# 

# **Отчет по**

# **Лабораторной работе №3.**

# **По предмету: Artificial Intelligence**

# 

# 

# 

# **Подготовил студент группы IA2303: Gutu Nicolae**

# **Проверяющий преподаватель: V. Trebiș**

# 

# 

# **Сhișinau, 2025**

**Задание:**

**Моделирование движения пешеходов**

**Цель:**

Исследовать поведение толпы людей при движении по ограниченному пространству, реализовать и визуализировать алгоритм «Boids для пешеходов» с добавлением цели движения (выход), сравнить влияние параметров (плотность, веса правил) на проходимость узких мест.

# **1)Постановка задачи**

Реализовать 2D-среду с выходами и препятствиями; создать набор агентов (пешеходов) с начальными позициями и скоростями; на каждом такте:

* находить соседей в радиусе восприятия;
* вычислять векторы **разделения**, **выравнивания**, **сближения** (правила Boids) и **движения к цели**;
* суммировать с весами, ограничить ускорение/скорость, обновить положение;
* реализовать корректные контакты со стенами/дверями и визуализировать движение;
* собирать простые метрики (время, число эвакуированных).

# **2) Краткая теория (Boids + Goal)**

Каждый пешеход iii на шаге ttt формирует желаемое направление как взвешенную сумму «инстинктов»: движения к цели, разделения (личное пространство), выравнивания (усреднение скорости во избежание столкновений), сближения с массой, отталкивание от / скольжение вдоль стен.

На пути пешеходов располагаются искусственно созданные препятствия (стены), которые необходимо обойти чтоб попасть к выходу. В некоторых местах стены образуют узкие коридоры, в которых становится меньше места и сильнее проявляется фактор сепарации.

# 

# 

# 

# **3) Реализация (ключевые моменты)**

# Стэк: Python 3 + Pygame (визуализация), NumPy (векторы).

## **Архитектура проекта**

# **config.py** — параметры (веса, радиусы, скорости, dt, размеры окна).

# **world.py** — описание мира: двери (Exit), прямоугольные и круглые стены; вычисление направлений к выходам; отталкивание от стен; «скольжение» по внешней рамке; «прозрачность» рамки в зонах дверей; жёсткая коррекция при попадании внутрь прямоугольной стены.

