МИНОБРНАУКИ РФ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Волгоградский государственный технический университет»

Институт архитектуры и строительства

Факультет архитектуры и градостроительного развития

Направление (специальность) 09.03.02 Информационные системы и технологии  
Кафедра цифровых технологий в урбанистике, архитектуре и строительстве

Дисциплина Качество и надежность ИС

|  | Утверждаю  зав. кафедрой Д.С. Парыгин |
| --- | --- |
|  | «\_\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20 \_\_\_ г. |

**ЗАДАНИЕ**

**на курсовую работу**

Групповая работа студентов: Сагалаев М.С., Подмосковнов И.Ю., Свиридов Д.Е., Чекашов М.П, Березовский И.П.

Группа ИСТ-1-22

1. Тема: «Разработка информационной системы для заданной предметной области с учетом требований к качеству. Строительный VR тренажёр.»

2. Срок представления работы к защите « » 20 г.

3. Содержание расчетно-пояснительной записки: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4. Перечень графического материала: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5. Дата выдачи задания « » 20 г.

Руководитель работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Рашевский Н.М.

подпись, дата

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Сагалаев М.С.

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Свиридов Д.Е.

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Чекашов М.П.

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Березовский И.П.

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Подмосковнов И.Ю.

МИНОБРНАУКИ РФ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Волгоградский государственный технический университет»

Институт архитектуры и строительства

Факультет архитектуры и градостроительного развития

Кафедра цифровых технологий в урбанистике, архитектуре и строительстве

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**к курсовой работе**

по дисциплине Качество и надежность ИС

на тему «Разработка информационной системы для заданной предметной области с учетом требований к качеству. Строительный VR тренажёр.»

Студенты Сагалаев Михаил Сергеевич, Чекашов Матвей Павлович, Березовский Пётр Иванович, Свиридов Денис Евгеньевич, Подмосковнов Илья Юрьевич.

(фамилия, имя, отчество)

Группа ИСТ-1-22

Руководитель работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись и дата подписания) (инициалы и фамилия)

Члены комиссии:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись и дата подписания) (инициалы и фамилия)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись и дата подписания) (инициалы и фамилия)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись и дата подписания) (инициалы и фамилия)

Нормоконтролер \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Рашевский Н.М.

(подпись, дата подписания) (инициалы и фамилия)

Волгоград 2023 г.

Содержание

[1. Исследование предметной области. 4](#_heading=h.30j0zll)

[2. Использование системы контроля версий 5](#_heading=h.1fob9te)

[3. Тестирование информационной системы 9](#_heading=h.3znysh7)

[4. Кодирование. 14](#_heading=h.2et92p0)

[5. Результат работы программы 17](#_heading=h.tyjcwt)

# 1. Исследование предметной области.

Наша программа является строительным тренажёром. Главной ее целью является обучение будущих строителей практической части. Также наша программа может помочь уже состоявшимся строителям отточить свои навыки.

На данном этапе разработки были разработаны следующие функции:

* Меню
* Передвижение и телепортация игрока
* Подбор предметов
* Подготовка (очистка) пола
* Заполнение кюветки грунтовкой
* Грунтовка стен и пола
* Наполнение ведра
* Открытие крана
* Смешивание смеси
* Нанесение смеси на стену
* Установка ПГП
* Установка распорки
* Заливка монтажной пеной

Также наша программа имеет следующие особенности:

* Атмосферное звуковое сопровождение
* Система квестов
* Красивая визуальная составляющая
* Возможность просмотра статистики по завершении сессии

Работа над нашей программой совершается группой из пяти человек.

Задачи равномерно распределены между членами команды, что помогает добиться продуктивности и улучшения качества продукта.

Свиридов Денис Евгеньевич – тим-лид, разработчик

Чекашов Матвей Петрович – тех-лид, разработчик, работа с документами

Подмосковнов Илья Юрьевич – UI designer, разработчик

Сагалаев Михаил Сергеевич – Sound designer, разработчик

Березовский Петр Иванович – VFX designer, разработчик

В ходе работы наша команда столкнулась с необходимостью создания

макета программы. Данная проблема была решена успешно.

