Software V&V Project Documentation

Student Name(s): Dominic Ciliberto, John Aquino, Kevin Zelaya, Matthew Fuller

Proposed System Under Test (SUT): Dark Souls Item Randomizer v0.3

Link to SUT Source Code: https://github.com/HotPocketRemix/DarkSoulsItemRandomizer

Link to SUT School Source Code:
https://github.com/csun-comp587-s20/DarkSoulsItemRandomizer

SUT Size: 21,232 total lines / approx. 3,689 lines of computational code / approx. 17,943 lines of dictionary elements

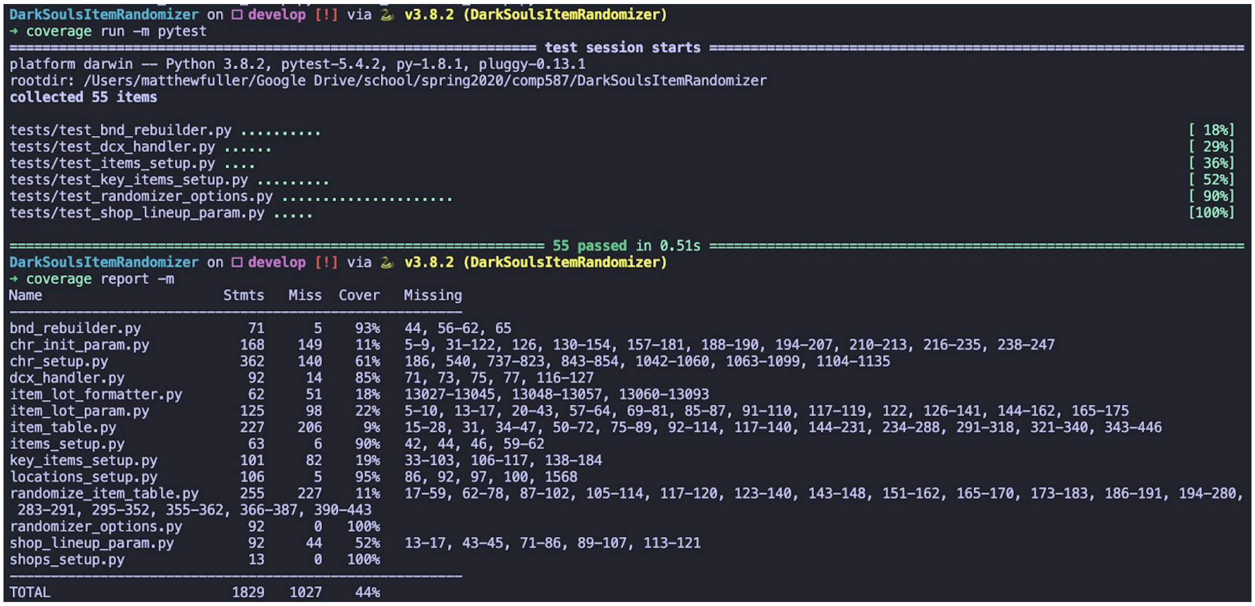
Unit Tested Files:

- bnd_rebuilder.py
- chr_init_param.py
- chr_setup.py
- dcx_handler.py
- item_lot_formatter.py
- item_lot_param.py
- item_table.py
- items_setup.py
- key_items_setup.py
- randomize_item_table.py
- randomizer_gui.py
- randomizer_options.py
- shop_lineup_param.py
- shops_setup.py

Github link to Tests Folder:
https://github.com/csun-comp587-s20/DarkSoulsItemRandomizer/tree/develop/tests

Coverage Screenshot:

Below are the tested files and the amount of unit tests for each test fixture. Also below please find the overview of the coverage of each tested file. This includes information about missing statements, total statements, percentage of statements covered, and which statements are specifically missing coverage.

