Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Саратовский государственный технический университет

имени Гагарина Ю.А.»

Институт прикладных информационных технологий и коммуникаций

Кафедра «Прикладные информационные технологии»

**ОТЧЕТ**

**о прохождении учебной (НИР (получение первичных навыков в НИР)) практики**

|  |  |
| --- | --- |
| Место прохождения  практики | Кафедра «Прикладные информационные технологии» |
| Время прохождения практики | С 09.07.2022 по 22.07.2022 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | ФИО | Подпись | Дата |
| Выполнил студент группы б1-ИФСТ-14 | Песчанов Арсений Андреевич |  |  |
| Руководитель от кафедры | Торопова Ольга Анатольевна |  |  |

|  |  |
| --- | --- |
| Итоговая оценка по защите результатов деятельности на практике |  |

Саратов 2022

**Федеральное государственное бюджетное образовательное**

**учреждение высшего образования**

**«Саратовский государственный технический университет имени Гагарина Ю.А.»**

**Институт прикладных информационных технологий и коммуникаций**

Кафедра «Прикладные информационные технологии»

**ЗАДАНИЕ**

**НА УЧЕБНУЮ (НИР (ПОЛУЧЕНИЕ ПЕРВИЧНЫХ НАВЫКОВ В НИР)) ПРАКТИКУ**

Студенту учебной группы \_б1-ИФСТ-14\_ института прикладных информационных технологий и коммуникаций

()Песчанову Арсению Андреевичу

(фамилия, имя, отчество)

Практика проходит в организации СГТУ им. Гагарина Ю.А.

(наименование организации)

расположенной по адресу г. Саратов, ул. Политехническая, д. 77

(фактический адрес)

Срок практики с 09.07.2022 г. по 22.07.2022 г.

Основание: Приказ СГТУ имени Гагарина Ю.А. № 1401-С\_\_от 17.06.2022 г.

**Индивидуальное задание**

В рамках практики необходимо освоить навыки в соответствии со следующими компетенциями: ПК-3:

|  |  |
| --- | --- |
| Код и наименование компетенции (результат освоения) | Код и наименование индикатора достижения компетенции (составляющей компетенции) |
| **ПК-3** Способность разрабатывать программное обеспечение (ПО), включая проектирование, отладку, проверку работоспособности и модификацию ПО | **ИД 1.1 пк3** иметь навыки: анализа возможностей реализации требований к программному обеспечению; оценки времени и трудоемкости реализации требований к программному обеспечению; согласования требований к программному обеспечению с заинтересованными сторонами; оценки и согласования сроков выполнения поставленных задач  **ИД 2.1 пк3** уметь: проводить анализ исполнения требований; вырабатывать варианты реализации требований; проводить оценку и обоснование рекомендуемых решений; осуществлять коммуникации с заинтересованными сторонами  **ИД 3.1 пк3** знать: возможности существующей программно-технической архитектуры; возможности современных и перспективных средств разработки программных продуктов, технических средств; методологии разработки программного обеспечения и технологии программирования; методологии и технологии проектирования и использования баз данных  **ИД 2.2 пк3** уметь: выбирать средства реализации требований к программному обеспечению; вырабатывать варианты реализации программного обеспечения; проводить оценку и обоснование рекомендуемых решений; осуществлять коммуникации с заинтересованными сторонами  **ИД 3.2 пк3** знать: языки формализации функциональных спецификаций; методы и приемы формализации задач; методы и средства проектирования программного обеспечения; методы и средства проектирования программных интерфейсов; методы и средства проектирования баз данных  **ИД 2.3 пк3** уметь: использовать существующие типовые решения и шаблоны проектирования программного обеспечения; применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов; осуществлять коммуникации с заинтересованными сторонами |

В рамках освоения компетенций необходимо закрепить и углубить теоретические знания, полученные студентами при изучении дисциплин 1 и 2 семестров учебного плана, получить первичные профессиональные умения и навыки в области разработки программных продуктов: веб - приложений и компьютерной графики.

Задачи практики:

- закрепить и углубить теоретические знания, полученные студентами по изучаемым дисциплинам;

* развить навыки использования современных средств вычислительной техники в решении инженерных задач;

- привить навыки самостоятельной информационно-поисковой работы.

* закрепить теоретические и практические знаний, полученные при изучении общепрофессиональных и специальных дисциплин;
* изучить технологические и программные средства автоматизации и управления;
* развить навыки выполнения самостоятельной исследовательской работы.
* использовать современные компьютерные технологии для разработки веб-приложений;

- изучить способ создания игровых приложений с помощью игрового движка Unity и языка C#.

В результате прохождения учебной (НИР (получение первичных навыков в НИР)) практики следует приобрести практические навыки использования современных информационных технологий для разработки веб-приложений.

**Руководитель практики от кафедры**  / Торопова О.А. /

(подпись) (Ф.И.О.)

**План-график проведения практики**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | Наименование вопросов подлежащими изучению в период практики или вид выполняемой работы | Количество дней/  акад. часов | Форма отчетности | Отметка  руководителя о выполнении |
| 1 | Инструктаж по технике безопасности и охране труда. Постановка задачи на производственную практику, обсуждение. Вводная лекция. | 1 | Запись в журнале инструктажей по технике безопасности |  |
| 2 | Изучение основ ООП и синтаксиса в языке C# | 5 | Письменный отчет |  |
| 3 | Ознакомление с интерфейсом Unity | 2 | Письменный отчет |  |
| 4 | Работа с готовыми скриптами на платформе Unity | 4 | Письменный отчет |  |
| 5 | Изучение графического редактора Adobe Photoshop | 1 | Письменный отчет |  |
| 6 | Оформление и сдача отчета, подготовка необходимой документации | 1 | Письменный отчет |  |

ИТОГО: 14

**Согласовано:**

**Руководитель практики от кафедры** 09.07.2022 /Торопова О.А. /

(дата согласования) (подпись) (Ф.И.О.)

**Ознакомлен:**

**Студент**\_\_\_\_ Песчанов А. А.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ «09» июля 2022г.

(фамилия, инициалы) (подпись)

**Отзыв руководителя практики от кафедры**

Подпись / / Дата 08.07.2022\_

Содержание

[Введение 9](#_Toc109330742)

[1. Описание предметной области и постановка задачи 10](#_Toc109330743)

[2. Разработка игры 10](#_Toc109330744)

[2.1. Создание файла и персонализация 10](#_Toc109330745)

[2.2. Добавление персонажей в игру 11](#_Toc109330746)

[2.3. Скрипт передвижения персонажей 14](#_Toc109330747)

[2.4. Управление героем 15](#_Toc109330748)

[2.5. Анимация передвижения и атаки персонажей 17](#_Toc109330749)

[2.6. Вывод 21](#_Toc109330750)

[Заключение 22](#_Toc109330751)

[Список использованных источников 23](#_Toc109330752)

# Введение

С появлением и развитием информационных технологий, привычные для нас вещи, такие как игры, стали переходить в цифровой формат. Сейчас компьютерные игры стали неотъемлемой частью жизни современного человека. Они не теряют своей популярности, а наоборот, только набирают ее, благодаря непрерывному развитию технологий, позволяющих создавать все более качественные игры, способности удовлетворить любые запросы потребителя, и возможности провести время с друзьями, находящимися далеко друг от друга.

