

определяется как половина суммы разностей между парами соседних пикселей. Посчитав значения CN, каждый пиксель может быть классифицирован как один из видов особых точек. Изображение отпечатка пальца с выделенными особыми точками приведено на рисунке 3б [3].

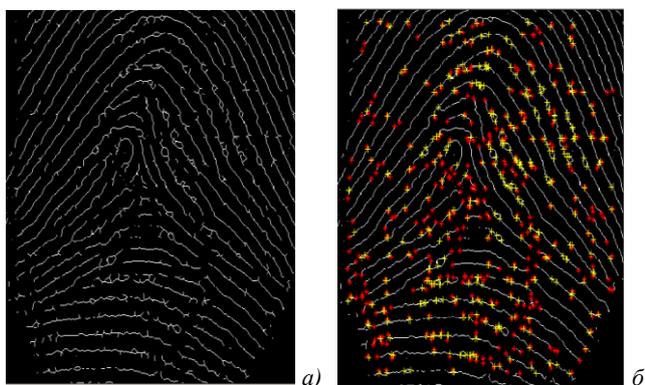


Рис 3. Изображение отпечатка пальца после процедуры скелетизации (а) и изображение отпечатка пальцев с выделенными особыми точками (б).

Таким образом, в ходе работы полученные со сканера отпечатков пальцев изображения прошли этапы предварительной обработки и на заключительном этапе были выделены особые точки, координаты которых можно хранить в базе данных дактилоскопических изображений.

Литература

1. Задорожный В., “Идентификация по отпечаткам пальцев”, Часть 1, 2004.
2. Farah Dhib Tatar, Mohsen Machhout, Preprocessing algorithm for digital fingerprint image, International Journal of Computational Science and Information Technology (IJCSITY) Vol.6, No.1/2/3, August 2018.
3. Roli Bansal , Priti Sehgal, Punam Bedi, Minutiae Extraction from Fingerprint Images, International Journal of Computer Science Issues, Vol. 8, Issue 5, No 3, September 2011.

УДК 004.92

СОЗДАНИЕ ИГРОВОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ С ИСПОЛЬЗОВАНИЕМ ПРОГРАММНОГО СРЕДСТВА UNITY

студентка Тарасевич К.С.,

Научный руководитель - доктор техн. наук Безродный А.А.

Белорусский государственный университет

Минск, Беларусь

Игровая индустрия стремительно вливается в жизнь человека, что в совокупности с информатизацией общества вызывает экспоненциальный рост количества видеоигр. Причины, по которым люди играют в игры, включают в себя снижение стресса, получение удовольствия, способ социализации и коммуникации с окружающими. Отдельным направлением является создание обучающих игровых систем и симуляций, позволяющих проводить отработку определенных навыков и знаний. Например, Zoo Tycoon: Ultimate Animal Collection позволяет не только детям, но и взрослым создавать свой зоопарк и управлять им, а также вместе с этим узнавать новые факты о животных. Algotica

представляет собой игру-пазл в жанре приключения, с помощью которой возможно интересно обучаться программированию. Программное представление видеоигры по своей сути является симуляцией некоего виртуального мира. Шотландские исследователи изучали игровую мотивацию студентов геймеров, в результате чего получили семифакторную модель: контроль, любопытство, фантазия, вызов, кооперация, соревнование, признание [1]. В Будапештском университете пришли к семи шкалам мотивации гейминга, но со следующими составляющими: социальная, копинг (действия, которые совершает человек, чтобы справиться со стрессом), эскапизм (стремление уйти от реальности в выдуманный мир), соревнование, отдых, развитие и фантазия [2]. Целью настоящей работы является создание алгоритма разработки и прототипа 2D-игры в жанре платформер как не теряющую популярность и не требующую особых навыков для неподготовленных пользователей.

Разработка выполнялась с помощью программного обеспечения Unity, языка программирования C#, для написания программного кода использовалась среда разработки Visual Studio 2019.

Последовательность действий на подготовительном этапе может выглядеть следующим образом:

1) Организация командной работы посредством распределенной системы контроля версий Git.

2) Генерация идеи, а также всех главных компонентов. На данном этапе происходит обсуждение, запись и систематизация высказанных предложений.

3) Изображение набросков для персонажей и уровней.

