



Федеральное государственное бюджетное образовательное учреждение
высшего образования «МИРЭА – Российский технологический университет»
Детский технопарк «Альтаир»

Симулятор современного танка «Modern Tanks»

Презентацию подготовили

Ученики группы 3-Д2:

Абуд Р. Н.

Сотник П. Д.

Руководитель: Борисов А.И.

Преподаватель Детского технопарка «Альтаир»

Цели и задачи

Цели:

Цель проекта состоит в том, чтобы сделать реалистичную игру, помогающую понять, как работает современная бронетехника от лица экипажа (а именно наводчика).

Задачи:

Реализовать псевдо 3D мир;

Реализовать весь основной функционал современных танковых систем управления огнем;

Сделать понятное обучение, помогающее понять, как работают главные механики игры;

Сделать несколько уровней, раскрывающих механики игры.

Актуальность

В связи с кратным увеличением производства и модернизации современных танков и, следовательно, увеличением потребности в обученных танковых экипажах, данный симулятор будет очень полезен для подобных задач.



Возможности проекта

Заходя в игру, пользователь видит основное меню, в котором можно перейти в окно с настройками, где можно менять разрешение, выбирать ФПС, регулировать громкость и т.д., и окно с выбором уровня, техники, боеприпасов с кратким описанием;

В самой игре пользователь может включать/выключать СУО (система управление огнем), наводить орудие по вертикали и горизонтали с помощью стрелок на клавиатуре или с помощью мышки, игрок может переключаться между тепловизионным и оптическим каналами прицела и менять их приближение. Также доступны, такие функции СУО как замер дистанции до цели и захват на АСЦ (автомат сопровождения цели). Доступен выбор боеприпасов и выстрел из основного орудия на левую кнопку мыши. Пользователь может перемещаться по карте со спецификой техники.

Ход разработки

Разработка началась с создания главного меню игры и настроек. Был реализован класс Settings со всеми параметрами и константами, база данных, где сохраняются выбранные пользователем настройки.



```
13 class Settings:
14     # Ryan +2 *
15     def __init__(self):
16         # база данных
17         self.bd = DBController('ModernTanksDB')
18         # разрешение
19         self.monitors = screeninfo.get_monitors()
20         self.monitor = int(self.bd.select(table: 'monitor_table', titles: 'id')[0][0])
21         self.monitor_on_text = self.monitor
22         self.width_m = self.monitors[self.monitor].width
23         self.height_m = self.monitors[self.monitor].height
24         self.WIDTH = self.bd.select(table: 'size_table', titles: 'width')[0][0]
25         self.HEIGHT = self.bd.select(table: 'size_table', titles: 'height')[0][0]
26         if (self.width_m < self.WIDTH or self.height_m < self.HEIGHT) and self.bd.select(table: 'full_table', titles: 'off')[0][0]:
27             self.WIDTH = self.width_m
28             self.HEIGHT = self.height_m
29             self.bd.update_to_db(table: "size_table", title: "(width, height)", value: f"({self.WIDTH}, {self.HEIGHT})")
30         self.size_list = list(filter(lambda x: x[0] / x[1] == 16 / 9, pygame.display.list_modes(display=self.monitor)))
31         # размер текста
```

