испытаний, а также для проверки их профессиональных компетенции на основе выполнения заданий по методам ГОСТ Р 51369-99. Данная программная система может найти широкое применение в центрах проведения климатических испытаний, а программное обеспечение и алгоритмы, использованные при ее реализации, могут послужить основой для создания симуляторов работы и программ-тренажеров и для других образцов испытательной техники.

1. Иващенко, А. В. Сценарное представление операции в хирургическом тренажере / А. В. Иващенко, Н. А. Горбаченко, А. В. Колсанов // Системы управления и информационные технологии. – 2016. – Т. 64. – № 2. – С. 91-97.
2. Устюжанина, А. Ю. Разработка и создание веб-приложения по моделированию чрезвычайных ситуаций на опасных производственных объектах нефтегазового комплекса / А. Ю. Устюжанина, А. А. Галкина, Д. С. Фукалов, А. А. Шарафутдинов, И. А. Хайретдинов, И. Ф. Хафизов // Проблемы сбора, подготовки и транспорта нефти и нефтепродуктов. – 2017. – № 1 (107). – С. 210-218.
3. Косаговская, И. И. Современные подходы к симуляционному обучению медицинских кадров / И. И. Косаговская, В. В. Мадьянова, Ю. В. Королева // Проблемы стандартизации в здравоохранении. – 2016. – № 5-6. – С. 22-28.
4. Лапшин, Э. В. Проектирование авиационных тренажеров с распараллеливанием вычислительных процессов / Э. В. Лапшин, Б. К. Кемалов, Б. Ж. Куатов // Надежность и качество сложных систем. – 2016. – № 4 (16). – С. 128-141.
5. Комолова, О. А. Разработка и использование компьютерных тренажеров в металлургии / О. А. Комолова, К. В. Григорович, Д. В. Горкуша, Д. О. Теребикина // Тяжелое машиностроение. – 2017. – № 1-2. – С. 55-61.
6. Climate Test Chambers WK3 WKS3 [Электронный ресурс]. – Режим доступа: <https://www.weiss-technik.com/fileadmin/Redakteur/Mediathek/Broschueren/WeissTechnik/Umweltsimulation/Weiss-Technik-WK3-WKS3-EN.pdf> (дата обращения: 19.07.2021)
7. SIMPATI Software. Installation and Operation Manual [Электронный ресурс]. – Режим доступа: http://weiss-na.com/wp-content/uploads/Simpat_4.50_user_guide.pdf (дата обращения: 19.07.2021)
8. Sketchup Help Center [Электронный ресурс]. – Режим доступа: <https://help.sketchup.com/ru/content/sketchup-make-and-sketchup-pro> (дата обращения: 25.07.2021)
9. Лизяев, С. Д. Особенности создания анимации при разработке обучающих симуляторов в среде Unity / С. Д. Лизяев, Р. С. Молотов // Вестник Ульяновского государственного технического университета. – 2016. – № 3 (75). – С. 41-43.
10. Абляев, М. Р. Unity как средство разработки программ с возможностью визуализации технологий виртуальной реальности / М. Р. Абляев, Ф. Р. Аметов, И. Ш. Мевлют // Информационно-компьютерные технологии в экономике, образовании и социальной сфере. – 2016. – № 4 (14). – С. 71-75.
11. ГОСТ Р 51369-99. Методы испытаний на стойкость к климатическим внешним воздействующим факторам машин, приборов и других технических изделий. Испытания на воздействие влажности. [Электронный ресурс]. – Режим доступа: <http://docs.cntd.ru/document/1200006116> (дата обращения: 11.07.2021)

Шавтикова Л.М., Мусакаев И.М.

Разработка компьютерной игры в жанре «Аркада»

*Северо-Кавказская государственная академия
(Россия, Черкесск)*

doi: 10.18411/lj-08-2021-34

Аннотация

В данной статье рассматривается разработка аркадной игры, которая представляет собой простое, интуитивно понятное управление игровыми объектами, где пользователю не нужно настраивать и изучать управление игрой, что позволяет практически сразу начать игровой процесс.

Ключевые слова: компьютерная игра, аркадная игра, программирование, дизайн игр, функциональное тестирование.

Abstract

This article discusses the development of an arcade game, which is a simple, intuitive control of game objects, where the user does not need to configure and learn how to control the game, which allows you to start the gameplay almost immediately.

Keywords: computer game, arcade game, programming, game design, functional testing.

Разработка или дизайн игр - это процесс, который начинается с идеи или концепции. Идея игры может относиться к одному или нескольким жанрам. У разных компаний разные формальные процедуры в отношении дизайна и разработки игр. Не существует стандартизированного метода разработки, однако есть общие черты.

Приложение аркадной игры - это приложение, в котором игрок управляет объектом, вертикально перемещающимся по игровому полю, разрушая препятствия или преодолевая их. Аркадные игры включают в себя все проекты жанра «файтинг», некоторые игры жанра «гонки» и некоторые игры жанра «шутер».

Разработанное игровое приложение рассчитано на использование одним игроком. Игра начинается с игрового поля, в левой части которого находится управляемый объект. Игрок управляет самолетом, который может двигаться вертикально вверх и вниз. Прямых выигрышных условий нет. Цель игрока - набрать как можно больше очков за проход объектов, движущихся навстречу игроку. Игра заканчивается, если персонаж сталкивается с объектом.

Разработка велась на языке программирования C# в среде разработки Visual Studio. Microsoft Visual Studio - это линейка продуктов Microsoft, включающая интегрированную среду разработки программного обеспечения и ряд других инструментов.

