Table of Contents

Быстрый старт

Программы для работы

Для начала вам потребуется установить на свой компьютер следующие программы:

1. UnityHub. Программа управления версиями Unity. После установки данной программы, необходимо в ней установить стабильные версии во вкдладке Installs.

   Note

   Совет! Устанавливайте версии с пометкой LTS. Так же не поднимайте и не опускайте версию существующего проекта без необходимости. Минимальная версия Unity для работы с проектами 2018

2. VisualStudio. Данная среда разработки поставляется вместе с Unity. Но можно поставить и отдельно. Настраивается персонально под разработчика. Также советуем поставить CodeMaid для чистоты оформления вашего кода.

3. SourceTree или ForkGit одно по желанию. Программы для работы с системой контоля версий SCM.

Работа с проектом

Для создания вашего проекта с нуля или работы с уже существующим Вам необходимо запросить создание и разрешение на доступ репозитория в системе SCM у ведущих программистов Александра и Ивана.

Далее требуется импортировать необходимые для вас файлы, а также последней версией PLUnity. Если PLUnity нет в проекте, то вы можете установить из сетевой папки \\192.168.1.10\public\PLCore, в другом случае PLUntiy обновиться автоматически при входе в систему PLServer.

Для разработки или модификации проекта вы должны максимально использовать функции представленные в PLUnity. Это требуется для унификации наших проектов, как визуальной, так и технической. Далее будут расписаны важные моменты, на которые вы должны обращать внимание и использовать в каждом вашем проекте.

Менеджер конфигураций

В данном менеджере идет настройка проектов и их конфигураций.

Менеджер конфигураций

Теория данной системы:

- В репозитории может быть несколько проектов
- У каждого проекта есть несколько конфигураций
- Конфигурация это отдельный выходной .exe файл
- Конфигурации имеют версии
- Все версии конфигурации доступны конечному пользователю

Окно менеджера конфигураций

На данном изображение представлено окно менеджера конфигураций. Окно поделено на 3 блока:

- Список проектов
- Список конфигураций
- Настройка конфигураций

В нижней части окна расположены 3 кнопки:

- Открыть окно BuildSetting
- Открыть настройки PLUnity
- Открыть окно авторизации в системе PLServer

Окно настройки PLUnity

В окне настроек PLUnity необходимо указать путь до InnoSetup Compil32.exe файла. Если он отсутствует, можно его сразу же скачать и установить.

Также необходимо установить путь до папки, куда будут складываться ваши сборки.

Можно выставить лог от системы Moodle (при работе с дистанционкой) и лог от PLServer.

Important

ВАЖНО! Настройки PLUnity глобальный! Их можно установить один раз и больше не трогать на данном компьютере!

Окно авторизации в системе PLServer

Для работы с системой PLServer необходимо авторизироваться в системе. Чтобы узнать, в каком вы статусе авторизации, достаточно посмотреть на кнопку откртыия окна авторизации. Если кружок зеленый - то вы авторизированы, если красный - нет.

Далее вы вводите логин и пароль в системе.

Авторизация держиться сутки, поэтому Вам необходимо будет каждый раз авторизироваться.

Warning

ВНИМАНИЕ! На данный момент этот функционал работает только под логином и паролем администраторов. При работе Вам необходимо запросить логин и пароль у ведущих программистов Александра и Ивана.

Открытие менеджера

Для того чтобы открыть менеджер конфигураций необходимо открыть вкладку PLUnity на верхней панеле и найти PL Project Manager или нажать комбинацию клавиш Ctrl+Shift+I

Настройка конфигурации

Параметры

В данной вкладке Вам необходимо установить только ссылки на сцены.

Сцены в поле "Сцены в меню сцен" - это главные сцены. Если вы устанавливаете одну сцену, нет необходимости назначать ей имя: она будет автоматически запускаться при нажатии кнопки "Запустить" в лаунчере PLUnity. При работе с несколькими сценами нет необходимости делать главную сцену меню. Просто установите все сцены в данное поле и назовите их, тогда PLUnity автоматически создаст список работ в лаунчере.

В ситуации, когда одна сцена должна запускать другую сцену, которая не должна находится в основном списке работ в лаунчере, все запускаемые таким образом сцены должны быть добавлены в список "Доп. сцены для проекта". Эти сцены добавятся в список сцен BuildSettings.

