

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение  
высшего образования

**«Южно-Уральский государственный университет»  
(национальный исследовательский университет)  
Высшая школа электроники и компьютерных наук  
Кафедра системного программирования**

РАБОТА ПРОВЕРЕНА

Рецензент  
Генеральный директор компании  
ООО «АппБит Софтвейр»

\_\_\_\_\_ Д.А. Диков

“ \_\_\_ ” \_\_\_\_\_ 2017 г.

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой,  
д.ф.-м.н., профессор

\_\_\_\_\_ Л.Б. Соколинский

“ \_\_\_ ” \_\_\_\_\_ 2017 г.

**РАЗРАБОТКА ВИРТУАЛЬНОГО СТЕНДА  
ДЛЯ МОДЕЛИРОВАНИЯ ПРОЦЕССА СБОРКИ  
ПЕРСОНАЛЬНЫХ КОМПЬЮТЕРОВ НА МОБИЛЬНЫХ  
УСТРОЙСТВАХ**

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА  
ЮУрГУ – 02.03.02.2017.13-016-1382.ВКР

Научный руководитель  
старший преподаватель кафедры СП  
\_\_\_\_\_ А.И. Семенов

Автор работы,  
студент группы КЭ-401  
\_\_\_\_\_ Криушичев А.Г.

Ученый секретарь  
(нормоконтролер)  
\_\_\_\_\_ О.Н. Иванова

“ \_\_\_ ” \_\_\_\_\_ 2017 г.

Челябинск-2017

**5. Дата выдачи задания: 09.02.2017.**

**Научный руководитель**

Старший преподаватель кафедры СП

**Задание принял к исполнению**

А.И. Семенов

А.Г. Криушичев

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	5
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ .....	8
1.1. Обучающие тренажеры и виртуальные программные комплексы.....	8
1.2. Постановка задачи.....	9
1.3. Обзор программных средств разработки.....	10
1.3.1. Платформа Unity .....	11
1.3.2. Платформа Unreal Engine .....	11
1.3.3. Выбор платформы .....	12
1.4. Анализ аналогичных проектов .....	12
2. ПРОЕКТИРОВАНИЕ ВИРТУАЛЬНОГО СТЕНДА .....	14
2.1. Определение требований.....	14
2.2. Проектирование архитектуры приложения.....	14
2.2.1. Диаграмма прецедентов .....	15
2.2.2. Архитектура модуля выбора комплектующих.....	16
2.2.3. Архитектура модуля сборки персонального компьютера .....	17
2.2.4. Архитектура модуля настройки BIOS .....	19
2.2.5. Архитектура модуля установки операционной системы.....	20
2.2.6. Проектирование пользовательского интерфейса.....	21
3. РЕАЛИЗАЦИЯ .....	24
3.1. Реализация сцены выбора комплектующих .....	24
3.2. Реализация сцены сборки компьютера .....	26
3.3. Реализация сцены предварительной настройки BIOS .....	28
3.4. Реализация сцены симуляции установки операционной системы .....	30
4. ТЕСТИРОВАНИЕ .....	31
4.1. Функциональное тестирование.....	31
4.2. Юзабилити-тестирование.....	38
ЗАКЛЮЧЕНИЕ .....	39
ЛИТЕРАТУРА.....	40

## **ВВЕДЕНИЕ**

В настоящее время для всех образовательных уровней необходимы специальные учебные средства, позволяющие эффективно обеспечивать процесс получения знаний, умений и навыков. Для обучения школьников и специалистов часто используют такие средства как: презентации, анимационные ролики, модели инструментов, приборов и станков, электронные лекции, учебные пособия и практикумы, а также компьютерные имитаторы приборов или производственных процессов. Последняя категория является наиболее новой и неосвоенной, но активно развивается в связи с развитием компьютерной техники и методов разработки ПО.

На данный момент, на рынке представлено множество комплектующих к персональным компьютерам. В связи с этим, возникает потребность в обучении по проверке совместимости, установки и подключения выбранных компонентов компьютера. Благодаря появлению новых методов и инструментов разработки полноценных трехмерных интерактивных приложений, позволяет с высокой степенью достоверности воспроизводить на современных компьютерах, а также телефонах высоко детализированные модели изучаемых объектов и систем. Обучение сопряжено со значительными материальными и временными затратами, появление виртуальных стендов имитаторов, позволяет в значительной мере ускорить процесс обучения школьников и подготовки специалистов и удешевить его, без потери качества, а зачастую с повышением уровня подготовки. За последнее время виртуальные комплексы эволюционировали из простых наглядных демонстраций в полноценные инструменты комплексного обучения, работающие в диалоговом режиме с обучаемым и преподавателем.

Очень важным аспектом при проектировании систем такого рода, является возможность максимально прозрачного и наглядного взаимодействия человека с системой. Связующим компонентом между пользователем и комплексом является графический интерфейс.

Графический интерфейс виртуально программных комплексов является очень важным звеном, так как именно он определяет качество восприятия информации обучаемым, а главное ее достоверность. Обучаемый школьник или студент, попадая в реальные условия и встречаясь с реальным комплектующим, в первую очередь пытается использовать полученный ранее во время обучения на тренажере-имитаторе пользовательский опыт. В первую очередь работает визуальная память и пользователь пытается найти знакомые слоты и разъемы, комплектующие и др. Отсюда вытекает одно из основных требований к интерфейсам систем такого рода: максимальная визуальная достоверность и реалистичность отображаемых объектов, начиная с расположения и заканчивая физически достоверным внешним видом, а также идентичность способа и вида установки и действий пользователя-оператора.

Второй основной задачей является максимальное соответствие информации получаемой с помощью интерфейса пользователя с реальными параметрами. Для этого необходимо тщательно ознакомиться с моделируемым объектом, изучить назначение компонента и его характеристики.

Заказчиком данного проекта является ООО НПП «Учтех-Профи», занимающееся разработкой, серийным и мелкосерийным производством инновационной учебной продукции для вузов, техникумов, колледжей и учебных центров предприятий.

Таким образом, разработка виртуального стенда по сборке персональных компьютеров является актуальной задачей на сегодняшний день. Стенд будет полезен как школьникам и студентам, так и обычным людям. Из-за возможности запуска стенда на мобильных устройствах, он станет доступен для всех.

Целью работы является разработка виртуального стенда для моделирования сборки персональных компьютеров на мобильных устройствах.

Для достижения этой цели нужно решить следующие задачи:

- 1) провести анализ аналогичных проектов;

- 2) выполнить проектирование приложения;
- 3) реализовать приложение;
- 4) провести тестирование;
- 5) внедрить в опытную эксплуатацию.

### **Структура и объем работы**

Работа состоит из введения, четырех основных разделов, заключения и библиографического списка. Объем работы составляет 41 страницы, а объем библиографического списка – 18 наименований.

В первой главе работы, «Анализ предметной области», проведен анализ аналогичных проектов, обзор программных средств разработки и постановка задачи.

Вторая глава работы, «Проектирование виртуального стенда», включает в себя определение требований и проектирование приложения.

В третьей главе работы, «Реализация», содержится описание реализации основных элементов приложения.

В четвертой главе работы, «Тестирование», содержится описание тестирования приложения.

В заключении сделаны выводы о проделанной работе.

## **1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ**

### **1.1. Обучающие тренажеры и виртуальные программные комплексы**

Разработка виртуально-программных комплексов [18] – это новое направление в области разработок образовательного программного обеспечения, возникшее в процессе совершенствования современных методов обучения. Все больше обретает актуальность создание специализированных комплексов-имитаторов работы реального оборудования или виртуальных лабораторий. Зачастую обучение персонала, взаимодействующего со сложным или дорогостоящим оборудованием, занимает много времени и порождает ряд проблем, от экономических до логистических. Виртуально-программные стенды позволяют решить весь спектр проблем в получении и закреплении навыков и знаний при обучении новых кадров и позволяет значительно ускорить процесс.

Для того чтобы обучить оператора станка с ЧПУ, достаточно владеть определенной информацией и иметь доступ к самому станку или его аналогу, что не всегда возможно, когда речь идет о сложных инфраструктурных объектах или линиях с неразделимыми комплексами сложного оборудования. Задача освоения таких комплексов и систем усложняется невозможностью иметь доступ к производству непосредственно для обучаемого ввиду экономических или других причин. В случае ошибки обучаемого существует риск понести экономические убытки в связи с выходом из строя дорогостоящего оборудования с последующим его простоем. Очевидным выходом является глубокая теоретическая подготовка и ограничение на использование оборудования в недопустимых режимах, но возникает другая проблема: отсутствие опыта взаимодействия с оборудованием в аварийной или нештатной ситуациях, возможных в ходе эксплуатации, но недостижимых во время обучения. Для решения таких задач и существуют виртуально-программные комплексы и стенды. Такие комплексы позволяют наглядно продемонстрировать производственный процесс или эмулировать работу того или иного сложного оборудования без риска реальной

поломки, но с возможностью получить аварийную ситуацию и отработать комплекс действий по ее устранению.

