**가. SW 기본정보 (팀명: TSTL)**

**□ SW 개요**

TSTL; TypeScript-STL.

TSTL 은 C++ 표준화 위원회 (C++ Standard Committee) 가 정의한 STL (Standard Template Library) 규약의 TypeScript 구현체이다. TSTL은 STL 규약과 동일하게, 다음 네 가지 영역의 라이브러리를 제공한다.

|  |  |
| --- | --- |
| Module | Description |
| Containers | Collection Framework |
| Iterators | Iterator Pattern |
| Algorithms | Functional programming with iterator pattern |
| Functors | Miscellaneous objects |

**□ Test 환경 정보**

그 어떤 컴퓨터이던, NodeJS 와 TypeScript 및 Git 만 설치되어 있으면 테스트를 수행할 수 있다.

**##########################################**

**# TSTL 설치, 빌드 및 테스트 수행**

**##########################################**

**# NodeJS 와 TypeScript 가 반드시 설치되어 있어야 함**

**brew** install node

**npm** install -g typescript

**# 깃허브로부터 TSTL 을 다운받자**

git **clone** https://github.com/samchon/tstl.git

**cd** tstl

**# Dependency 를 설치하고, TSTL 을 빌드한 후,**

**npm install** --save-dev

**node** build/build

**# 테스트 자동화를 돌려주면 끝.**

**npm** test

**나. SW 기능 리스트**

**□ Containers & Iterators**

Java 의 Collection Framework 에 해당하는 모듈.

다채로운 Containers 와 정규화된 Iterator Pattern 을 제공함.

* **Linear Containers**
  + Vector vector
  + Deque deque
  + List list
* **Adaptor Containers**
  + Stack stack
  + Queue queue
  + PriorityQueue priority\_queue
* **Associative Containers**
  + Atomic Containers
    - TreeSet set
    - TreeMultiSet multiset
    - HashSet unordered\_set
    - HashMultiSet unordered\_multiset
  + Dictionaries
    - TreeMap map
    - TreeMultiMap multimap
    - HashMap unordered\_map
    - HashMultiMap unordered\_multimap

**□ Algorithms**

다양한 알고리즘 함수들의 모음.

STL의 알고리즘 함수들은, 아래와 같이 일관된 패턴을 지녀, 용이성과 확장성이 우수함

* Iterator Pattern
* Functional Programming

TSTL 에서 제공되는 알고리즘 함수들은, 크게 다음과 같이 분류할 수 있다.

* iterations
* modifiers
* sortings
* heaps
* binary\_searches
* partitions
* union\_sets
* mathematics

**□ Functors**

유틸리티성 객체들의 모음 (다른 말로는 miscellaneous).

현재 TSTL 이 제공하고 있는 Functors 는 다음과 같다.

* <exception>
  + logic\_error
  + runtime\_error
  + system\_error
* <functional>
  + unary functions
  + binary functions
  + hash functions
* <utility>
* Asynchronous critical sections
  + mutex
  + timed\_mutex
  + shared\_mutex
  + shared\_timed\_mutex
  + conditional\_variable

