

Разработка информационной системы прохождения интеллектуальной игры МБДОУ Детский сад №5 п. Смидович

Бушманова Татьяна Сергеевна

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В статье описана разработка информационной системы прохождения интеллектуальной игры для воспитанников МБДОУ Детский сад №5 п. Смидович, реализованной на игровом движке Unity как кроссплатформенное приложение. Рассмотрены структура сцен, принципы навигации и единые механики трёх уровней, а также модуль автоматического сохранения прогресса и статистики попыток. Показано, как предложенное решение упрощает организацию интерактивных занятий и обеспечивает педагогу наглядный контроль результатов без ручного учёта.

Ключевые слова: Unity, кроссплатформенное приложение, обучающая игра, дошкольники, интеллектуальная игра, навигация сцен, прогресс, статистика попыток, UI/UX.

Development of an information system for passing the intellectual game Smidovich Kindergarten No. 5

Bushmanova Tatiana Sergeevna

Sholom-Aleichem Priamursky State University

Student

Abstract

The article describes the development of an information system for passing intellectual games for pupils of Smidovich Kindergarten No. 5, implemented on the Unity game engine as a cross-platform application. The structure of the scenes, the principles of navigation and the unified mechanics of the three levels are considered, as well as the module for automatically saving progress and attempt statistics. It is shown how the proposed solution simplifies the organization of interactive classes and provides the teacher with visual control of the results without manual accounting.

Keywords: Unity, cross-platform application, educational game, preschoolers, intellectual game, scene navigation, progress, attempt statistics, UI/UX.

1 Введение

1.1 Актуальность

Интерактивные интеллектуальные игры в дошкольном образовании становятся востребованным инструментом развития внимания, памяти и

логического мышления, а также повышают мотивацию детей за счёт игрового формата. Вместе с тем в образовательных организациях сохраняется проблема практического сопровождения таких занятий: воспитателю требуется быстро запускать задания, повторять нужные уровни и фиксировать результаты без трудоёмкого ручного учёта. Поэтому актуальной является разработка кроссплатформенной информационной системы прохождения интеллектуальной игры, которая обеспечивает простой и предсказуемый сценарий «меню → уровень → результат», автоматическое сохранение прогресса и сбор базовой статистики попыток. Использование Unity как основы реализации позволяет получить единый программный продукт для разных устройств, упростить внедрение в условиях детского сада и обеспечить стабильную работу без зависимости от сети и сторонних сервисов.

1.2 Обзор исследований

А.С. Дедюкина, Д.Г. Мандарова описали создание приложения для углубленного изучения повседневного быта якутской культуры при помощи межплатформенной среды разработки Unity [1]. Д.Ю. Тазабеков, Н.В.Бужинская исследовали особенности разработки обучающей компьютерной игры в среде Unity. Данная игра предназначена для обучения школьников информатике в занимательной форме. Графическая оболочка игры создавалась в Adobe Photoshop CS6, игровая механика - на игровом движке Unity. Для написания скриптов использовался язык программирования C# [2]. А.В. Бардин, Т.Н. Томчинская описали процесс создания интерактивной сцены в игровом движке Unity. Объяснен путь получения модели для экспорта в программном обеспечении Autodesk Infraworks. Модель создана на основе воспроизведенного ранее рельефа поверхности части Зеленского съезда Нижнего Новгорода и его близлежащей инфраструктуры. Рассмотрены способы импорта полученной модели в сцену Unity с дальнейшим взаимодействием с ней [3]. Н.Г. Дюкина, А.С. Шутов рассмотрели разработку интерактивного приложения на платформе UNITY. Интерактивное приложение рассматривается как компьютерная игра для мобильных устройств [4]. С.В. Козлов, И.О. Меженцев описали проблемы создания видеоигр, которые, в свою очередь, подталкивают к созданию новых решений для развития компьютерной науки. В качестве примера рассматривается разработка видеоигры с помощью межплатформенной среды Unity. Для взаимодействия со средой разработки используется язык программирования C#, который имеет свои особенности в данной среде [5].

