

SokobanSolver

Программа поиска решений игры “Sokoban”

Выполнил: Кудашев Тимур Алексеевич, 10-2
Руководитель: Лянгузов Ф.А.

Содержание

Описание задачи.....	3
Решение.....	3
Технологии и библиотеки.....	3
Алгоритмы.....	3
А*.....	3
Интерфейс.....	4
Хранение и загрузка данных.....	4
Обобщенная блок-схема программы.....	4
Файловая система программы.....	4
Пример работы программы.....	6
Вывод.....	6

Описание задачи

Реализовать как игру Sokoban по классическим правилам(правила: <https://ru.wikipedia.org/wiki/Sokoban>), так и возможность получить решение конкретного уровня. Также важно научиться использовать алгоритмы на графах в реальных задачах.

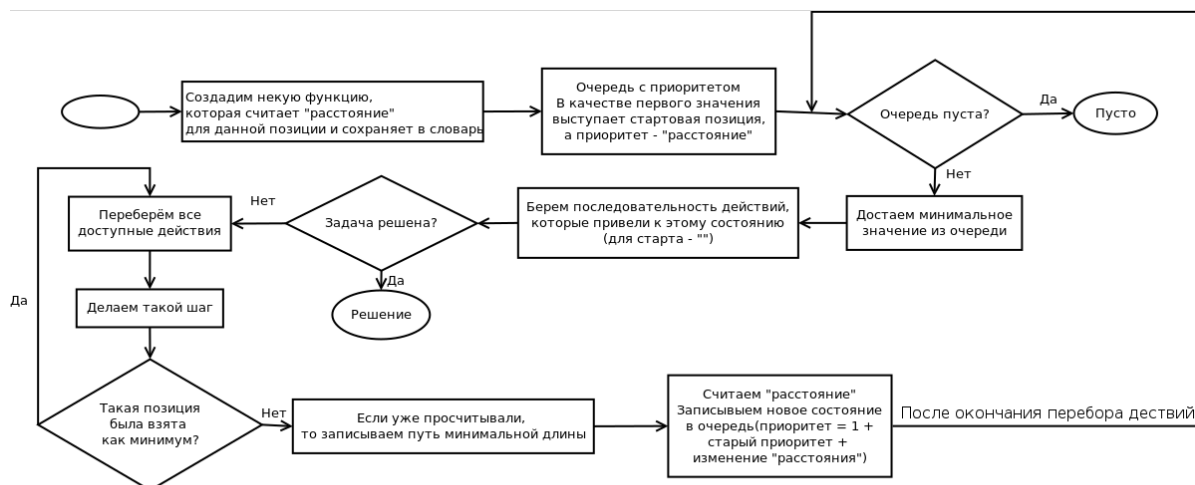
Решение

Технологии и библиотеки

- Cocos2D, для изображения графического интерфейса
- Heapq, для реализации очереди с приоритетом для алгоритма A*
- Unittest, для автоматического тестирования различных элементов программы

Алгоритмы

A*



«Расстояние» - сумма расстояний от каждого ящика до ближайшей к нему цели + от пользователя до ближайшего ящика.

Пусто - отсутствие решения

Интерфейс

Пользователь запускает программу, перед ним предстаёт первый уровень. Фиолетовые прямоугольники — ящики, точки — конечные цели, человек — игрок, тёмные прямоугольники — стены, светлые - пустоты. Он может:

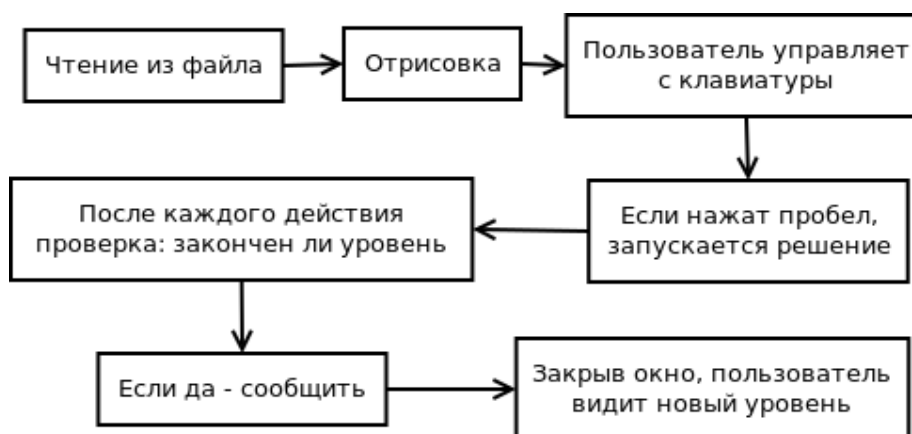
- управлять героем с помощью «стрелочек»
- с помощью «пробела» получить и запустить решение уровня