Ход разработки

Следующим шагом стала реализация 2D-составляющей игры. Была реализована карта с соответствующим классом, началась разработка класса Tank, пока что только с движением по карте с коллизиями. В нынешнем положении игры это можно заметить на примере миникарты.



```
class Map:
    def __init__(self, mini_map, tile_w, tile_h, width=0, color=(50, 60, 50)):
        # миникарта
        self.mini_map = mini_map
        self.world_map = set()
        self.world_map_dict = {}
        # параметры миникарты
        self.tile_w = tile_w
        self.tile_h = tile_h
        self.width = width
        self.color = color
        # словарь и множество с координатами стен
        for j, row in enumerate(self.mini_map):
            for i, char in enumerate(row):
                if char != '.':
                    self.world_map.add((i, j))
```

```
class Tank:
    def collisions(self, dx, dy):
        self.stuck = False

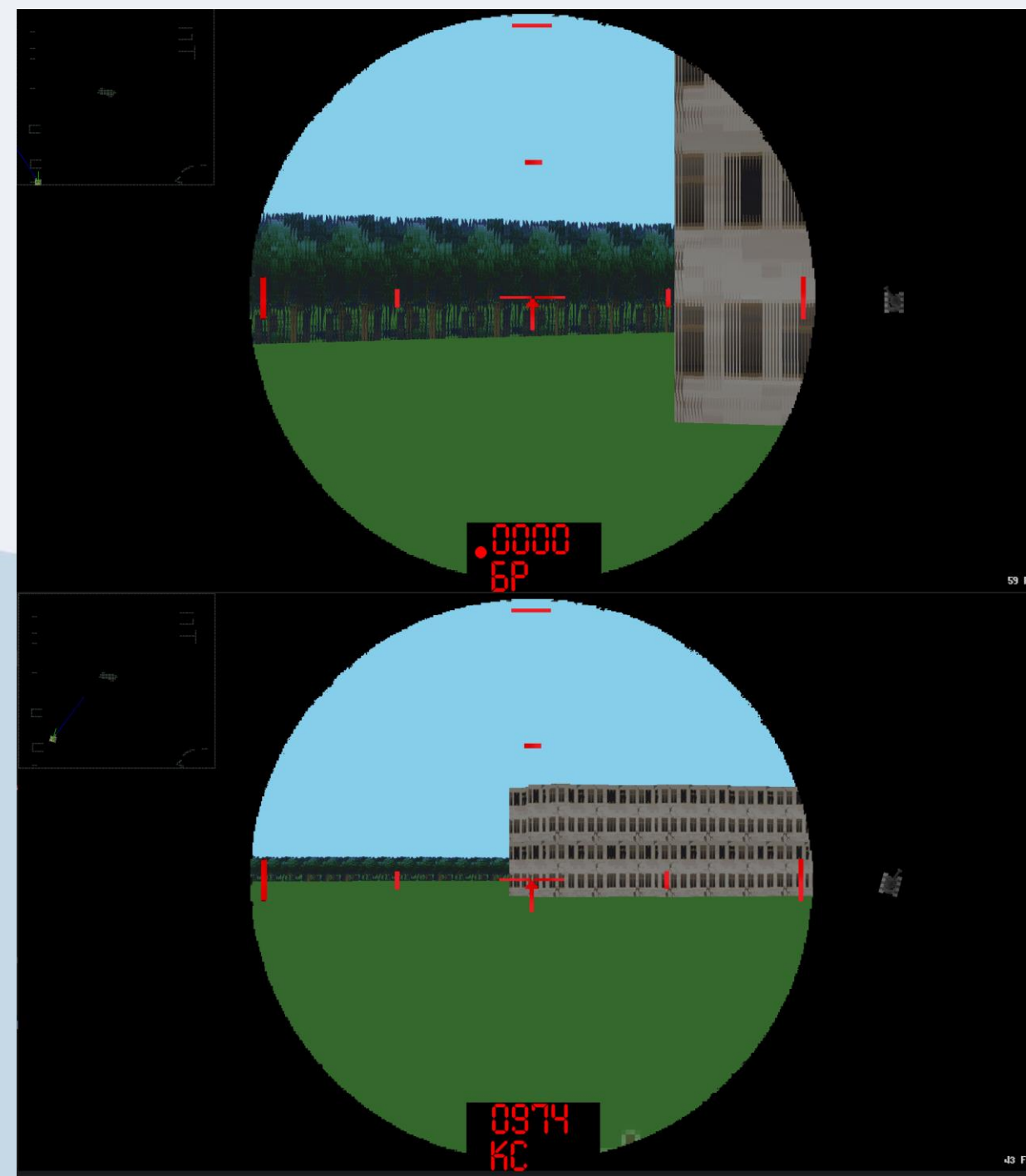
        ty = 0
        if dy != 0:
            ty = self.side // 2 * abs(dy) / dy
        tx = 0
        if dx != 0:
            tx = self.side // 2 * abs(dx) / dx

        # проверка по x
        if self.check_wall((self.x + tx + dx) // self.s.tile_w, (self.y) // self.s.tile_h):
            if tx > 0:
                temp = ((self.x + tx + dx) // self.s.tile_w) * self.s.tile_w
            else:
                temp = ((self.x + tx + dx) // self.s.tile_w + 1) * self.s.tile_w
            self.x = temp - tx
            self.stuck = True

        # проверка по y
        if self.check_wall(self.x // self.s.tile_w, (self.y - dy - ty) // self.s.tile_h):
```

