ДОДАТОК А

Код реалізації оптимізованого алгоритму A*

```
private ArrayList<Cell> open;
private ArrayList<Cell> closed;
private HashMap<Cell, Cell> parents;
private Cell current;
private Cell[] neighbours;
private Direction curDir;

public void runFastestPath(int goalRow, int goalCol) {
        long start = System.nanoTime();
        do {
                current = minimumCellCost(goalRow, goalCol);
                if (parents.containsKey(current)) curDir = getTargetDir(parents.get(current).getRow(), parents.get(current).getCol());

                closed.add(current);
                open.remove(current);
                if (checkIfGoarFound(goalRow, goalCol, start)) return;
                if (exploredMap.checkValidCoordinates(current.getRow() + 1, current.getCol())) {
                        neighbours[0] = exploredMap.getCell(current.getRow() + 1, current.getCol());
                        if (!canBeVisited(neighbours[0])) neighbours[0] = null;
                }
                if (exploredMap.checkValidCoordinates(current.getRow() - 1, current.getCol())) {
                        neighbours[1] = exploredMap.getCell(current.getRow() - 1, current.getCol());
                        if (!canBeVisited(neighbours[1])) neighbours[1] = null;
                }
                if (exploredMap.checkValidCoordinates(current.getRow(), current.getCol() - 1)) {
                        neighbours[2] = exploredMap.getCell(current.getRow(), current.getCol() - 1);
                        if (!canBeVisited(neighbours[2])) neighbours[2] = null;
                }
                if (exploredMap.checkValidCoordinates(current.getRow(), current.getCol() + 1)) {
                        neighbours[3] = exploredMap.getCell(current.getRow(), current.getCol() + 1);
                        if (!canBeVisited(neighbours[3])) neighbours[3] = null;
                }

                for (int i = 0; i < 4; i++) {
                        if (neighbours[i] != null) {
                                if (closed.contains(neighbours[i])) continue;
                                if (!open.contains(neighbours[i])) {
                                        parents.put(neighbours[i], current);
                                        double v = gCosts[current.getRow()][current.getCol()] + costG(current, neighbours[i], curDir);
                                        gCosts[neighbours[i].getRow()][neighbours[i].getCol()] = v;
                                        open.add(neighbours[i]);
                                } else {
                                        double currentG = gCosts[neighbours[i].getRow()][neighbours[i].getCol()];
                                        double tempG = gCosts[current.getRow()][current.getCol()] + costG(current, neighbours[i], curDir);
                                        if (tempG < currentG) {
                                                gCosts[neighbours[i].getRow()][neighbours[i].getCol()] = tempG;
                                                parents.put(neighbours[i], current);
                                         }
                                }
                        }
    } while (!open.isEmpty());
}


private Cell minimumCellCost(int goalRow, int getCol) {
    int size = open.size();
    double minCost = RobotConstants.INFINITE_COST;
    Cell result = null;
    for (int i = size - 1; i >= 0; i--) {
        double gCost = gCosts[(open.get(i).getRow())][(open.get(i).getCol())];
        gCost = gCost / DECREASE_G_COST;
        double cost = gCost + costH(open.get(i), goalRow, getCol);
        if (cost < minCost) {
            minCost = cost;
            result = open.get(i);}}
    return result;
}
```

ДОДАТОК Б

Демонстраційний матеріал

Формат	Позиція.	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кільк.	Примітки
			Текстові документи		
А4	1	ГЮИК 468150.030 ПЗ	Пояснювальна записка	55	с.
			Інші документи		
А4	2		Код реалізації оптимізованого	1	с.
			алгоритму А*		
А4	3		Демонстраційний матеріал	14	*у форматі *.ppt*

					ГЮИК 468150.030 ВД		
Змін	А	№ докум.	Підпи	Дата			
Розроб.		Огарь А.А.			Розроблення системи автоматизації для визначення оптимальної траєкторії руху за допомогою комп'ютерного зору роботизованої мобільної платформи на виробництві	Літер	Аркуш	Аркушів
Перевір		Аллахверанов Р. Ю.				Н	1	1
						Кафедра КІТАР ХНУРЕ		
Н.конт.		Демська Н.П.			Відомість кваліфікаційної роботи			
Затв.		Невлюдов І.Ш.						