

ПРОЕКТИРОВАНИЕ СИСТЕМЫ ОТСЛЕЖИВАНИЯ ПРОГРЕССА ПОЛЬЗОВАТЕЛЯ В РАМКАХ ОБУЧАЮЩЕГО ТРЕНАЖЕРА «РЕМОНТ УЧАСТКА ТРУБОПРОВОДА»

Ермолаев Е.М.¹, Лоскутов В.В.²

¹ ТПУ, ИШИТР, 8ИМ31, e-mail: puchek01@yandex.ru

² ТПУ, ОИТ ИШИТР, ст. преподаватель, e-mail: rewenger@tpu.ru

Аннотация

В данной статье рассматривается задача по реализации системы отслеживания прогресса пользователя. Задача решена созданием системы отслеживания прогресса на основе связного графа. В результате работы получена система отслеживания прогресса, и апробирована ее работа в рамках обучающего тренажера «Ремонт участка трубопровода».

Ключевые слова

Квестовая система, последовательность шагов, тренажер, система прогресса, VR, виртуальная реальность.

Введение

Технологии виртуальной реальности активно развиваются в последнее десятилетие и продолжают укреплять свои позиции в сфере образования. Причина такого интереса кроется в наглядности процесса обучения, его удешевлении, возможности безопасно совершать ошибки, набирать механическую память и так далее [1,2].

В рамках научной работы была поставлена задача разработки интерактивного тренажера, обучающего последовательности операций ремонта участка трубопровода. Во время разработки тренажера появилась необходимость спроектировать и разработать систему, отслеживающую выполнение пользователем последовательности шагов. Выполнение всей последовательности шагов означает успешное завершение тренажера. Отсутствие подобной системы приводит к дополнительным временным затратам, сложно изменяемой и трудно масштабируемой архитектуре. Шаги, созданные с помощью данной системы, должны иметь возможность состоять из нескольких действий, либо выполняться параллельно с другими шагами. Система должна быть легко масштабируемой и удобно расширяемой.

На основе поставленных требований был проведен анализ уже имеющихся систем, но найденные варианты оказались слишком перегруженными не нужной для работы функциональностью и не обладали достаточной гибкостью и удобством в модификации, рассчитанными на игры [3,4]. Это обуславливает необходимость данной работы.

Основная часть

Решение производственной задачи по ремонту участка трубопровода состоит из большой последовательности шагов. На основе них была создана блок-схема (рис. 1) последовательности действий, которые необходимо выполнить пользователю для прохождения тренажера.



Рис. 1. Часть блок-схемы

При проектировании системы было решено представить последовательность действий в виде связного графа, состоящего из узлов и рёбер [5]. Работа конечной системы представлена в виде схемы на рисунке 2.