# **agent.py** — логика одного пешехода: вычисление правил, ограничение ускорения/скорости, шаг движения.

# **sim.py** — шаг симуляции: поиск соседей в радиусе, обновление всех агентов, удаление попавших в Exit.

# **viz.py** — отрисовка мира, агентов и HUD (FPS | agents | t | evacuated), подсветка агентов и зон спавна.

# **main.py** — запуск, генерация мира/агентов, цикл.

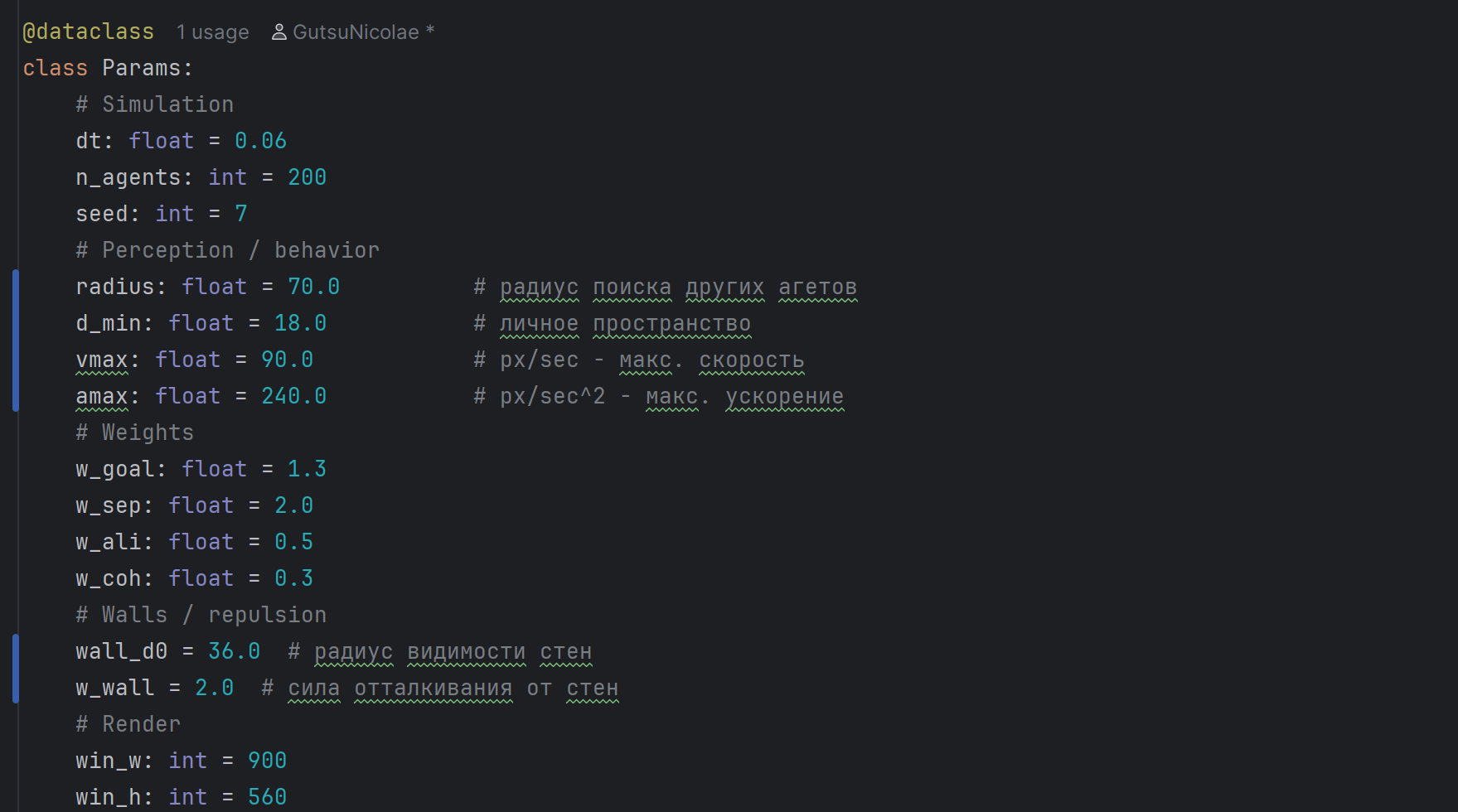
# **3.1 Движение агента**

# *рис.1 Движение агента*

Один сим-шаг для агента: смотрим на соседей и на мир (выходы/стены), обновляем скорость и позицию:

* Считаем «инстинкты» (векторы поведения) и взвешиваем
* Собираем итоговое «желание двигаться» и ограничиваем стимул
* Интегрируем скорость и делаем ограничение по скорости
* Предлагаем новый шаг позиции (без коллизий)
* Дальше — в коде мира — применяется скольжение /выталкивание о стены и проверка выхода.

