](#)
 - [Approval and Request Changes Workflow](#)
 - [Step 11: Landing](#)
 - [Continuous Integration Testing](#)
 - [Code Coverage Testing](#)
 - [Code Quality Checking](#)

▶ 프로젝트 기여 방법을 문서로 제공하기

Commit message guidelines

A good commit message should describe what changed and why. RosettaStone project uses [semantic commit messages](#) to streamline the release process.

Before a pull request can be merged, it **must** have a pull request title with a semantic prefix.

Examples of commit messages with semantic prefixes:

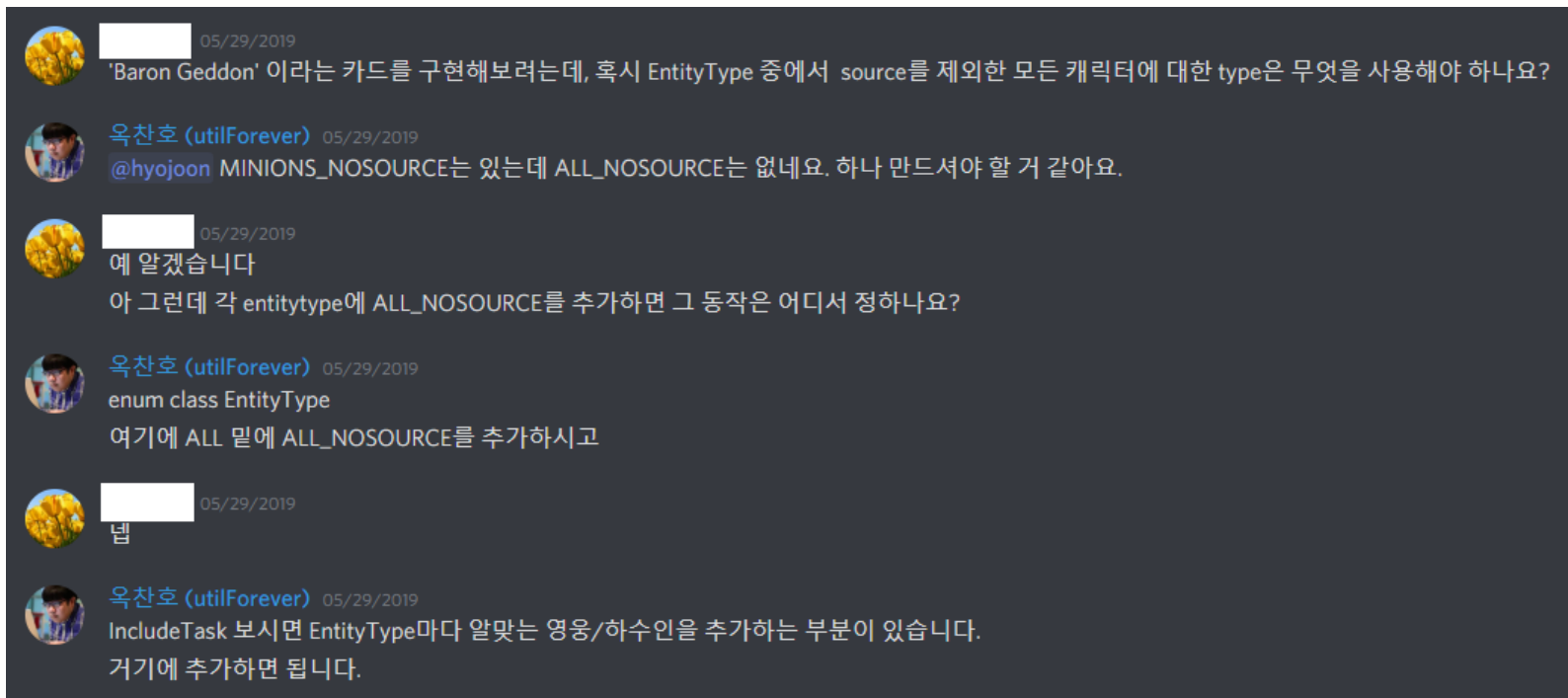
- feat: allow provided config object to extend other configs
- docs: correct spelling of CHANGELOG
- feat(lang): add polish language
- fix: correct minor typos in code

Common prefixes:

- fix: A bug fix
- feat: A new feature
- docs: Documentation changes
- test: Adding missing tests or correcting existing tests
- build: Changes that affect the build system
- ci: Changes to our CI configuration files and scripts
- perf: A code change that improves performance
- refactor: A code change that neither fixes a bug nor adds a feature
- style: Changes that do not affect the meaning of the code (linting)

▶ 질문/답변을 위한 창구 만들기

이슈를 통해 프로젝트와 관련된 질문/답변을 할 수도 있지만,
자유롭게 이야기할 수 있는 별도의 창구를 만들어 두면 좋습니다.
Gitter, Slack, Discord 등의 커뮤니티를 활용해 보면 어떨까요?



A screenshot of a Gitter chat conversation. The chat is on a dark background with white text. It shows a series of messages between several users, including a user with a yellow flower profile picture and a user named '옥찬호 (utilForever)'. The messages discuss adding 'ALL_NOSOURCE' to an 'EntityType' enum.

05/29/2019
'Baron Geddon' 이라는 카드를 구현해보려는데, 혹시 EntityType 중에서 source를 제외한 모든 캐릭터에 대한 type은 무엇을 사용해야 하나요?

옥찬호 (utilForever) 05/29/2019
@hyojoon MINIONS_NOSOURCE는 있는데 ALL_NOSOURCE는 없네요. 하나 만드셔야 할 거 같아요.

05/29/2019
예 알겠습니다
아 그런데 각 entityType에 ALL_NOSOURCE를 추가하면 그 동작은 어디서 정하나요?

옥찬호 (utilForever) 05/29/2019
enum class EntityType
여기에 ALL 밑에 ALL_NOSOURCE를 추가하시고

05/29/2019
넵

옥찬호 (utilForever) 05/29/2019
IncludeTask 보시면 EntityType마다 알맞는 영웅/하수인을 추가하는 부분이 있습니다.
거기에 추가하면 됩니다.

▶ PR이 올라왔을 때 코드 리뷰하기

개발자가 버그를 수정하거나 새로운 기능을 구현해 PR을 올렸습니다.

이제 여러분의 차례입니다. 올라온 PR을 살펴 보고 코드 리뷰를 해주세요!

utilForever reviewed on Sep 15 [View changes](#)

Sources/Rosetta/PlayMode/Tasks/SimpleTasks/DiscoverTask.cpp Outdated [Hide resolved](#)

```
... @@ -356,6 +356,30 @@ std::vector<Card*> DiscoverTask::Discover(Game* game, Player* player,
356 356 }
357 357 }
358 358 break;
359 + case DiscoverType::FILTER_STACK:
```

utilForever on Sep 15 • edited Owner

Suggested change ⓘ

```
- case DiscoverType::FILTER_STACK:
+ case DiscoverType::JANDICE_BAROV:
```

utilForever on Sep 15 Owner

Why don't you annotate the way this code works to help you understand?

doobee98 on Sep 15 Author Collaborator

I changes DiscoverType::FILTER_STACK to DiscoverType::JANDICE_BAROV, so DiscoverType::JANDICE_BAROV doesn't need a process for removing duplicates. Finally, I don't add annotation.

And I tried to add ChoiceAction::JANDICE_BAROV (because it already loses generality), but I thought writing a specific id(SCH_351a, b) in Choose.cpp is a little bad design. Also, I cannot get a unchosen card id in Choose.cpp. So I didn't change it and use ChoiceAction::STACK continuously.

If you have any comment, notice me. Thanks.