В результате разработки приложения была спроектирована диаграмма классов – UML (рисунок 1). UML (Unified Modeling Language) - это язык графического описания для объектного моделирования в области разработки программного обеспечения, моделирования бизнес-процессов, системного проектирования и отображения организационных структур.

Рисунок 1 - Диаграмма классов игрового приложения

Игровое приложение состоит из двух сцен: меню и игрового поля. Игровые объекты можно разделить на три категории: те, которые отвечают за создание и уничтожение объектов (GeneratorEnemy, GeneratorAmmo, DestroyObject); графический

интерфейс (Canvas и вложенные объекты); и игровое поле (Background, Middleground, Foreground и вложенные объекты).

Объект «Player» отвечает за поведение и логику игрока на игровом поле. К нему подключены компоненты - Sprite Renderer, BoxCollider2D, Rigidbody2D, Player (Script), Health (Script), Destroy (Script).

Компонент «Sprite Renderer» отвечает за двухмерное графическое представление игрового объекта. Компонент «BoxCollider2D» отвечает за обработку столкновений с другими объектами. Компонент «Rigidbody 2D» добавляет игровому объекту физические свойства, такие как масса, скорость и ускорение.

Компонент «Player (Script)» отвечает за перемещение игрока по игровому полю. Он позволяет назначать управляющие клавиши, допустимые направления движения (Project Aris), возможность поворота объекта по направлению к указателю мыши (LookAtCursor), а также дополнительную скорость движения.

Компонент «Health (Script)» отвечает за здоровье игрока. Он определяет максимальное значение здоровья (Hp), способ отображения уровня здоровья в графическом интерфейсе игры (Lifes), урон (CountDam - возраст). Когда уровень здоровья установлен на 0, игра заканчивается и выполняется переход в меню (загружается сцена «Меню»). Компонент «Destroy (Script)» отвечает за уничтожение вражеских персонажей, подсчет очков. Текущее количество снарядов отображается в верхней части экрана.

Объект «Background» отвечает за отображение фона игрового приложения. К нему подключен компонент «Scrolling (Script)».

Компонент «Scrolling (Script)» отвечает за непрерывное движение фона с той же заданной скоростью (Speed) и направлением (Direction), который использует два идентичных элемента, чтобы применить их в качестве фона. При движении справа налево в момент, когда правый край фона пересекается с правым краем камеры, создается новый фоновый объект, левый край которого соединяется с правым краем предыдущего объекта. В момент, когда фоновый объект покидает камеру, этот объект уничтожается. Благодаря этому создается впечатление непрерывного движения фона внутри экрана.

Объект «Middle Ground» отвечает за отображение декоративных элементов игрового фона. К нему также подключен компонент «Scrolling (Script)», отвечающий за постоянное перемещение и обновление декоративных элементов.

Объект «Generator Enemy» отвечает за случайную генерацию вражеских персонажей на игровом поле. К нему подключен компонент «Generator (Script)», который отвечает за случайное расположение препятствия на игровом поле в изначально заданном пространстве. Оно задается параметрами «SpawnPos (X, Y, Z)», представляя собой параллелепипед. Координаты X и Z указываются точно, координата Y указывается по модулю (случайная генерация осуществляется в любой точке пространства с координатами (X; от -Y до Y; Z)). Далее указывается количество типов вражеских персонажей (Size), отличающихся скоростью передвижения и размерами. Также указывается время простоя генератора после начала игры, по истечении которого, генератор начинает случайно располагать препятствия на игровом поле. Последним параметром компонента указывается среднее время между генерацией объектов.

Объект «DestroyObject» отвечает за уничтожение игровых объектов, покинувших игровое поле. К нему подключен компонент «BoxCollider2D».

Объект «Canvas» отвечает за отображение элементов графического интерфейса, показанных на рисунках 2 и 3. Он содержит объекты «Score», «withLifes» и «безLifes».

Объект «Score» отвечает за отображение текста с количеством баллов вверху экрана.

Рисунок 2 - Интерфейс игры

Рисунок 3 – Интерфейс меню

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

Функциональное тестирование - это тестирование программного обеспечения с целью проверки выполнимости функциональных требований, то есть способности программного обеспечения решать задачи, которые необходимы пользователям при определенных условиях.

Юзабилити - тестирование - это метод оценки юзабилити продукта, основанный на привлечении пользователей в качестве тестировщиков и обобщении полученных на их основе выводов [3, с.198].

1. Баджет, Т. Введение в объектно-ориентированный анализ, проектирование и итеративную разработку [Текст]/ Т. Баджет – М.: Вильямс, 2016. – 736 с.
2. Вигенс, К. Разработка требований к программному обеспечению [Текст]/ К. Вигенс. – М.: Русская Редакция, 2018. – 736 с.
3. Майерс Г. Искусство тестирования программ [Текст]/ Г. Майерс – М.: Вильямс, 2017. – 272 с.
4. Официальный сайт Unity3D. [Электронный ресурс] URL: <https://unity3d.com/ru> (дата обращения: 28.06.2021).
5. Официальный сайт Visual Studio. [Электронный ресурс] URL: <https://www.visualstudio.com/ru/vs/> (дата обращения: 28.06.2021).
6. C# Programming Guide. [Электронный ресурс] URL: <https://msdn.microsoft.com/en-us/library/67ef8sbd.aspx> (дата обращения: 30.06.2021).
7. Unity3D Manual. [Электронный ресурс] URL: <http://docs.unity3d.com/Manual/index.html> (дата обращения: 29.06.2021).