**다. 기능 시험 항목**

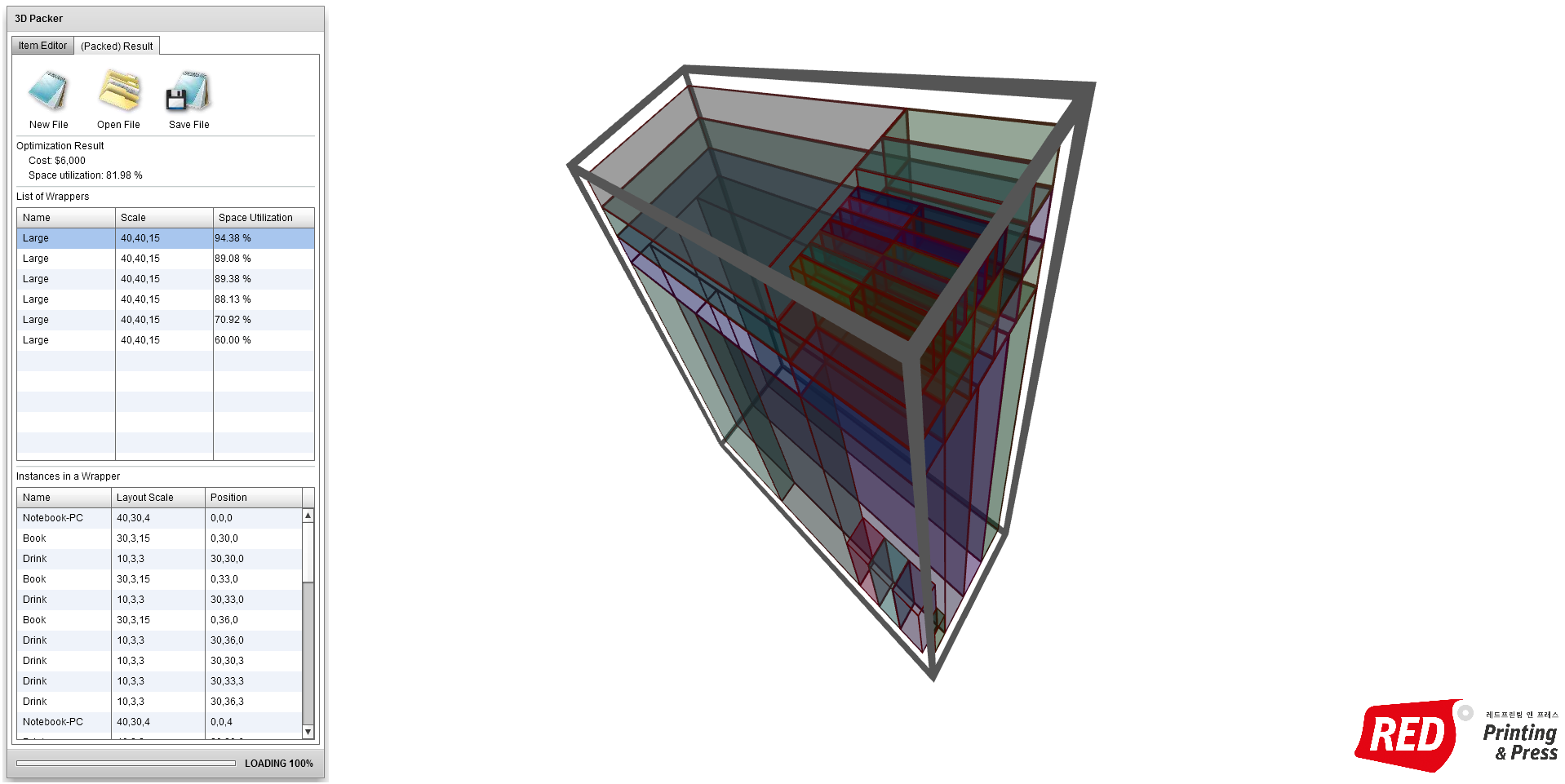
|  |  |  |  |
| --- | --- | --- | --- |
| 대분류 | 중분류 | 소분류 | Pass/Fail |
| Containers | Linear | Vector |  |
| Deque |  |
| List |  |
| Adaptor | Stack |  |
| Queue |  |
| PriorityQueue |  |
| Associative | TreeSet |  |
| TreeMultiSet |  |
| HashSet |  |
| HashMultiSet |  |
| TreeMap |  |
| TreeMultiMap |  |
| HashMap |  |
| HashMultiMap |  |
| Algorithms | Iterations | |  |
| modifiers | |  |
| partitions | |  |
| sortings | |  |
| binary searches | |  |
| union sets | |  |
| heaps | |  |
| mathematics | |  |
| Functors | exceptions | LogicError |  |
| RuntimeError |  |
| SystemError |  |
| functional | unary functions |  |
| binary functions |  |
| hash functions |  |
| utilities | |  |
| Asynchronous CV | (Timed)Mutex |  |
| Shared(Timed)Mutex |  |
| ConditionVariable |  |

**라. 간접 테스트**

아래 프로젝트들은 TSTL을 사용하고 있는 프로젝트들이다.