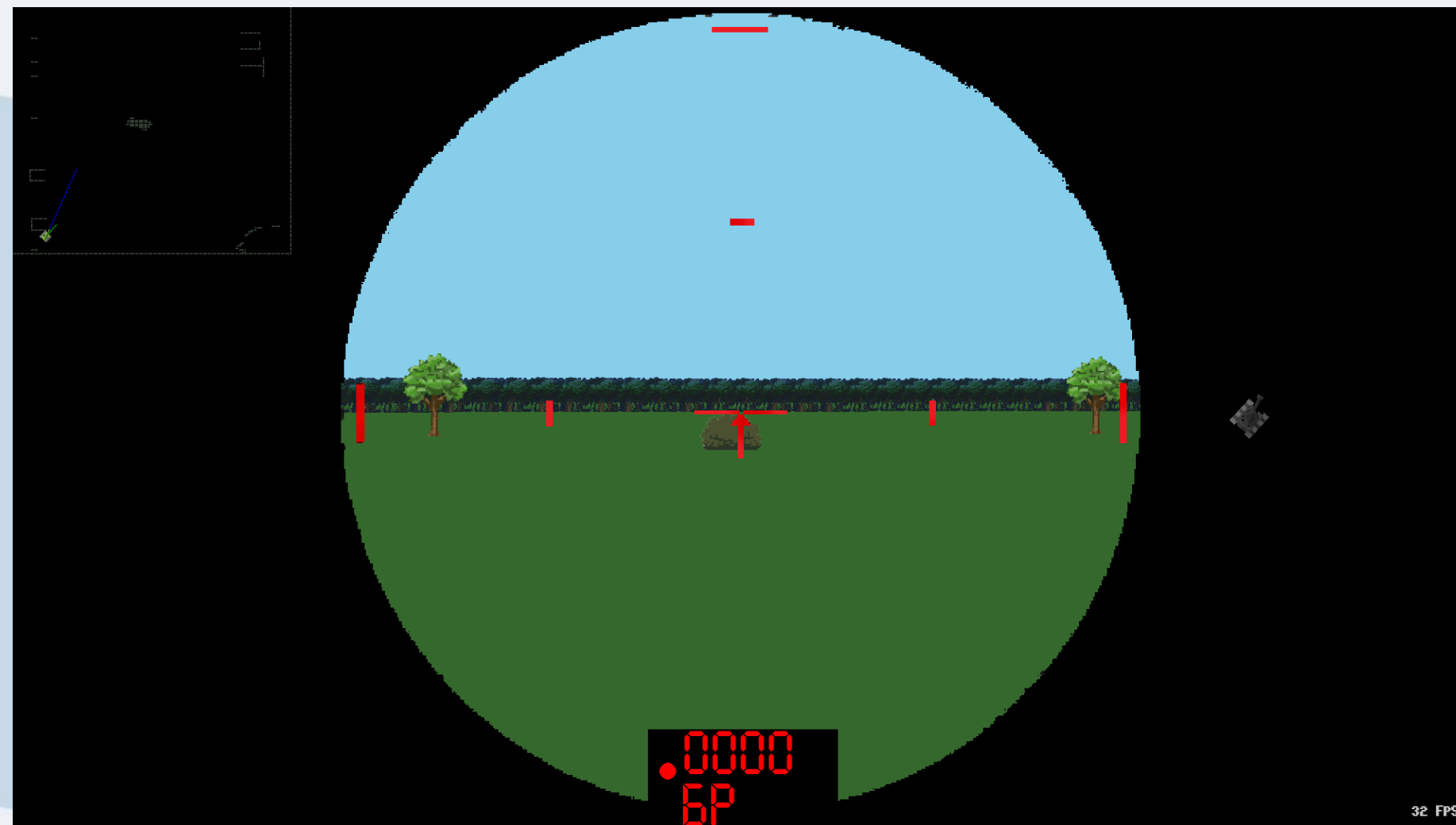

Ход разработки

Далее была реализована главная сложность - Ray casting. На разработку и тестирование на баги пришлось потратить немало времени, но в итоге все получилось. После этого были сделаны прицелы с изменением кратности и некоторыми функциями СУО.