Примером таких комплексов могут служить авиасимуляторы, используемые для подготовки пилотов гражданских и военных авиалиний. Данный вид ПО позволяет избежать жертв и поломок дорогостоящей техники и подготовить пилотов к экстренным ситуациям и выходу из строя воздушного судна.

Другой пример: виртуальные образовательные лаборатории. Например, виртуальная лаборатория по химии. Данный стенд представляет из себя трехмерный симулятор лаборатории, состоящий из 3D-моделей реального оборудования и химических веществ. Комплекс позволяет проводить химические опыты, выполнять лабораторные работы и формировать отчет для преподавателя. Данный стенд позволяет качественно повысить уровень дистанционного обучения.

Таким образом, на основании вышесказанного, задача создания и развития виртуально-программных комплексов и их совершенствование остается актуальной. Современные графические технологии позволяют достичь необходимого уровня достоверности и реализма процесса обучения без значительных затрат.

## **1.2. Постановка задачи**

Виртуальный стенд по сборке персонального компьютера [14] – это образовательное приложение, направленное на обучения школьников и студентов этапам сборки ПК. В приложении должно быть подробные описание и характеристики всех комплектующих. Пользователь должен наглядно увидеть 3D-модель компонента ПК и правильно ее разместить.

В приложении должно быть 4 этапа сборки: выбор комплектующих, сборка персонального компьютера, предварительная настройка BIOS, установка операционной системы.

**Выбор комплектующих** – на этом этапе пользователь должен выбрать центральный процессор, материнскую плату, оперативную память,



жесткий диск, видеокарту, кулер, блок питания и корпус. В специальном окне прорисовывается 3D-модель выбранного компонента, которую пользователь может осмотреть со всех сторон для более детального осмотра. После выбора всех комплектующих компьютера происходит проверка совместимости, и если проверка пройдена, то происходит переход на следующий этап, в противном случае появляется окно об ошибке выбора.

**Сборка ПК** – на этом этапе пользователь должен из выбранных ранее комплектующих собрать персональный компьютер. Перед ним расположены материнская плата и корпус компьютера. Он должен в правильной последовательности разместить компоненты компьютера в нужные слоты и разъемы. У каждого комплектующего есть всплывающие информационные окна для пояснения. После прохождения всех этапов сборки, происходит переход на следующий этап.

**Предварительная настройка BIOS** – на этом этапе пользователь должен выбрать способ установки операционной системы: либо с USB накопителя, либо с компакт-диска. После этого необходимо пройти в нужную вкладку BIOS [15, 16] и поставить приоритет загрузки. Далее можно сохранить изменения и выйти. Если приоритет загрузки был выставлен правильно, то пользователь перейдет к установке операционной системы, в противном случае появится окно об ошибке.

**Установка операционной системы** – на этом этапе пользователь должен поэтапно установить операционную систему. На экране будет выведен каждый шаг с пояснением. По окончании установки появится окно с сообщением, что компьютер готов к использованию. Пользователь сможет начать сборку компьютера с самого начала или выйти из приложения.

### **1.3. Обзор программных средств разработки**

Для разработки приложения было решено использовать инструмент для разработки мобильных и компьютерных игр, т.к. он уже имеет встроенные компоненты для работы с 3D- и 2D-графикой. Самыми доступными и наиболее мощными являются Unity и Unreal Engine.

### 1.3.1. Платформа Unity

Unity [9] – платформа для разработки компьютерных и мобильных игр. Данная платформа позволяет создавать проекты для большинства операционных систем и платформ: Microsoft Windows, Linux, Mac OS и Mac OS X, консолей Xbox, Xbox 360, PlayStation 2, PlayStation 3, PSP, PS Vita, Wii, Dreamcast, GameCube и др., а также на различных мобильных устройствах. Также присутствует возможность создания проектов для запуска в браузерах. Платформа имеет удобный для работы редактор, который можно дополнять своими расширениями. Unity имеет систему подписок для распространения своего продукта. Можно совершенно бесплатно создавать свои приложения и получать коммерческую прибыль до определенного лимита. Присутствует поддержка языка программирования C#.

Достоинства:

- кроссплатформенность;
- легок для освоения;
- удобный редактор;
- регулярные обновления.

Недостатки:

- условно-бесплатный;
- без открытого кода.

### 1.3.2. Платформа Unreal Engine

Unreal Engine [10] – платформа для разработки компьютерных и мобильных игр. Данная платформа позволяет создавать проекты для большинства операционных систем и платформ, как и Unity. Unreal Engine имеет возможность создания алгоритмов внутри своего графического редактора. Платформа распространяется бесплатно при условии, что прибыль от проекта частично отчисляется производителю данной платформы. Присутствует поддержка языка программирования C++.

Достоинства:

- кроссплатформенность;

- возможность создания алгоритмов внутри графического редактора;
- регулярные обновления.

Недостатки:

- условно-бесплатный;
- высокие аппаратные требования;
- высокий порог вхождения.

### **1.3.3. Выбор платформы**

Самым доступным и удобным для пользования является Unity3D [5]. Эта игровая платформа позволяет создавать игры в трехмерном и двумерном пространстве и легко развертывать готовые проекты на мобильных платформах. Инструмент Unity использует компонентно-ориентированный подход [7]. Это развитие объектно-ориентированного подхода, где все объекты являются наборами компонентов. Каждый компонент выполняет свою строго ограниченную задачу. Основой платформы Unity является специальный редактор, который предоставляет интерфейс для создания игр и приложений. В редакторе создаются сцены, на которые добавляются игровые объекты, с прикрепленными к ним компонентами. Компонентами являются скрипты, написанные на C# [2, 13] или на JavaScript.

### **1.4. Анализ аналогичных проектов**

В ходе анализа аналогичных проектов было найдено следующее.

Cisco IT Essentials Virtual Desktop [3] – виртуальный образовательный стенд по сборке персонального компьютера. Стенд демонстрирует процесс сборки, а также позволяет проводить тестирование. В проекте реализовано только по одному наименованию каждого компонента.

Виртуальный тренажер по сборке персонального компьютера [12] – сайт, созданный для изучения внутреннего устройства ПК. Тренажер позволяет отрабатывать навыки сборки конфигураций ПК в зависимости от типа решаемых задач. Из недостатков стоит отметить отсутствие 3D-визуализации.

PC Builder Simulator [8] – симулятор сборки компьютера. Проект реализован на Unity3D. Управление происходит от первого лица. В игре доступен только режим игры обучения, где пошагово объясняется, что нужно делать. В проекте так же, как и в Cisco IT Essentials Virtual Desktop, реализовано по одному наименованию каждого компонента.

## **2. ПРОЕКТИРОВАНИЕ ВИРТУАЛЬНОГО СТЕНДА**

### **2.1. Определение требований**

В ходе проектирования виртуального стенда были определены функциональные возможности, предоставляемые пользователю, а также нефункциональные требования к разрабатываемому приложению.

Функциональные требования:

- приложение должно предоставлять пользователю выбор комплектующих персонального компьютера;
- приложение должно предоставлять графическое представление всех комплектующих;
- приложение должно предоставлять информацию по каждому из комплектующих;
- приложение должно производить проверку выбранных комплектующих на совместимость;
- приложение должно предоставлять сборку персонального компьютера;
- приложение должно предоставлять возможность предварительной настройки BIOS для установки операционной системы;
- приложение должно иметь интуитивно понятный интерфейс;
- приложение должно выводить подсказки по сборке персонального компьютера.

Нефункциональные требования:

- приложение должно быть реализован в Unity3D;
- приложение должно функционировать на устройствах с операционной системой Android, IOS и Windows.

### **2.2. Проектирование архитектуры приложения**

В процессе проектирования была использована парадигма компонентно-ориентированного программирования. Согласно этой парадигме, приложение состоит из базовых сущностей, функциональность которых расширяется с помощью специализированных компонентов.