utilForever on Sep 15 Owner

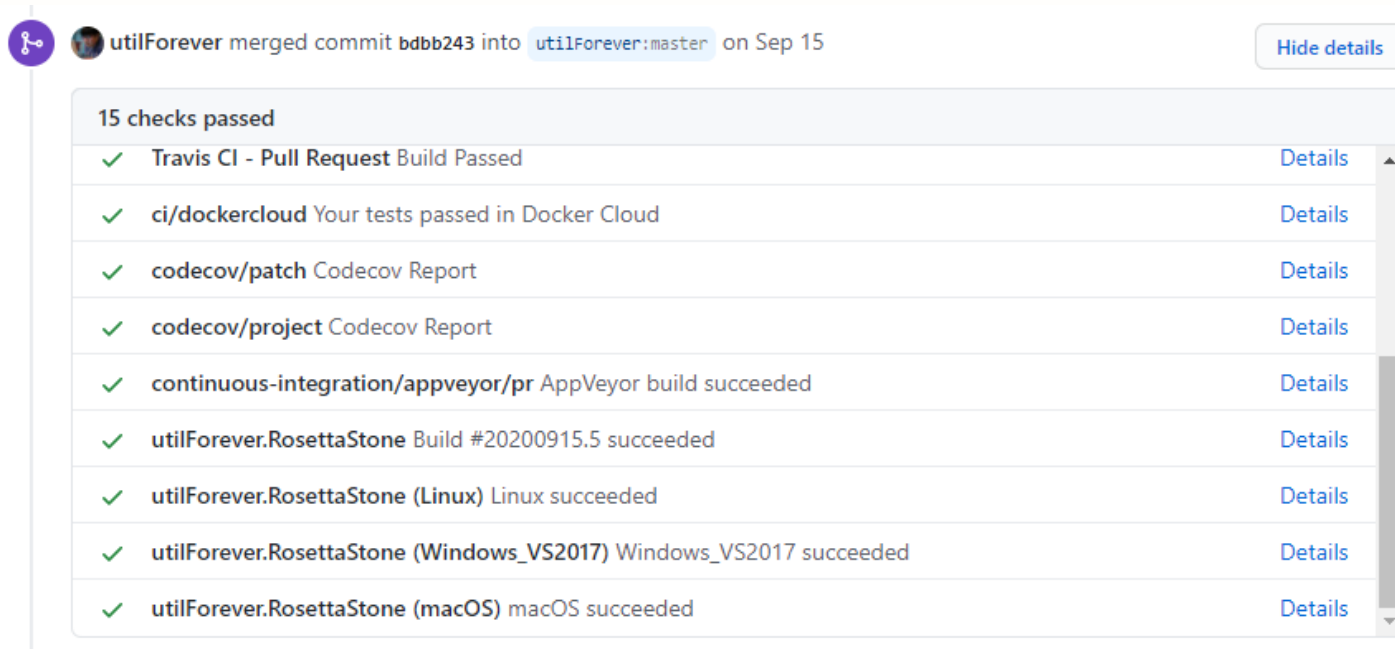
Alright. I agree. 🙌

Reply...

▶ PR이 올라왔을 때 코드 리뷰하기

PR이 올라왔을 때 각종 툴들을 사용하면 도움이 됩니다.

예를 들어, CI를 통해 여러 플랫폼에서 빌드가 되는지 테스트하거나
정적 분석을 통해 코드 품질을 검사하는 방법 등이 있겠네요.

