СОЗДАНИЕ ЧИТОВ ДЛЯ WARCRAFT III

Введение

В ходе выполнения работы была поставлена задача разработать вспомогательную модификацию для игры Warcraft III, обеспечивающую расширенное отображение информации о противнике. Основная цель заключалась в создании мапхака, позволяющего игроку видеть расположение вражеских юнитов и построек в режиме реального времени. Для достижения этой цели проект был разделён на несколько взаимосвязанных подзадач:

1) Обеспечить постоянную видимость вражеских юнитов и построек;

2) Отключить туман войны (Fog of War);

3) Снять маску видимости (Visibility Mask) с миникарты.

Далее каждая из подзадач будет рассмотрена отдельно, так как для их выполнения применялись разные методы и приёмы.

Выбор версии игры

Для проведения работы была выбрана версия Warcraft III: Reign of Chaos & The Frozen Throne 1.26a. Это одна из самых распространённых и стабильных сборок игры, которая до сих пор активно используется сообществом.

Использованные инструменты

При выполнении всех подзадач мы использовали программу Cheat Engine, которая позволяет анализировать память игры, отслеживать обращения к нужным адресам и изучать работу связанных с ними функций.

Постоянная видимость вражеских юнитов и построек

Мы исходили из предположения, что видимость юнита определяется определённым значением в памяти: когда объект отображается, соответствующий байт принимает значение 1, а когда скрыт – 0. Чтобы подтвердить это, мы выполняли сканирование памяти в двух состояниях: когда юнит виден и когда невидим. Повторив процедуру несколько раз, мы получили набор адресов, изменяющих своё значение с 0 на 1 и обратно.

Затем мы перебрали найденные варианты и определили тот адрес, который действительно отвечает за флаг видимости. Через встроенный функционал Cheat Engine мы просмотрели функции, которые обращаются к этому адресу. Логично предположить, что функция отрисовки вызывает это значение намного чаще остальных, поэтому мы выбрали именно тот вызов, у которого наблюдалось максимальное количество обращений.

После дизассемблирования мы определили, какие аргументы принимает нужная функция. Экспериментально было установлено, что идентификатор текущего игрока равен 1, и именно это значение необходимо подставлять для корректной работы. В результате удалось добиться того, что юниты и сооружения противника отображаются постоянно, независимо от условий обзора.

Отключение тумана войны

Эта подзадача оказалась заметно сложнее, поскольку изначально было неясно, существует ли единый параметр, отвечающий за наличие тумана, или же каждый участок карты хранит собственное значение.

Чтобы найти его, мы использовали встроенную в игру команду, временно отключающую туман. Сканирование проводилось по методу поиска неизвестного значения: сначала при включённом тумане, затем – после его отключения. Далее мы загружали сохранение, повторяли процесс и отсекали значения, которые менялись нестабильно. Дополнительно мы опирались на гипотезу, что при наличии тумана параметр должен быть больше, а при его отключении – уменьшаться. Это позволило сузить выборку, но после нескольких циклов в списке по-прежнему оставалось более 265 адресов.

Поскольку автоматическая фильтрация перестала давать результат, мы перебрали оставшиеся значения вручную, предполагая, что отсутствие тумана соответствует нулю. Такой подход позволил обнаружить нужный адрес, управляющий глобальным состоянием Fog of War. После его нахождения дальнейшие действия повторяли методику из предыдущей подзадачи: через Cheat Engine проанализировали функции, обращающиеся к найденному

адресу, выделили наиболее активное обращение, изучили параметры и использовали их для корректного отключения тумана войны.

Снятие маски видимости с миникарты

Поиск параметра, отвечающего за маску видимости на миникарте, оказался самым проблемным. Мы пытались применить ту же методику, что и в предыдущих подзадачах: проводили сканирование памяти в Cheat Engine, меняли игровые условия и фиксировали изменения. Однако прямой поиск не дал результата – подходящие значения либо не менялись, либо сразу сбрасывались игрой. После этого мы вручную перебрали порядка 300 найденных адресов, но ни один из них не давал стабильного влияния на отображение миникарты.

Анализ показал, что механизм работы миникарты отличается от обычной логики видимости: данные о затемнённых областях не хранятся в виде одного числового флага или простого массива, как это было в предыдущих задачах. Значения пересчитываются самим движком и обновляются через систему отрисовки, из-за чего найденные адреса либо моментально перезаписывались, либо оказывались промежуточными и не управляли итоговым изображением. Из-за этого изменить маску видимости напрямую не удалось, и реализация данной подзадачи на текущем этапе оказалась невозможной.