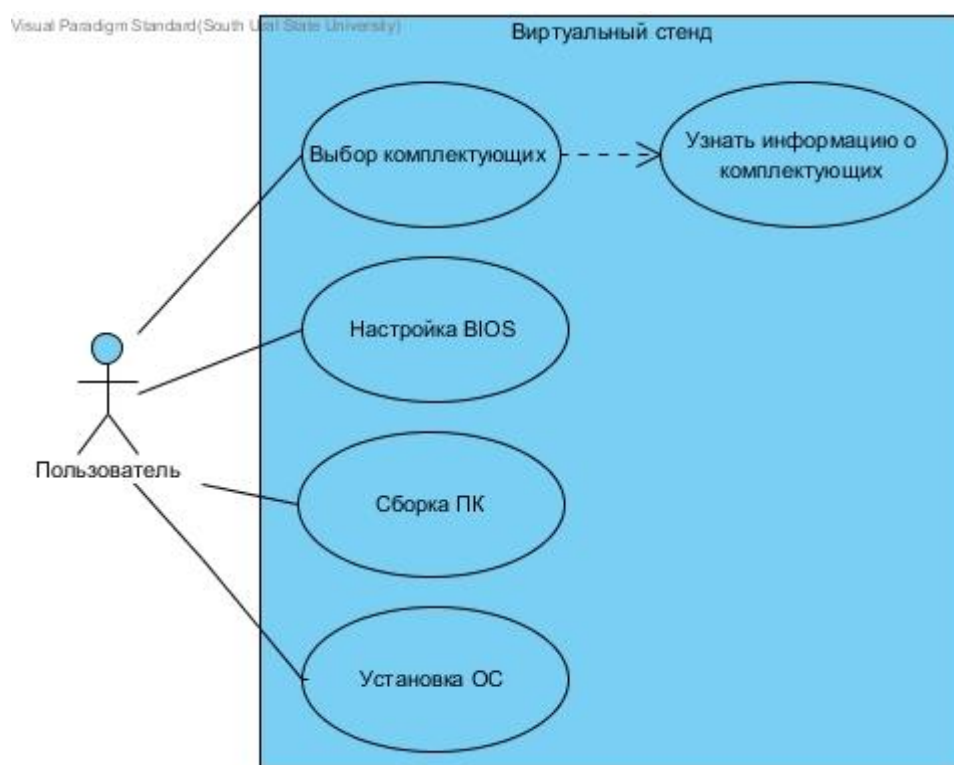
Во время проектирования было решено разделить реализуемое приложение на 4 модуля:

- модуль выбора комплектующих;
- модуль сборки персонального компьютера;
- модуль настройки BIOS;
- модуль установки операционной системы.

### 2.2.1. Диаграмма прецедентов

Диаграмма прецедентов – диаграмма, отражающая отношения между актерами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.

Для проектирования приложения был использован язык графического описания для объектного моделирования UML [11]. Была построена модель взаимодействия актера «Пользователь» с приложением в виде диаграммы вариантов использования (рис. 1).



**Рис. 1.** Диаграмма прецедентов

В рамках реализуемого приложения предусмотрен следующий актер – пользователь.

Пользователь с помощью данной системы может:

- производить выбор комплектующих;
- производить настройку BIOS;
- производить сборку персонального компьютера;
- производить установку операционной системы.

### 2.2.2. Архитектура модуля выбора комплектующих

Данный модуль предусматривает возможность выбора комплектующих, а также проверку на совместимость. Диаграмма классов изображена на рис. 2.

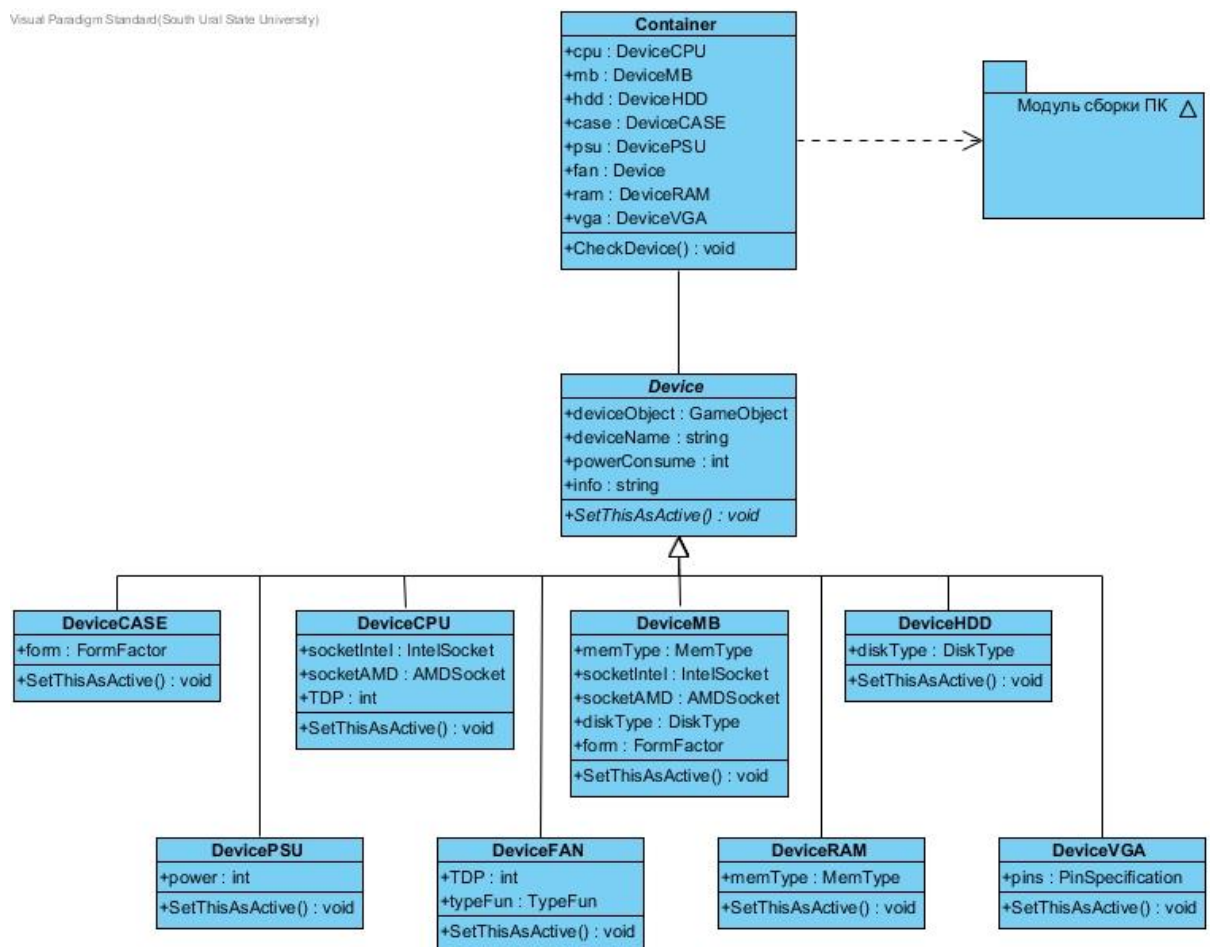


Рис. 2. Архитектура модуля выбора комплектующих

**Device** – абстрактный класс, в котором хранится ссылка на объект комплектующего, его имя, информация и потребляемая мощность. Абстрактный метод `SetThisAsActive()` служит для передачи значения выбранного комплектующего в класс **Container**.

*DeviceMB* – класс, экземпляром которого является материнская плата. Унаследован от класса Device. В нем содержатся: тип памяти, тип сокета, тип жесткого диска и форма корпуса.

*DeviceCPU* – класс, экземпляром которого является центральный процессор. Унаследован от класса Device. В нем содержатся: тип сокета и вырабатываемое количество теплоты.

*DeviceHDD* – класс, экземпляром которого является жесткий диск. Унаследован от класса Device. В нем содержится тип диска.

*DeviceCASE* – класс, экземпляром которого является корпус компьютера. Унаследован от класса Device. В нем содержится его форма.

*DevicePSU* – класс, экземпляром которого является блок питания. Унаследован от класса Device. В нем содержится его мощность.

*DeviceFAN* – класс, экземпляром которого является кулер. Унаследован от класса Device. В нем содержится его тип подключения и количество рассеиваемого тепла.

*DeviceRAM* – класс, экземпляром которого является оперативная память. Унаследован от класса Device. В нем содержится тип памяти.

*DeviceVGA* – класс, экземпляром которого является видеопамять. Унаследован от класса Device. В нем содержится тип подключения дополнительного питания.

*Container* – класс служит для проверки совместимости выбранных комплектующих. При правильной собранной конфигурации информация о выбранных комплектующих, а именно объект для сборки, передается в модуль сборки ПК, в противном случае появляется окно ошибки.