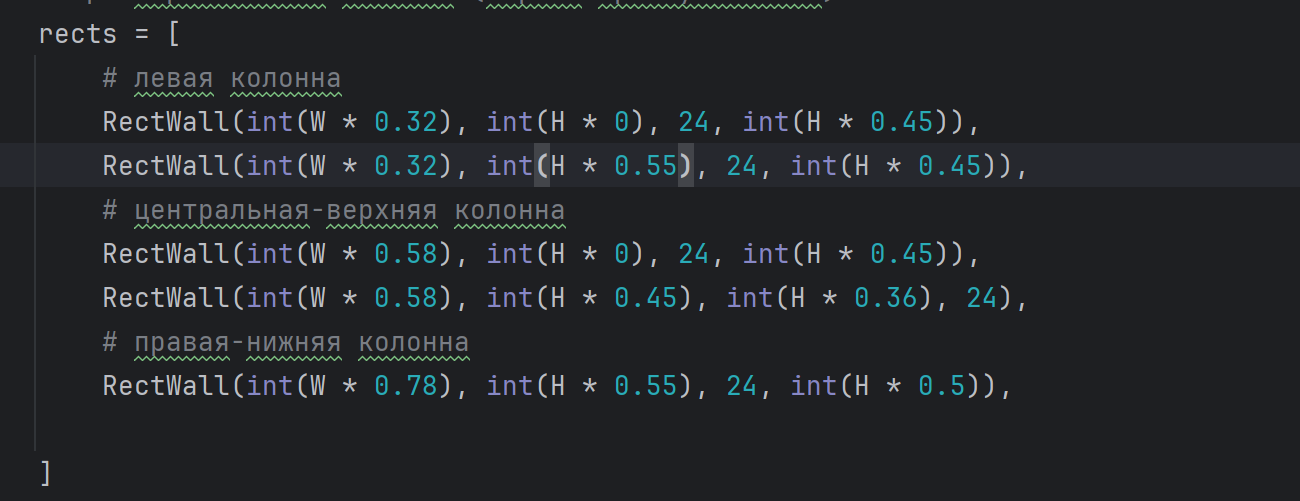
**3.2 Конфигурационный файл:**

**

*рис.2 Файл конфигурации ( “веса” инстинктов/ силы репульсии от стен и т.д)*

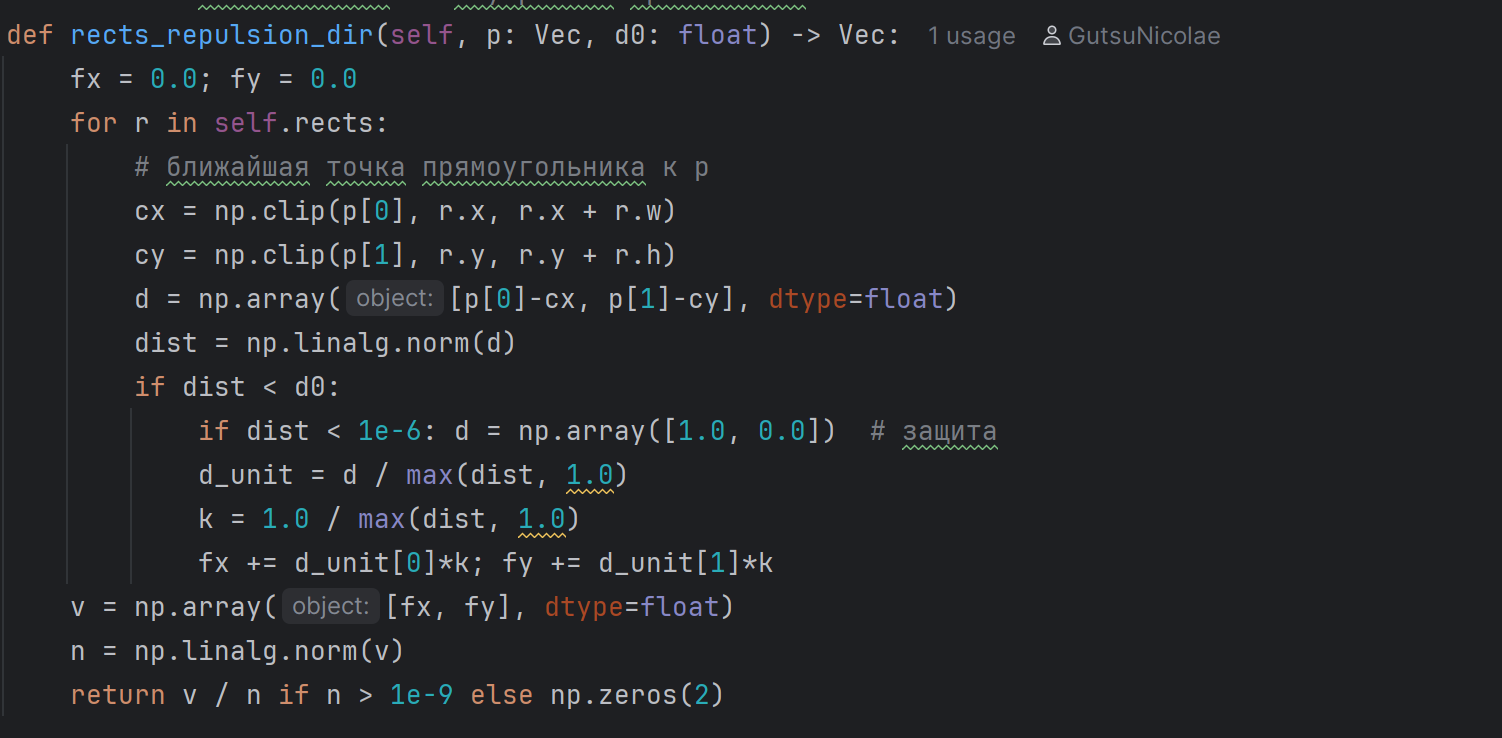
В данном файле задаются главные параметры для весов инстинктов, скорость, радиус видимости агентов/стен и т.д  
Изменяя параметры можно отдавать приоритет разным инстинктам и моделировать соответствующее поведение, экспериментировать с поведением агентов.

**3.3 Создание искусственных преград:**

*рис.3 Генерация стен*

Генерация стен в виде прямоугольников, которые потом для отрисовки передаются дальше в файл [viz.py](http://viz.py)

