

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра САПР**

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОЙ РАБОТЕ**  
**по дисциплине «Алгоритмы и структуры данных»**  
**Тема: Песчаная куча**

Студент гр. 2302

\_\_\_\_\_

Николаев В.Ю.

Преподаватель

\_\_\_\_\_

Калмычков В.А.

Санкт-Петербург  
2023 г.

## Оглавление

<b>Исходная формулировка</b>	<b>5</b>
<b>Математическая постановка задачи</b>	<b>5</b>
<b>Реализация</b>	<b>5</b>
Особенности системы программирования	6
Используемые переменные	6
Используемые функции	6
<b>Организация UI</b>	<b>7</b>
<b>Горячие клавиши</b>	<b>8</b>
<b>Пример работы</b>	<b>8</b>

## Исходная формулировка

Реализация и визуализация модели песчаной кучи с возможностью выбора различных сеток.

### Математическая постановка задачи

Рассмотрим квадратную сетку. На этой сетке расположена песчаная куча: в каждом узле этой сетки помещается стопка из нескольких песчинок. Если на некотором узле в стопке 4 песчинки или больше, то куча нестабильна, и происходит обвал: из этого узла в 4 соседних узла перемещается по 1 песчинке.

Обвалы происходят до тех пор, пока куча не станет стабильной, то есть пока в каждом узле не останется менее 4 песчинок. Порядок обвалов может быть любым. Доказательство:

Есть нестабильная песчаная куча. Рассмотрим два не равных порядка обвала  $x_1, x_2, \dots, x_n$  и  $y_1, y_2, \dots, y_m$ , где  $x_i$  и  $y_i$  - нестабильные клетки на сетке. Рано или поздно во второй последовательности обвалов тоже должна встретиться клетка  $x_1$ , ведь иначе она так и останется нестабильной, т.е. есть какая-нибудь клетка  $y_i = x_1$ . Заметим, что если в двух нестабильных клетках могут случиться обвалы подряд — сначала в одной, потом в другой, причём вторая была нестабильной ещё до первого обвала, — то они могут случиться и в обратном порядке с тем же результатом. Поэтому можно протащить обвал  $y_i$  в самое начало последовательности, не меняя итоговый результат, получая:  $x_2, x_3, \dots, x_n$  и  $y_1, y_2, \dots, y_{i-1}, y_{i+1}, \dots, y_m$ . Теперь повторяем то же самое рассуждение до тех пор, пока от одной из последовательностей ничего не останется — а значит, и от второй тоже, ибо пустая последовательность обвалов означает, что куча стабильна.

Не обязательно рассматривать модель песчаной кучи именно на квадратной сетке. Можно вместо квадратной сетки взять другую (в этом случае обвал должен происходить не при 4 песчинках в узле, а при числе песчинок, равном числу соседей), например треугольную или шестиугольную сетку.

Заметим, что при обычной реализации если в клетке  $2 * neighborsCount$  или больше песчинок, то обвал необходимо проводить 2 и больше раз, однако если в каждую соседнюю клетку поместить  $\text{floor}(\text{sandCount}/\text{neighborsCount})$ , то в рассматриваемой клетке останется  $\text{sandCount} \% \text{neighborsCount}$  песчинок, что меньше  $neighborsCount$ . При этом проводить дополнительные обвалы в этой клетке не понадобится.

### Реализация

В программе реализованы три различные сетки: треугольная, квадратная и шестигранная.

## Особенности системы программирования

Программа написана на языке программирования c++ с использованием дополнительной графической библиотеки SFML. Проект собран на g++ (Ubuntu 11.4.0-1ubuntu1~22.04) 11.4.0.

### Используемые переменные

`map<pair<int, int>, int> grid` - сетка;

`set<pair<int, int>> unstableCells` - список нестабильных клеток;

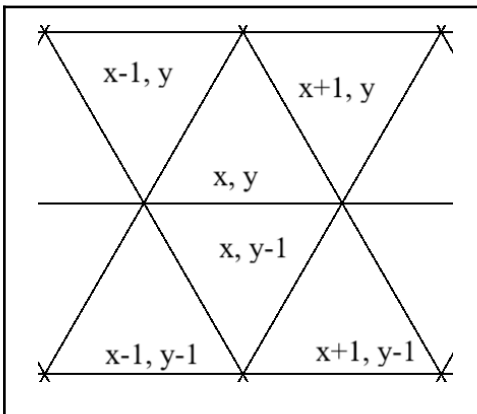
`int neighborsCount` - количество соседей;

`vector<pair<int, int>> neighbors` - список соседних клеток;

`bool oriented` - есть ли ориентация для клеток;

`vector<pair<int, int>> topNeighbors` - список соседей для ориентированных вверх клеток;

`vector<pair<int, int>> bottomNeighbors` - список соседей для ориентированных вниз клеток.