1.3 Цель исследования

Целью исследования является разработка и описание информационной системы прохождения интеллектуальной игры для воспитанников МБДОУ Детский сад №5 п. Смидович, реализованной на игровом движке Unity как кроссплатформенное приложение.

В работе рассматриваются проектирование структуры сцен и навигации, реализация трёх игровых уровней с различными механиками, а также

организация подсистемы хранения прогресса и статистики попыток. Дополнительно описываются принципы унификации интерфейса, механизмы обратной связи и логика экранов результата, обеспечивающие повторяемость сценария прохождения и удобство использования в условиях дошкольной образовательной организации.

2 Материалы и методы

В качестве основы для разработки использовался игровой движок Unity (версия 2022.3 LTS), обеспечивающий кроссплатформенную сборку приложения и удобную реализацию интерактивных сценариев. Функциональная часть системы реализована на языке C# с применением компонентного подхода Unity (MonoBehaviour).

3 Результаты и обсуждения

Возрастные особенности дошкольников (4–7 лет) быстро потеря внимания при сложной навигации и перегруженных экранах, поэтому интерфейс интеллектуальной игры должен использовать минималистичную верстку, крупные элементы управления, ограниченную палитру и стабильные паттерны расположения кнопок. Текстовые инструкции должны сопровождаться голосовыми подсказками и визуальными маркерами (анимация выделения нужной зоны). Восприятие времени у детей младшего возраста затруднено, поэтому длительные экраны ожидания и сложные цепочки действий недопустимы. Важна немедленная обратная связь: позитивный звук/анимация при верном действии, мягкий нейтральный сигнал при ошибке.

Интерфейс должен опираться на предсказуемую структуру: главный экран → выбор уровня → прохождение → экран результата (успех/повтор). Каждая сцена должна иметь 1–2 целевых действия, без побочных меню и всплывающих окон. Визуальный язык — яркие, но не кислотные цвета; контрастное разделение активных элементов; отсутствие мелкого текста. Контент (задания, персонажи, предметы) должен быть релевантен опыту детей: бытовые объекты, животные, формы, простые логические связи. Озвучка — короткие нейтральные или доброжелательные реплики, без сложных терминов. Навигационные кнопки (“назад”, “повторить”, “продолжить”) располагаются в фиксированных местах на всех экранах, чтобы формировать привычку.

В системе участвуют две основные роли:

- Ребёнок: выполняет задания уровня, получает мгновенную обратную связь, видит простой результат (успех/нужно попробовать ещё раз).
- Воспитатель: выбирает уровень, помогает с инструкцией, при необходимости наблюдает за прогрессом (количество попыток, время на выполнение).