Обычно компьютерные игры делятся на несколько типов: квесты, ролевые игры (рпг), стратегии, симуляторы, логические, экшн. Они могут быть как на одного игрока, так и на двух и более, появилась возможность играть по сети. Данное многообразие игр существует благодаря наличию множества сред их разработки, поскольку одна среда не может подходить для всех типов игр.

Нашу игру мы будем делать на кроссплатформенном игровом движке Unity, с использованием языка C#.

Целью учебной (ознакомительной) практики является разработка и создание игры с использованием движка Unity и полученным в ходе первой учебной (ознакомительной) практики материалом.

## 

## Описание предметной области и постановка задачи

Игра будет выполнена в жанре RTS (real-time strategy), особенностью которого является полное отсутствие «очерёдности ходов», присущих классическим пошаговым стратегиям. Игра предназначена для одного игрока, который будет размещать свои объекты (армия и сооружения), на поле, с целью защитить свои участки карты и атаковать/уничтожить противника. Кроме того, под управлением игрока находится герой, управляемы с помощью клавиатуры. Здесь будет рассмотрена работа с персонажами игры.

Чтобы достичь нашей цели, необходимо выполнить следующие задачи:

* Добавить персонажей в проект;
* Реализовать анимацию ходьбы и атаки персонажей;
* Написать скрипты передвижения персонажей;
* Написать скрипт управления героем с клавиатуры.

В проекте использовались: Unity, C#.

## Разработка игры

Сюжет игры «Knights of Legends»: главной целью игры является защита своего замка от вражеской армии. Для этого герой собирает армию бойцов, готовых защищать свой дом и отважно ринуться в битву. Вам предстоит сражаться с множеством разнообразных противников, расставляя своих бойцов на поле и продумывая стратегию битвы. Готовы ли вы взять на себя роль командира светлой армии?

### 2.1. Создание файла и персонализация

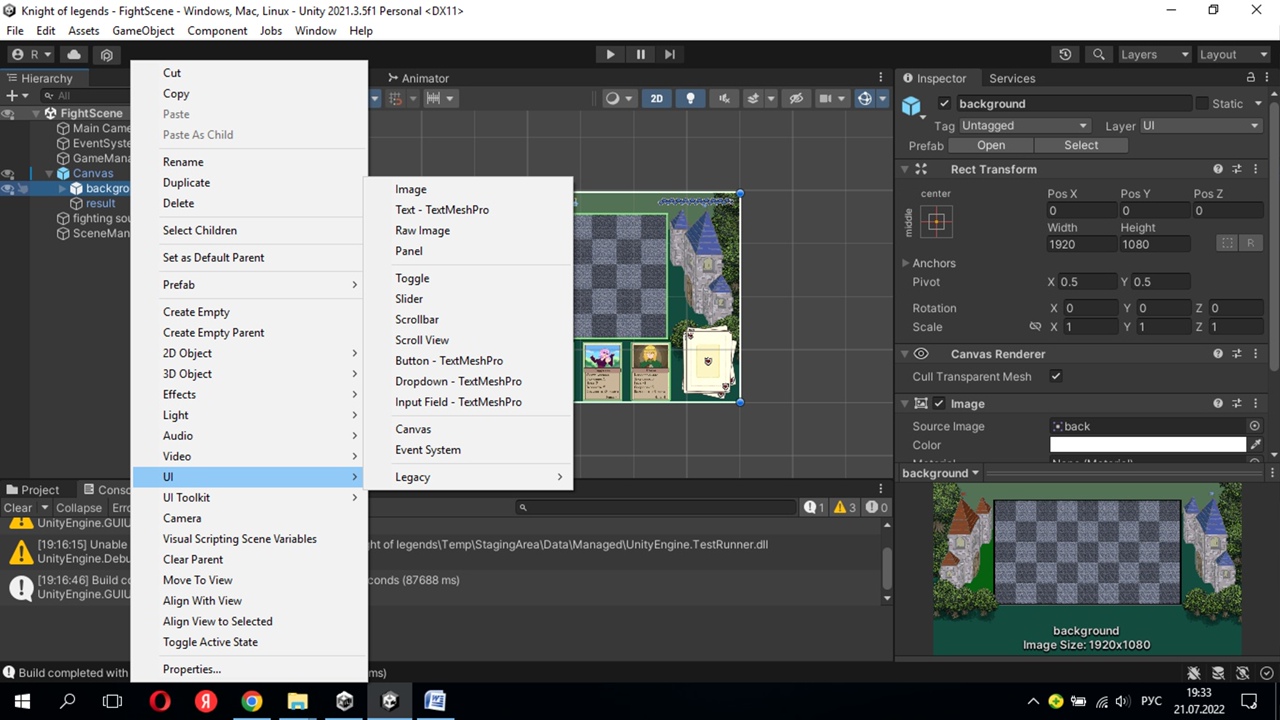
Для начала нужно создать файл для игры и называем его «Knights of Legends». После создания файла открываем его и настраиваем рабочую область. В папке с проектом нужно создать папку Animations – тут будут все спрайты анимаций, которые используются в проекте. Кроме того создаем папку Textures в которой будут храниться все спрайты, фоны и дополнительные изображения.

После подготовки всех файлов, можно приступать к созданию игры.

### 2.2. Добавление персонажей в игру

Чтобы игра не была пустой, необходимо добавить туда персонажей, с которыми мы и будем играть. Для этого надо:

1. В Canvas выбрать раздел background→пкм→UI→Image (рис.1);
2. Перемещаем спрайт в нужное место поля (рис.2);
3. В окне инспектора выбираем Add Component→Box Colider 2D (рис.3);
4. Настраиваем размер и ставим галочку напротив строки Is Trigger (рис.4). Это нужно для вызова скрипта при прикосновении к персонажу;
5. Добавляем Rigidbody 2D и в Body Type выбираем Kinematic (рис.5). Это необходимо для придания объекту физики;
6. Добавляем скрипт, отвечающий за движение персонажа;
7. Переносим персонажа в папку Prefabs, чтобы в дальнейшем настраивать его взаимодействие с другими персонажами.