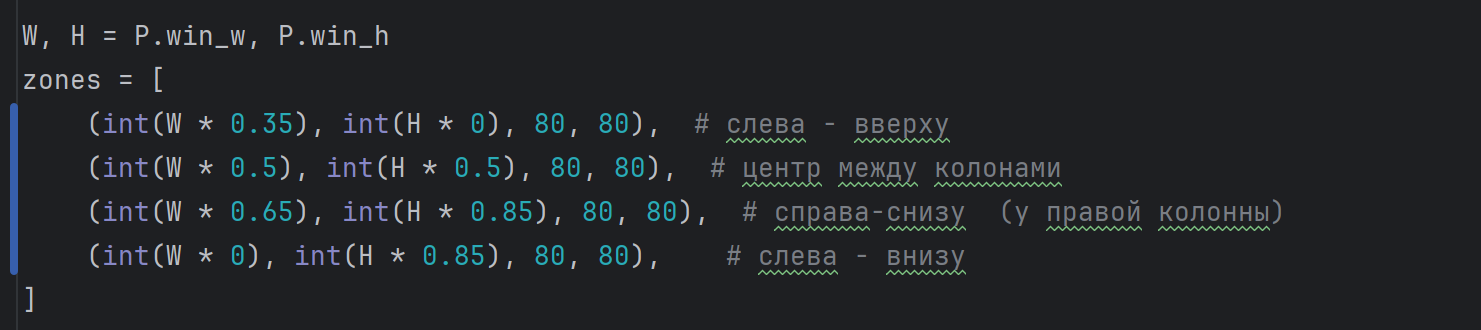
**3.4) Механизм отталкивания от стен**

*рис.4 Отталкивание от стен*

Как работает:

1. Берём каждую прямоугольную стену и находим ближайшую к агенту точку этой стены.
2. Смотрим, насколько близко агент к стене.
3. Если близко — добавляем небольшой импульс **от стены** наружу.  
    Чем ближе — тем сильнее импульс; если далеко — отталкивания нет.
4. Складываем все такие импульсы от всех стен и получаем **общее направление, куда лучше сместиться**, чтобы не сталкиваться об стены.
5. Возвращаем направление (а не большую силу) — дальше вес этого правила задаётся параметром w\_wall.