Хранение и загрузка данных

Каждый уровень хранится в отдельном текстовом файле с помощью специальной «азбуки»: «#» - стена, « » - пустота, «.» - конечная цель, «@» - игрок, «\$» - ящик, «*» - ящик на конечной цели. В начале каждого уровня файл читается и на его основании рисуется карта. Пример уровня(графическое отображение ниже):

```
#####xxx
#@ .####
# . . #
#$$ $ #
# #####
# #xxxx
####xxxx
```

Обобщенная блок-схема программы

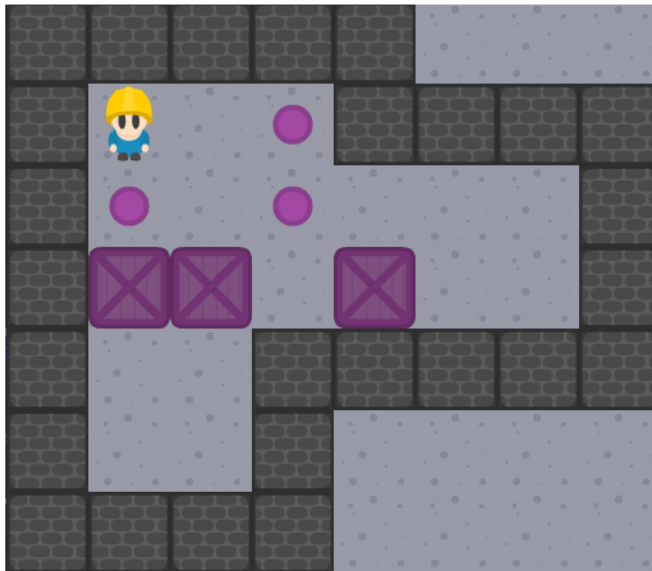


Файловая система программы

- game.py — реализован класс Game. Его объекты строят графическую карту, отвечают на запрос решён ли уровень, а также вызывают алгоритм поиска.

- keyboard.py — ядро программы. Реализовано отображение графической карты, взаимодействие пользователя с клавиатурой и графическое отображение решения.
- Level.py — два класса: Level и Matrix. Первый отвечает за чтение из файла и сохранение в объект второго. Второй, в свою очередь, меняет карту в режиме реального времени(как при действиях пользователя, так и при решении), отвечает какие действия доступны пользователю, то есть по сути дела объект этого класса — сама карта уровня.
- main.py — основной цикл программы. Уровни по очереди читаются из файлов. Запускается функция play из файла keyboard.
- move.py — реализована функция, отвечающая может ли пользователь сделать шаг при нажатии соответствующей клавиши, и какой ящик он при этом двигает.
- PriorityQueue.py — реализован класс PriorityQueue(очередь с приоритетом).
- solve_test.py — файл с тестами. Тестируются метод move и алгоритм поиска
- useful.py — реализован ключевой алгоритм поиска и нахождения «расстояния»

Пример работы программы



Основное окно программы

Вывод

Разработка самой игры и алгоритма была интересна, а также интересным было внедрение алгоритма поиска решения. Получив нужную последовательность действий, не было понятно, как именно её отображать. В итоге решение следующее: поскольку для управления с клавиатуры используется словарь клавиш, то можно заменить этот словарь, на словарь, в котором только у одного из 4 направлений стоит 1, а у остальных 0. Положение 1 определяется текущим символом в найденной последовательности действий.