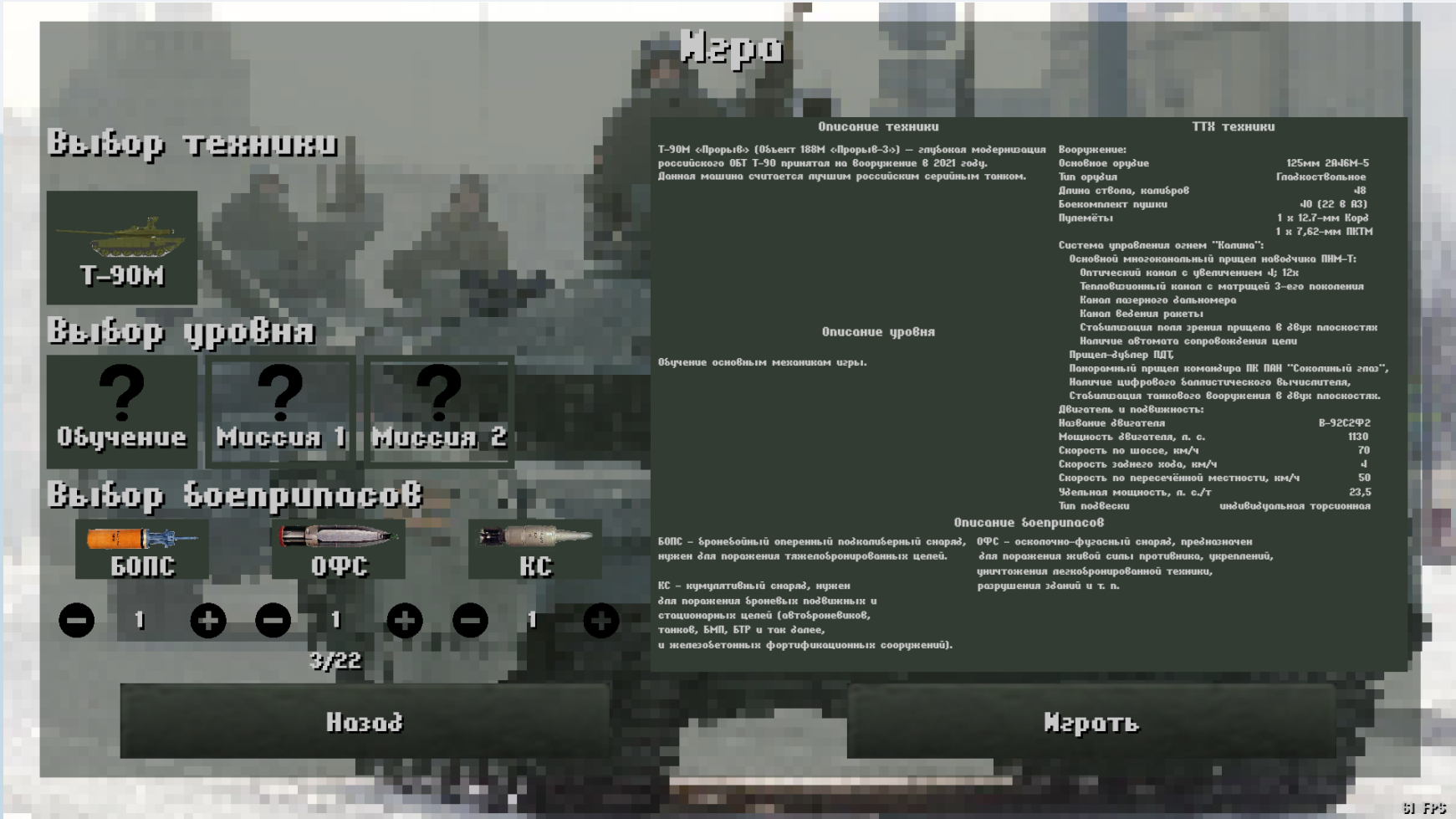


Ход разработки

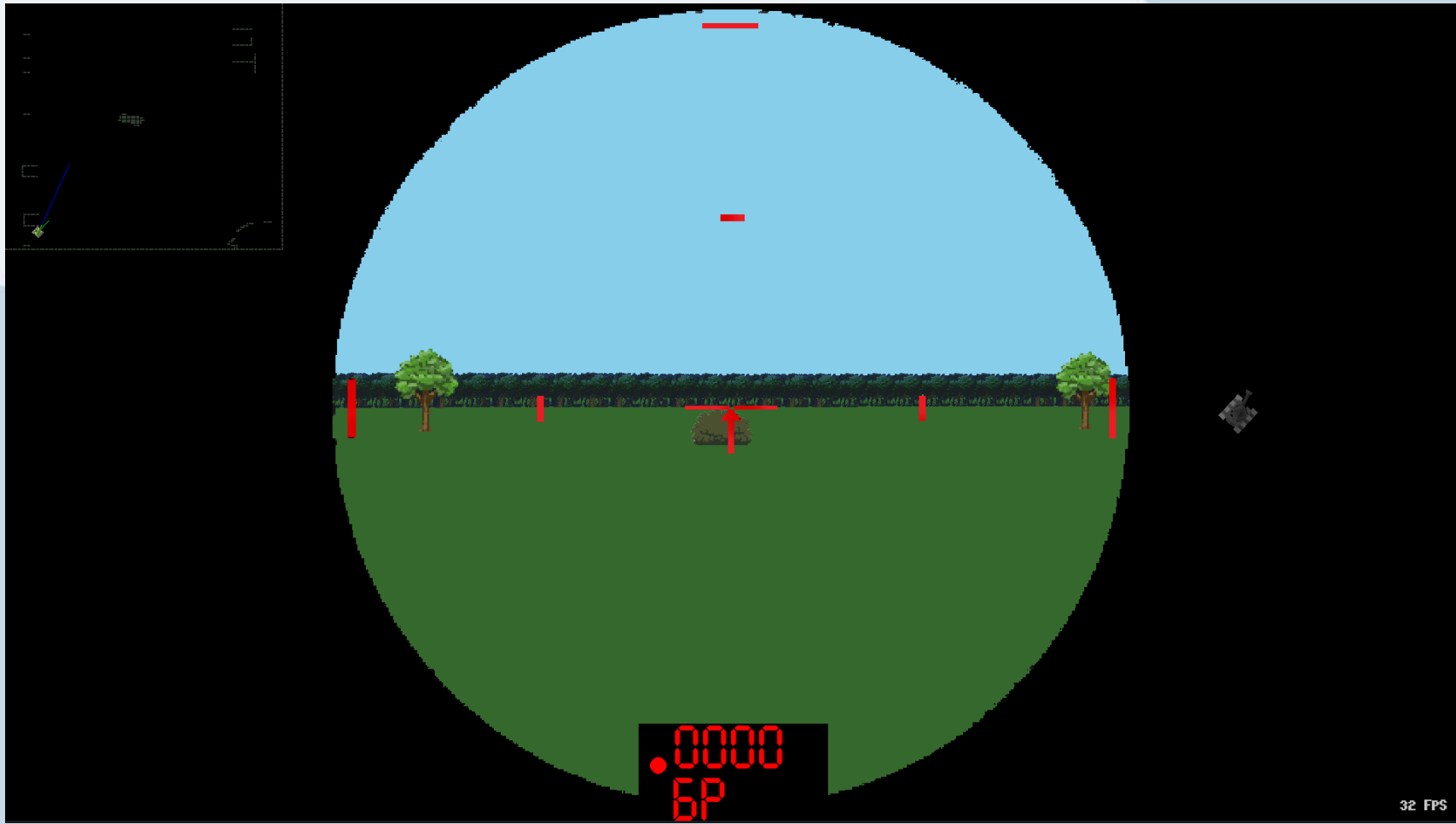
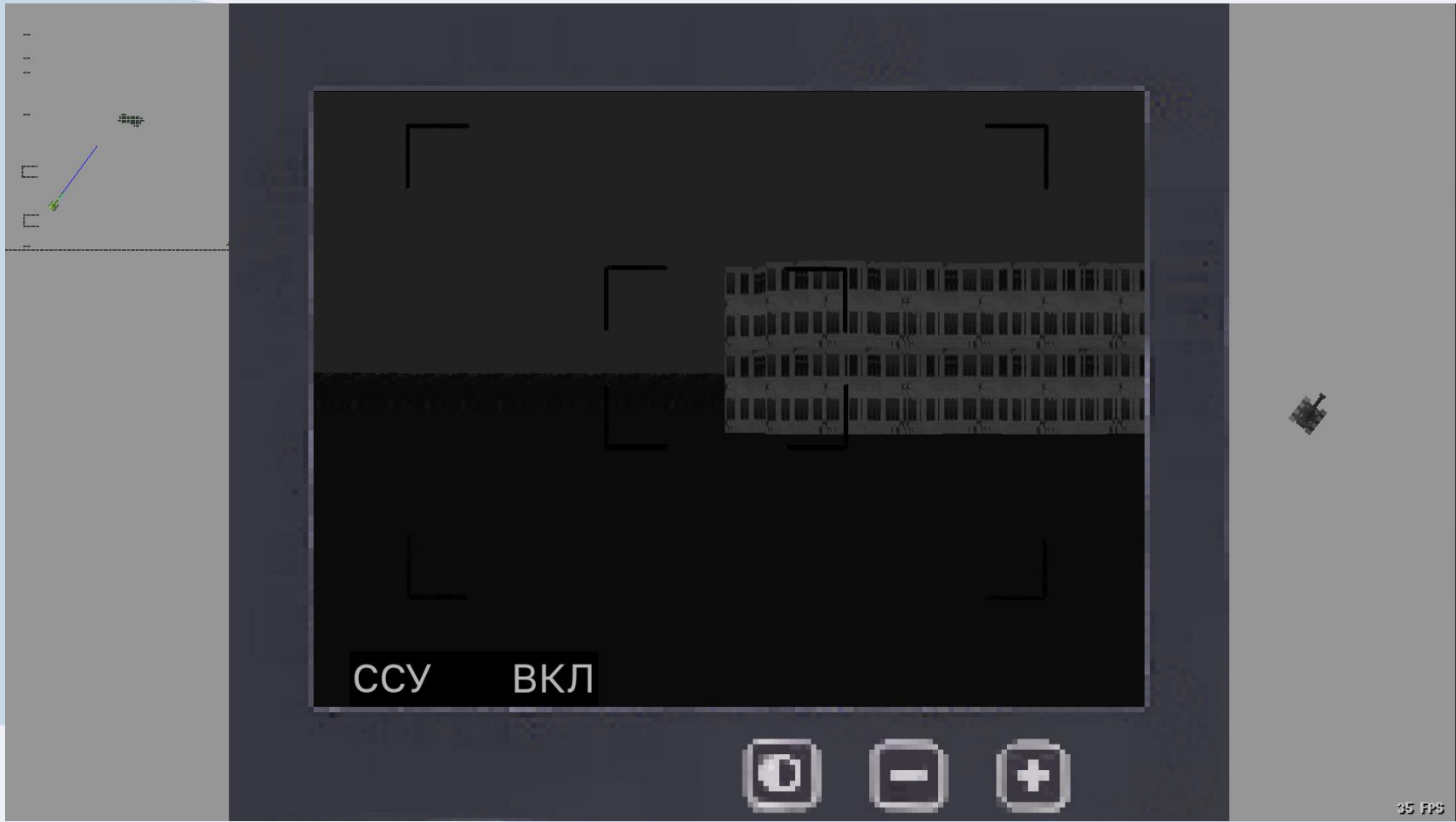
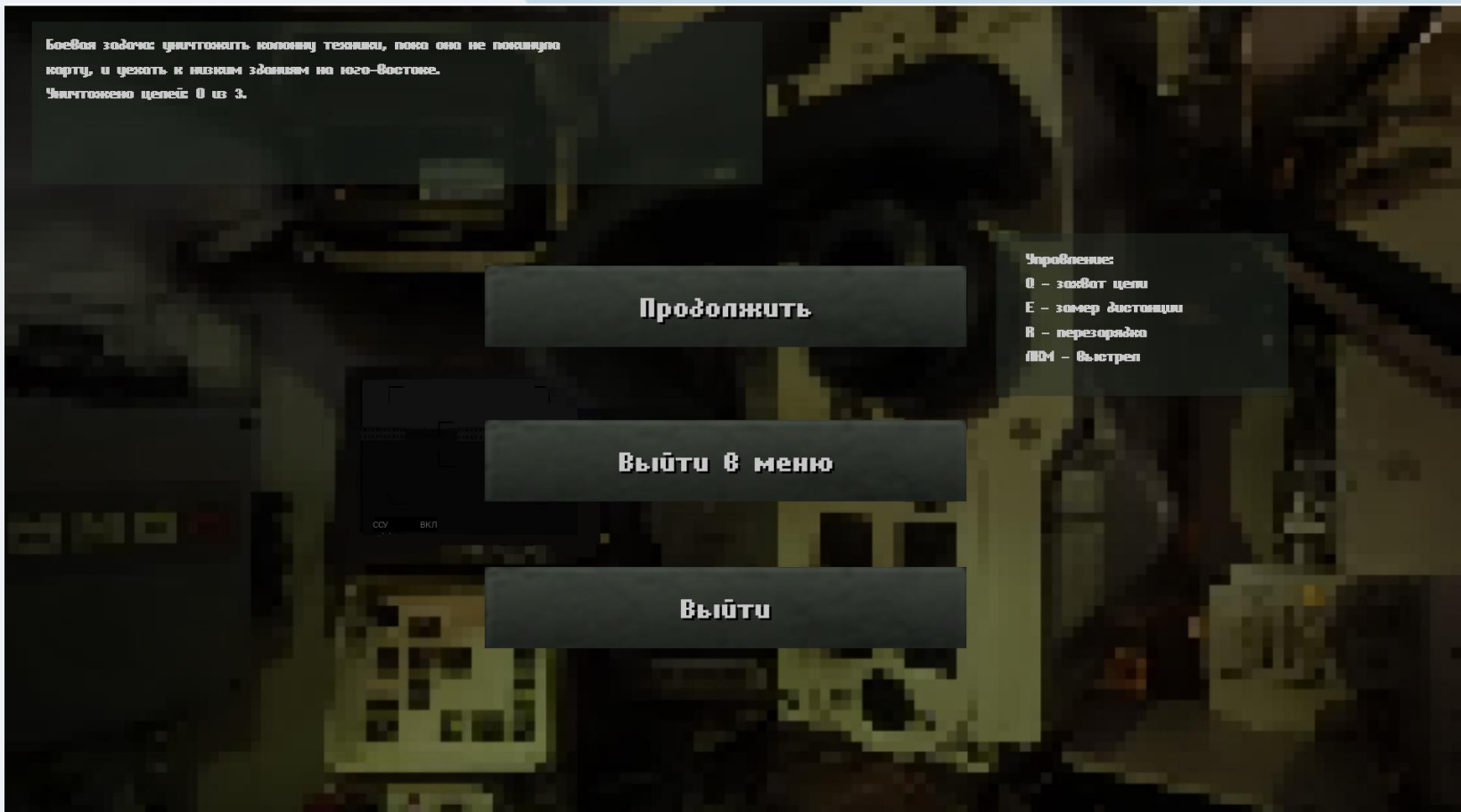
После реализации остальных функций и механик следующей сложностью стали спрайты. Для отображения спрайтов в Ray casting'e были созданы отдельные классы со всей их логикой, также были созданы объемные спрайты, состоящие из набора по восемь картинок. В конце разработки были доделаны финальные окна, уровни и урон по спрайтам.