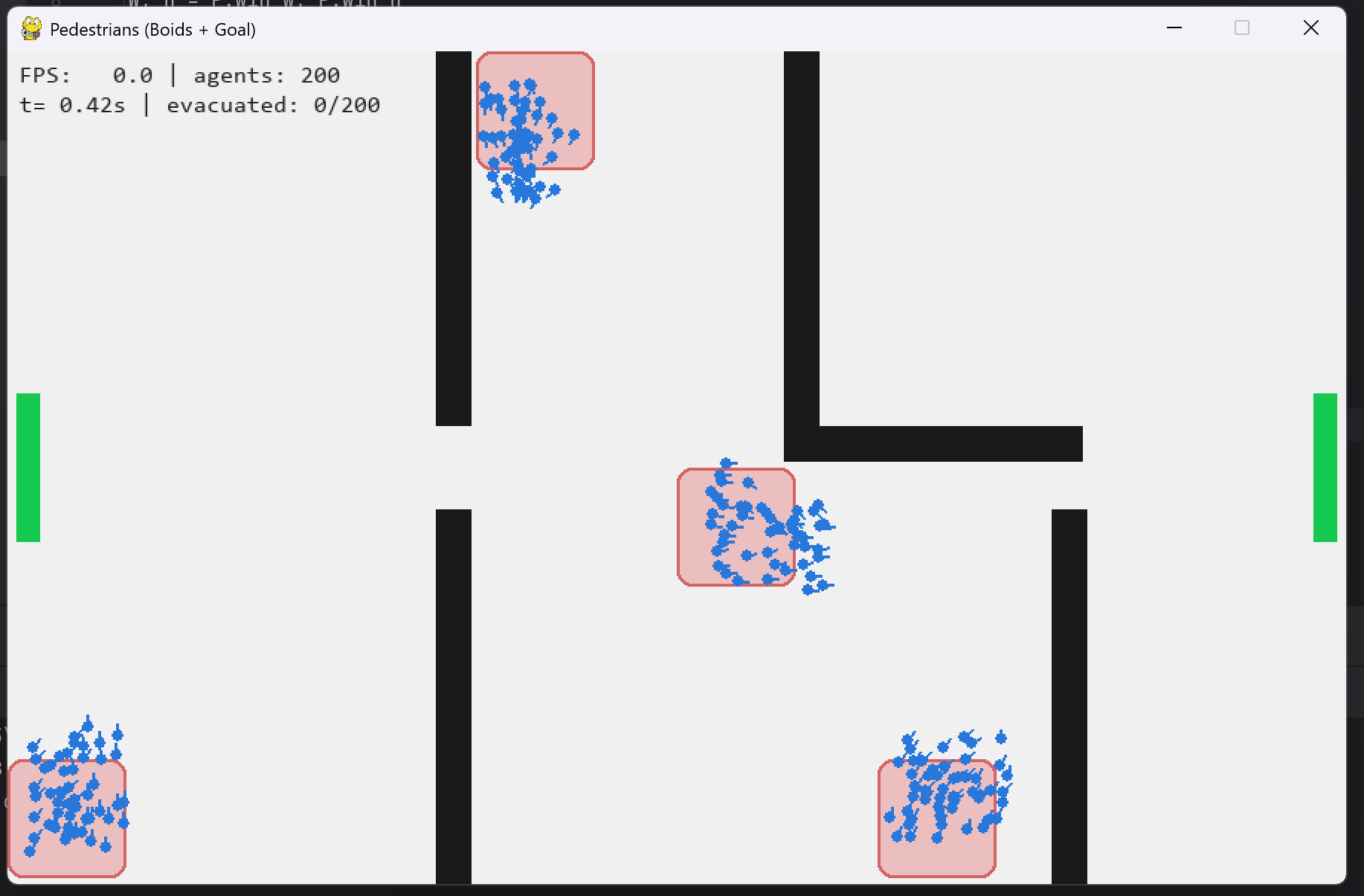
**3.5 Точки спавна агентов**

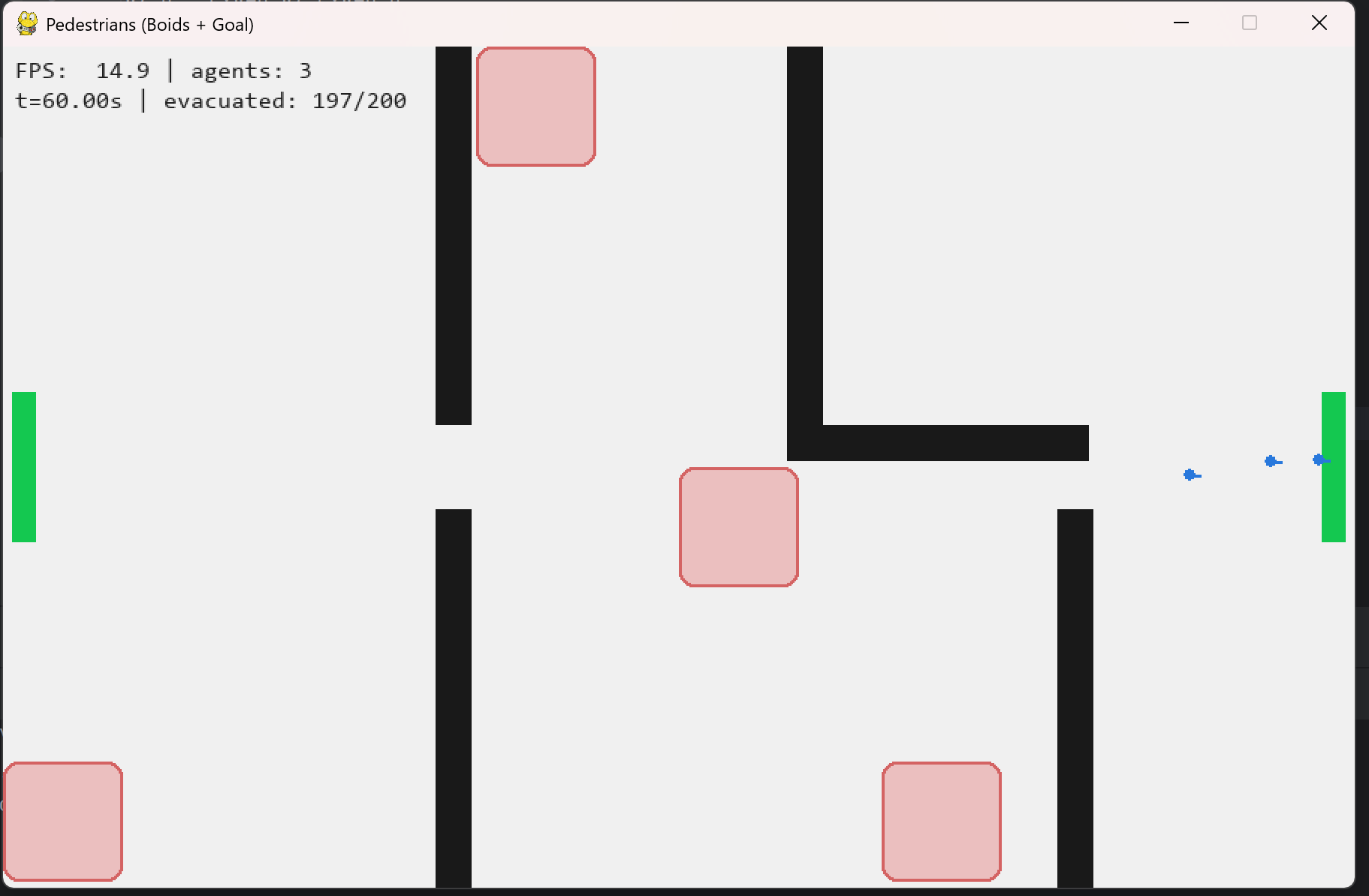
*рис.5 Спавн агентов*

Всего таких мест 4 - для демонстрации как агенты реагируют на разные препятствия и как стартовые условия оказывают влияние на их поведение.