### **2.2.3. Архитектура модуля сборки персонального компьютера**

Данный модуль предусматривает возможность симуляции сборки персонального компьютера. Пользователь должен в правильной последовательности выбрать места подключения комплектующих. После сборки приступить к предварительной настройке BIOS. Диаграмма классов модуля сборки персонального компьютера изображена на рис. 3.



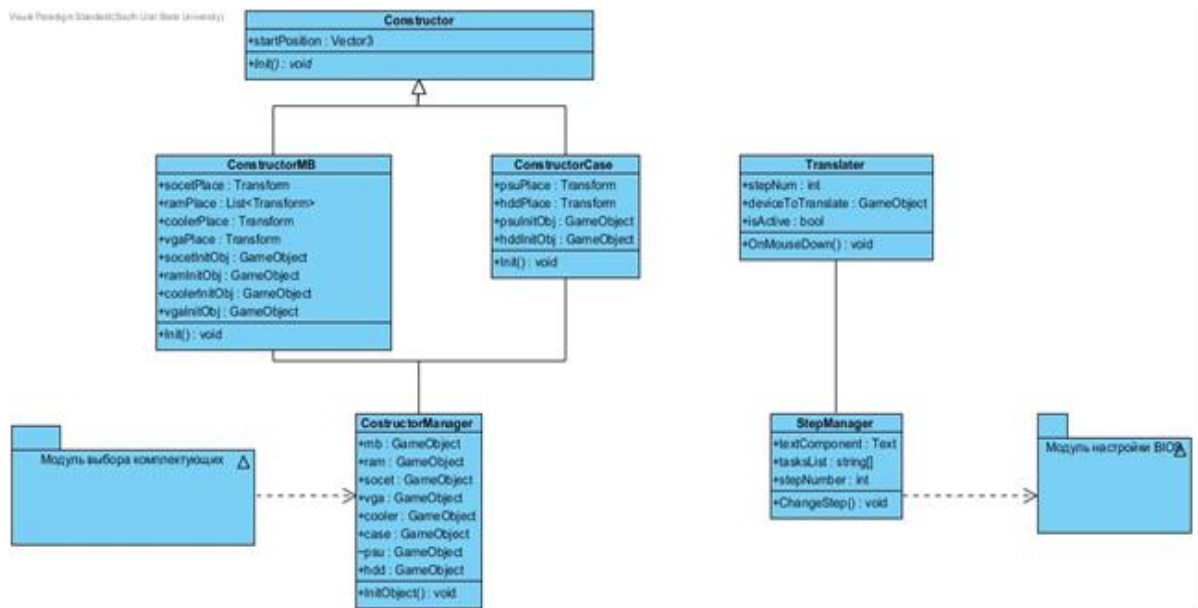


Рис. 3. Архитектура модуля сборки персонального компьютера

**ConstructorManager** – класс инициализирует материнскую плату и корпус персонального компьютера и передает им выбранные объекты из модуля выбора комплектующих.

**Constructor** – абстрактный класс. Имеет абстрактный метод `Init()` для инициализации объектов.

**ConstructorMB** – класс конструктора материнской платы. Унаследован от класса `Constructor`. В нем переопределен метод `Init()` для инициализации центрального процессора, кулера, оперативной памяти и видеокарты.

**ConstructorCase** – класс конструктора корпуса персонального компьютера. Унаследован от класса `Constructor`. В нем переопределен метод для инициализации блока питания и жесткого диска.

**Translator** – класс, необходимый для перемещения комплектующего в его позицию. В нем хранится номер шага последовательности сборки, комплектующее и флаг, который показывает активно ли оно в данное время или нет. Этот класс находится на объектах, являющимися местами подключения комплектующего. По нажатию на этот объект вызывается метод `OnMouseDown()`, и если шаг последовательности сборки равен текущему

шагу и объект не активен, то объект становится активным и перемещается на свое место. Шаг последовательности увеличивается на один.

*StepManager* – класс отвечает за последовательность сборки. В нем хранится номер шага и его описание. По окончании сборки класс передает информацию в модуль настройки BIOS.

#### 2.2.4. Архитектура модуля настройки BIOS

Данный модуль предусматривает возможность симуляции предварительной настройки BIOS (рис. 4). Пользователь должен выбрать способ установки: с DVD-диска или USB-накопителя. После выбора он должен пройти в нужную вкладку BIOS и выбрать приоритет загрузки устройств, сохранить изменения и если приоритет выставлен правильно, то приступить к установке операционной системы.

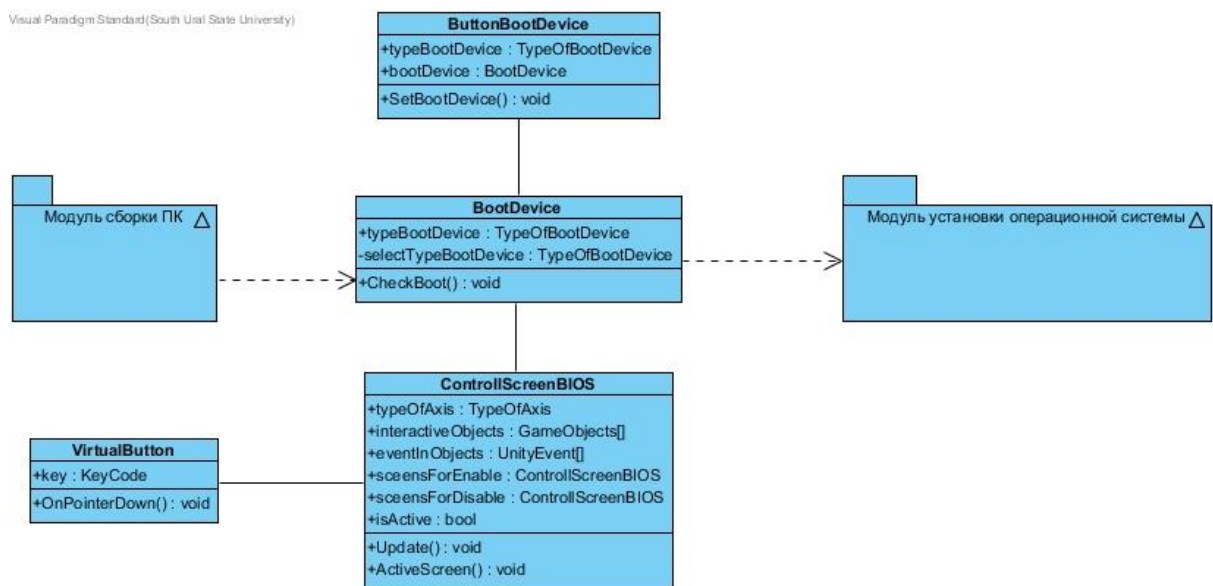


Рис. 4. Архитектура модуля настройки BIOS

*ControllScreenBIOS* – класс, экземпляры которого являются панели BIOS. В нем хранится тип перемещения: вертикальный или горизонтальный, объекты, которые должны включаться или переключаться при перемещении в этой панели, события, которые должны происходить при нажатии кнопки Enter в выбранной позиции и другие панели, которые должны активироваться или деактивироваться при нажатии кнопки Enter или Escape. Также флаг активности данной панели. Класс имеет два метода: Ur-

date() и ActivateScreen(). Update() вызывается каждый кадр. Если панель активна, то он ожидает нажатия кнопок навигации Enter и Escape и производит соответствующие манипуляции. Метод ActivateScreen() делает панель активной.

**VirtualButton** – класс отвечает за виртуальные кнопки, т.к. на мобильном устройстве отсутствуют кнопки навигации Enter и Escape. По нажатию вызывается метод OnPointerDown(), который сообщает классу ControllScreenBIOS о нажатии определенной кнопки.

**BootDevice** – класс, который отвечает за проверку правильного выбора приоритета загрузки.

**ButtonBootDevice** – класс, экземпляром которого является кнопка выбора приоритета загрузки. По нажатию на нее вызывается метод SetBootDevice() который передает выбранный приоритет в класс BootDevice.

### 2.2.5. Архитектура модуля установки операционной системы

Данный модуль предусматривает возможность пошаговой симуляции установки операционной системы (рис. 5). Было предложено пошагово показывать изображение этапа установки операционной системы.

