

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра математической кибернетики и компьютерных наук

РАЗРАБОТКА МОБИЛЬНОЙ ВЕРСИИ НАСТОЛЬНОЙ ИГРЫ
АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 451 группы
направления 09.03.04 — Программная инженерия
факультета КНиИТ
Яшина Егора Александровича

Научный руководитель
доцент, к. ф.-м. н.

А. С. Иванова

Заведующий кафедрой
к. ф.-м. н., доцент

С. В. Миронов

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 Теоретическая часть	4
1.1 Основные использованные компоненты движка Unity	4
1.1.1 Сцена	4
1.1.2 Camera	4
1.1.3 EventSystem	4
1.1.4 Sprite	5
1.1.5 BoxCollider	5
1.2 Портитрованные игры на движке Unity	5
1.2.1 Картографы	5
1.2.2 Каркассон	5
1.2.3 Talisman	5
2 Практическая часть	7
2.1 Правила игры	7
2.2 Реализация игры	7
2.2.1 Главное меню	7
2.2.2 Фигуры	7
2.2.3 Колода	8
2.2.4 Поля	8
2.2.5 Карта	8
2.3 Реализация партии	8
2.4 Реализация партии на одном устройстве	9
2.5 Реализация партии против бота	9
ЗАКЛЮЧЕНИЕ	11
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	12

ВВЕДЕНИЕ

В наше время развлекательная сфера занимает немалую часть приложений на мобильных устройствах. Также достаточно много людей интересуются настольными играми. Но к сожалению, не всегда удаётся найти партнера или когда он есть нет игры под рукой, чтобы скоротать время. Из-за этих причин начало популяризироваться портирование настольных игр для мобильных устройств или для персональных компьютеров [1].

Целью данной работы является разработка мобильной версии настольной игры. Для достижения данной цели были поставлены следующие задачи:

1. рассмотреть особенности создания мобильных приложений на движке Unity;
2. создать игру с возможностью игры на одном устройстве;
3. создать бота для созданной игры;

Для бота поставлены следующие задачи:

1. перебор возможных состояний игры;
2. определение эффекта на игру при ходе;
3. выбор и совершение наилучшего хода;

1 Теоретическая часть

В данное время существует огромное множество движков для разработки приложений. Наиболее популярными являются следующие движки:

1. Construct 3 — простой движок, в котором идёт визуальное программирование, но из-за него идёт ограничение в возможностях разработчика;
2. Unreal Engine — движок, который направлен на разработку 3D приложений;
3. Unity — движок, который направлен на межплатформенную разработку, ведущая в мире платформа направленная на разработку в реальном времени;
4. и другие;

В данной работе был использован движок Unity, из-за его популярности и простоте в разработке на мобильные платформы.

1.1 Основные использованные компоненты движка Unity

1.1.1 Сцена

Сцена — это самый большой блок, в котором описывается организация объектов. В самом простом виде это отдельно взятый уровень, в более продвинутом виде обычно загружено несколько сцен. Это позволяет сцены загружать и выгружать в память [2].

1.1.2 Camera

Параметр Camera отвечает за то, что видит пользователь. Camera позволяет настроить расстояние, с которого будет происходить отрисовка, цвет заднего фона, проекцию и прочие.

1.1.3 EventSystem

Параметр EventSystem позволяет объектам взаимодействовать с друг другом без сильного изменения кода. Благодаря EventSystem объекты могут общаться с помощью сигналов. Таким образом при отдаче сигнала одним объектом, те объекты, что имеют функцию принимающую этот сигнал, могут получить его и выполнять свой функционал [3].

1.1.4 Sprite

Sprite — это визуальное представление объекта. Оно позволяет выбрать разные настройки среди которых: изображение, цвет, слой отрисовки и т.п.

1.1.5 BoxCollider

BoxCollider — это представление границ, в которых происходит взаимодействие с объектом.

1.2 Портированные игры на движке Unity

В данном разделе представлены существующие портированные настольные игры. Следующие порты были выбраны из-за их популярности в сообществе настольных игр. И мною было проведено сравнение с их мобильной версией.