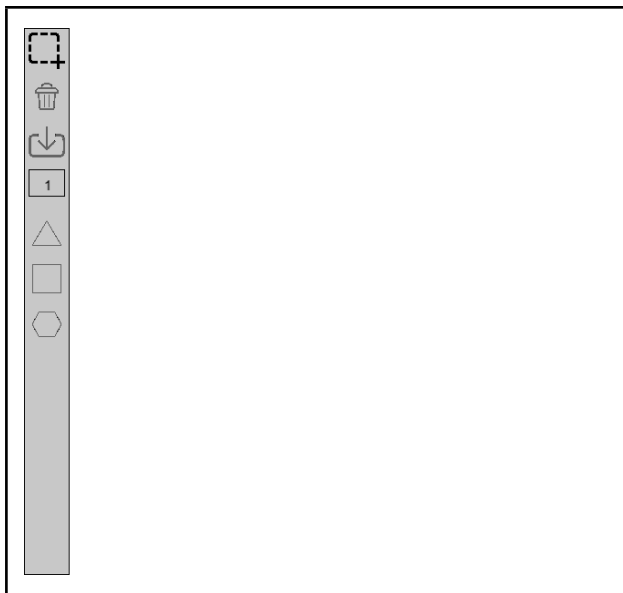
Для треугольной сетки необходима ориентация клеток: соседями клетки (x, y) являются (x-1, y), (x+1, y), (x, y-1), соседями клетки (x, y-1) являются (x, (y-1)+1), (x-1, y-1), (x+1, y-1). Если в сетке используется ориентация клеток, то для определения соседей используются массивы `topNeighbors` и `bottomNeighbors`. Иначе используется массив `neighbors`.

### Используемые функции

<pre>void addSand(int x, int y, int sandNumber) {     grid[{x, y}] += sandNumber;     if (grid[{x, y}] &gt;= neighborsCount)     {         unstableCells.insert({x, y});     } }</pre>	Добавляет <code>sandNumber</code> песчинок в клетку с координатами (x, y). Если после добавления песчинок их количество в клетке стало больше количества соседей этой клетки, то рассматриваемая клетка заносится в список нестабильных.
<pre>void toppleNotOriented(int x, int y) {     for (std::pair&lt;int, int&gt; neighbor : neighbors)     {</pre>	Проводит обвал в клетке с координатами (x, y) при неориентированной сетке.

<pre>         addSand(x + neighbor.first, y + neighbor.second, grid[{x, y}] / neighborsCount);     }      grid[{x, y}] %= neighborsCount; } </pre>	
<pre> void toppleOriented(int x, int y){...} </pre>	Проводит обвал в клетке с координатами (x, y) при ориентированной сетке.
<pre> void updateGrid() {     while (!unstableCells.empty()) {         std::pair&lt;int, int&gt; cell = *unstableCells.begin();         topple(cell.first, cell.second);         unstableCells.erase(cell);     } } </pre>	Проводит обвал во всех нестабильных клетках.

## Организация UI



При запуске появляется окно. Справа находится панель управления. На ней расположены следующие инструменты: скриншот, очистить доску, произвести обвал, выбор количества насыпаемых за раз песчинок, выбор треугольной сетки, выбор квадратной сетки, выбор шестиугольной сетки. Кнопка скриншот создаёт снимок окна и сохраняет его в папку screenshots/ (если папка отсутствует, то скриншот не будет сохранён). Кнопка очистить доску очищает доску от песчинок. Кнопка произвести обвал производит обвал песчинок. При нажатии сохраняет своё

положение до того момента, пока не будет нажата повторно (после нажатия обвалы будут происходить до тех пор, пока кнопка не будет нажата повторно). Счётчик выбор количества песчинок позволяет выбрать количество насыпаемых за раз песчинок. При нажатии на него правой кнопкой мыши количество песчинок увеличивается, при нажатии левой кнопкой мыши - уменьшается. Кнопки выбор треугольной сетки, выбор квадратной сетки, выбор шестиугольной сетки меняют сетку на выбранную. Изначально никакая сетка не является выбранной, поэтому окно не рисует сетки.

## Горячие клавиши

w, a, s, d - перемещение по карте. Стрелочки вверх и вниз - приближение карты. р - сохранить скриншот. Пробел - произвести разовый обвал. с - очистить доску. ЛКМ по доске - перемещение по карте. ПКМ по доске - насыпать п песчинок, где п - количество песчинок на счётчике. Скрол колёсика мыши - приближение карты.

## Текст программы

Программа находится на моём github:

<https://github.com/Lolevtexa/3rdSemester-AISD-CourseWork>

## Пример работы