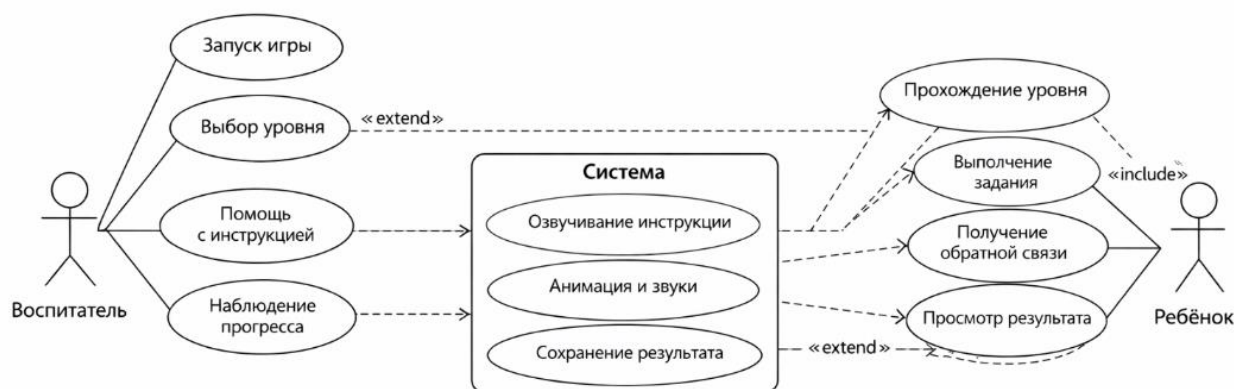


Рис. 1. Диаграмма ролей и взаимодействий

Ключевым элементом системы является контекст `I18nProvider`, отвечающий за хранение текущей локали и предоставление функций управления языком. Внутри провайдера используется состояние `locale`, инициализируемое значением по умолчанию, а также дополнительная логика загрузки сохранённой локали из `localStorage` при первом монтировании приложения. Такой подход обеспечивает восстановление пользовательских настроек между сессиями.

При проектировании важно обеспечить:

1. Минимальное число переходов и действий со стороны ребёнка;
2. Возможность для воспитателя быстро запускать нужный уровень, возвращать в меню и повторять задания;
3. Корректную работу в условиях отсутствия интернета и ограниченных ресурсов мобильного устройства.

Архитектура системы ориентирована на использование в дошкольной образовательной организации, поэтому приоритетными являются следующие принципы:

Простота и предсказуемость навигации. Сценарий прохождения игры должен быть линейным: «меню → выбор уровня → выполнение задания → результат → продолжение/повтор/меню». На каждом экране допускается 1–2 целевых действия. Система избегает вложенных меню, сложных настроек и «скрытых» переходов.

Повторяемость UX и единый язык управления. Независимо от механики уровня (перетаскивание, выбор, соединение пар) пользователь должен видеть знакомую структуру экрана и одинаково расположенные управляющие элементы. Повторяемость снижает когнитивную нагрузку, ускоряет привыкание и уменьшает число ошибок, вызванных интерфейсом, а не заданием.

Офлайн-режим и автономность. Система не требует сети Интернет, регистраций и сторонних сервисов. Все данные о прохождении сохраняются локально. Это повышает надёжность внедрения в детском саду и исключает зависимость от качества связи.

Отсутствие рекламы и отвлекающих факторов. Поскольку система применяется в образовательной среде, исключаются рекламные блоки,

всплывающие окна и внешние переходы. Также ограничивается использование избыточных анимаций, мерцаний и резких звуков.

Автоматическое сохранение и минимизация «административных» действий. Учёт попыток, фиксация прогресса, запись времени выполнения должны выполняться автоматически. Воспитателю не требуется вручную отмечать результаты или вводить данные.

Система реализована как набор Unity-сцен, связанных едиными правилами переходов и общими вспомогательными компонентами. При реализации 3 уровней целесообразно выделить следующие сцены:

- MainMenu — главное меню (кнопки «Начать/Продолжить», «Выбор уровней», «Выход» при необходимости).
- LevelMenu — меню уровней (карта), отображающее доступность уровней и прогресс.
- LV1, LV2, LV 3 — игровые сцены трёх уровней (разные механики при единых паттернах UI).
- Success — сцена успешного выполнения задания (поощрение, кнопки «Далее», «Повторить», «Меню»).
- Fail — сцена повторной попытки (мягкая обратная связь, кнопки «Повторить», «Меню»).

Навигация строится по предсказуемому правилу: после завершения уровня система переводит пользователя в один из результатов (Success/Fail), а затем — либо на следующий уровень, либо обратно в LevelMenu, либо на повтор текущего задания.

Фиксация целевых переходов реализуется через служебные значения следующей и повторной сцен (NextSceneName, RetrySceneName). Это исключает необходимость создавать отдельный «успех» для каждого уровня: сцены Success/Fail универсальны и получают информацию о том, куда возвращаться и куда переходить дальше.