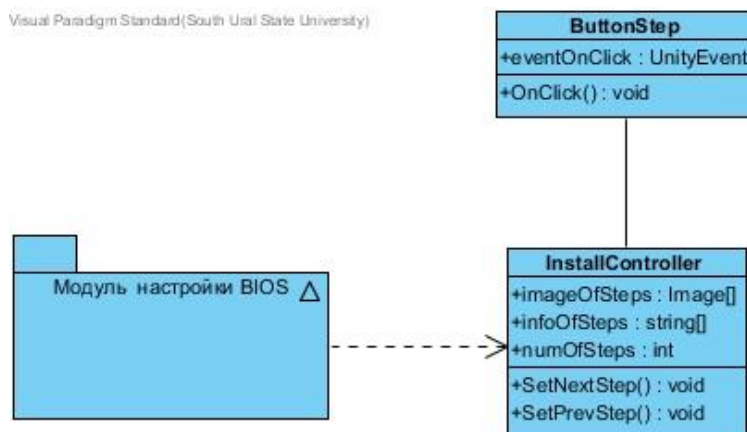


Рис. 5. Архитектура модуля установки операционной системы

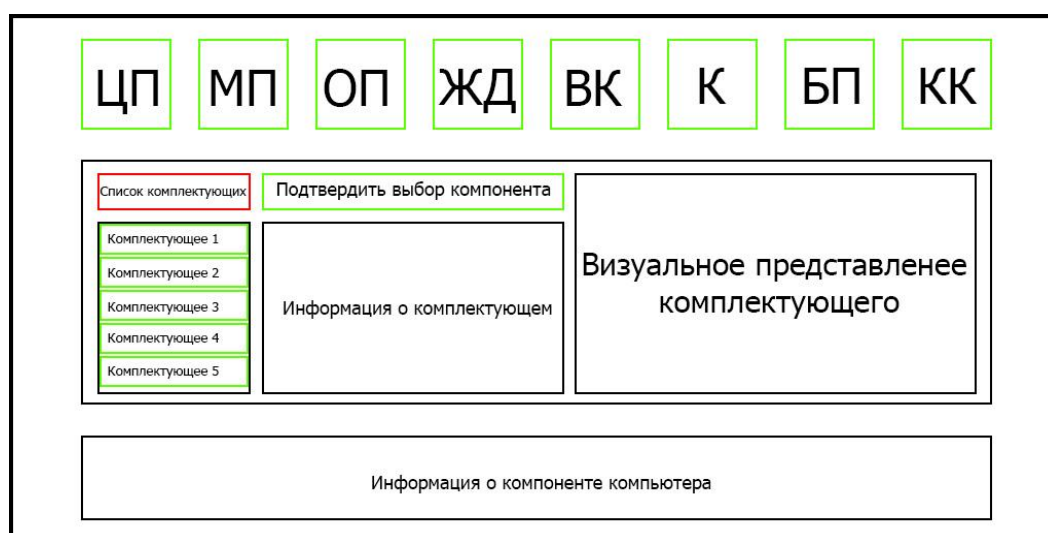
**InstallController** – класс отвечает за контроль этапов установки. В нем хранятся изображения и описания каждого шага. Присутствуют методы SetNextStep() и SetPrevStep() для показа следующего и предыдущего шага.

**ButtonStep** – класс, экземпляром которого является кнопка шага. В редакторе настраивается событие `eventOnClick`, которое должно произойти по вызову метода `OnClick()`.

### 2.2.6. Проектирование пользовательского интерфейса

Интерфейс проекта также разделен на 4 составляющие для каждого модуля. Было создано визуальное представление о будущем каждого из них. Главной особенностью должно быть удобство пользования на мобильных устройствах. Текст должен быть читабельным, а кнопки большими для удобства нажатия.

На рис. 6 представлен эскиз экрана для выборка комплектующих.



**Рис. 6.** Эскиз экрана для выбора комплектующих

Зеленым помечены кнопки, а красным – выпадающий список подтипов компонента компьютера.

На данном экране в верхней части расположены кнопки компонентов компьютера: ЦП – центральный процессор, МП – материнская плата, ОП – оперативная память, ЖД – жесткий диск, ВК – видеокарта, К – кулер, БП – блок питания, КК – корпус компьютера. При нажатии на них в нижней части экрана появляется информация о назначении данного компонента. Выпадающий список подтипов компонента необходим для более удобной навигации по комплектующим. При выборе подтипа появляются кнопки самих комплектующих. По нажатию на них в правой части экрана появля-

ется визуальное представление комплектующего, а в середине – его описание и характеристики. При нажатии кнопки «Подтвердить выбор компонента», выбранный компонент сохраняется для дальнейшей сборки.

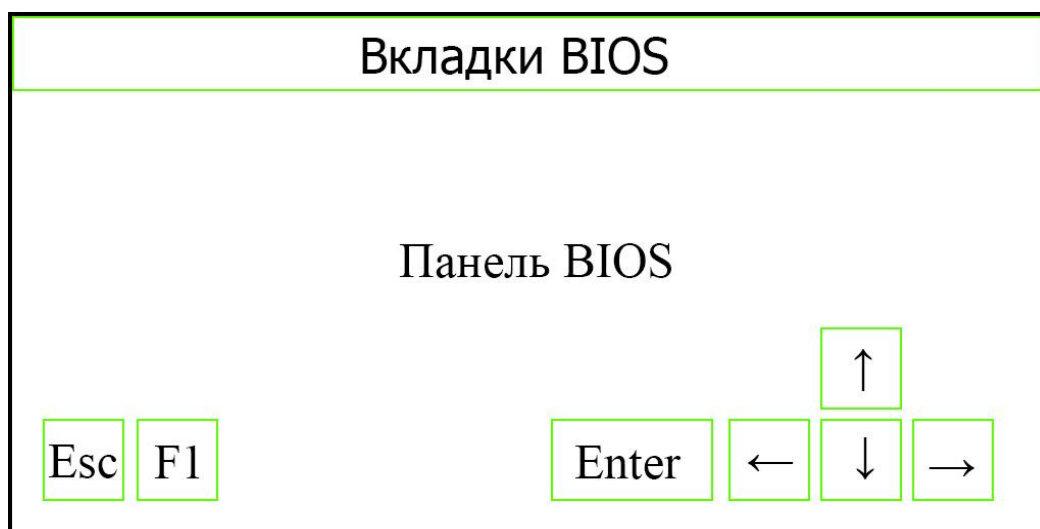
На рис. 7 представлен эскиз экрана для сборки компьютера.



**Рис. 7.** Эскиз экрана для сборки компьютера

На данном экране в нижней части расположен текст с информацией об этапе сборки. В верхнем правом углу расположена кнопка переключения вида со сборочного стола на материнскую плату, для удобства подключения комплектующих.

На рис. 8 представлен эскиз экрана предварительной настройки BIOS.



**Рис. 8.** Эскиз экрана предварительной настройки BIOS

На данном экране в верхней части расположены вкладки переключения между панелями BIOS, а в центре – панель BIOS. В нижней части расположены виртуальные кнопки навигации Enter, Escape и F1.

На рис. 9 представлен эскиз экрана симуляции установки операционной системы.



**Рис. 9.** Эскиз экрана симуляции установки операционной системы

На данном экране в нижней части расположена информация о шаге установки операционной системы. В центре расположено изображение самого окна установки. В верхнем левом углу кнопка «Назад», при нажатии на которую показывается предыдущий шаг установки. В верхнем правом углу кнопка «Вперед», при нажатии на которую показывается следующий шаг установки.

### 3. РЕАЛИЗАЦИЯ

Для реализации приложения было решено использовать систему сцен, которая предоставляется платформой Unity. Проект был разбит на 4 сцены: сцена выбора комплектующих, сцена сборки компьютера, сцена предварительной настройки BIOS и сцена симуляции установки операционной системы.

Для реализации проекта использовались 3D-модели существующих комплектующих компьютера. Модели были предоставлены предприятием. Для их визуального представления в среде разработки Unity был использован встроенный компонент MeshRenderer, который воспринимает модель, как объект, состоящий из треугольников, с наложенной текстурой.

Для реализации интерфейса приложения был использован компонент Canvas. Этот компонент представляет собой абстрактное пространство, в котором производится настройка и прорисовка объектов интерфейса.

#### 3.1. Реализация сцены выбора комплектующих

Для реализации сцены выбора комплектующих было решено разбить процесс выбора на два этапа. На первом этапе пользователь выбирает одного из производителя центрального процессора: AMD [1] или Intel [6]. На рис. 10 представлен первый шаг.



Рис. 10. Выбор платформы для сборки

После выбора центрального процессора пользователь переходит к шагу выбора компонентов компьютера (рис. 11).