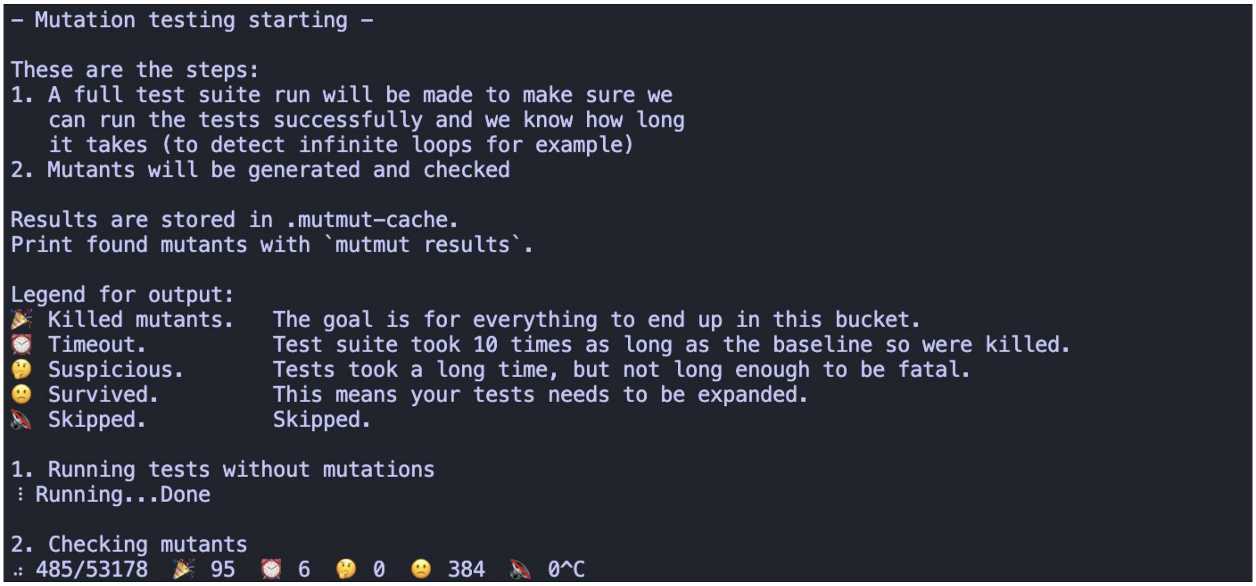


```
DarkSoulsItemRandomizer on □ develop [!] via 🐍 v3.8.2 (DarkSoulsItemRandomizer)
➜ coverage run -m pytest
============================================================= test session starts ==============================================================
platform darwin -- Python 3.8.2, pytest-5.4.2, py-1.8.1, pluggy-0.13.1
rootdir: /Users/matthewfuller/Google Drive/school/spring2020/comp587/DarkSoulsItemRandomizer
collected 55 items

tests/test_bnd_rebuilder.py ..........                                  [ 18%]
tests/test_dcx_handler.py ......                                        [ 29%]
tests/test_items_setup.py ....                                          [ 36%]
tests/test_key_items_setup.py .........                                 [ 52%]
tests/test_randomizer_options.py .....................                  [ 90%]
tests/test_shop_lineup_param.py .....                                   [100%]

============================================================== 55 passed in 0.51s ==============================================================
DarkSoulsItemRandomizer on □ develop [!] via 🐍 v3.8.2 (DarkSoulsItemRandomizer)
➜ coverage report -m
Name                      Stmts   Miss  Cover   Missing
-------------------------------------------------------
bnd_rebuilder.py             71      5    93%   44, 56-62, 65
chr_init_param.py           168    149    11%   5-9, 31-122, 126, 130-154, 157-181, 188-190, 194-207, 210-213, 216-235, 238-247
chr_setup.py                362    140    61%   186, 540, 737-823, 843-854, 1042-1060, 1063-1099, 1104-1135
dcx_handler.py               92     14    85%   71, 73, 75, 77, 116-127
item_lot_formatter.py        62     51    18%   13027-13045, 13048-13057, 13060-13093
item_lot_param.py           125     98    22%   5-10, 13-17, 20-43, 57-64, 69-81, 85-87, 91-110, 117-119, 122, 126-141, 144-162, 165-175
item_table.py               227    206     9%   15-28, 31, 34-47, 50-72, 75-89, 92-114, 117-140, 144-231, 234-288, 291-318, 321-340, 343-446
items_setup.py               63      6    90%   42, 44, 46, 59-62
key_items_setup.py          101     82    19%   33-103, 106-117, 138-184
locations_setup.py          106      5    95%   86, 92, 97, 100, 1568
randomize_item_table.py     255    227    11%   17-59, 62-78, 87-102, 105-114, 117-120, 123-140, 143-148, 151-162, 165-170, 173-183, 186-191, 194-280,
 283-291, 295-352, 355-362, 366-387, 390-443
randomizer_options.py        92      0   100%
shop_lineup_param.py         92     44    52%   13-17, 43-45, 71-86, 89-107, 113-121
shops_setup.py               13      0   100%
-------------------------------------------------------
TOTAL                      1829   1027    44%
```

We did not include mutation testing in our efforts because the runtime for the mutants was far to slow. We had 53,000 mutants and it took the fastest machine we had thirty minutes to get to mutant number 485.



```
- Mutation testing starting -

These are the steps:
1. A full test suite run will be made to make sure we
   can run the tests successfully and we know how long
   it takes (to detect infinite loops for example)
2. Mutants will be generated and checked

Results are stored in .mutmut-cache.
Print found mutants with `mutmut results`.

Legend for output:
🎉 Killed mutants.   The goal is for everything to end up in this bucket.
⏰ Timeout.          Test suite took 10 times as long as the baseline so were killed.
🤔 Suspicious.       Tests took a long time, but not long enough to be fatal.
🙁 Survived.         This means your tests needs to be expanded.
🔇 Skipped.          Skipped.

1. Running tests without mutations
⠸ Running...Done

2. Checking mutants
⠴ 485/53178  🎉 95  ⏰ 6  🤔 0  🙁 384  🔇 0^C
```

Automated Tested Features: We sought out to create an automated test suite for the Repository's capability to prevent softlocks in the game. However, due to oversight in time and external factors we were not able to deliver the automated testing suite in time.

Lessons Learned: While this project was definitely very interesting to work on. We had many issues in testing the repository due to the way that the SUT was laid out. We believe this was due to how the System Under Testing was written entirely by one person. The SUT features many values that are hardcoded and is overall very difficult to read. Overall, we underestimated the complexity and illegibility of the SUT. Something we also should have considered was integration testing considering the size of the SUT and how many methods feed into one another. Overall, we learned to appreciate the work and craftsmanship that an individual may put in their own personal project, but also understand that they may be writing code with legibility as an afterthought. If we could restart the project, we would dedicate much more time into refactoring the SUT.