Анализ требований

Игровой движок – это программный комплекс, который упрощает разработку игр, предоставляя вам набор необходимых для разработки инструментов.

Обобщённо говоря, игровой движок ответственен за организацию и поведение игровых объектов, а также за их отображение на экране. Ваша же задача – выбрать, как они будут выглядеть и как себя вести. Для этого движок предоставит вам возможность создавать и удалять объекты, задавать их параметры, добавлять логику и управлять ресурсами.

Касательно внутреннего устройства игровые движки делятся на:

* Полноценное ПО – это такая специальная программа, которая может собрать игру воедино за вас, а вам предоставляет полноценный редактор.
* Фреймворк – это набор полезных классов и функций, которые вы можете использовать при создании своей игры с нуля. Игру вам придётся собирать самостоятельно, но зато у вас будет больше контроля за процессом.

Возможности, которые может предоставлять или не предоставлять игровой движок (список неокончательный):

* Система уровней
* Физика
* Система частиц
* Карты тайлов
* Редактор местности (terrain)
* Воспроизведение звука
* Источники освещения и тени
* Шейдеры
* Диалоговая система

Игровой мир состоит из игровых объектов (GameObject). К этой базовой категории можно практически отнести всё, что находится в игре, в том числе игрока, его инвентарь, камеру, землю под ногами, каждый отдельный кустик и даже небо. Не стоит думать, что все объекты обязательно должны быть видимы – всякие триггеры (объекты, вызывающие события при прикосновении), барьеры, источники освещения и даже части интерфейса являются такими же объектами. Все игровые объекты обладают несколькими базовыми свойствами: положение в пространстве (Transform), включены ли они (Active), какой у них родительский объект и есть ли он (Parent).

Игровые объекты так же могут быть дополнены поведением (Behaviour или Component). Поведение – это отдельный код, который привязан к объекту и выполняется при определённых условиях. Условия могут быть самыми разными, а количество поведений, привязанных к объекту, ничем не ограничено. В таком коде вы, например, можете двигать объект по движению мыши или перекрашивать его цвет. А ещё у каждого поведения могут быть свои отдельные параметры (выраженные в переменных).

Например, мы можем создать для картинки поведение «Персонаж», у которого будут очки здоровья и возможность прыгать. И когда персонаж падает со слишком большой высоты, эти очки здоровья у него отнимать.

Помимо своих собственных поведений в игровом движке есть несколько стандартных типов поведений: форма столкновения (Bounding Box/Sphere/Capsule/…), физическое тело (Rigidbody), отрисовщик (Renderer), камера (Camera), создатель частиц (Particle Manager), аниматор (Animator) и ещё десятки других типов. Всеми этими поведениями вы можете управлять на лету.

Очень важным концептом является событие (Event). Это сигнал, который возникает при соблюдении каких-то условий. Поведения объектов в игре могут порождать эти события и реагировать на них. Например, столкновение – это событие, причём одно из самых частых по использованию. Именно на событиях строится основной игровой процесс, разработчик игры может навешивать действия (Action) одних поведений на события других и так, например, делать кнопки, рычаги, точки сохранений и так далее.

Но это и не единственный способ заставить игру работать, ещё есть раздел Update, в который можно написать код и который будет выполняться постоянно, в каждый игровой тик (tick). Тик – это самая минимальная единица времени, которую игра может обеспечить. Обычно тик составляет 16 миллисекунд, но если у вас плохо с оптимизацией, то он увеличится. Без этой функции не обойтись, и некоторые вещи, например плавное передвижение и проверка столкновений, пишутся именно там. Но чем меньше кода написано в этой секции – тем лучше.

Место, в котором находятся игровые объекты, называется уровень или сцена (Level или Scene). Уровни можно менять в любой момент, а в некоторых движках ещё и совмещать между собой. Ваши игровые объекты будут распределены по уровням, чтобы друг другу не мешать. Например, это будут локации и их наполнение. Но определённые универсальные для всех уровней объекты, например главный персонаж или интерфейс, лучше хранить в отдельном месте.

В вашем проекте должна быть отдельная папка, в которой вы будете хранить сохранённые объекты (Prefab). Любой объект в игре вы можете сконструировать всего один раз, а затем сохранить в эту папку для дальнейшего, в том числе многократного, использования. Например, это могут быть деревья или враги. Во время игры вы можете создать любое количество объектов из этой папки, но лучше не переборщить и не использовать тысячи объектов, иначе движок начнёт лагать.

Объекты в игре могут выглядеть самым разным способом. И дело даже не в отдельный настройках, а в самом способе их отображения на экране. Это могут быть 2D-объекты, например различные простейшие геометрические формы (Shape) или картинки (Sprite). А могут и 3D-объекты, которые состоят из 3D-модели (Mesh). Все видимые объекты в игре обязаны иметь материал (Material) – набор параметров, влияющий на отображение объекта. Такими параметрами могут являться текстуры (Texture), цвета (Color) и обычные числа (Float). Некоторые движки дают доступ ограниченный доступ материалу, давая лишь задать текстуру и цвет окрашивания, другие же дают полный доступ. В основе материала лежит шейдер (Shader) – особая программа, которая проводит математические вычисления и проецирует объекты в пространстве на плоский экран камеры.

Описание компонентов

Из анализа требований можно выделить несколько основных модулей:

* Система частиц;
* Редактор местности;
* Модуль воспроизведение звука;
* Модуль отображения и взаимодействия со сценой в реальном времени;
* Модуль шейдера;
* Модуль диалоговых окон.
* Редактор написания кода.