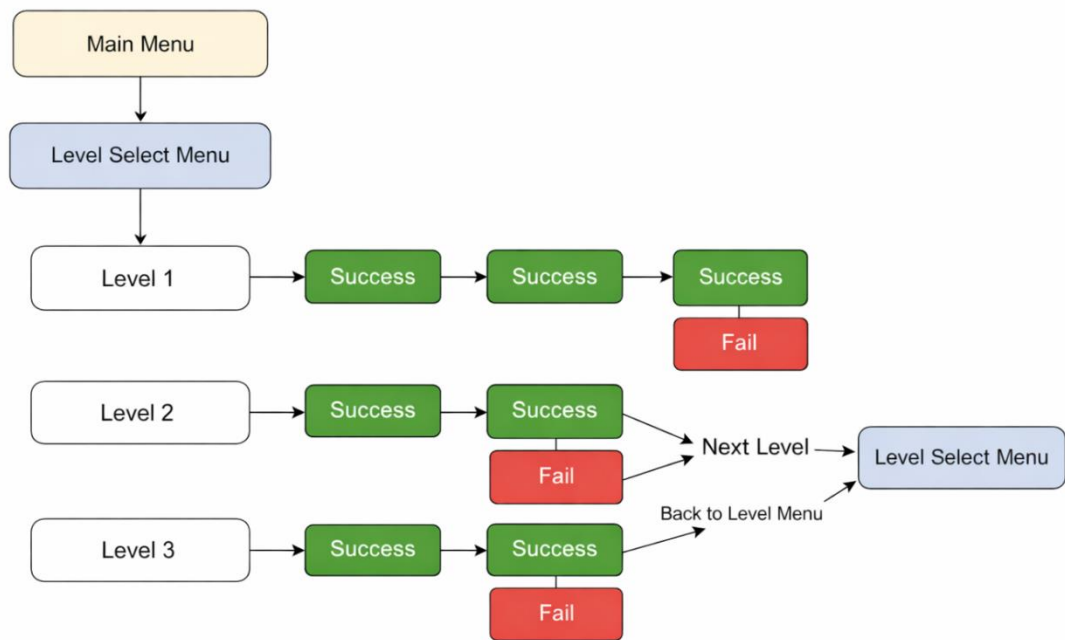


Рис. 2. Карта навигации сцен

Ограничение доступа к уровням реализовано в `LevelMenuController`: на старте меню уровней он считывает значение прогресса `GameProgress.MaxUnlockedLevel` и делает кнопки уровней активными только для доступных уровней. Недоступные уровни остаются неинтерактивными и визуально “приглушаются” стандартными средствами Unity (настройка `Disabled-состояния` у кнопки). Для визуального указания “точки продолжения” применяется `LevelMenuHeroMover`: изображение героя перемещается к последнему доступному уровню, помогая пользователю быстро понять, куда переходить дальше.

```
1  using UnityEngine;
2  using UnityEngine.UI;
3
4  0 references
5  public class LevelMenuController : MonoBehaviour
6  {
7      3 references
8      [SerializeField] private Button[] levelButtons; // Level1 = 0, Level2 = 1, Level3
9
10     0 references
11     private void Start()
12     {
13         int maxLevel = GameProgress.MaxUnlockedLevel;
14
15         for (int i = 0; i < levelButtons.Length; i++)
16         {
17             int levelNumber = i + 1;
18
19             if (levelNumber <= maxLevel)
20             {
21                 levelButtons[i].interactable = true;
22             }
23             else
24             {
25                 levelButtons[i].interactable = false;
26             }
27         }
28     }
29 }
```