Рисунк 1 – Добавление изображения

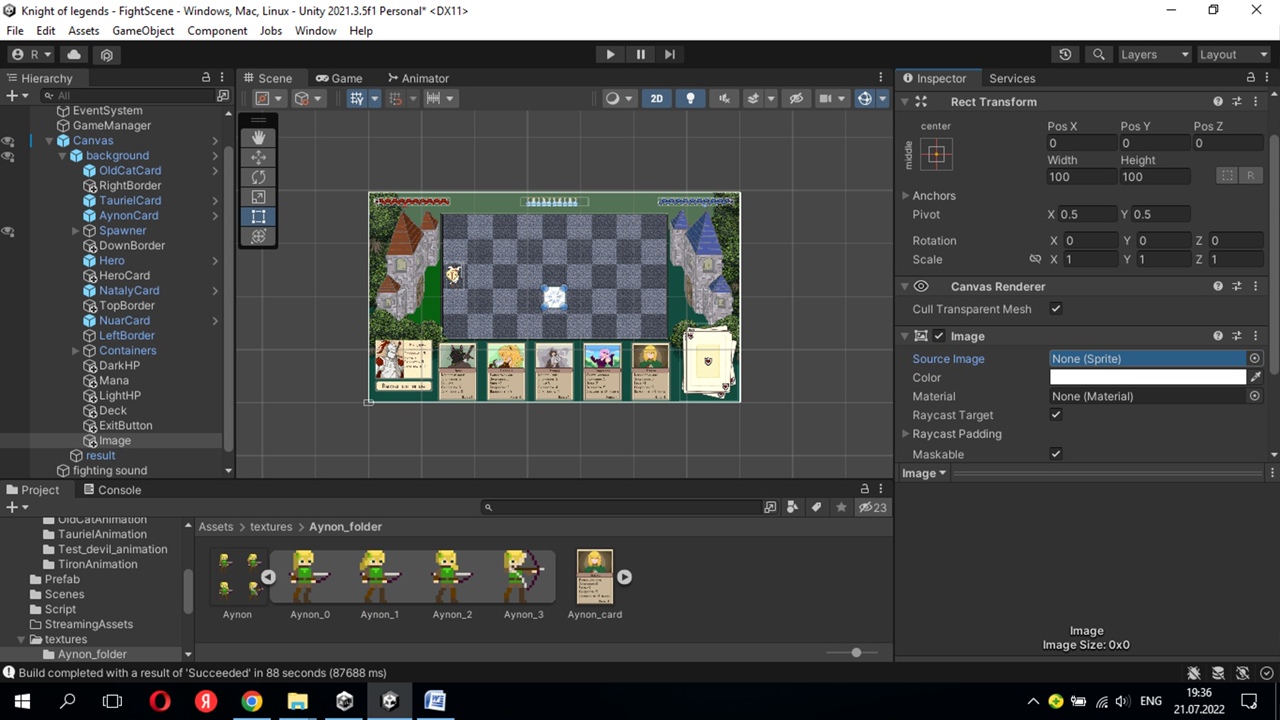


Рисунок 2 – Размещение спрайта на поле

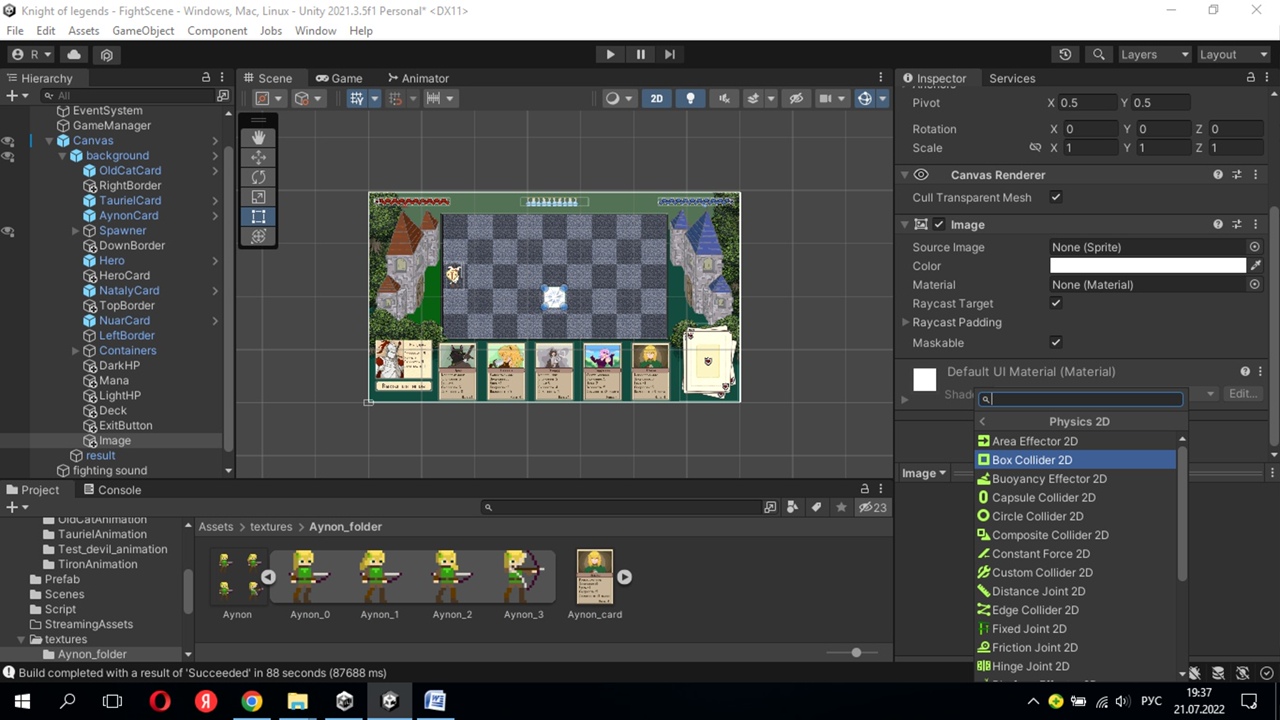


Рисунок 3 – Добавление Box Colider 2D

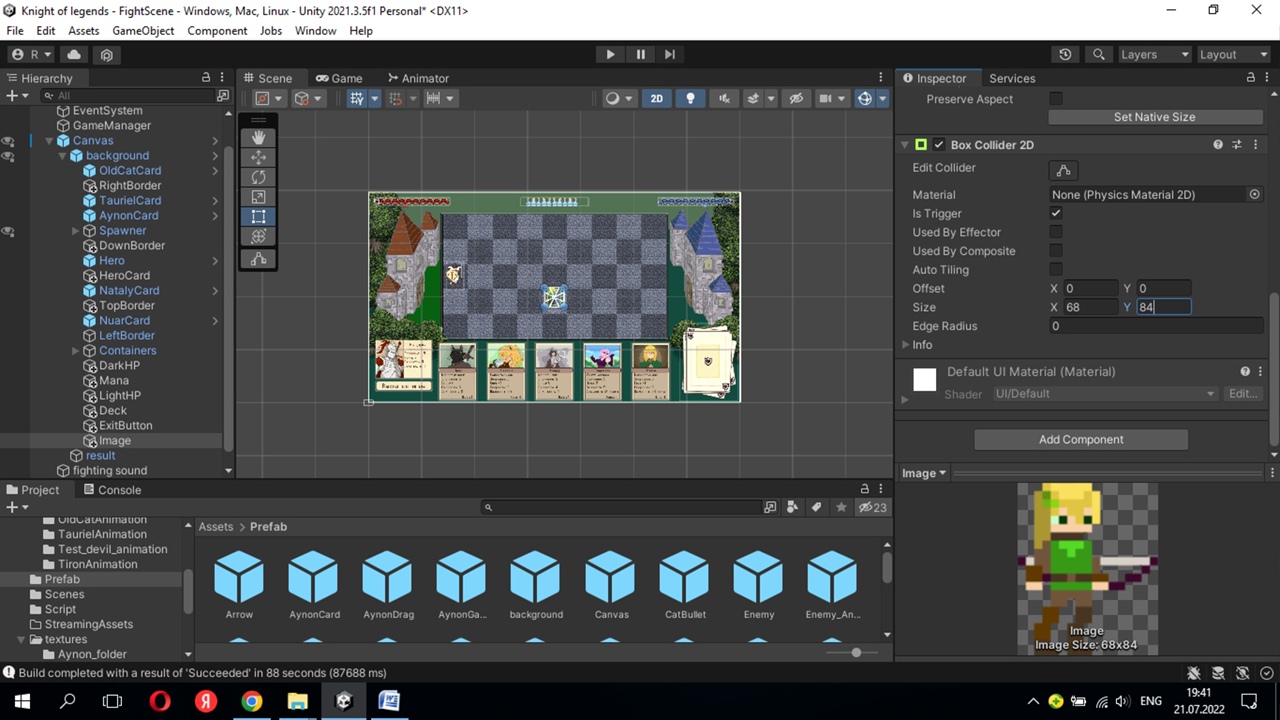


Рисунок 4 – Настройка размера; Trigger

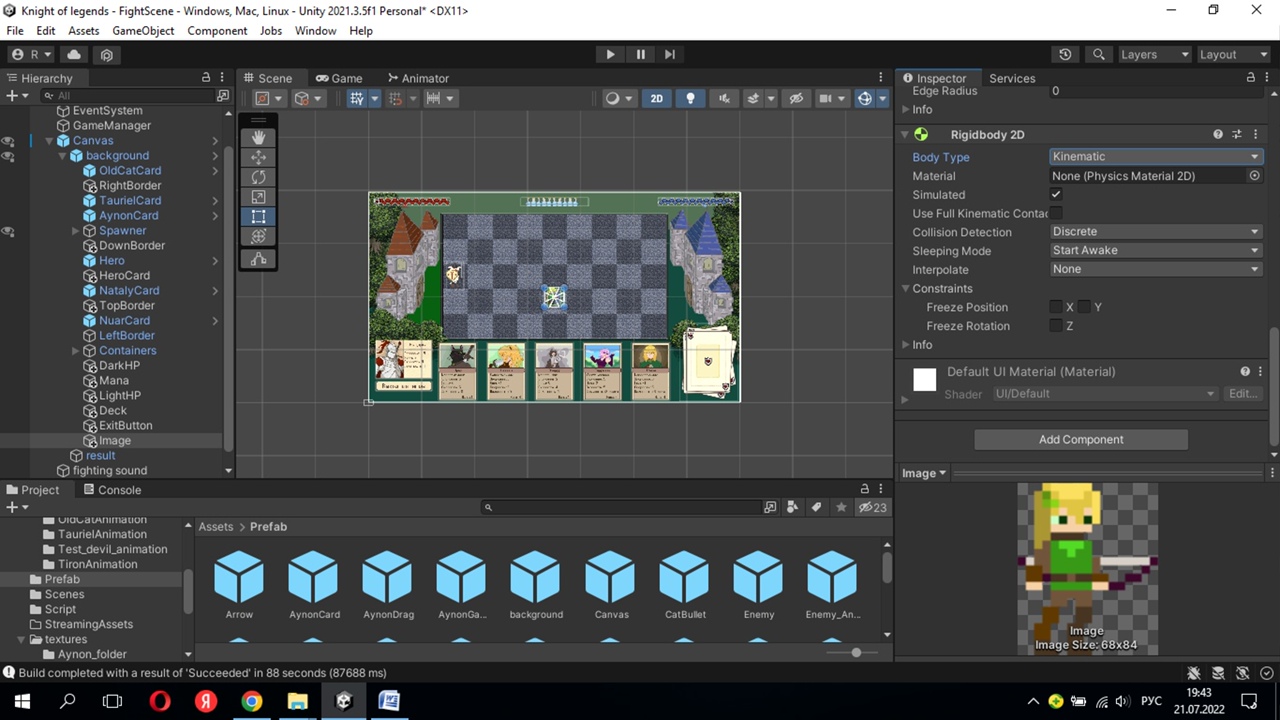


Рисунок 5 – Придание физики персонажу

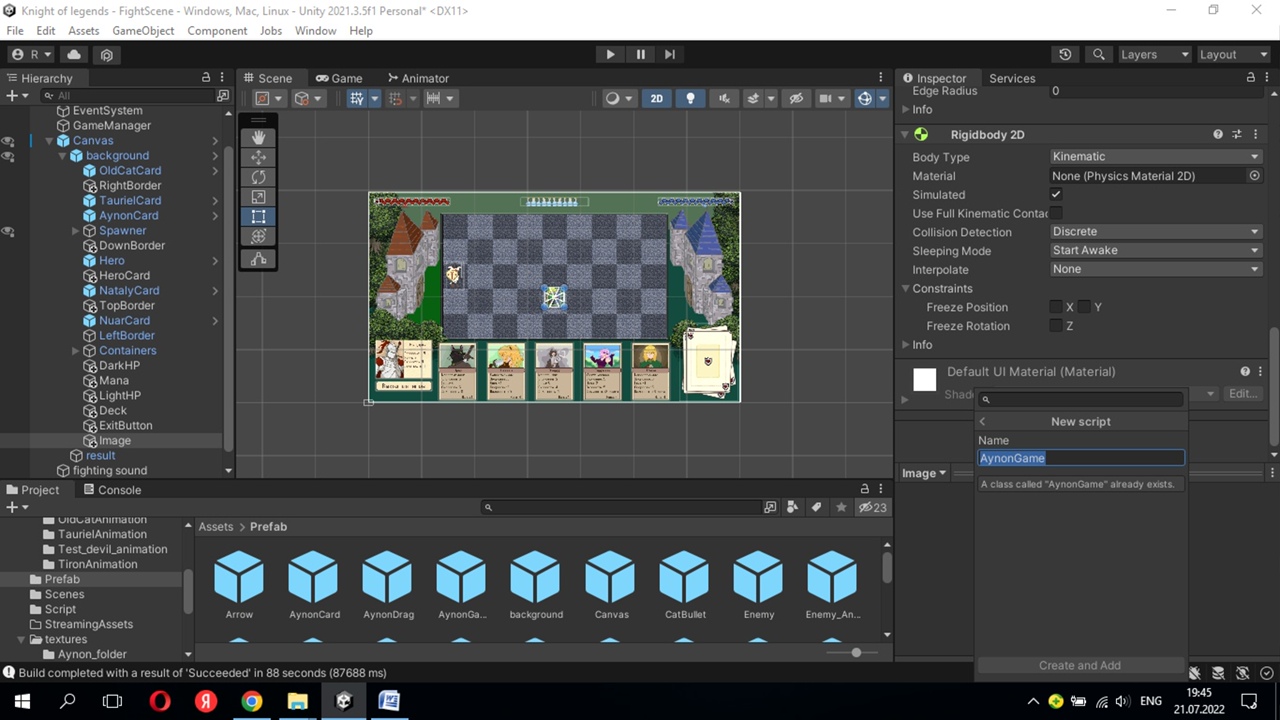


Рисунок 6 – Добавление скрипта

### 2.3. Скрипт передвижения персонажей

Персонажи (игрока и вражеские), находясь в определенной ячейке поля могут двигаться только вперед, при этом останавливаясь перед вражеским бойцом для нанесения тому урона. Для того чтобы добавленные в игру персонажи могли передвигаться, нужно прописать им скрипт:

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class AynonGame : MonoBehaviour

{

public int Health;

public int Damage;

public float movementSpeed;

public bool isStopped;

public Animator anim;

public bool moving = true;

// Движение персонажа по полю

void Update()

{

if(!isStopped)

{

transform.Translate(new Vector2(movementSpeed, 0));

}

anim.SetBool("Moving", moving);

}

// Взаимодействие персонажа с другим персонажем или стеной

public void OnTriggerEnter2D(Collider2D collision)

{

if(collision.gameObject.layer == 9)

{

isStopped = true;

moving = false;

}

if(collision.gameObject.layer == 12)

{

Destroy(gameObject);

}

}

// Убийство моба и выход из триггера (взаимодействия с персонажем)

public void OnTriggerExit2D(Collider2D collision)

{

isStopped = false;

moving = true;

}

}

### 2.4. Управление героем

Рыцарь, находящийся под прямым управлением игрока, должен двигаться в любые стороны, переходя по всем ячейкам, а не идя в направлении только одной линии. Для этого необходимо его контролировать с помощью клавиш на клавиатуре, добавив следующий скрипт:

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class HeroMovement : MonoBehaviour

{

public Animator anim;

public int Health;

public int Damage;

public float DamageCooldown;

public float moveSpeed;

private Rigidbody2D rb;

private float x;

private float y;

private Vector2 input;

private bool moving;

public bool isAttacking = false;

private void Start()

{

rb = GetComponent<Rigidbody2D>();

}

private void Update()

{

GetInput();

Animate();

}

private void FixedUpdate()

{

rb.velocity = input \* moveSpeed;

}

private void GetInput()

{

x = Input.GetAxisRaw("Horizontal");

y = Input.GetAxisRaw("Vertical");

input = new Vector2(x, y);

input.Normalize();

}

private void Animate()

{

if(input.magnitude > 0.1f || input.magnitude < -0.1f)

{

Moving = true;

}

else

{

moving = false;

}

if(moving)

{

anim.SetFloat("X", x);

anim.SetFloat("Y", y);

}

anim.SetBool("Moving", moving);

}

Кроме того герой должен уметь атаковать соперника, для этого надо написать следующий скрипт:

public void OnTriggerEnter2D(Collider2D collision)

{

if(collision.gameObject.layer == 9)

{

isAttacking = true;

anim.SetBool("isAttacking", isAttacking);

moveSpeed = 0;

moving = false;

StartCoroutine(Attack(collision));

}

}

// Убийство моба и выход из триггера (взаимодействия с персонажем)

public void OnTriggerExit2D(Collider2D collision)

{

isAttacking = false;

anim.SetBool("isAttacking", isAttacking);

moveSpeed = 15f;

}

### 2.5. Анимация передвижения и атаки персонажей

Для создания анимации персонажей, необходимы изображения покадровой ходьбы и атаки персонажей, которые уже добавлены в проект. Далее необходимо:

1. Нажимаем на персонажа и в окне Window выбираем Animation (рис.7);

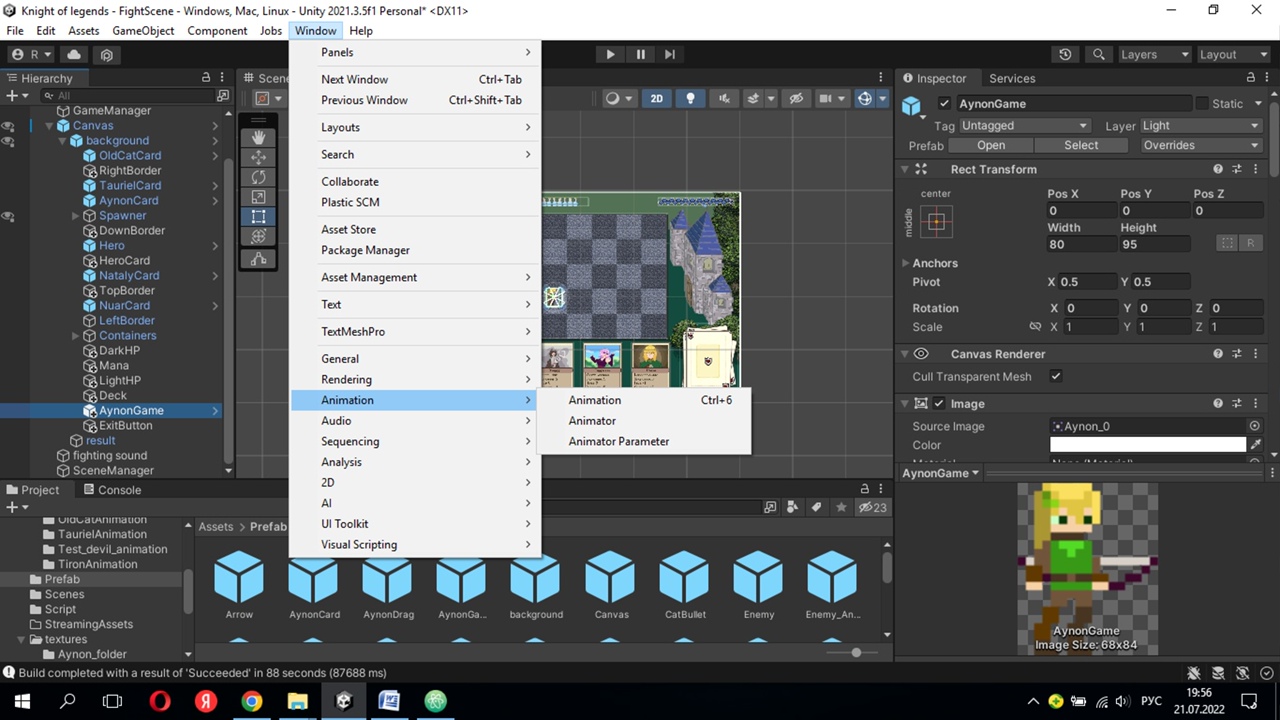


Рисунок 7 - Animation

1. Нам предложат создать аниматор и новый клип, на что мы соглашаемся (рис.8);

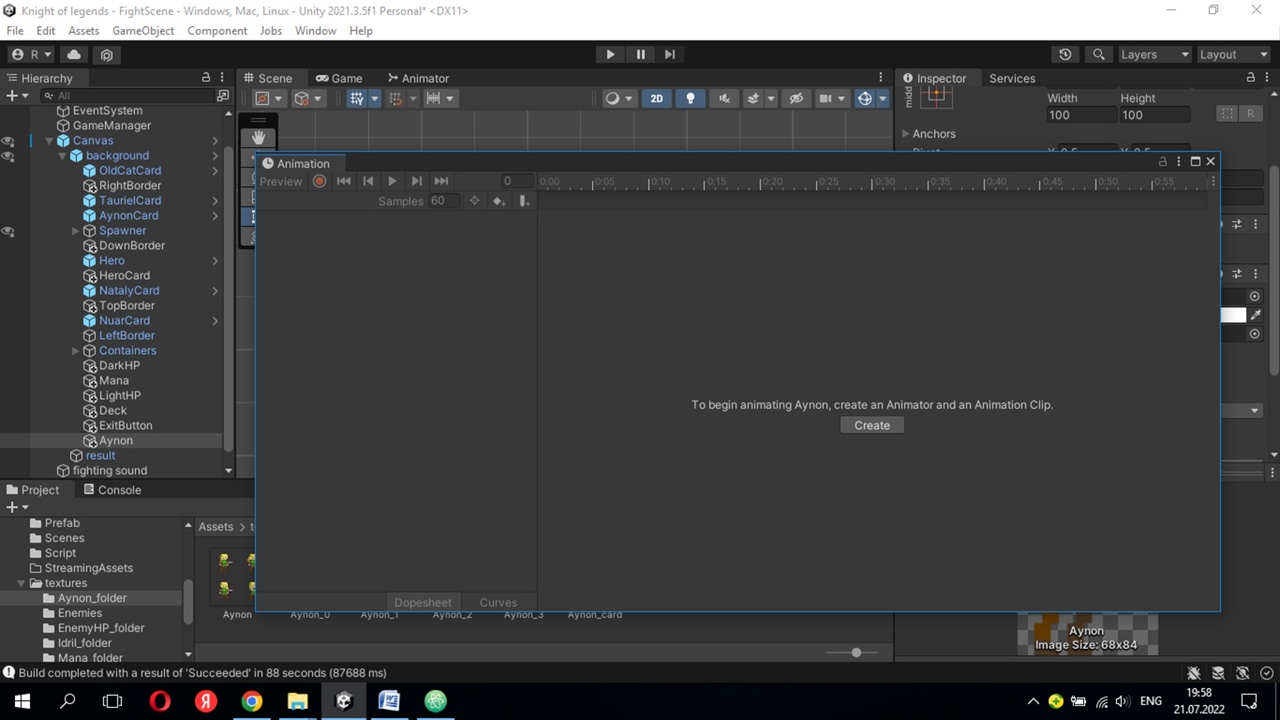


Рисунок 8 – Создание нового клипа

1. Создаем отдельную папку для анимаций, внутри которой будут папки для анимаций отдельных персонажей (рис.9);