Программы, работающие с [трёхмерной графикой](https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D1%91%D1%85%D0%BC%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D0%BA%D0%B0) и видео ([игры](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%B8%D0%B3%D1%80%D0%B0), [GIS](https://ru.wikipedia.org/wiki/%D0%93%D0%B5%D0%BE%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0), [CAD](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%B0%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82%D0%B8%D0%B7%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F), [CAM](https://ru.wikipedia.org/wiki/CAM) и др.), используют шейдеры для определения параметров геометрических объектов или изображения, для изменения изображения (для создания эффектов сдвига, отражения, [преломления](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B5%D0%BB%D0%BE%D0%BC%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5_%D1%81%D0%B2%D0%B5%D1%82%D0%B0), затемнения с учётом заданных параметров [поглощения](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%B3%D0%BB%D0%BE%D1%89%D0%B5%D0%BD%D0%B8%D0%B5_%D1%81%D0%B2%D0%B5%D1%82%D0%B0) и [рассеяния света](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D1%81%D1%81%D0%B5%D1%8F%D0%BD%D0%B8%D0%B5_%D1%81%D0%B2%D0%B5%D1%82%D0%B0), для наложения [текстур](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BA%D1%81%D1%82%D1%83%D1%80%D0%B0_(%D1%82%D1%80%D1%91%D1%85%D0%BC%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D0%BA%D0%B0)) на геометрические объекты и др.).

Модуль отображения и взаимодействия со сценой является интерфейсом пользователя и через взаимодействие с ним нажатием горячих клавиш и кнопок мыши можно вызвать методы модуля шейдера, которые перерисуют изображение на сцене. Также на нее можно добавлять объекты, изменять, перемещать и прочее.

Модуль воспроизведения звука представляет собой пользовательский интерфейс для добавления звуковых дорожек в создаваемый проект, их изменения, удаления и пр.

Редактор местности представляет собой удобный инструмент для быстрого создания рельефа карты, ее высот, добавление объектов и их генерацию. Система уровней представляет собой интерфейс для удобного создания карт (уровней) проекта, их изменения в сцене, удобное переключение уровней между собой (осуществление навигации по проекту).

Система частиц – используемый в компьютерной графике способ представления объектов, не имеющих чётких геометрических границ (различные облака, туманности, взрывы, струи пара, шлейфы от ракет, дым, снег, дождь и т. п.). Системы частиц могут быть реализованы как в двумерной, так и в трёхмерной графике.

Модуль диалоговых окон – это инструмент для добавления в игру различных панелей элементов (главных и второстепенных меню, настроек графики и персонажа, всплывающих подсказок во время прохождения игры, карты местности и пр.).

Редактор написания кода представляет собой модуль, в котором описывается основная логика приложения, которую нельзя реализовать с помощью ранее описанных инструментов. Это вызов от элементов меню, изменения показателей игрока в зависимости от его действий и их отображение в соответствующем диалоговом окне или изменения звуковой дорожки в зависимости от ситуации, с которой столкнулся игрок и пр.

Архитектура игрового движка

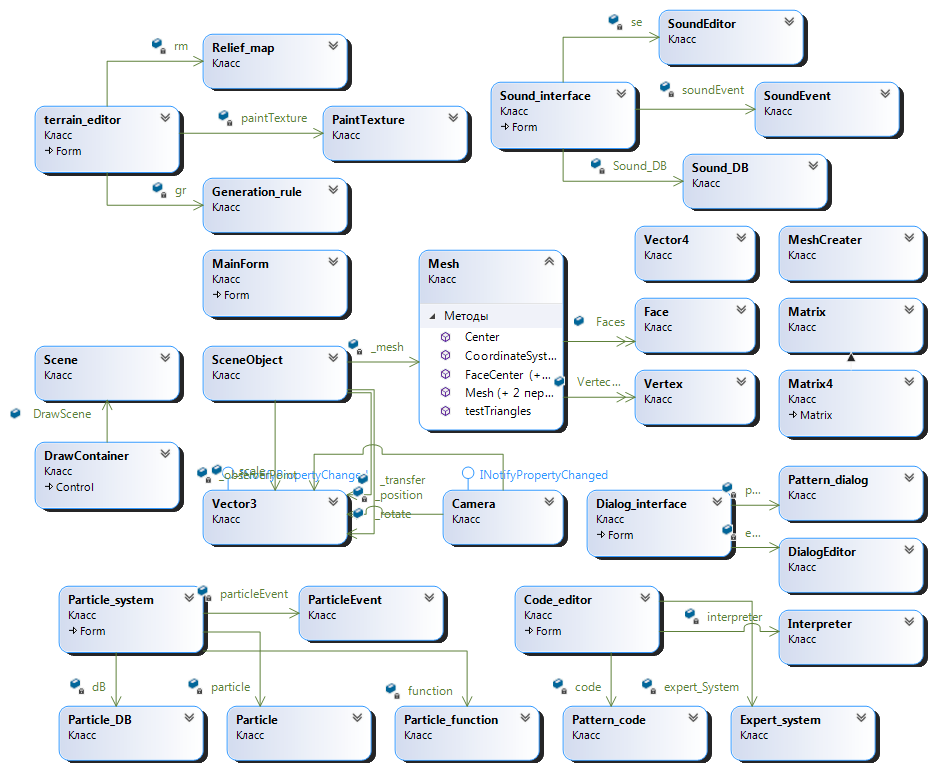


Рисунок 1 – Архитектура проекта