Механізми обробки кладки

Ми виділили 5 механізми кладки:

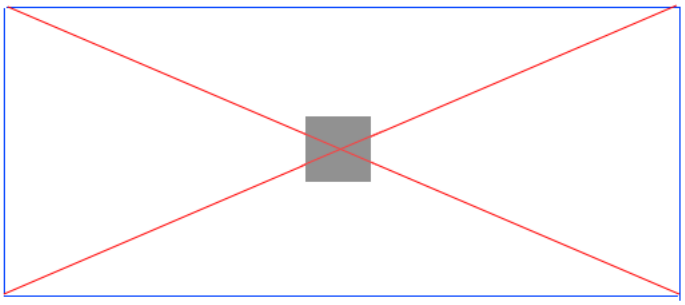
1) Прямий. Край до краю (Straight)
2) Прямий.Вертикальний відступ(StraightIndentTop)
3) Прямий.Горизонтальний відступ(StraightIndentLeft)
4) З поворотом 90 градусів.Квадратна плитка (AnglewiseCube) доступно в меню при a=b
5) З поворотом 90 градусів.Прямокутна плитка(AnglewiseRectangle) доступно при a = b/2 або b = a/2

Тепер конкретніше

Для укладки плитки використано наступний алгоритм:

1) Визначаємо крайні точки кімнати,
2) Визначаємо, яка стіна задана в параметрі, якщо задано від центру див пункт 3, задано від стіни див пункт 4
3) Для укладки від центру використовуємо наступний алгоритм:

3.1) Визначаємо кординати центру, і встановлюємо перший елемент по центру



3.2) Відраховуємо кількість ітерацій до лівого та кількість ітерацій до правого краю

3.3) Відраховуємо кілбкість ітерацій до верхнього краю

3.4) Далі проходимо 4 цикли і закладаємо проміжні плитки

4) Для укладки від стіни використовуємо аналогічний метод, різниця полягає тільки у наступному

4.1) Визначаємо тип стіни горизонтальна чи вертикальна

4.2) Якщо горизонтальна відраховуємо так щоб плитка не вирізалася, рівняємо по верху і відраховуємо кількість ітерацій до низу і до верху

4.3) Якщо вертикальна, рівняємо по лівому краю. Рахуємо кількість ітерацій зліва і з права

4.4) Проходимо 2 цикла

Примітка. Ідея укладання плитки програмою наступна: ми закладаємо усю площу , а функція cutTile (tile.php) створена для відсіювання плиток які не попадають в область укладки

Перебір точок циклом

Алгоритми перебору точок зображені на рисунках
top – у кова кордината крайньої лівої точки

Блок Третій

Принцип роботи чарівної функції cutTile() (tile.php) надалі всі функції лежать у файлі tile.php

1) На вході отримуємо чотири точки розпочинаємо цикл в чотири ітерації – стандарт плитки
2) Беремо першу точку, спочатку перевіряємо чи входить вона до площини кімнати, за допомогою функції pointFixedToRoomPlane ();
3) Якщо точка входить додаємо її у вихідний масив координат, якщо ні, пропускємо
4) У цьому ж кроці циклу обираємо наступну точку проводимо відрізок і перевіряємо, чи ніяка сторона не перетинає його
5) Якщо перетинає слідуємо наступним крокам,

5.1) Якщо перетинів 2, записуємо першу точку входження, перетин цих двох відрізків, і останню точку входження
5.2) Перетин 1 і прямої немає в буфері – записуємо точку входження записуємо пряму в буфер
5.3) Перетин 1 і пряма в буфері ідентична знайденій прямій – записуємо точку входження очищаємо буфер
5.4) Пряма не ідентична заданій прямій – записуємо точку їхню перетину, записуємо точку входження
6) наступа ітерація