**4) Результаты и экспериментирование с конфигурационными данными**

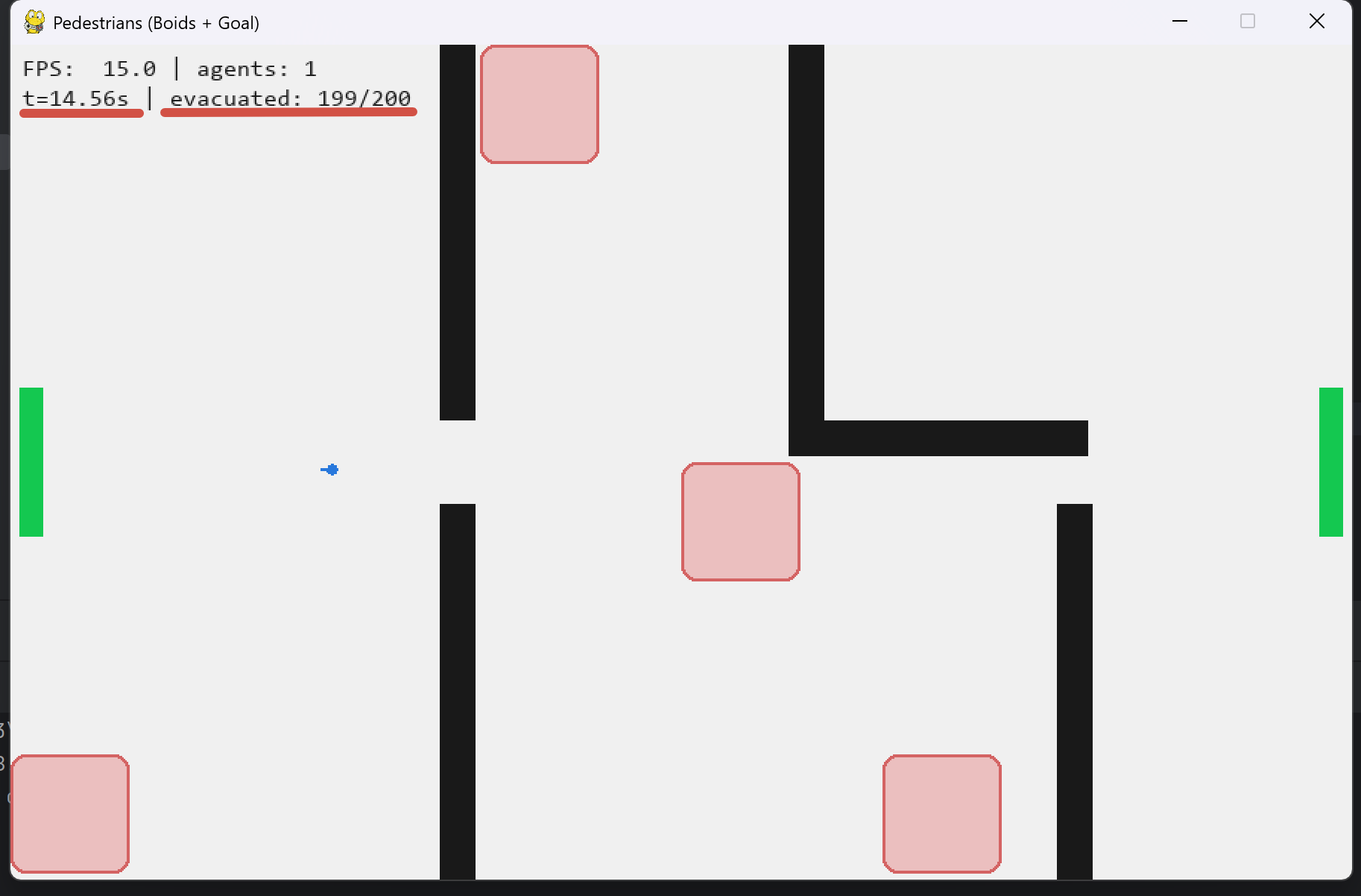
**4.1 “Стартовые” условия агентов:**

*******рис. 6.1 Начальная позиция при базовых вводных*

*рис. 6.2 Результат 1-го прохождения*

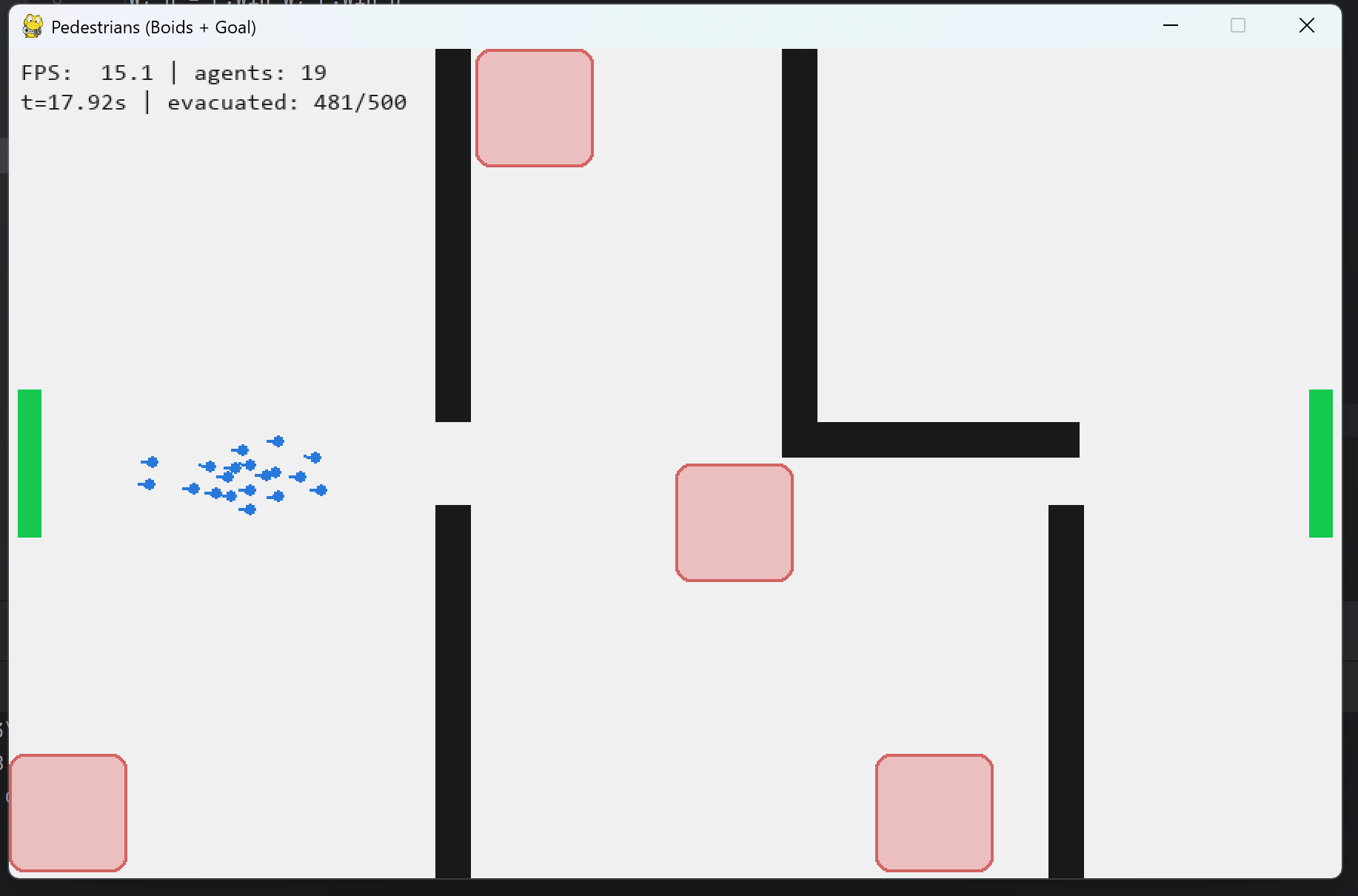
При первом прохождении комнаты у агентов был выставлен высокий “коэффициент воздействия” стен на их поведение. Как итог - многие долго проходили коридоры даже в одиночку, из-за узких пространств, из-за чего последнему агенту, чтоб покинуть комнату потребовалось ~61 секунда.