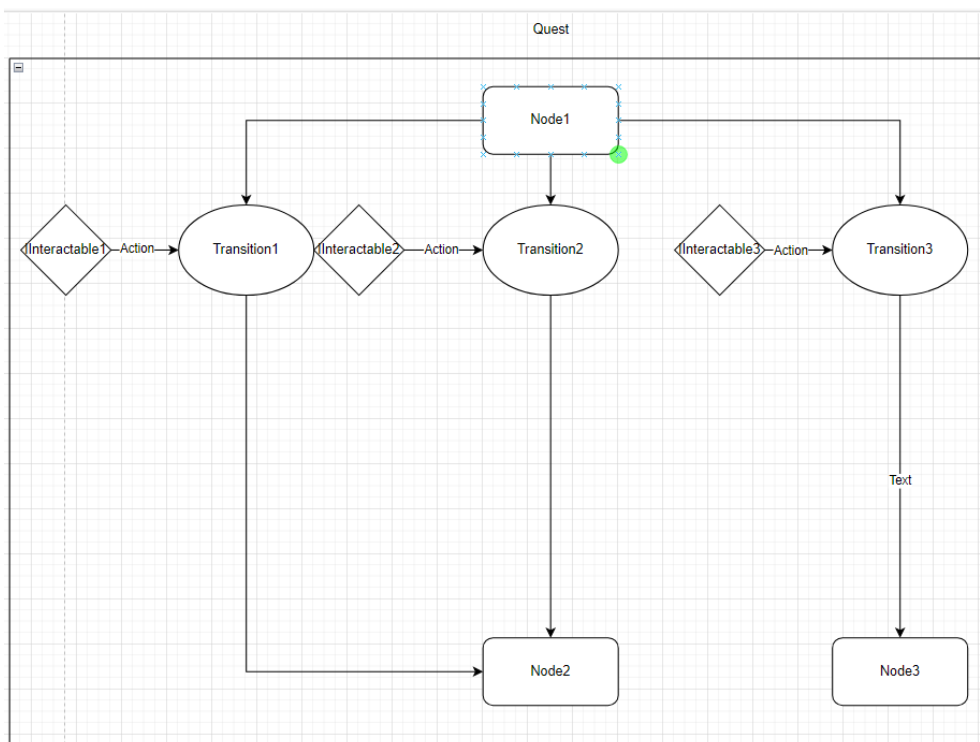


Рис. 2. Схема работы системы

В системе, есть три базовых класса: Transition, Node, Quest и объекты, реализующие интерфейс Interactable. Класс Transition — представляет из себя переход между двумя объектами класса Node, а Quest контролирует данный процесс. Интерфейс Interactable реализуется непосредственно объектом, который указывает параметры для проверки правильности выполнения шага, а также событие, которое вызывает проверку внутри подписанного на него перехода. Если параметры совпадают с эталонными, сохраненными в Transition, то срабатывает событие перехода и активируется следующий узел, ссылка на который хранится в Transition. Если ссылка пуста или такого узла не существует, то происходит событие завершения последовательности, и система деактивируется.

Для данной системы была создана система автоматического прохождения, которая запускается при необходимости проведения тестирования. Это позволит значительно ускорить отладку создаваемой последовательности шагов на базе основной системы. Она реализована с помощью класса Tester, схема ее работы представлена на рисунке 3.

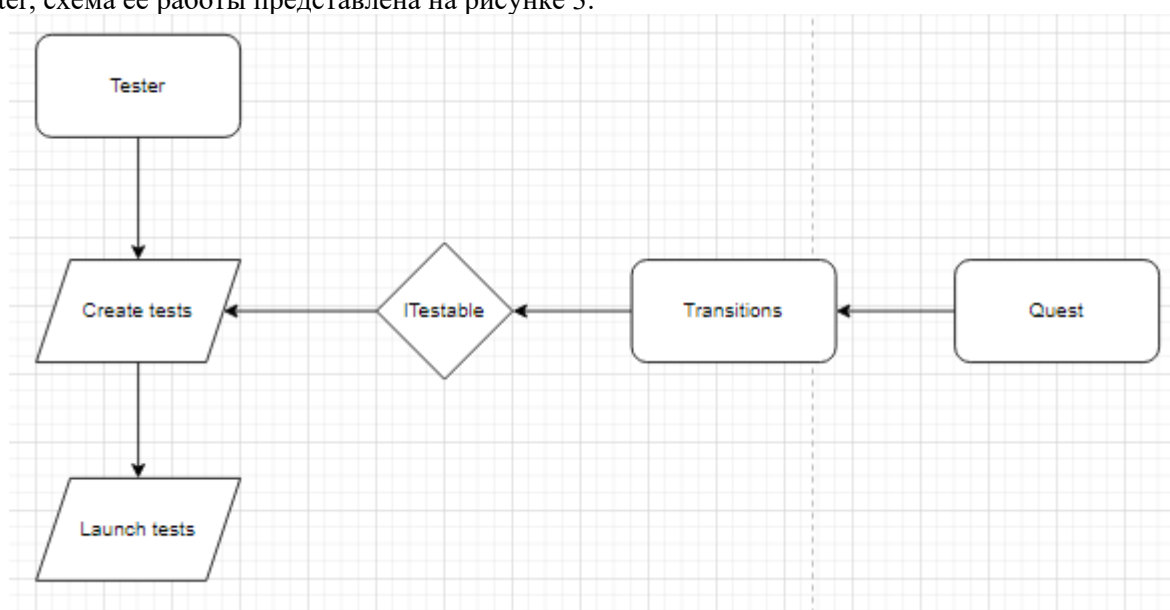


Рис. 3. Схема работы системы автоматического прохождения

В объект класса Tester подается ранее созданный Quest, хранящий в себе узлы и переходы, там из переходов берутся объекты, за которыми за которыми они следят, и, если они реализуют интерфейс ITestable, добавляются в список тестов. Далее запускаются тесты в той очередности, в которой были шаги в Quest.