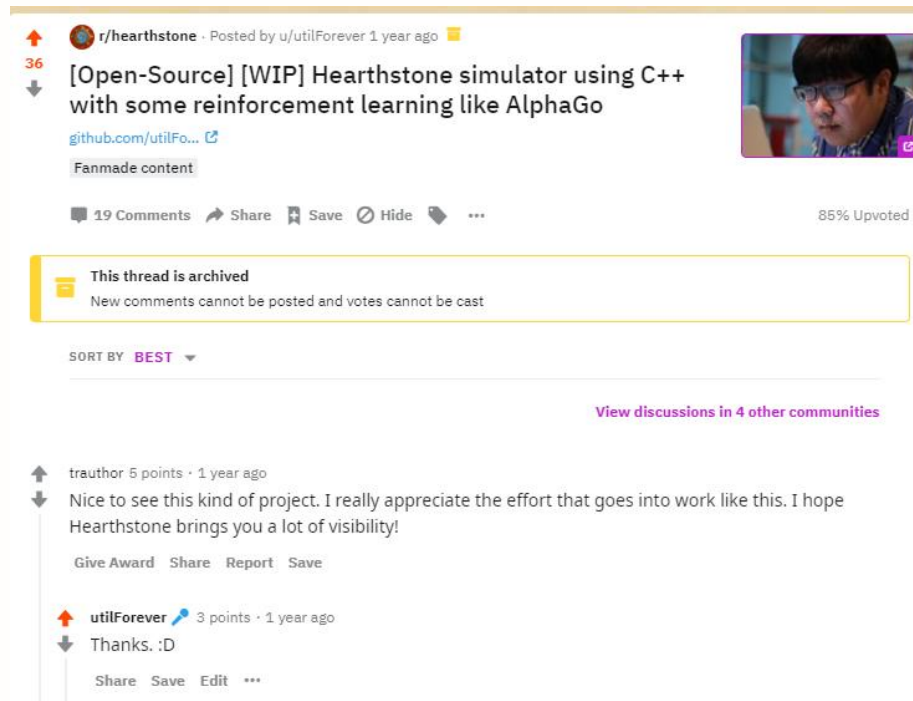


A screenshot of a GitHub pull request interface. At the top, it says 'utilForever merged commit bdbb243 into utilForever:master on Sep 15' with a 'Hide details' button. Below this, a section titled '15 checks passed' lists various CI and code quality checks, each with a green checkmark and a 'Details' link. The checks include Travis CI, Docker Cloud, Codecov, AppVeyor, and RosettaStone builds for Linux, Windows, and macOS.

Check Name	Status	Details
Travis CI - Pull Request Build Passed	✓	Details
ci/dockercloud Your tests passed in Docker Cloud	✓	Details
codecov/patch Codecov Report	✓	Details
codecov/project Codecov Report	✓	Details
continuous-integration/appveyor/pr AppVeyor build succeeded	✓	Details
utilForever.RosettaStone Build #20200915.5 succeeded	✓	Details
utilForever.RosettaStone (Linux) Linux succeeded	✓	Details
utilForever.RosettaStone (Windows_VS2017) Windows_VS2017 succeeded	✓	Details
utilForever.RosettaStone (macOS) macOS succeeded	✓	Details

▶ 오픈소스 프로젝트 알리기

대부분의 다른 사람들은 여러분이 만든 프로젝트의 존재를 모릅니다.
SNS나 커뮤니티에 여러분이 만든 오픈소스 프로젝트를 알리세요.
Facebook, Twitter, Reddit 등 여러 커뮤니티가 있습니다.



어느새 멋지게 성장한 프로젝트를 보게 될 것입니다.

main 4 branches 0 tags

Go to file Add file Code

utilForever Merge pull request #528 from utilForever/update-data 058690b 23 hours ago 7,315 commits

.github	ci: Add step to install packages 'python3-setuptools'	15 days ago
Builds	fix: Add CMake argument '-DROSETTA_ROOT' to Windows build	15 days ago
Datas	Update Datas - Fix nickname part of test users	2 years ago
Documents	feat: Implement card 'Jandice Barov'(SCH_351)	2 months ago
Extensions	refactor: Change the logic of method 'CreateDeckWithCode()'	2 months ago
Includes/Rosetta	feat(update-data): Apply card update by Hearthstone Patch 18.6	yesterday
Libraries	build: Update the version to 'v2.6.0rc1'	16 days ago
Medias	[ci skip] docs: Upload presentations	2 years ago
Resources	feat(update-data): Update card data by Hearthstone Patch 18.6	yesterday
Scripts	ci : Remove Travis CI script files	16 days ago
Sources/Rosetta	feat(update-data): Apply card update by Hearthstone Patch 18.6	yesterday
Tests	fix: Rollback changes of the card 'Tortollan Pilgrim' (ULD_236)	3 days ago
.clang-format	style: Set 'AfterCaseLabel' to 'true'	6 months ago
.codacy.yml	ci: Add '.github/**' to ignore list	15 days ago
.gitattributes	style: Add .gitattributes	2 months ago
.gitignore	build: Add 'setup.py' to .gitignore	9 months ago
.gitmodules	refactor: Init submodule 'Lyra'	9 months ago
.lgtm.yml	fix: Correct CodeFactor issues	2 years ago
.travis.yml	ci: Update Travis CI configuration file	16 days ago

About

Hearthstone simulator using C++ with some reinforcement learning

utilforever.github.io/rosettastone/

hearthstone reinforcement-learning cpp17 hearthstone-api rl rl-environment cpp cplusplus python-api simulator-game hearthstone-simulator

Readme

AGPL-3.0 License

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Contributors 51

+ 40 contributors



감사합니다.

utilForever@gmail.com

Facebook, Twitter: @utilForever