При необходимости можно установить дополнительные конфигурации в виде кастмного ScriptableObject и обращаться через к нему через PLUnity.

Файлы

В данной вкладке Вам необходимо установить все пути до файлов. Данные пути сериализуются относительно расположения проекта.

Каталог модуля - папка, в которую будет положен собранный проект из Unity.

Каталог руководства пользователя - папка, в которой расположена методичка в формате .htm.

Иконка PNG – изображение иконки в формате PNG.

Иконка ICO - изображение иконки в формате ICO.

Задник - общее изображение комплекса в форма PNG.

Справка - при включении данной опции можно добавить справочный материал. Этот материал будет отображаться в общем меню программы при нажатии на кнопку Esc. В меню будет добавлена кнопка "Справочный материал". В данном матерале должна быть только теория по данной работе.

Доп. ресурсы - при включении данной опции, вы можете добавить дополнительные ресурсы из указанной папки к финальному .exe файлу. Это могут быть сторонние файлы конфига и прочее.

Доп. общие ресурсы - при включении данной опции, вы так же можете добавить дополнительные ресурсы из указанной папки, но данные файлы расположаться в общих документах. Эти файлы можно изменять не под админом. Сюда можно класть базы данных, сохранения и всё то, что программа должа изменять.

Драйвера - список .exe фалов драйверов, необходимых для работы с программой. Драйвера автоматически установятся при запуске установочного файла модуля.

Important

ВАЖНО! Располагайте файлы в соответствующие папки. Модуль – в папку Builds. Изображения – в папку Images. Руководство – в папку Help. Если у вас несколько проектов и необходимо иметь разные справки и изображения, то создавайте соответственно различные подкаталоги, к примеру: /Images/Project1 и /Images/Project2, /Help/Project1 и /Help/Project2, и т.д.

Настройки сборки

В данной вкладке Вам необходимо настроить сборку проекта.

IL2CPP - компиляция под C++. Необходимо чтобы ваш проект поддерживал данную компиляцию. За некоторым исключением ее можно выключить.

Профиль графики - стоит автоматически. Трогать его не надо. Впоследствии добавится полноценный редактор.

Иконка - иконка проекта. Необходимо взять файл иконки PNG, который вы ранее выставили в файлах, и поместить его в папку Assets/Sprites, после чего прокинуть ссылку на данный файл.

Лого - лого компании. Трогать нет необходимости.

Релиз версии

Так как все версии храняться в одной конфигурации и доступны заказчику, Вам необходимо внимательно тестировать проект при отправки его на сервер. Для создания финального билда версии необходимо использовать компьютер с лицензией UnityPlus. Это необходимо для того чтобы при загрузке проекта отсутствовало лого Unity.

Панель релиза

На данной панели вы можете без отправки файла на сервер, собрать модуль и протестировать. Для этого используются кнопки "Собрать", "Упаковать", "Собрать и упаковать".

Для отправки версии необходимо вписать нововведения в версии. Данный текст будет виден заказчику. Пишите с большой буквы ваши нововедния, для перечисления используйте дефис "-". В первой версии необходимо написать "Базовая конфигурация".

Далее нужно выбрать тип повышения проекта:

- Мажор
- Минор
- Микро

Первая версия проекта должна быть 1.0.0.

Далее нажмите на соответсивующую кнопку публикации.

Note

Собрать - это сделать билд Unity. Упаковать - сделать выходной exe файл установщика. Опубликовать - отправить на сервер.

Сборка проекта на компьютере для билдов

Все проекты необходимо собирать под UnityPlus.

Для этого необходимо:

1) Запустить подключение к удаленному рабочему столу

2) Вбить IP 192.168.1.145

3) Вбить пользователя: User

4) Вбить пароль PLDeveloper

Important

ВАЖНО! Не забывайте проверить финальную версию, которую вы отправили на сервер! Новая версия проекта не должна быть новым проектом или конфигурацией! Клиент видит все версии и может установить любую, а также прочитать её нововведения! Версия обновится у всех клиентов, которым доступна данная конфигурация!