Результаты

В результате проделанной работы была реализована расширяемая и масштабируемая система, позволяющая успешно выполнять последовательность операций по блок-схеме, часть которой представлена на рисунке 1. Она имеет возможность гибкого расширения и добавления объектов взаимодействия (таких, как рация, полумуфты и т.п.). Также была реализована система автоматического прохождения квеста, которая значительно упрощает и ускоряет процесс создания и отладки последовательности шагов.

Чтобы доказать удобство и простоту расширения созданной системы, создадим простую последовательность шагов, состоящую из двух узлов и перехода между ними. В первом случае реализованная система использоваться не будет, во втором будет использована система. (рис. 4-5).

```
public class TransmitterTool : DiveTool
{
    public Action<int, int> OnStateChanged;

    // Some TransmitterTool code...
}

public class Quest : MonoBehaviour
{
    public int CurrentTransmitterRecipient = 0;
    public int CurrentTransmitterMessage = 0;
    public TransmitterTool Transmitter;
    public Action NextStepAction;

    public void NextStepActivate()
    {
        NextStepAction?.Invoke();
    }

    public void CreateFirstStepQuest()
    {
        NextStepAction += CreateSecondStepQuest;
        Transmitter.OnStateChanged += CheckTransmitterParams();
        CurrentTransmitterRecipient = 0;
        CurrentTransmitterMessage = 1;
    }

    public void CreateSecondStepQuest()
    {
        NextStepAction -= CreateSecondStepQuest;
    }

    private void CheckTransmitterParams(int inRecipient, int inMessage)
    {
        if(CurrentTransmitterRecipient == inRecipient && CurrentTransmitterMessage == inMessage)
        {
            Transmitter.OnStateChanged -= CheckTransmitterParams();
            NextStepActivate();
        }
    }
}
```

Рис. 4. Создание последовательности шагов без использования системы

```

class InteractableTransmitterParams : IInteractableParams
{
    public int ReceptientNumber;
    public int MessageNumber;

    public InteractableTransmitterParams(int receipientNumber, int messageNumber) { /* Constructor logic */ }
    public bool Equal(IInteractableParams param) { /*Check equal params logic*/ }
}

public Quest QuestSequence = new Quest();

public void CreateQuest()
{
    QuestSequence.CreateNodes(2);

    QuestSequence.Nodes[0].AddTransition(new QuestTransition(parameters: new InteractableTransmitterParams(0, 0),
                                                                target: Transmitter,
                                                                nextNode: QuestSequence.Nodes[1],
                                                                callback: null));
}

```

Рис. 5. Создание последовательности шагов с использованием системы

Разработанная система является легко масштабируемой ввиду того, что она позволяет расширять количество и функциональность интерактивных объектов и легко создавать новые последовательности действий пользователя без внесения правок в саму систему.

Заключение

Система разработана в рамках тренажера виртуальной реальности «Ремонт участка трубопровода», апробирована и успешно введена в учебный процесс. В дальнейшем планируется оптимизация системы, создание удобного графического интерфейса для создания квестов, а также внедрение данной системы в другие тренажеры.

Список использованных источников

1. VR-технологии в сфере образования // Svetak: сайт. – 2024. – URL: <https://svetak.ru/blog/virtual-reality>
2. Виртуальная реальность в образовании // Высшая школа бизнеса: сайт. – 2024. – URL: <https://hsbi.hse.ru/articles/virtualnaya-realnost-v-obrazovanii/>
3. Пример квестовой системы // Github: сайт – 2024. – URL: <https://github.com/devdogio/Quest-System-Pro>
4. Плагин с квестовой системой для Unity // Unity asset store: сайт. – 2024. – URL: <https://assetstore.unity.com/packages/tools/game-toolkits/quests-system-for-unity-218596>
5. Теория графов, связанные графы // Hexlet: сайт. – 2024. – URL: https://ru.hexlet.io/courses/graphs/lessons/connectivity/theory_unit