В данном случае для основной цветовой гаммы сцен меню, заключения и первого уровня были выбраны оттенки фиолетового. Идея заключается в следующем: главный герой (кот-путешественник) пытается вернуться домой, но на корабле происходит поломка, и он падает на ближайшую планету. Чтобы починить свой корабль главному герою необходимо собрать несколько существенных деталей, которые находятся у другого персонажа. Для того, чтобы получить детали, герою приходится выполнять поручения.

После завершения подготовительных работ можно приступать к написанию программного кода и сборке компонентов.

Алгоритм сборки всех необходимых компонентов может быть представлен в подобном виде:

Шаг 1. Сбор воедино визуальных составляющих фона и локации: сам фон; несколько разных частей платформ, чтобы в дальнейшем было удобно собирать из них цельные объекты; несколько декоративных объектов.

Шаг 2. Создание фона на сцене непосредственно в игровом движке с помощью компонента Canvas (холста). На него добавляется компонент Image и растягивается по всему экрану. Необходимо, чтобы Canvas оставался на заднем плане, чтобы не закрывать игровое пространство. Для этого устанавливается режим отрисовки на Screen Space – Camera.

Шаг 3. Добавление платформ. На пустой объект GameObject прикрепляется дочерний объект Sprite, который будет содержать рисунок. Для того, чтобы понимать, стоит ли персонаж на платформе, а также для применения 2D физики, необходимо добавить на нее физический компонент твердого тела Rigidbody 2D и, для возможности реагировать на столкновение с другими спрайтами, Box Collider 2D.

Шаг 4. Создание персонажа. На пустой GameObject добавляется дочерний объект Sprite, который будет содержать рисунок. Также добавляется коллайдер и Rigidbody 2D. Данные действия аналогичны для всех игровых персонажей, платформ и почти всех декораций.

Шаг 5. Для реализации геймплея и некоторых свойств объектов прикрепляется компонент Script. В данном компоненте для главного персонажа указываются переменные для скорости, силы прыжка, а также ссылки на такие объекты как Rigidbody 2D, Sprite Renderer, Animator.

Шаг 6. Создание анимации для главного героя. Используются стандартные инструменты для создания анимации.

Шаг 7. Движение камеры за объектом. Реализуется через скрипт, который прикрепляется к камере, с добавлением ссылки на объект игрока, за которым она должна следовать.

Шаг 8. Размещение объектов на сцене (рис. 1а).

Шаг 9. Добавление возможности перемещения платформ происходит через скрипт с указанием смещения по координатным осям.

Шаг 10. Появление персонажа на точке старта после падения. Для этого необходимо создать пустой объект с координатами стартовой точки. Чтобы фиксировать падение надо использовать коллайдер пустого объекта, растянутый на все игровое пространство (рис. 1б).

Шаг 11. Добавление первой и последней сцен, кнопок Retry и Exit.

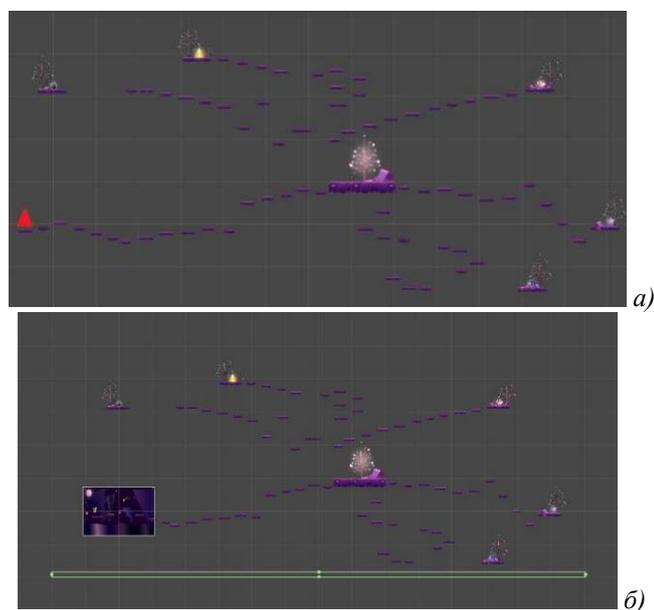


Рис. 1. Игровая сцена первого уровня, где красным треугольником показана стартовая точка, (а) и игровая сцена, где зеленым цветом показаны границы коллайдера (б)

В результате работы прототип созданной игры опубликован на своей странице сайта itch.io. Получено несколько отзывов от независимых пользователей, которые помогут в улучшении и устранении недочетов игры. На этом заканчивается этап разработки основных элементов и механик. Таким образом, игровой движок Unity позволяет без особых проблем создавать игры, удобен как в одиночной разработке, так и в командной. На базе предложенного алгоритма построения игрового взаимодействия возможна разработка образовательных и узкоспециальных обучающих приложений.