Основные классы

PLUnityModel

PLUnityModel. Головной класс для работы с PLUnity. В данном классе хранится вся информация о выбранном проекте, конфигурации и её версии. Здесь можно привязаться к основным параметрам, таким как: кастомные ScriptableObjects, использование BP, дистанционного обучения и т.п. Таким образом, для сборки новой версии не требуется тратить время на изменение каких-либо внутренних параметров вашей системы. Данный класс также позволяет получить путь до папки проекта в общих документах на компьютере.

```
// Получить путь до общей папки проекта
string path = PLUnityModel.GetPathToSpecialFolder();

// Получить кастомный ScriptableObject конфигурации
ScriptableObject custom = PLUnityModel.unityConfiguration.projectScriptableObjects[0];
```

PLREModel

PLREModel. Класс для работы с системой дистанционного образования на базе Moodle и с поддержкой SCORM спецификации.

```
// Используется ли в данной конфигурации дистанционное обучение
bool useRE = PLREModel.isRE;

// Отправка результата для SCO
PLREModel.SendScoResult("Lab1", 100, LessonStatus.Completed);
```

PLUnityEventsDispatcher

PLUnityEventsDispatcher. Класс для отслеживания событий Unity. Подходит для решения задач, когда класс не является MonoBehaviour, а методы Update, StartCoroutine и т.п. необходимы.

```
// Подписаться на Unity Update
PLUnityEventsDispatcher.OnUpdate += UpdateYourCustomClass;

// Запустить коррутину
PLUnityEventsDispatcher.StartCoroutine(ExampleCoroutine());
```

Клиент-серверное взаимодействие

Для создания клиент-серверного взаимодействия необходимо использовать на серверном приложении класс ServerConnection, а на клиентском ClientConnection. Пакет общения должен быть унаследован от класса NetworkPackage.

ServerConnection

ServerConnection. Класс для создания приложения-сервера. Обработка пакетов осуществляется не в потоке, поэтому для использования в Unity, необходимо вызывать метод Update(). Это необходимо для обработки пакетов в главном потоке программы. Пакеты должны быть унаследованы от NetworkPackage.

```
// Создать сервера
ServerConnection serverConnection = new ServerConnection(port, new Network.UnityJSONMemStreamSerializer(), ConnectionLogsLevel.Connections);

// Подписаться на получение пакета от сервера
serverConnection.AddListener<CPPackage>(GetClientPackage);

// Отправить пакет данных всем клиентам
serverConnection.Send(new SPPackage());

// В пакете полученного от клиента, вшит сам клиент, поэтому можно напрямую отвечать ему
client.Send(new SPPackage());
```

```
// Обновление сервера
void Update()
{
  serverConnection.Update;
}
```

ClientConnection

ClientConnection. Класс для создания клиентского подключения. Обработка пакетов осуществляется не в потоке, поэтому для использования в Unity, необходимо вызывать метод Update(). Это необходимо для обработки пакетов в главном потоке программы. Пакеты должны быть унаследованы от NetworkPackage.

```
// Создать клиентское подключение
ClientConnection connection = new ClientConnection("localhost", port, new Network.UnityJSONMemStreamSerializer(), ConnectionLogsLevel.Connections);

// Подписаться на получение пакета от сервера
connection.AddListener<SPPackage>(GetServerPackage);

// Отправить пакет данных на сервер
connection.Send(new CPPackage());
```

```
// Обновление клиента
void Update()
{
  connection.Update;
}
```

Компоненты для Unity

Данные компоненты можно смело располагать на сцене в вашем проекте. Вызывать методы можно при помощи станадртной кнопки и события onClick.

BrowserOpenComponent

BrowserOpenComponent. Компонент для откртия URL ссылки. Подходит для открытия локальных htm и html страниц.

CanvasGWPriorityComponent

CanvasGWPriorityComponent. Компонент для определение приоритета канваса для глобальных окон. Необходим для работы в проектах со множеством канвасов, для определения главного канваса и вывода на него глобальных окон.

MenuOpenComponent

MenuOpenComponent. Компонент для вызова открытия меню.

SceneLoadComponent

SceneLoadComponent. Компонент для загрузки сцены с прогресс баром.

Дополнительные возможности

SceneReference

SceneReference. Класс для сериализации ссылок на сцене. При помощи данного класа можно спокойно серилизовать ссылку на сцену и загрузить ее, не боясь переименования файла сцены.

UnityJsonSerializer

UnityJsonSerializer. Класс для сериализации и десериализации данных в формате JSON.

INIManager

INIManager. Класс для работы с ini файлами. Подходит для создания конфигурационных файлов проекта.