Рис. 3. Логика разблокировки кнопок уровней в LevelMenuController

Уровень 1 (LV1) реализует механику перетаскивания предметов на соответствующие тени. Логика разделена на три компонента: Game2Controller управляет сценарием уровня, DraggableShadowItem отвечает за перетаскивание объекта и возврат на старт при ошибке, а ShadowTarget принимает “drop” и проверяет корректность по совпадению идентификаторов. При успешном размещении всех элементов контроллер фиксирует параметры возврата/продолжения (через LevelResultData), разблокирует следующий уровень и загружает экран успеха. Любая ошибка приводит к переходу на Fail с возможностью повторного запуска этого же уровня.

```
1 using UnityEngine;
2 using UnityEngine.SceneManagement;
3
4 2 references
5 public class Game2Controller : MonoBehaviour
6 {
7     [Header("Элементы уровня")]
8     2 references
9     [SerializeField] private DraggableShadowItem[] draggableItems; // все перетаскиваемые жив
10
11     [Header("Сцены результатов")]
12     1 reference
13     [SerializeField] private string successSceneName = "Success";
14     1 reference
15     [SerializeField] private string failSceneName = "Fail";
16
17     [Header("Куда идём после успеха")]
18     [Tooltip("Следующая сцена после успешного прохождения (например, Game2_Level2 или LevelMe
19     1 reference
20     [SerializeField] private string nextSceneName = "Game2_Level2";
21
22     3 references
23     private int correctlyPlacedCount = 0;
24
25     0 references
26     private void Start()
27     {
28         correctlyPlacedCount = 0;
29
30         foreach (var item in draggableItems)
31         {
32             if (item != null)
33             {
34                 item.Init(this);
35             }
36         }
37     }
38
39     1 reference
40     public void OnItemPlacedCorrect(DraggableShadowItem item)
41     {
42         correctlyPlacedCount++;
43
44         // когда все на своих тенях – победа
45         if (correctlyPlacedCount >= draggableItems.Length)
46         {
47             LevelResultData.RetrySceneName = SceneManager.GetActiveScene().name;
48             LevelResultData.NextSceneName = nextSceneName;
49         }
50     }
51 }
```

Рис. 4. Фрагмент кода Game2Controller: инициализация элементов и обработка переходов Success/Fail



Рис.5. Итоговый вид первого уровня

Уровень 2 (LV2) реализует механику “найди отличия”. Управляющий скрипт Game4Controller хранит набор подсветок отличий на обеих картинках (leftOutlines и rightOutlines), а также массив флагов найденных элементов и счётчик прогресса. При правильном клике соответствующее отличие отмечается найденным и подсвечивается сразу на обеих картинках. Повторные клики по уже найденному отличию игнорируются. Если найдено необходимое количество отличий, происходит переход на Success и (при необходимости) разблокируется следующий уровень. Неверный клик обрабатывается отдельно: пользователь сразу переводится на экран Fail, откуда доступна повторная попытка.

```
43 public void OnDifferenceClicked(int index)
44 {
45     if (index < 0 || index >= found.Length)
46     {
47         Debug.LogWarning("Game4Controller: неправильный index " + index);
48         return;
49     }
50
51     if (found[index]) return;
52
53     found[index] = true;
54     foundCount++;
55
56     if (leftOutlines[index] != null) leftOutlines[index].SetActive(true);
57     if (rightOutlines[index] != null) rightOutlines[index].SetActive(true);
58
59     if (foundCount >= found.Length)
60     {
61         if (unlockLevelOnWin > 0)
62         {
63             GameProgress.UnlockLevel(unlockLevelOnWin);
64         }
65
66         LevelResultData.RetrySceneName = SceneManager.GetActiveScene().name;
67         LevelResultData.NextSceneName = nextSceneName;
68
69         SceneManager.LoadScene(successSceneName);
70     }
71 }
72
73 // вызывается при клике в пустое место
74 0 references
75 public void OnWrongClick()
76 {
77     LevelResultData.RetrySceneName = SceneManager.GetActiveScene().name;
78     LevelResultData.NextSceneName = SceneManager.GetActiveScene().name;
79
80     SceneManager.LoadScene(failSceneName);
81 }
82 }
83
```

Рис. 6. Game4Controller: обработка правильного клика, обработка неверного клика и переход на Fail