Литература

1. The differences in motivations of online game players and offline game players: A combined analysis of three studies at higher education level / T. Hainey, T. Connolly, M. Stansfield, E. Boyle // Computers and Education. – 2011. – Vol. 57. – Issue 4. – P. 2197-2211.

2. Why do you play? / Z. Demetrovics, R. Urban, K. Nagygyorgy, J. Farkas, D. Zilahy, B. Mervo, A. Reindl, C. Agoston, A. Kertesz, E. Harmath // The development of the motives for

online gaming questionnaire (MOGQ). Behavior Research Methods. – 2011. – Vol. 43. – Issue 3. – P. 814-825.

3. Unity и C#. Геймдев от идеи до реализации. 2-е изд. — СПб.: Питер, 2019. — 928 с.: ил. — (Серия «Для профессионалов»).

УДК 004.92

ТРЕХМЕРНАЯ РЕКОНСТРУКЦИЯ ЛИЦА ПО ФОТОГРАФИЯМ

магистрант гр. 10 Курочкин М. Н.

Научный руководитель – канд. техн. наук Головатая Е. А.

Белорусский Государственный Университет

Минск, Беларусь

Трехмерная реконструкция лица – сложная проблема, которую пытаются решить уже не одно десятилетие. Процесс трехмерной реконструкции заключается в восстановлении так называемой сцены, которая представляет собой множество объектов, отображаемых системой визуализации. Опирируя двумерными проекциями, такие системы способны извлекать сведения о форме и текстуре объектов. Суть процесса трехмерной реконструкции как раз и заключается в том, чтобы, во-первых, осуществить сбор необходимой информации из плоского изображения объекта, а во-вторых, преобразовать полученную информацию в требуемый вид. Фотографирование и видеосъемка являются одними из наиболее распространенных способов сбора сведений. В связи с увеличением спроса на сервисы, обеспечивающие быстрый и легкий способ получения трехмерной модели, в сфере маркетинга, разработки игр, медицине, безопасности и многих других в ближайшее десятилетие ожидается стремительное развитие в области трехмерной реконструкции и распознавания. Однако доступность такого рода сервисов остается весьма низкой из-за их дороговизны и сложности использования без предварительного обучения.

Альтернативой стереовидению и аналогичным методам, использующим камеры, являются методы трехмерной реконструкции, основанные на использовании только одного 2D-изображения для восстановления поверхности лица. Также возможно использование некоторого набора фотографий одного человека, но к ним не применяются различные требования, например, степень освещенности или направленность света, принимаемые позы и так далее. Основой алгоритмов, осуществляющих реконструкцию лица по одному изображению, является разработанная заранее параметрическая модель [2].

Существует несколько подходов осуществления данной идеи. Так, например, в некоторых алгоритмах используется открытый пакет библиотек eos, который был разработан для подгонки 3D-моделей с использованием параметрической модели лица. Алгоритмы трехмерной реконструкции набора программных компонент eos использует в роли входных данных одно изображение лица и набор ключевых точек, которые можно получить при помощи библиотека dlib. Она была разработана специально для детекции ключевых точек.

Наибольшее распространение получил метод, называемый регрессионной карточной сетью (PRN), для совместного прогнозирования плотного выравнивания и восстановления трехмерной формы лица. Данный метод превосходит предыдущие работы по 3D-выравниванию лица и реконструкции на нескольких наборах данных. Между тем, метод имеет несложную структуру, которая обеспечивает результат за один проход в 9,8 мс. Все это достигается благодаря тщательно продуманному дизайну двумерного представления трехмерной структуры лица и соответствующей функции ошибки. В частности, происходит проектирование UV-карты положения, которая представляет собой 2D-изображение, записывающее 3D-координаты полного лицевого облака точек, и в то же время сохраняет смысловую нагрузку каждой точки UV-карты. Затем обучается простая