1.2.1 Картографы

Картографы — это игра выпущенная в 2019 и портированная на мобильные устройства годом позднее студией Brettspielwelt GmbH [4]. В ней игрокам предоставлены 4 цели дающие победные очки, которые меняются походу игры. В каждом раунде вытягивается карта с местностью, которую игрокам предстоит нарисовать на своём поле. В мобильной версии полностью реализован игровой процесс, который целиком передаёт игровой опыт [5].

1.2.2 Каркассон

Каркассон — это игра выпущенная в 2000 году и портированная на мобильные устройства в 2017 году студией Asmodee Digital [6]. В данной игре по очереди игроки берут случайный тайл и выкладывают его на поле так, чтобы он граничил с другими тайлами. После этого игрок может поставить рабочего, который дает победные очки в зависимости от места, и после этого передаётся ход [7]. Мобильная версия включает в себя режим игры против ботов, игру на одном устройстве и игру по сети. В игре на одном устройстве подразумевается передача устройства между игроками в течении партии.

1.2.3 Talisman

Talisman — игра выпущенная в 2007 и портированная на мобильные устройства в 2014 году студией Nomad Games [8]. В данной игроки передвигаются по игровому полю броском кубика, в зависимости от конечной локации

игрок выполняет разные действия написанные на ней [9]. Победной целью является дойти до центра карты. Talisman имеет всего два режима игры на одном устройстве или по сети.

2 Практическая часть

В рамках данной работы разработано приложение, представляющее настольную игру Onitama. Цель игры — довести своего мастера до храма оппонента или захватить мастера оппонента.

2.1 Правила игры

Игрой для разработки является настольная игра Onitama. Игра происходит на поле размером 5 на 5, в двух противоположных центрах расположены храмы. Также у каждого игрока есть 5 фигур (1 учителя и 4 ученика). После начальной расстановки каждому игроку раздается по 2 карты передвижения и 1 карта кладется сбоку. Карты показывают как могут передвигаться фигуры. После этого случайным образом выбирается первый игрок. После этого каждый ход активный игрок выбирает карту, которой он совершит ход, передвигает фигуру в соответствии с выбранной картой, если фигура оказывается на клетке с фигурой противника, то она съедает фигуру противника. Игра продолжается до тех пор пока один из игроков не съест учителя противника или он не закончит ход учителем на клетке храма противника [10].

2.2 Реализация игры

Движком для создания игры был выбран Unity. Он поддерживает разработку на различные платформы, в числе которых есть Android .

2.2.1 Главное меню

Первым что видит пользователь является главное меню, из которого он может выбрать режим игры [11].

2.2.2 Фигуры

Рассмотрим принцип работы игровых фигур. Она представляет собой Sprite позволяющий отображать фигуру на поле, также у неё есть компонент Box Collider 2D для определения области доступной для взаимодействия [12].

В момент инициализации на экране фигура получает цвет соответствующий её стороне и текстуру соответствующую её рангу. После инициализации фигура находится в состоянии ожидания нажатия на экран.

При нажатии на экран происходит проверка, было ли это нажатие на фигуру. Если пересечение произошло фигура переходит в фазу перемещения и меняет своё положение на место, где продолжается нажатие [13]. После того как происходит окончание нажатия фигура переходит в позицию, которая соотносится с местом на поле. После того как происходит завершение перемещения вызывается функция для каждой фигуры, которая либо перемещает фигуру в новое место, либо уничтожает её, если она была съедена.

2.2.3 Колода

Колода карт является словарем где ключ это номер карты в колоде, а значение это клетки перемещения на карте. Также она имеет две функции: перемешивания и выбора карты. Функция перемешивания выбирает случайный элемент с первого по последний и меняет их местами, этот процесс повторяется но диапазон меняется с первого до предпоследнего и так далее. Функция выбора карты возвращает значение словаря по ключу.

2.2.4 Поля

Поле представлено двумя видами: игровое и карточное. Для полей были созданы родительский класс `FieldBehaviour` и два дочерних класса: `GameFieldBehaviour`, `CardFieldBehaviour`. Родительский класс обладает основными функциями для поля такими как: очистка, определение границ, поворота, размера и положения на нём [14]. Различия в игровых полях в том, как они показывают свои компоненты.