**4.2 Воздействие стен увеличено, но радиус их видимости уменьшен**

*рис. 7 Воздействие стен изменено*

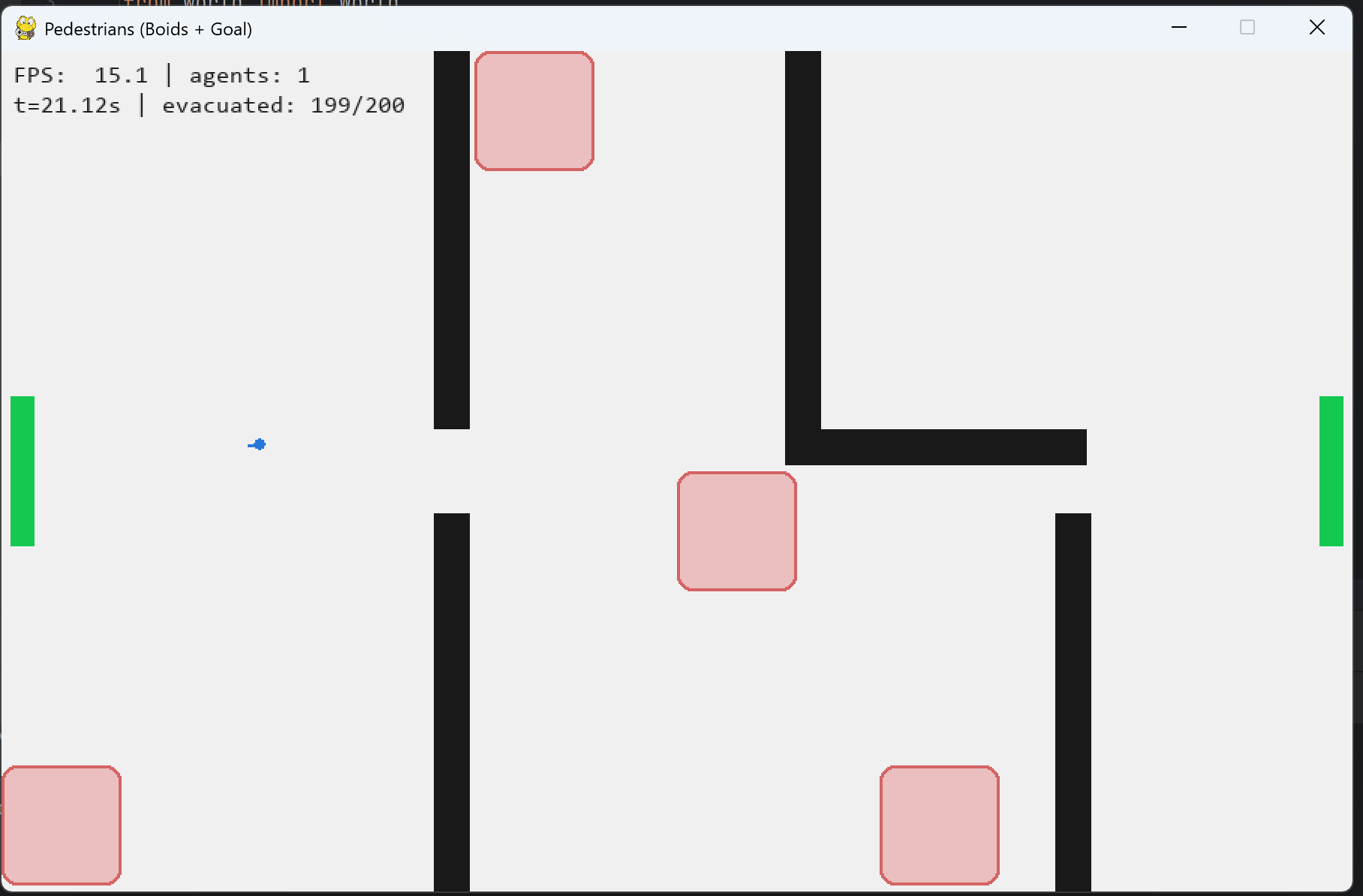
Радиус стен был уменьшен в 2 раза, но сила отталкивания увеличена в полтора раза. Таким образом, агенты стали меньше “бояться” узких коридоров, но стали в них аккуратнее.