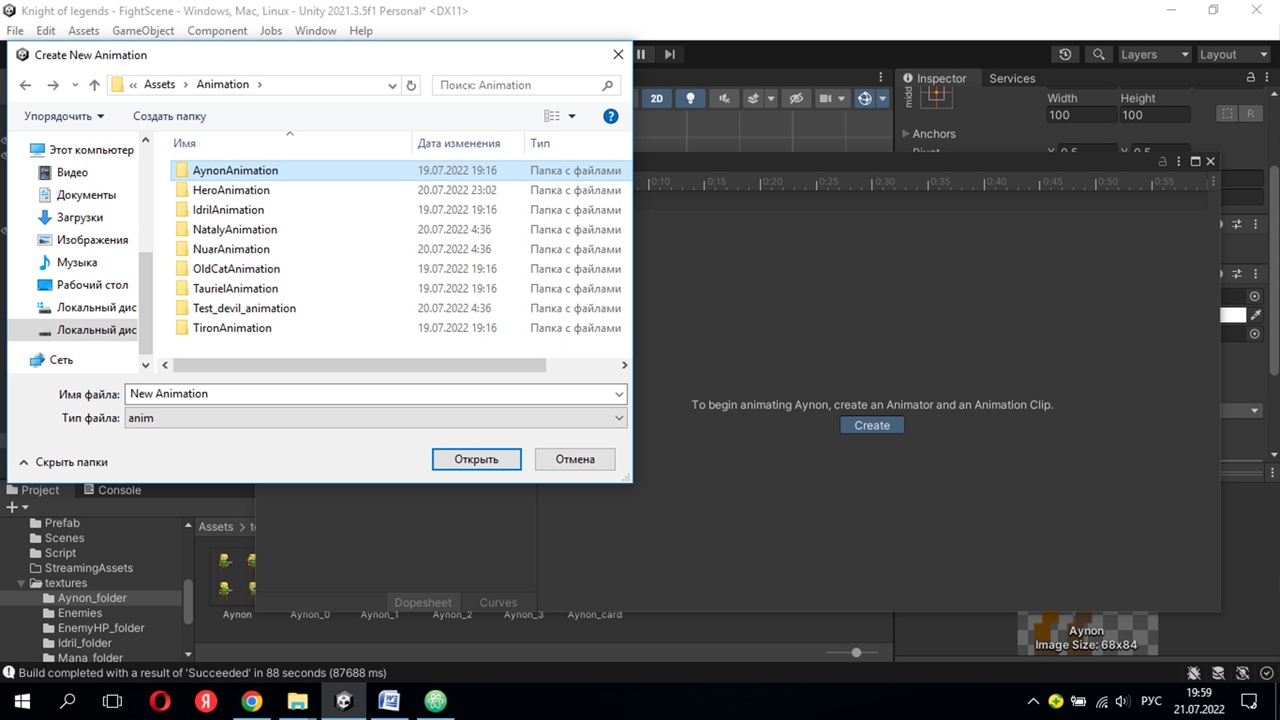


Рисунок 9 – Папка для анимаций

1. Возвращаемся к клипу и размещаем спрайты на временной линии с определенным порядком и необходимым интервалом (рис.10);

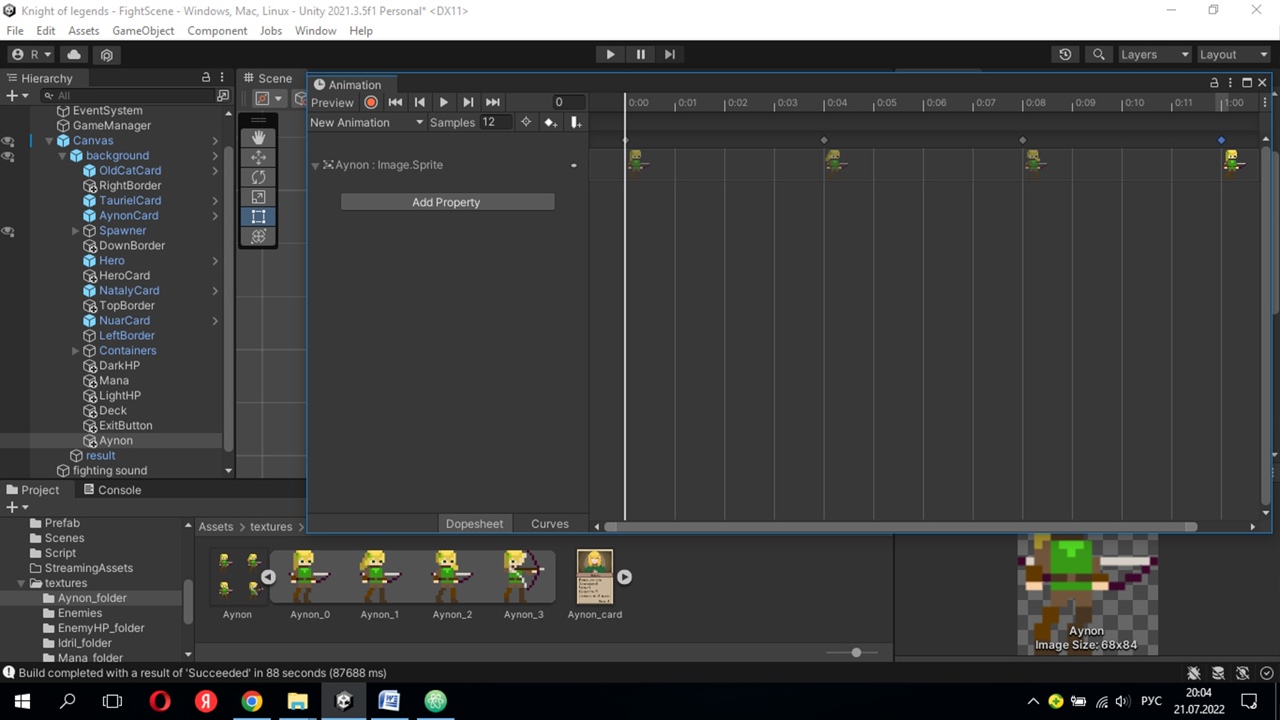


Рисунок 10 – Размещение спрайтов

1. Переходим в аниматор, где можно настроить переход от одной анимации к другой (рис.11);