2.2.5 Карта

Для взаимодействия между картой и игроком был создан класс `CardBehaviour`. При инициации происходит первичная установка карточного поля с правильной позицией и поворотом. Также этот класс отвечает за взаимодействие с нажатиями пользователем. Если нажатие происходит в руке активного пользователя, то карта становится активной. Если же карта там не находится то она переворачивается под вид текущего игрока, и в обратное положение при конце нажатия.

2.3 Реализация партии

Для реализации игры созданы родительский класс `GameController` и два дочерних класса:

GameFieldBehaviour, CardFieldBehaviour. При инициализации происходит установка поля в стартовое положение, создаётся поле, где первую линию занимают синие фигуры, а последнюю занимают красные, в центре каждой линии ставятся мастера и остальные ученики [15].

2.4 Реализация партии на одном устройстве

Одним из режимов игры является игра на одном устройстве. При инициализации поля происходит связь сигналов для получения позиции на поле и концом передвижения [16]. При игре после того как произошла попытка передвинуть фигуру, идет сигнал от нее к полю, которое проверяет может ли быть совершен этот ход. Идет запрос к игровому контролеру на массив с передвижением на выбранной карте. После этого берется позиция фигуры, определяются её границы передвижения и идет проход по всем клеткам с карты. Если положение, на которое фигуру хотели переместить, доступно, то подаётся сигнал о передвижении фигуры с начальной позиции на доступную, если же не доступно то возвращается стартовое положение фигуры.

2.5 Реализация партии против бота

Альтернативным режимом является игра против бота. Основной функцией бота является оценка игрового поля, за это отвечают функции `countSide` и `Evaluate`. Функция `countSide` получает на вход массив представляющий поле и цвет игрока, от которого идёт оценка, после этого идёт проход по всему массиву и подсчет фигур игроков, по 10 очков за ученика и по 1000 за учителя. Функция `Evaluate` подсчитывает разницу в состоянии игроков, сначала запускается `countSide` для двух игроков, подсчитывается разница между синим и красным игроком и она умножается на перспективу для игрока [17]. Эти функции используются для определения какой ход лучше всего совершить. При обдумывании хода определяются все возможные ходы с помощью функции `GetAllMoves`, она возвращает словарь с ключом начальной позицией и значением словарем с ключом номером карты и значением конечной позицией.

После этого идет поиск наилучшего хода, создается переменная хранящая значение с выигрышем лучшего хода, проходятся возможные ходы полученные ранее, и если достигнута глубина, то возвращается оценка конечного поля. При получении оценки идёт сравнение с выигрышем лучшего хода, ес-

ли они равны, то ход добавляется к словарю лучших, если оценка больше, то создается новый словарь с новым ходом. Если в итоге вышло больше чем один выгодный ход, то выбирается случайный из лучших [18–20]. Также для каждого полученного поля с набором карт идёт сохранение этого состояния и при попадании в это состояние снова берётся уже известный результат поиска.

Все эти функции включены в общую функцию `CommitMove`, которая инициализирует стартовое состояние карт, поля и глубину поиска, с этими переменными вызывается поиск следующего хода.

После того как бот совершает ход подается сигнал о совершении хода, после этого происходит смена карт и активного игрока аналогично игре на одном устройстве. Также при конце игры отображается победный экран с сообщением о победе/поражении и 3 кнопками аналогичными при игре на одном устройстве. В рамках разработки было проведено около 830 игр. Количество игр выигранных людьми составило: 333 игры. Количество игр выигранных ботом составило: 501 игра.