Результат работы



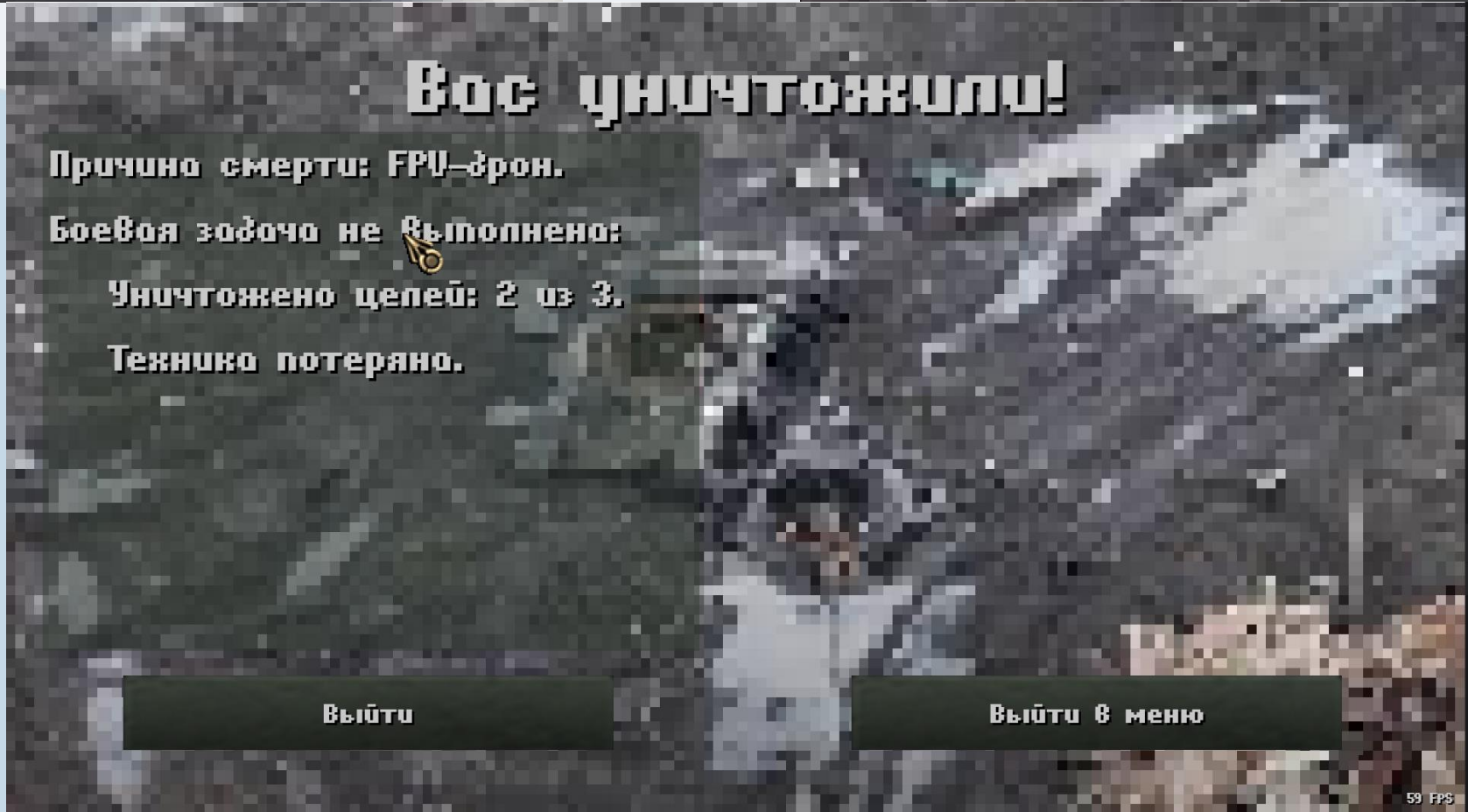
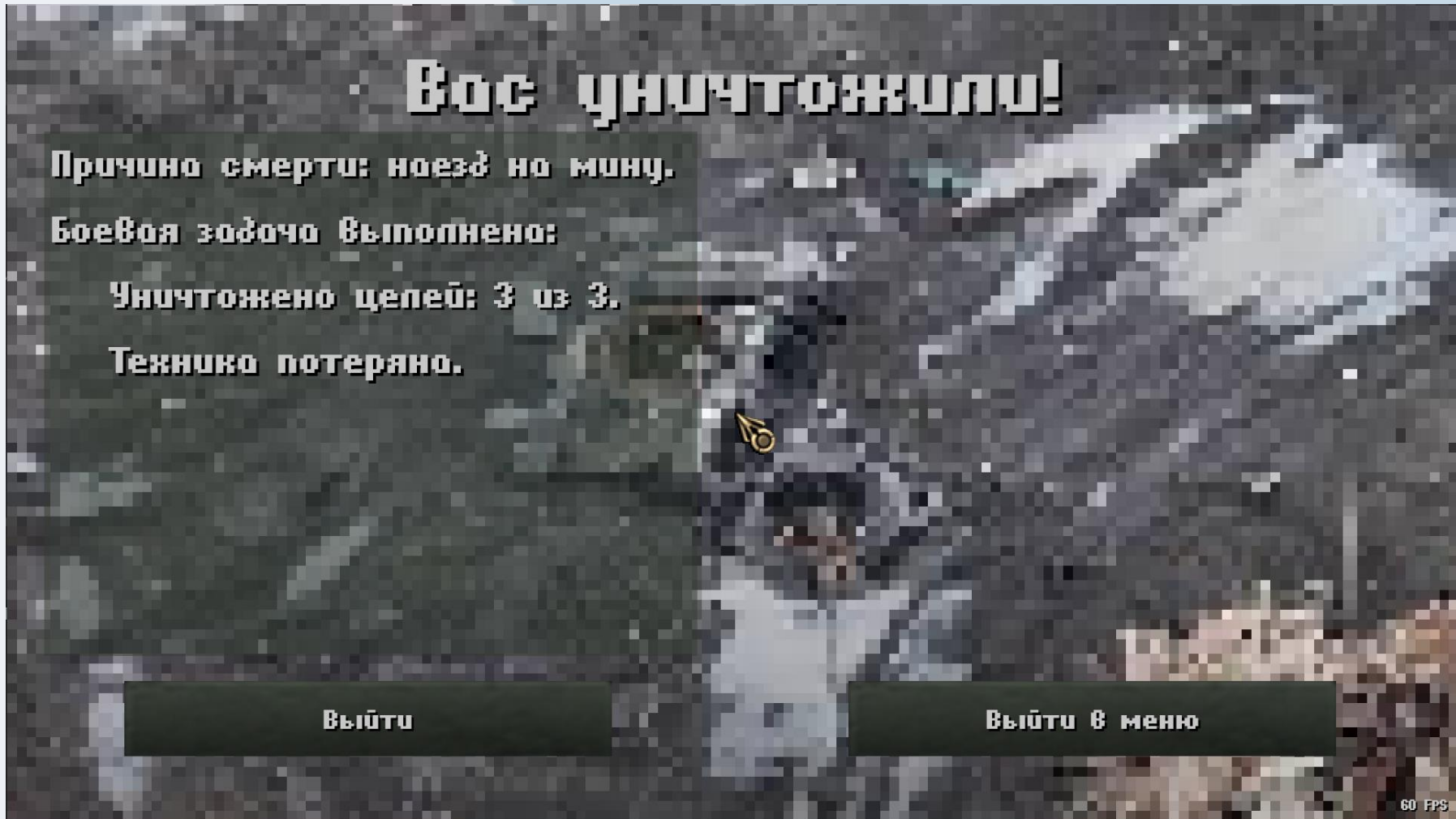
Результат работы