Рис. 11. Выбор компонента для сборки

К каждой кнопке комплектующего подключен соответствующий компонент, унаследованный от класса Device. В редакторе сцены настроены все его характеристики, и при нажатии на кнопку «Подтвердить выбор компонента» данный класс передается в класс Container.

По нажатию кнопок компонентов переключаются заранее настроенные панели категорий.

После подтверждения выбора компонента в каждой категории пользователь должен нажать кнопку «Далее», что вызовет метод CheckDevice() из класса Container для проверки совместимости выбранных комплектующих. Метод сравнивает характеристики выбранных комплектующих между собой. Центральный процессор с сокетом материнской платы. Тип оперативной памяти с разъемом для нее на материнской плате. Размеры материнской платы и выбранного корпуса. Тип подключения жесткого диска и разъем для его подключения на материнской плате. Выделяемое тепло центрального процессора и поглощаемое тепло кулера. Необходимую



мощность для всех комплектующих и мощность блока питания. При правильном выборе происходит запуск сцены для сборки компьютера, при неправильном выборе всплывает окно, информирующее о неверном выборе комплектующих.

### 3.2. Реализация сцены сборки компьютера

На сцене сборки компьютера находится стол, на котором расположены монитор, клавиатура, мышь, выбранный корпус компьютера и материнская плата (рис. 12). При наведении курсора или долгое нажатие на сенсорный экран мобильного устройства на объект, объект подсвечивается и появляется окно с его описанием.



Рис. 12. Сборочный стол

При нажатии на кнопку «МОБО» камера фокусируется на материнской плате для более удобного подключения комплектующих к ней (рис. 13).

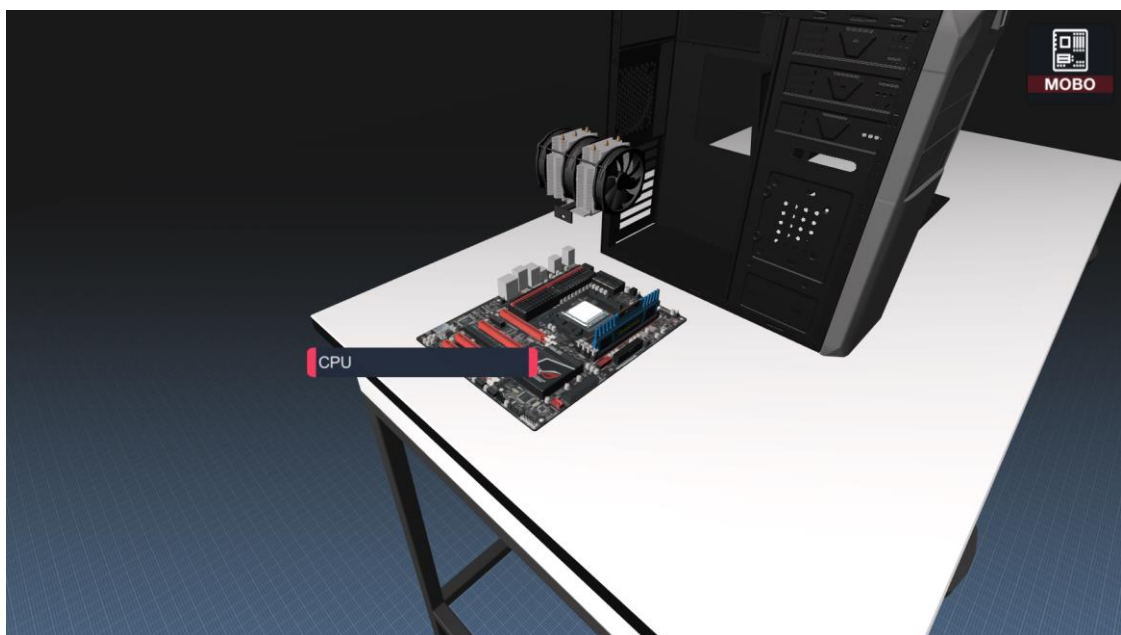
При нажатии на разъем или место подключения, комплектующее появляется на сцене и перемещается на свое место (рис. 14). Для реализации перемещений объектов был использован DoTween [4]. Это дополнение к

Unity для реализации анимации. Позволяет плавно изменять положение объекта в пространстве, его размеры, цвет и многое другое.



**Рис. 13.** Фокусировка камеры на материнской плате

Был реализован скрипт для перемещения камеры. Он позволяет вращать камеру вокруг объекта, а при нажатии на него происходит приближение. Так же приближать и отдалять камеру можно колесом мыши или движением пальцев по сенсорному экрану навстречу друг другу или в противоположную сторону.



**Рис. 14.** Перемещение объекта в свое положение

Для реализации подключения кабелей было решено использовать встроенный компонент LineRenderer. Он прорисовывает линию от основания подключения, до конечной точки (рис. 15).

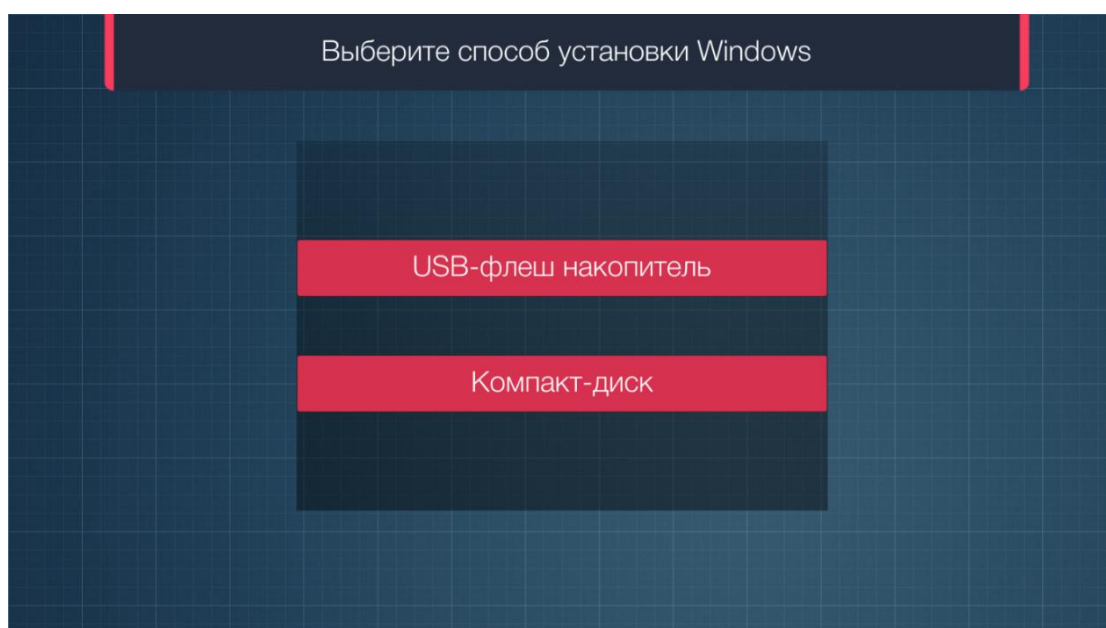


**Рис. 15.** Подключение кабеле к комплектующим

После полной сборки компьютера пользователь переходит в сцену предварительной настройки BIOS.

### 3.3. Реализация сцены предварительной настройки BIOS

В сцене предварительной настройки BIOS пользователь для начала должен выбрать способ установки операционной системы (рис. 16).



**Рис. 16.** Выбор способа установки операционной системы

Для управления BIOS на мобильных устройствах пришлось реализовать виртуальные кнопки навигации Enter, Escape и F1, которые дублируют настоящие клавиши (рис. 17).