아래의 프로젝트를 통하여, TSTL의 기능이 제대로 동작하고 있는지 간접적으로 테스트를 해 볼 수 있다.

**□ 3D Bin Packing**



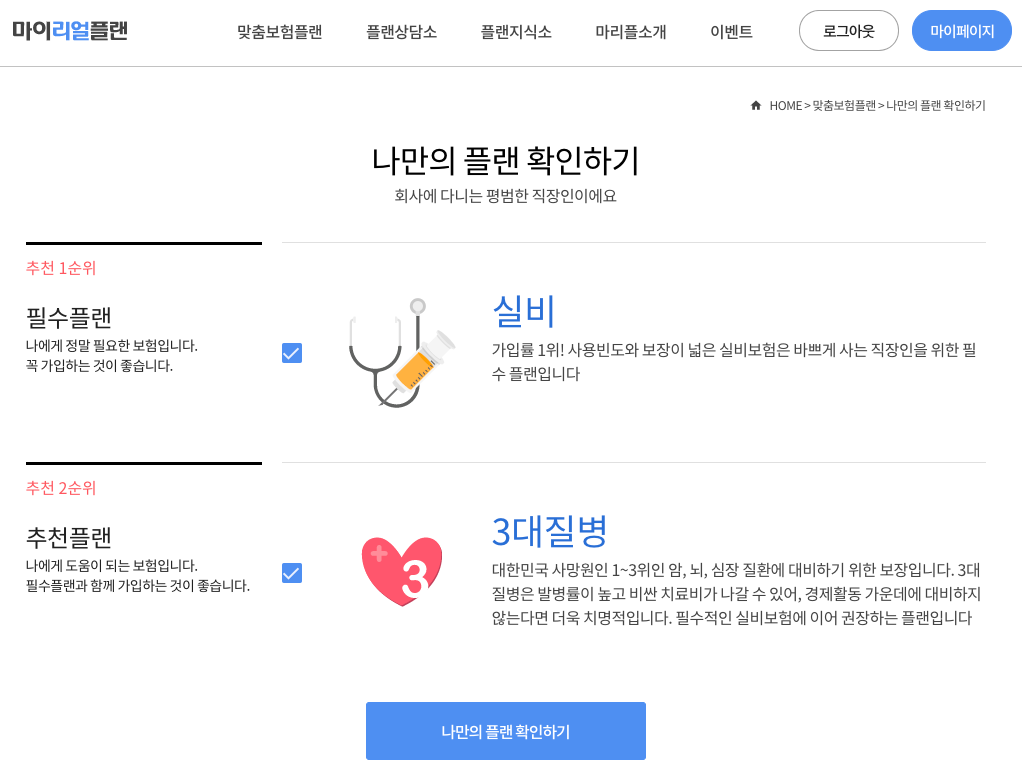
<http://betterwaysystems.github.io/packer/demo>

3D Bin Packing 은 공간배치 최적화 솔루션이다.

3D Bin Packing 이 직접적으로 사용하고 있는 TSTL 의 기능은 다음과 같다.

* Containers
  + Linear Containers
    - Vector
    - List
  + Associative Containers
    - HashSet
    - HashMap
* Algorithms
  + sortings
* Functors
  + exceptions
  + functional
  + utility

**□ Study Engine**



<https://www.myrealplan.co.kr/plan>

Study Engine은 보험 검색엔진의 일종으로써, 사용자에게 최적의 보험상품 패키지를 추천해준다.

Study Engine이 직접적으로 사용하고 있는 TSTL의 기능은 다음과 같다.

* Containers
  + Linear Conainters
  + Associative Containers
* Algorithms
  + iterations
  + sortings
  + union sets
* Functors
  + exceptions
  + functional
  + utility
  + Asynchronous Critical Sections
    - SharedMutex
    - ConditionVariable