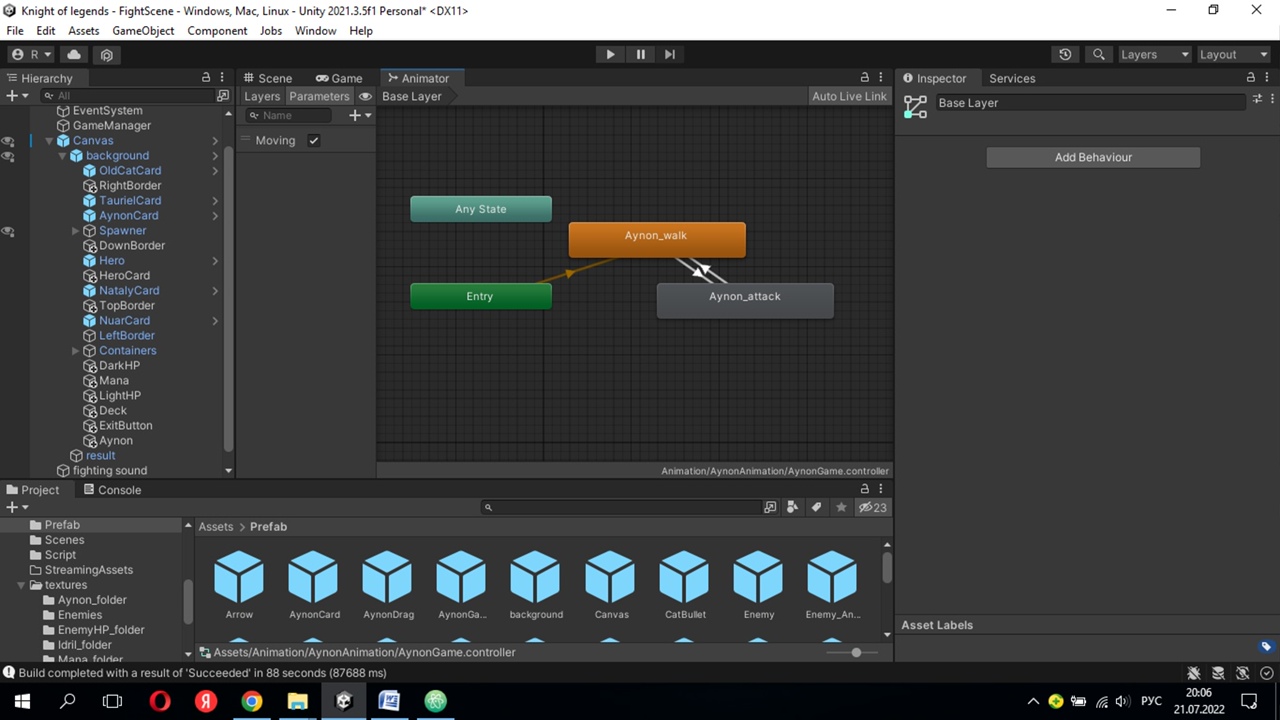


Рисунок 11 – Переход к аниматору

1. Желтая стрелочка указывает на то, какая анимация исполняется по дефолту. Кроме того можно создать переменную для условий переходов анимаций (рис.12);

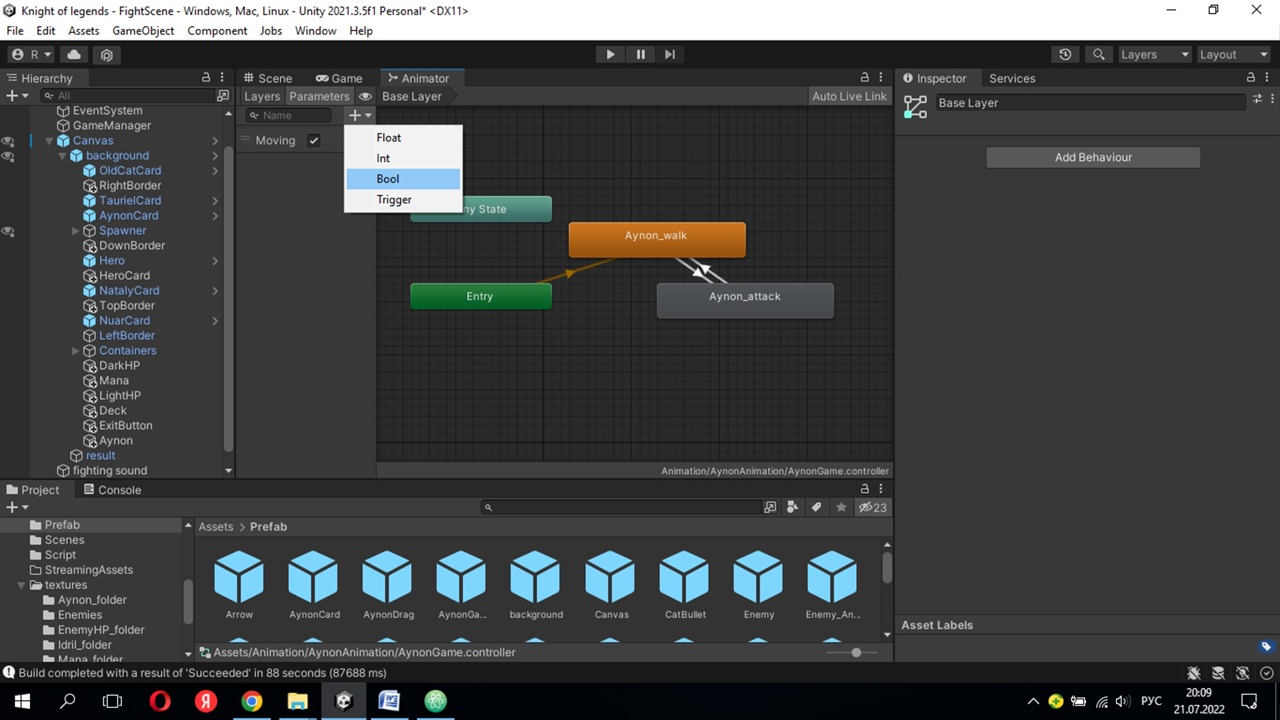


Рисунок 12 – Переменная для переходов анимаций

1. Создаем Transition, являющийся нашим переходом (рис.13);