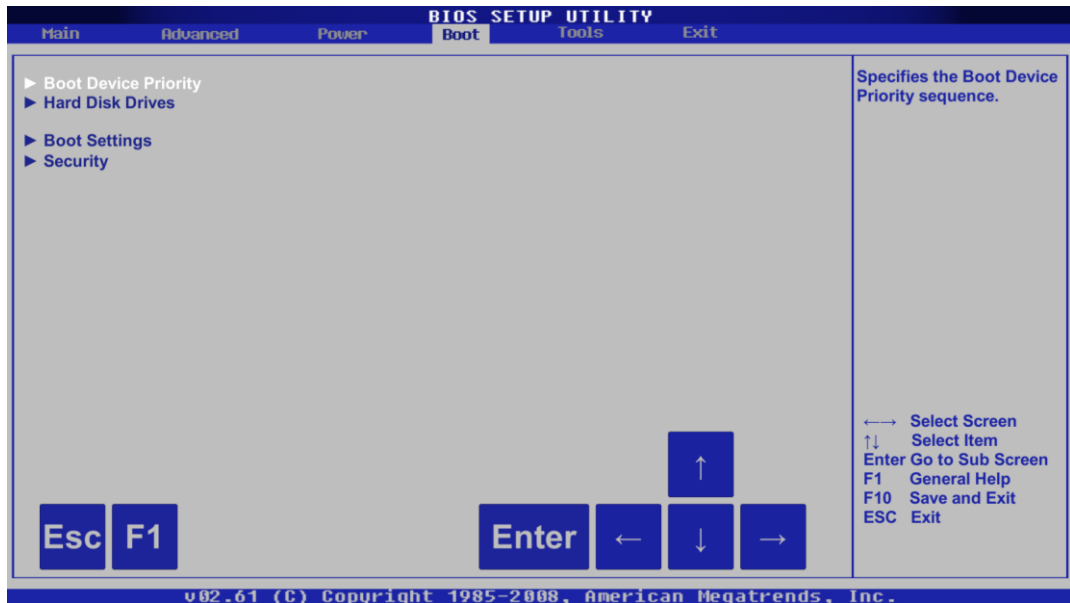


Рис. 17. Виртуальные клавиши

Пользователь выбирает нужный приоритет загрузки и выходит из настройки BIOS с сохранением изменений (рис. 18).

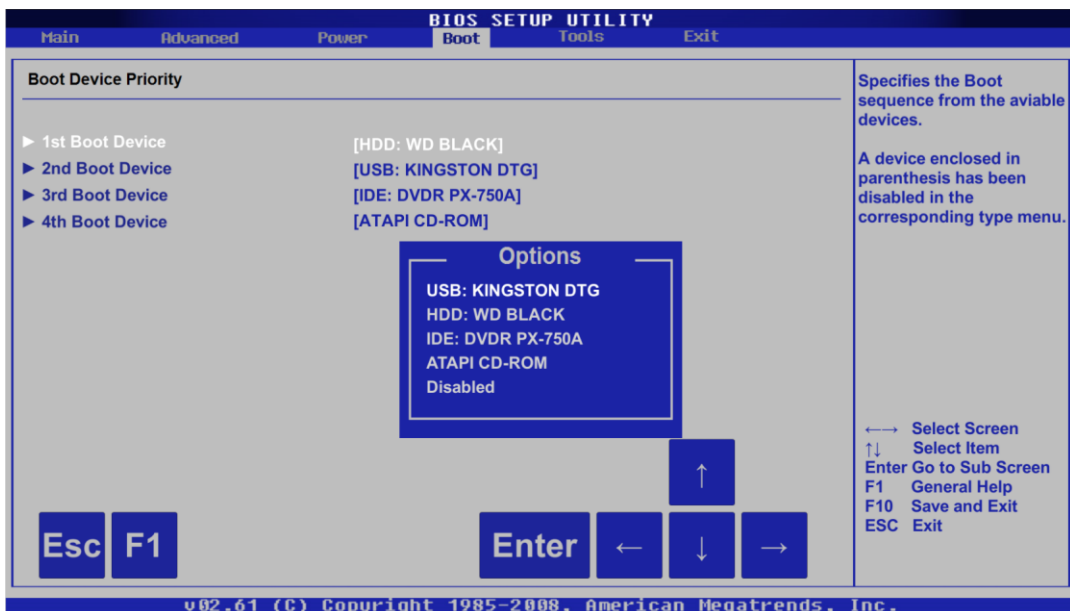
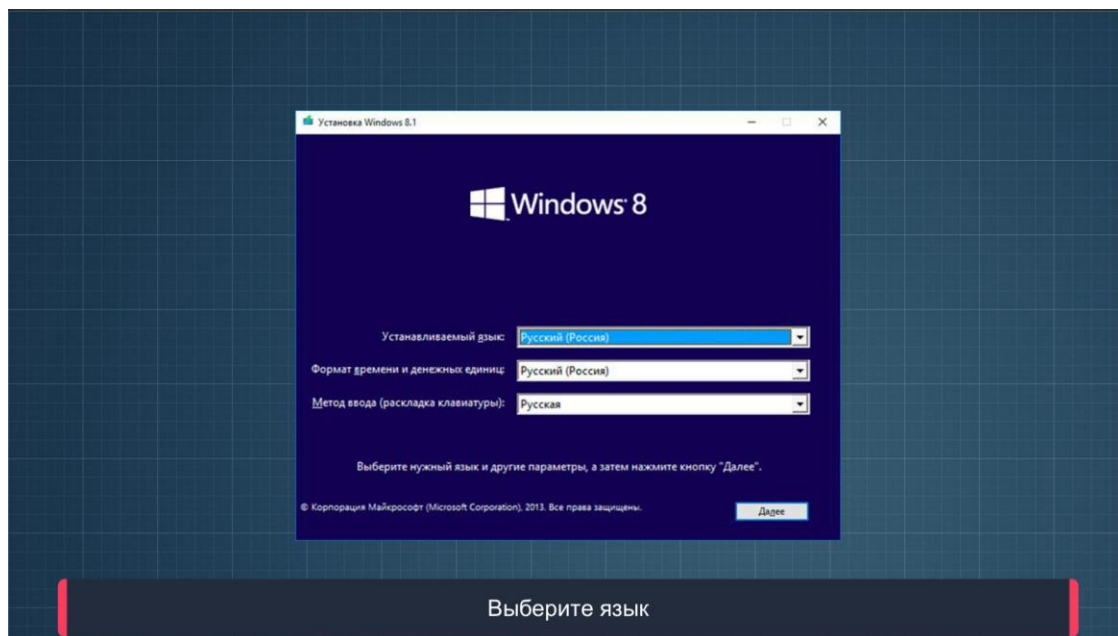


Рис. 18. Окно выбора приоритета загрузки

Если пользователь выполнил все верно, он переходит на сцену симуляции установки операционной системы.

### 3.4. Реализация сцены симуляции установки операционной системы

Для реализации сцены симуляции установки операционной системы было решено использовать установку Windows 8 (рис. 19). Пользователь проходит все этапы установки нажимая на кнопки «Далее».



**Рис. 19.** Окно симуляция установки операционной системы

После прохождения всех этапов приложение предлагает начать с начала или выйти из него.

## 4. ТЕСТИРОВАНИЕ

Для проверки работоспособности игрового приложения было проведено функциональное тестирование и юзабилити-тестирование.

### 4.1. Функциональное тестирование

Функциональное тестирование – это тестирование ПО в целях проверки реализуемости функциональных требований, то есть способности ПО в определенных условиях решать задачи, нужные пользователям [17].

Тест выбора комплектующих компьютера (табл. 1).

**Табл. 1.** Результат тестирования выбора комплектующих компьютера

Действие	Ожидаемый результат	Результат теста
Выбрать производителя платформы AMD	Появление в категории центрального процессора и материнской платы комплектующих с сокетом AMD	Да
Выбрать производителя платформы Intel	Появление в категории центрального процессора и материнской платы комплектующих с сокетом Intel	Да
Выбор компонента	Компонент помечается как выбранный	Да
Выбрать центральный процессор и материнскую плату с разными сокетами	Появление окна об ошибке с текстом “Мат. плата и процессор не совместимы”	Да
Выбрать материнскую плату и оперативную память с разными сло	Появление окна об ошибке с текстом “Мат. плата не поддерживает	Да

Действие	Ожидаемый результат	Результат теста
тами	данный тип оперативной памяти	
Выбрать материнскую плату и жесткий диск с разными типами подключения	Появление окна об ошибке с текстом “Мат. плата не поддерживает выбранный тип твердотельного накопителя”	Да
Выбрать центральный процессор с выделяемой теплотой выше, чем поглощаемая у выбранного кулера	Появление окна об ошибке с текстом “Система Охлаждения не удовлетворяет минимальным требованиям”	Да
Выбрать комплектующие с потребляемой мощностью больше, чем выделяемая выбранным блоком питания	Появление окна об ошибке с текстом “Мощности блока питания недостаточно для собранной системы”	Да
Выбрать материнскую плату меньше чем выбранный корпус компьютера	Появление окна об ошибке с текстом “Мат. плата и корпус разного форм-фактора”	Да
Выбрать совместимые комплектующие компьютера	Переход к сборке компьютера	Да

Тест сборки компьютера (табл. 2).