**4.3 Увеличение числа агентов**

*рис. 8 Увеличение числа агентов до 500.*

При росте числа агентов (с 200 до 500) большой разницы во времени не наблюдается, так как они все также стремятся к “центру массы”, но их скорость коррелируется соседями, из-за чего их движение остается структурированным

**4.4 Увеличение влияния фактора Separation**

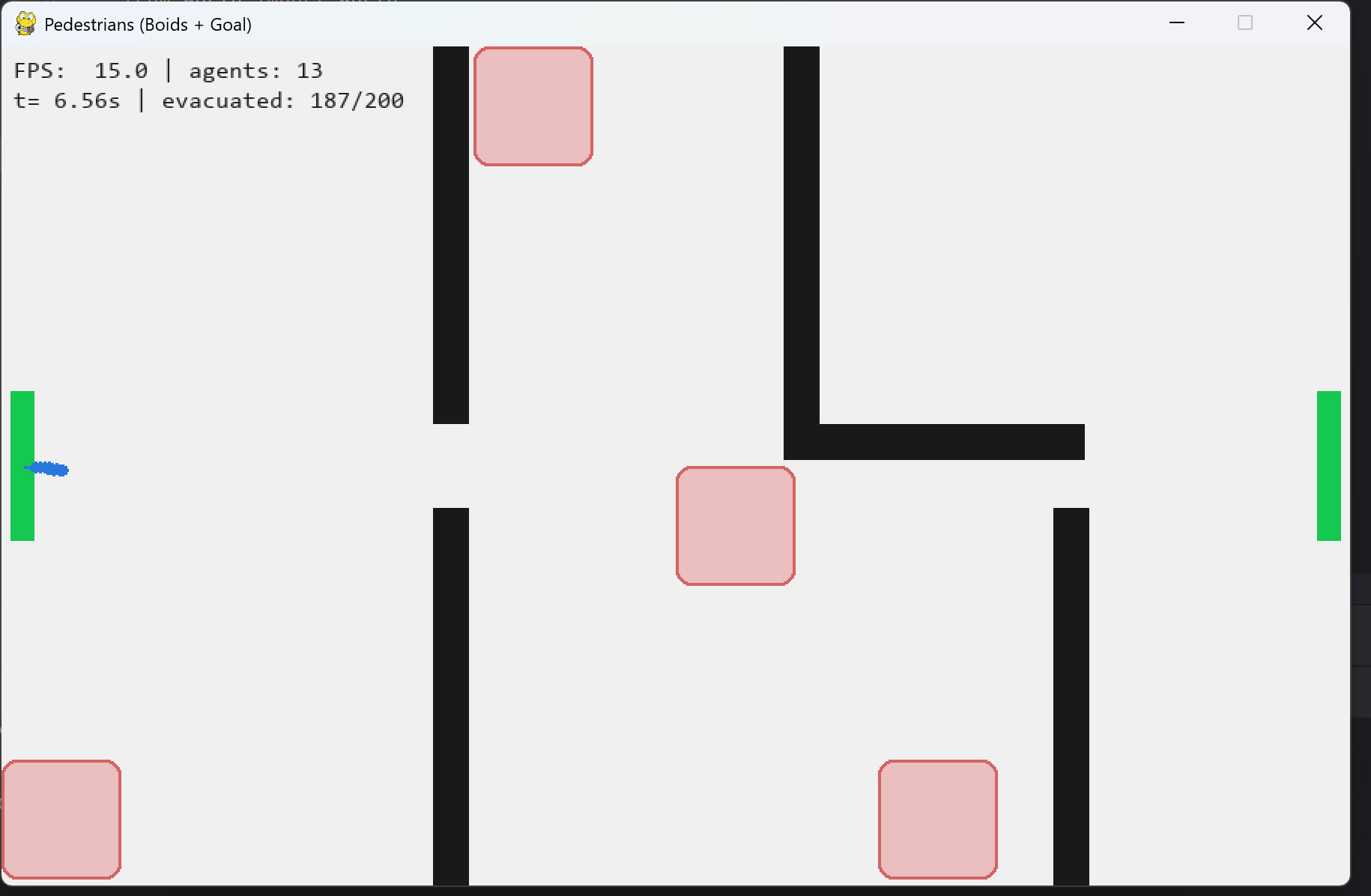


*рис. 9 Увеличение коэффициента Separation с 2 до 4*

При увеличении коэффициента сепарации время выполнения задания увеличилось

на ~40%. Это обусловлено тем, что сохранять строй в узких проходах стало гораздо труднее, так как ограничение пространства заставляло конфликтовать параметры “стремления к центру массы” и “Разделения”. Но так как инстинкт личного пространства в данном эксперименте был выведен на первое место, то скорость прохождения узких мест заметно упала, так как “живая масса” рассредоточилась.

**4.6 Приоритет - выход**

*рис. 9 Коэффициент важности выхода увеличен дважды.*

В данном эксперименте параметр Goal (стремление к выходу) был увеличен в 2 раза, что показывает значительный прорыв во времени эвакуации из комнаты.   
Результат улучшился на ~65%. Так как цель была выведена в приоритет, инстинкт сепарации частично подавлялся, аналогично как и избегание стен, что позволило увеличить “пропускную способность” узких проходов и большему числу агентов пройти по коридору одновременно.

**Вывод**

Модель *Boids + цель* адекватно воспроизводит поведение толпы: выстраивание очередей у узких мест, объезд препятствий и «скольжение» вдоль стен. В ходе экспериментов на результат сильнее всего сильнее всего повлияли: **увеличенная важность цели** — *w\_goal ×2*: потоки быстрее «собираются» к выходам, меньше колебаний возле дверей, итоговое время эвакуации сокращается, и сила/радиус влияния стен w\_sep и w\_wall, иначе ~~появляется клаустрофобия~~ возрастает “боязнь узких проходов” и локальные застревания.