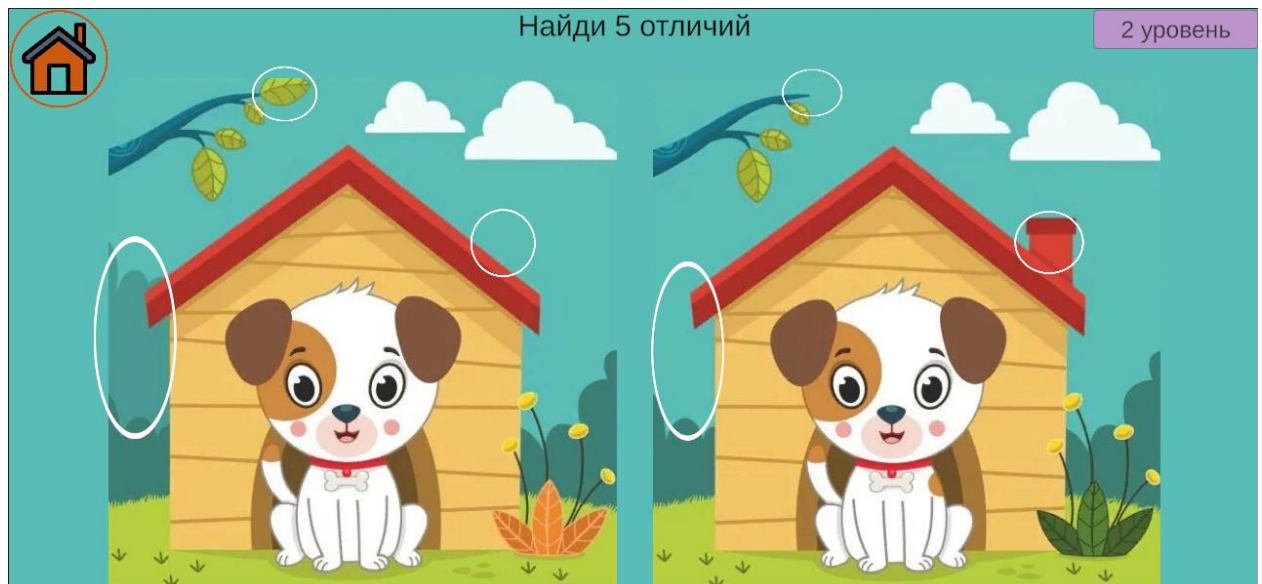


Рис. 7. Итоговый вид второго уровня

Уровень 3 (LV3) реализует механику соединения пар линиями. Каждый элемент представлен узлом `Game6Node` с идентификатором пары. Контроллер `Game6LinesController` управляет выбором первой и второй точки: первый клик выделяет узел (в интерфейсе используется мягкое визуальное подтверждение — увеличение масштаба), второй клик завершает проверку пары. При совпадении идентификаторов создаётся линия между узлами, оба узла помечаются соединёнными и перестают реагировать на дальнейшие нажатия. После соединения всех пар выполняется переход на `Success` с фиксацией данных навигации. При несовпадении пары уровень немедленно переводится на `Fail`, чтобы пользователь повторил попытку.

```
14 references
29 private Game6Node firstSelected;
3 references
30 private int correctConnections = 0;
31
0 references | Unity Message
32 private void Start()
33 {
34     foreach (var node in nodes)
35     {
36         if (node != null)
37             node.Init(this);
38     }
39
40     firstSelected = null;
41     correctConnections = 0;
42 }
43
1 reference
44 public void OnNodeClicked(Game6Node node)
45 {
46     if (node.IsConnected)
47         return;
48
49     if (firstSelected == null)
50     {
51         firstSelected = node;
52         firstSelected.SetSelected(true);
53         return;
54     }
55
56     if (firstSelected == node)
57     {
58         firstSelected.SetSelected(false);
59         firstSelected = null;
60         return;
61     }
62
63     bool isCorrectPair = firstSelected.Id == node.Id;
64
65     if (isCorrectPair)
66     {
67         CreateLineBetween(firstSelected, node);
68     }
```