Результат работы

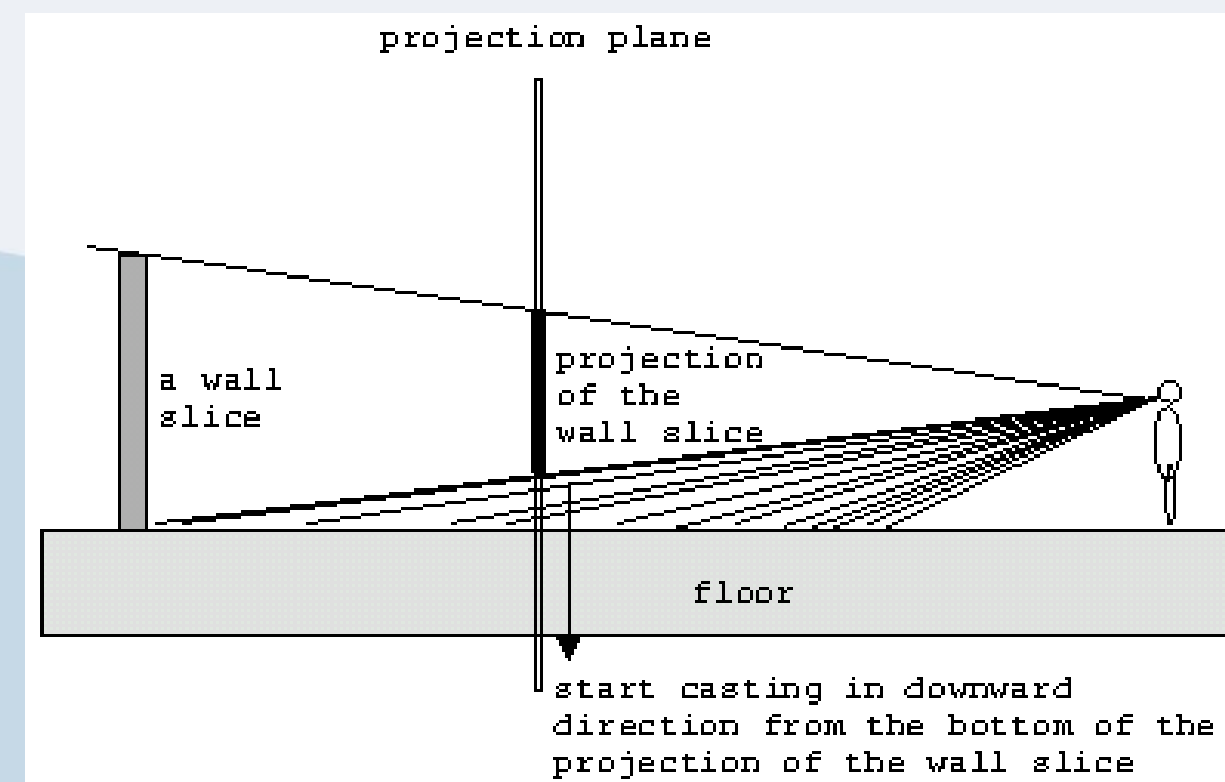


Результат работы



Перспектива развития

Перспективы проекта безграничны. Например, в игре может быть реализована как иная техника со своим функционалом, так и умные противники с использованием искусственного интеллекта. Также может быть улучшена основная технология Ray casting'a добавлением большего разнообразия текстур, противников, объектов с разными высотами или использованием технологии floorcasting для реалистичной отрисовки пола. В игре представлен широкий инструментарий для создания новых уровней.



**СПАСИБО
ЗА ВНИМАНИЕ!**

