

## Руководство для программистов по устройству в Targem Games

Мы делаем многопользовательские мобильные игры на движке **Unity**, а это значит, что нам приходится добиваться требуемого уровня качества в условиях жестко ограниченных ресурсов. Чтобы отвечать этим требованиям, программист должен хорошо понимать возможности движка (изучить его самостоятельно или [пройти обучение](#)), отдавать себе отчет в затратности используемых функций (с этим могут помочь [специализированные статьи](#) и [инструменты профилирования](#)).

Подавляющая часть игровой логики в наших проектах написана на **языке программирования C#**. При решении практических задач нам приходится программировать различные алгоритмы, и здесь важно понимать, что от выбранного набора инструкций напрямую зависит процессорное время и объем памяти, которые потребуются для выполнения этого алгоритма. Чтобы осознанно писать эффективный код, нужно четко понимать ресурсоемкость используемых инструкций, особенности механизмов управления памятью, назначение ключевых слов и то, как их интерпретирует компилятор. Основным подспорьем в этом деле является официальная [документация](#) и книги, такие как «C#. программирование для профессионала» за авторством Джона Скита (Jon Skeet, “C# in Depth”).

Современные среды разработки, такие как **Microsoft Visual Studio** и **JetBrains Rider**, также способны оказать поддержку в написании эффективного кода за счет расширений для анализа и рефакторинга (таких как JetBrains ReSharper). Кроме того, они незаменимы для отладки скриптов, а также быстрой навигации и поиска, особенно когда количество строк кода в проекте исчисляется десятками или даже сотнями тысяч.

Вопреки популярному мнению, наиболее сложной задачей в нашей работе является не разработка каких-то экзотических алгоритмов, а проектирование игровых систем таким образом, чтобы обеспечить их надежность и гибкость, минимизировать трудозатраты на дальнейшее их развитие. В этом смысле полезным будет знание **шаблонов проектирования**, которое можно почерпнуть, например, из [одноименной книги](#). Поможет здесь также изучение и даже участие в открытых проектах (найти можно на [github.com](https://github.com) или любых других тематических сайтах).

Что касается алгоритмов, то наиболее часто нам приходится программировать алгоритмы для решения задач, связанных с ориентацией в трехмерном пространстве. Причем сложность здесь возникает не с программированием, как таковым, а с постановкой, формализацией и аналитическим решением таких задач: для этого нужно отличное знание школьного курса математики — **геометрии, тригонометрии, векторной алгебры**. Подтянуть свои знания в этой области можно при помощи любого сборника задач, либо выполняя обучающие игровые проекты.

Неотъемлемой частью долгосрочной командной работы над проектом является регулярное взаимодействие с системами контроля версий (мы используем **git** и **svn**). Для решения повседневных задач достаточно понимания этих систем на концептуальном уровне, а также способность ориентироваться в их графических оболочках (мы используем **tortoise**).

Наконец, полезным (но не обязательным) будет:

- опыт программирования на C++ (для расширения серверной части проекта);
- опыт программирования на python (для разработки вспомогательных утилит);
- опыт программирования на javascript (для разработки инструментов оперирования);
- опыт разработки шейдеров, работы с графикой;
- работа с серверами баз данных, написание SQL запросов;
- администрирование сетей, знание сетевых протоколов;
- знание основ информационной безопасности (шифрование, обмен ключами);
- опыт публикации приложений в Google Play и App Store.

**Если вы считаете, что обладаете необходимыми знаниями и навыками, то отправьте нам свое резюме и попробуйте выполнить тестовое задание. Удачи!**