Рис. 8. Инициализация уровня LV3: передача контроллера в узлы и сброс состояния

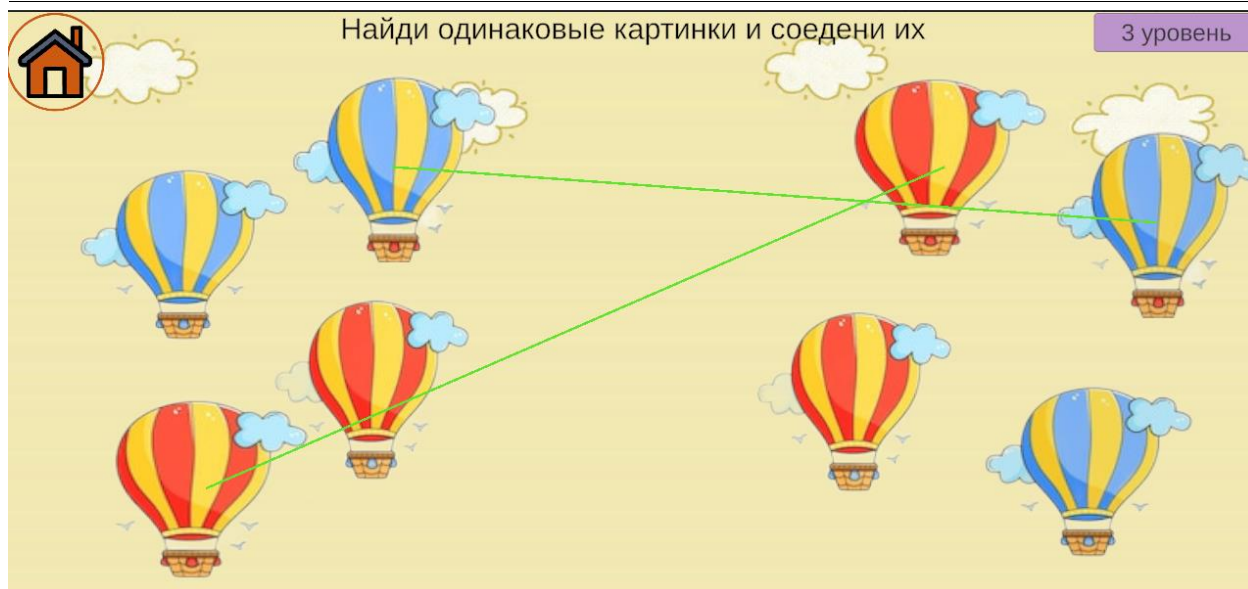


Рис. 9. Построение линии между элементами пары

4 Выводы

В работе разработана и реализована информационная система прохождения интеллектуальной игры для МБДОУ «Детский сад №5 п. Смидович» на базе Unity, обеспечивающая единый сценарий взаимодействия «меню → выбор уровня → прохождение → результат». Реализованы три обучающие механики (перетаскивание в тени, поиск отличий, соединение пар), а также общий каркас навигации и экранов результата с сохранением контекста переходов. В систему встроено локальное хранение прогресса, позволяющее автоматически фиксировать доступность уровней и статистику попыток без участия педагога. Полученное решение демонстрирует устойчивую работу в кроссплатформенной среде и может применяться как практический инструмент для сопровождения интерактивных занятий в дошкольной образовательной организации.

Библиографический список

1. Дедюкина А.С., Мандарова Д.Г. Разработка vr приложения «Якутский балаган» с помощью межплатформенной среды разработки Unity // DIGITAL EDU. Цифровые компетенции в образовании. Сборник материалов Всероссийского научного форума с международным участием. Киров, 2022. С. 228-231.
2. Тазабеков Д.Ю., Бужинская Н.В. Применение среды unity для разработки обучающих компьютерных игр // Тенденции развития науки и образования. 2020. № 58-2. С. 34-37.
3. Бардин А.В., Томчинская Т.Н. Разработка интерактивной сцены в игровом движке unity для автомобильного симулятора автошколы // КОГРАФ-2022. сборник материалов 32-й Всероссийской научно-практической конференции по графическим информационным технологиям и системам. Нижний Новгород, 2022. С. 144-152.

4. Дюкина Н.Г., Шутов А.С. Разработка интерактивного приложения на платформе unity // Вестник педагогического опыта. 2022. № 53. С. 21-24.
5. Козлов С.В., Меженцев И.О. Взаимодействие языка программирования с# с межплатформенной средой разработки Unity // Системы компьютерной математики и их приложения. 2019. № 20-1. С. 193-199.