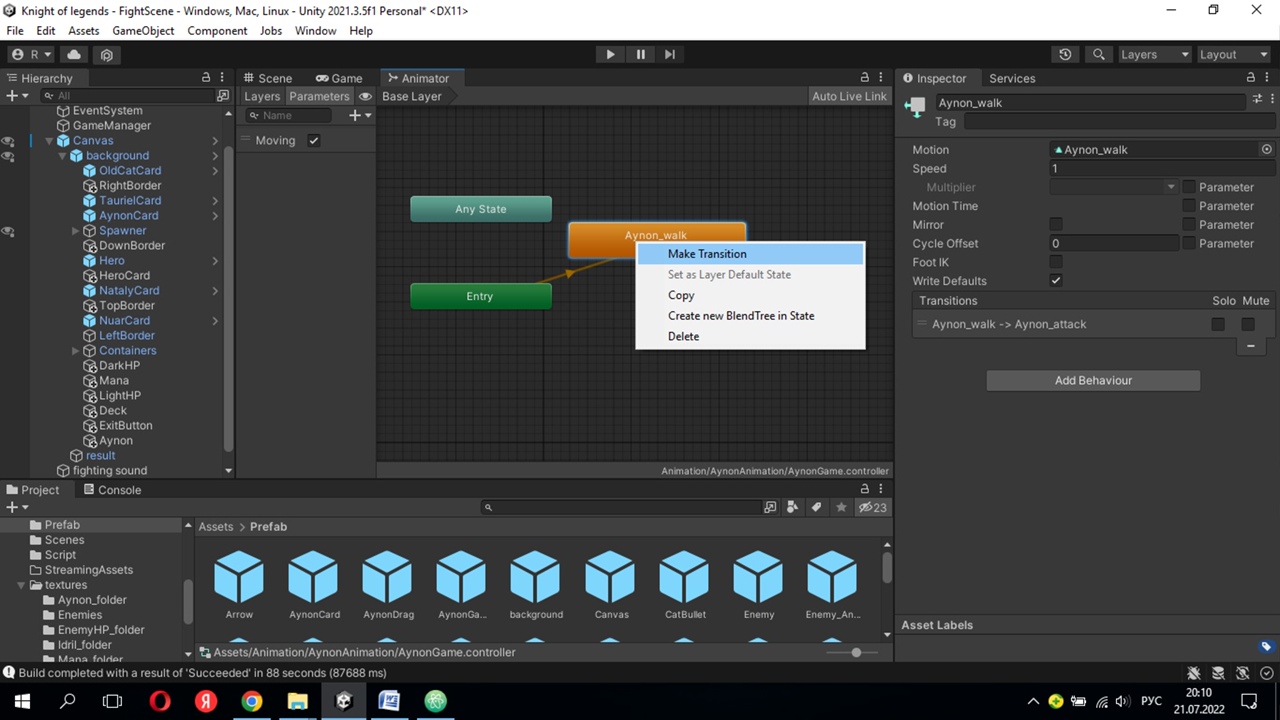


Рисунок 13 – Made Transition

1. Используя переменную указываем, в каком случае происходит переход между анимациями. В нашем случае это Moving == false (рис.14)

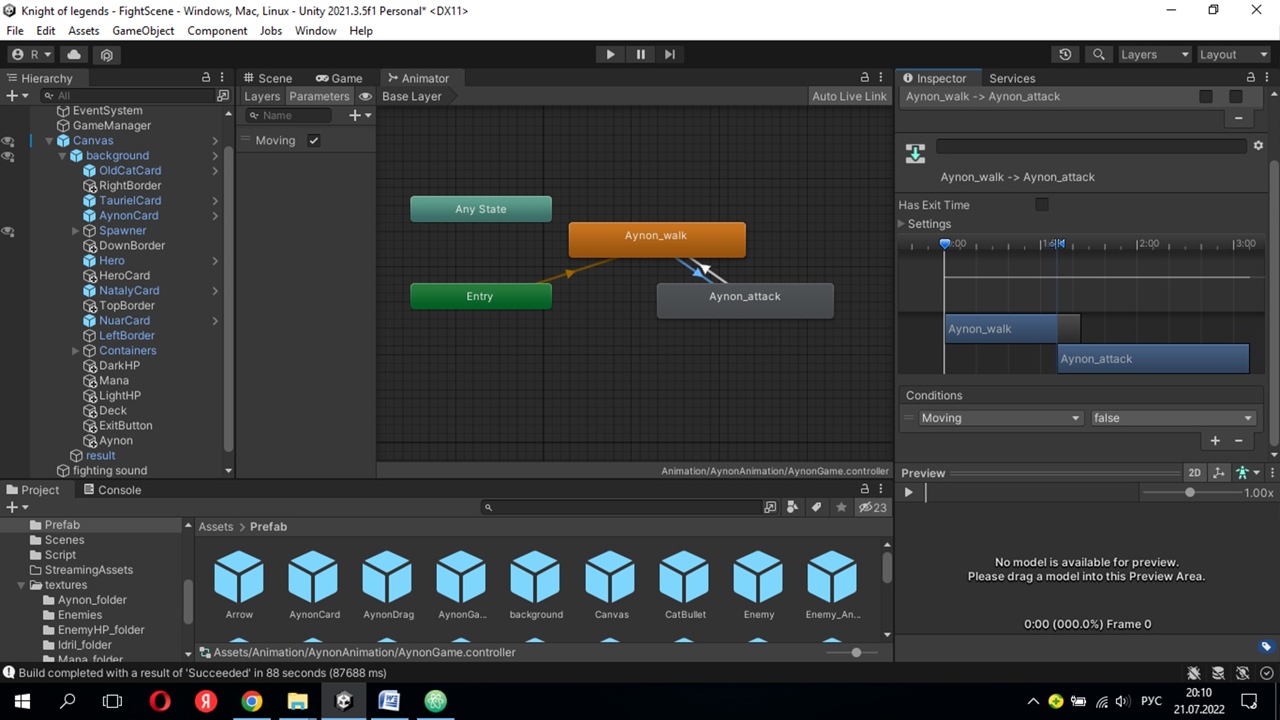


Рисунок 14 – Условие перехода анимаций

### 2.6. Вывод

Таким образом в игру были добавлены игровые персонажи. А так же написаны скрипты для их передвижения. Дополнительно реализована анимация ходьбы и атаки персонажей. Безусловно, можно было добавить также управляемого героя врага (наподобие рыцаря), добавить ему передвижение и управление, или же сделать игру на двух игроков. Но на основе полученных знаний можно создавать и другие проекты в будущем.

# Заключение

В конечном итоге за время учебной (ознакомительной) практики была создана игра с помощью движка Unity, с применением базовых знаний языка C#. Графический дизайн был реализован в редакторе Adobe Photoshope. Цели и задачи практики были выполнены.

# Список использованных источников

1. Первые шаги в Unity [Электронный ресурс] // habr.com. 2021. URL: <https://habr.com/ru/company/otus/blog/576434/> (Дата обращения: 24.06.2022);
2. Unity Essentials [Электронный ресурс] // learn.unity.com. 2020. URL: <https://learn.unity.com/pathway/unity-essentials?language=en> (Дата обращения: 25.06.2022);
3. Уроки C# Unity [Электронный ресурс] // itproger.com. 2022. URL: <https://itproger.com/course/unity-csharp> (Дата обращения: 26.06.2022);
4. Полное руководство по языку программирования С# 10 и платформе .NET 6 [Электронный ресурс] // metanit.com. 2022. URL: <https://metanit.com/sharp/tutorial/> (Дата обращения: 26.06.2022);
5. Объектно-ориентированное программирование. Часть 1. Что такое классы и объекты [Электронный ресурс] // skillbox.ru. 2019. URL: <https://skillbox.ru/media/code/oop_chast_1_chto_takoe_klassy_i_obekty/> (Дата обращения: 28.06.2022);
6. Основные сведения о рабочей среде [Электронный ресурс] // helpx.adobe.com. 2022. URL: <https://helpx.adobe.com/ru/photoshop/using/workspace-basics.html> (Дата обращения: 04.07.2022);
7. Как рисовать пиксель арт в фотошопе — часть 1 [Электронный ресурс] // internet-technologies.ru. 2022. URL: https://www.internet-technologies.ru//articles/sozdaem-8-bitnuyu-pikselnuyu-grafiku-s-pomoschyu-photoshop-chast-1.html (Дата обращения: 04.07.2022);
8. 4 преимущества Unity3D, а также полезный инструмент для ускорения процесса разработки приложений [Электронный ресурс] // habr.com. 2015. URL: https://habr.com/ru/company/ximad/blog/252525/ (Дата обращения: 05.07.2022);
9. Learn C# [Электронный ресурс] // dotnet.microsoft.com 2022 URL: <https://dotnet.microsoft.com/en-us/learn/csharp> (Дата обращения: 06.07.2022);
10. Пиксель-арт для начинающих: инструкция по применению [Электронный ресурс] // habr.com. 2019 URL: <https://habr.com/ru/company/playgendary/blog/482536/> (Дата обращения: 06.07.2022);