ЗАКЛЮЧЕНИЕ

В ходе данной работы была разработана мобильная версия настольной игра Onitama на движке Unity имеющая следующие режимы:

1. одиночный режим,
2. игра против противника.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 How to Create an RPG Game in Unity – Comprehensive Guide [Электронный ресурс].— URL: <https://gamedevacademy.org/how-to-create-an-rpg-game-in-unity-comprehensive-guide/> (Дата обращения 25.03.2021). Загл. экр. Яз. англ.
- 2 How to load a scene in Unity - Game Dev Beginner [Электронный ресурс].— URL: <https://gamedevbeginner.com/how-to-load-a-new-scene-in-unity-with-a-loading-screen/> (Дата обращения 25.04.2022). Загл. экр. Яз. англ.
- 3 Как работает Event System в Unity? | unity3dschool.com [Электронный ресурс].— URL: <https://unity3dschool.com/kak-rabotaet-event-system-v-unity.html> (Дата обращения 23.03.2022). Загл. экр. Яз. рус.
- 4 Brettspielwelt GmbH - Android Developer info on AppBrain [Электронный ресурс].— URL: <https://www.appbrain.com/dev/Brettspielwelt+GmbH/> (Дата обращения 21.03.2022). Загл. экр. Яз. англ.
- 5 Cartographers | Board Game | BoardGameGeek [Электронный ресурс].— URL: <https://boardgamegeek.com/boardgame/263918/cartographers> (Дата обращения 22.04.2022). Загл. экр. Яз. англ.
- 6 Carcassonne [Электронный ресурс].— URL: <https://www.asmodee-digital.com/en/carcassonne/> (Дата обращения 22.04.2022). Загл. экр. Яз. англ.
- 7 Carcassonne | Board Game | BoardGameGeek [Электронный ресурс].— URL: <https://boardgamegeek.com/boardgame/822/carcassonne> (Дата обращения 25.03.2022). Загл. экр. Яз. англ.
- 8 Talisman: Digital Edition - Nomad Games [Электронный ресурс].— URL: <https://nomadgames.co.uk/talisman-digital-edition> (Дата обращения 22.04.2022). Загл. экр. Яз. англ.
- 9 Talisman: Revised 4th Edition | Board Game | BoardGameGeek [Электронный ресурс].— URL: <https://boardgamegeek.com/boardgame/27627/talisman-revised-4th-edition> (Дата обращения 25.03.2022). Загл. экр. Яз. англ.

- 10 Onitama | Tesera [Электронный ресурс].— URL: <https://tesera.ru/game/onitama/> (Дата обращения 19.03.2021). Загл. экр. Яз. рус.
- 11 Unity documentation [Электронный ресурс].— URL: <https://docs.unity3d.com/ScriptReference/SceneManagement.SceneManager.html> (Дата обращения 22.04.2021). Загл. экр. Яз. англ.
- 12 *Бонд, . . .* Unity и C#. Геймдев от идеи до реализации / . . . Бонд. — Санкт-Петербург: Издательский дом «Питер», 2019.
- 13 How to Make a Chess Game with Unity | raywenderlich.com [Электронный ресурс].— URL: <https://www.raywenderlich.com/5441-how-to-make-a-chess-game-with-unity> (Дата обращения 19.04.2022). Загл. экр. Яз. англ.
- 14 *Хокин, . . .* Unity в действии. Мультиплатформенная разработка на C#. 2-е межд. издание / . . . Хокин. — Санкт-Петербург: Издательский дом «Питер», 2020.
- 15 Build your game in Unity - Android Developers [Электронный ресурс].— URL: <https://developer.android.com/games/develop/build-in-unity> (Дата обращения 19.04.2022). Загл. экр. Яз. англ.
- 16 *Ferrone, H.* Learning C# by Developing Games with Unity 2019: Code in C# and build 3D games with Unity, 4th Edition / H. Ferrone.— Birmingham: Packt, 2019.
- 17 *Aversa, D. D.* Unity Artificial Intelligence Programming / D. D. Aversa, A. S. Kyaw, C. Peters. — Birmingham: Packt, 2018.
- 18 A step-by-step guide to building a simple chess AI [Электронный ресурс].— URL: <https://www.freecodecamp.org/news/simple-chess-ai-step-by-step-1d55a9266977/> (Дата обращения 22.04.2021). Загл. экр. Яз. англ.
- 19 Chess-chess composition | Britannica [Электронный ресурс].— URL: <https://www.britannica.com/topic/chess/Chess-composition> (Дата обращения 19.04.2021). Загл. экр. Яз. англ.
- 20 *Liebana, D. P.* General Video Game Artificial Intelligence / D. P. Liebana, S. M. Lucas, R. D. Gaina. — California: Morgan and Claypool Publishers, 2020.