**Табл. 2.** Результат тестирования сборки компьютера

<b>Действие</b>	<b>Ожидаемый результат</b>	<b>Результат теста</b>
Переход из сцены выбора комплектующих к сцене сборки персонального компьютера	Появление в сцене выбранной материнской платы и корпуса	Да
Нажать на сокет материнской платы	Появление выбранного процессора и установка его в нужное место	Да
Нажать на слот оперативной памяти	Появление выбранной оперативной памяти и установка ее в нужное место	Да
Нажать на центральный процессор	Появление выбранного кулера и установка его в нужное место	Да
Нажать на боковую стенку корпуса компьютера	Установка материнской платы в корпус	Да
Нажать на место установки жесткого диска	Появление выбранного жесткого диска и установка его в нужное место	Да
Нажать на место установки блока питания	Появление выбранного блока питания и установка его в нужное место	Да



Действие	Ожидаемый результат	Результат теста
	место	
Нажать на слот подключения видеокарты	Появление выбранной ранее видеокарты и установка ее в нужное место	Да
Нажать на разъем подключения жесткого диска	Появление кабеля для подключения жесткого диска к материнской плате и установка его в нужное место	Да
Нажать на разъем подключения кабеля питания материнской платы	Появление кабеля питания материнской платы и установка его в нужное место	Да
Нажать на разъем подключения кабеля питания центрального процессора	Появление кабеля питания центрального процессора и установка его в нужное место	Да
Нажать на разъем подключения кабеля питания жесткого диска	Появление кабеля питания жесткого диска и установка его в нужное место	Да
Нажать на разъем подключения монитора на видеокарте	Появление кабеля подключения монитора к видеокарте и установка его в нужное место	Да

<b>Действие</b>	<b>Ожидаемый результат</b>	<b>Результат теста</b>
Нажать на разъем подключения клавиатуры на материнской плате	Появление кабеля подключения клавиатуры к материнской плате и установка его в нужное место	Да
Нажать на разъем подключения компьютерной мыши на материнской плате	Появление кабеля подключения компьютерной мыши к материнской плате и установка его в нужное место	Да
Прохождение всех этапов сборки	Появление сообщение об успешной сборке компьютера и переход к сцене предварительной настройки BIOS	Да

Тест предварительной настройки BIOS (табл. 3).

**Табл. 3.** Результат тестирования предварительной настройки BIOS

<b>Действие</b>	<b>Ожидаемый результат</b>	<b>Результат теста</b>
Переход из сцены сборки компьютера в сцену предварительной настройки BIOS	Появление окна выбора способа установки операционной системы	Да
Перемещение вправо по вкладкам BIOS по	Переход на вкладку справа	Да

Действие	Ожидаемый результат	Результат теста
нажатие виртуальной кнопки “Вправо”		
Перемещение влево по вкладкам BIOS по нажатие виртуальной кнопки “Влево”	Переход на вкладку слева	Да
Перемещение вниз по панели BIOS по нажатие виртуальной кнопки “Вниз”	Переход на позицию ниже	Да
Перемещение вверх по панели BIOS по нажатие виртуальной кнопки “Вверх”	Переход на позицию выше	Да
Перемещение в подразделы BIOS по нажатие виртуальной кнопки “Enter”	Переход в подраздел	Да
Перемещение из подразделов BIOS по нажатие виртуальной кнопки “Escape”	Переход из подраздела	Да
Нажать на виртуальную кнопку “F1”	Показ справки пользования BIOS	Да
Нажать на раздел Boot Device Priority	Показ раздела приоритета загрузки	Да

<b>Действие</b>	<b>Ожидаемый результат</b>	<b>Результат теста</b>
Выбрать приоритет загрузки	Поменяется приоритет загрузки	Да
Выйти из BIOS с неправильно настроенным приоритетом загрузки	Появления окна ошибки	Да
Выйти из BIOS с правильно настроенным приоритетом загрузки	Переход к сцене симуляции установки операционной системы	Да

Тест симуляции установки операционной системы (табл. 4).

**Табл. 4.** Результаты тестирования симуляции установки операционной системы

<b>Действие</b>	<b>Ожидаемый результат</b>	<b>Результат теста</b>
Нажать на кнопку “Далее”	Переход к следующему этапу	Да
Нажать на кнопку “Назад”	Переход к предыдущему этапу	Да
Прохождение всех этапов	Появление меню	Да
Нажать на кнопку “Начать заново”	Переход к сцене выбора комплектующих	Да
Нажать на кнопку “Выход”	Выход из приложения	Да

## 4.2. Юзабилити-тестирование

Юзабилити-тестирование (проверка эргономичности) – метод оценки удобства продукта в использовании, основанный на привлечении пользователей в качестве испытателей и суммировании полученных от них выводов [17].

При прохождении данного теста были привлечены четыре пользователя. Участникам тестирования было предложено решить следующие задачи:

- выбрать несовместимые комплектующие компьютера;
- выбрать совместимые комплектующие компьютера;
- собрать компьютер из выбранных комплектующих;
- настроить BIOS;
- пройти симуляцию установки операционной системы.

Все задачи были решены всеми участниками тестирования без затруднений. Было выявлено несколько косметических недочетов, которые были быстро устранены. Тест успешно завершен.

## **ЗАКЛЮЧЕНИЕ**

В результате разработки виртуального стенда для моделирования процесса сборки персональных компьютеров на мобильных устройствах были достигнуты следующие результаты:

- произведен анализ аналогичных проектов;
- выполнено проектирование приложения;
- приложение реализовано;
- выполнено тестирование разработанного приложения;
- внедрено в опытную эксплуатацию.

В дальнейшем приложение получит ряд нововведений:

- добавление новых BIOS;
- добавление новых операционных систем;
- добавление новых комплектующих.

## ЛИТЕРАТУРА

1. AMD. [Электронный ресурс] URL: <http://www.amd.com/ru-ru> (дата обращения: 15.04.2017).
2. C# Programming Guide. [Электронный ресурс] URL: <https://msdn.microsoft.com/en-us/library/67ef8sbd.aspx> (дата обращения: 15.04.2017).
3. Cisco IT Essentials: PC Hardware and Software v4.1 Overview [Электронный ресурс] URL: [http://www.cisco.com/web/offer/emea/6754/docs/ITEPC\\_Overview\\_09Mar10.pdf](http://www.cisco.com/web/offer/emea/6754/docs/ITEPC_Overview_09Mar10.pdf) (дата обращения: 24.04.2017).
4. DoTween. [Электронный ресурс] URL: <http://dotween.demigiant.com> (дата обращения: 02.04.2017).
5. Goldstone W. Unity Game Development Essentials. / Will Goldstone; пер. с англ. – М.: Вильме. – 266 с.
6. Intel. [Электронный ресурс] URL: <http://www.intel.ru/content/www/ru/ru/homepage.html> (дата обращения: 28.04.2017).
7. Kung-Kiu Lau. Software component models: past, present and future. // CBSE '14 Proceedings of the 17th international ACM Sigsoft symposium on Component-based software engineering, 2014. – P. 185-186.
8. PC Building Simulator. [Электронный ресурс] URL: <https://mcintyre.itch.io/pc-building-simulator> (дата обращения: 28.04.2017).
9. Unity Documentation. [Электронный ресурс] URL: <http://docs.unity3d.com/> (дата обращения: 28.03.2017)
10. Unreal Engine. [Электронный ресурс] URL: <https://www.unrealengine.com/> (дата обращения: 28.03.2017).
11. Арлоу Д., Нейштадт А. UML 2 и Унифицированный процесс. Практический объектно-ориентированный анализ и проектирование. – М.: Символ-Плюс, 2007. – 624 с.

12. Виртуальный тренажер по сборке персонального компьютера. [Электронный ресурс] URL: <http://assembly-pc.narod.ru> (дата обращения: 28.03.2017).
13. Гриффитс Й. Программирование на C# 5.0. – М.: Эксмо, 2014. – 1136 с.
14. Гузенко Е.Н., Сурядный А.С. Персональный компьютер. – М.: АСТ: Астрель, 2011. – 544 с.
15. Донцов Д.А. BIOS и тонкая настройка ПК. Легкий старт. – СПб.: Питер, 2007. – 160 с.
16. Кубин А.В., Дмитриев П.А., Финкова М.А. 1000 и 1 секрет BIOS по «тонкой» настройке, решению проблем и оптимизации работы компьютера. – СПб.: Наука и Техника, 2007. – 368 с.
17. Майерс Г., Баджет Т. Искусство тестирования программ. – М.: Вильямс, 2012. – 272 с.
18. Шабров Н.Н. Программно-аппаратные комплексы виртуального окружения – ключевые компоненты технологий виртуального инжиниринга. // CAD/CAM/CAE Observer, 2016. – № 3. – С. 83–86.