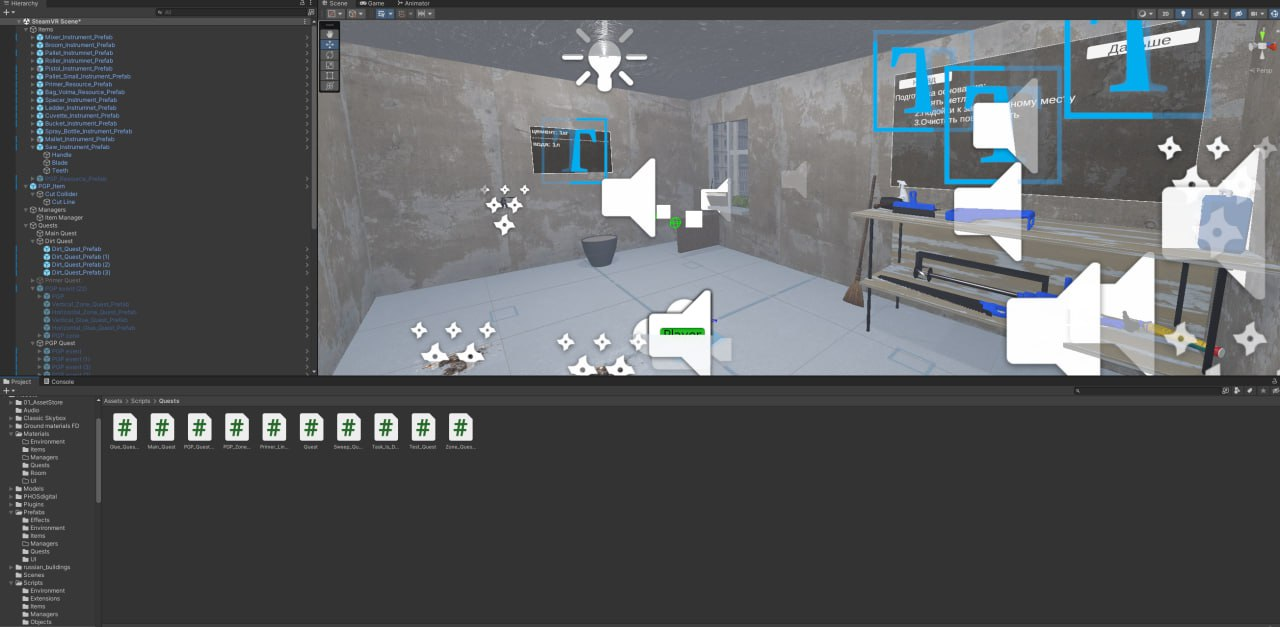


Рис.1. Макет программы.

Предложенный нами интерфейс удовлетворяет такому фактору качества

разработки ИС, как удобство использования, так как разработанная нами программа является интуитивно понятной и простой как для целевой аудитории, так и для других пользователей. Также важно отметить, что интерфейс нашего проекта не перегружен лишними объектами.

# 2. Использование системы контроля версий

Программа должна реализовываться с использованием распределенной системы контроля версий, поэтому членами нашей команды было принято решение хранить наш репозиторий на web-ресурсе GitHub. Данное решение обосновано тем, что данный продукт имеет понятный интерфейс, широкий функционал и удобную Desktop-версию.

Наша работа с GitHub началась с создания веток.



Рис.2. Пример веток проекта.

Main – главная ветка репозитория

Denis – ветка тим-лида

Sandbox – ветка для экспериментирования

Sounds – ветка для работы со звуками

SteamVR – ветка для работы с плагином SteamVR

VFX – ветка для работы с визуальными эффектами

VR – ветка с программой, адаптированной для VR

Iluha-UI – ветка для работы с интерфейсом

Каждый из участников создал для себя собственную ветку, в которой вел работу над проектом. Данное решение обусловлено тем, что использование отдельной ветки позволяет разработчику делать свою работу и избежать прекращения работы всего проекта в случае возникновения ошибок или конфликтов.

Изменения в ветках происходили путем коммита в них изменений

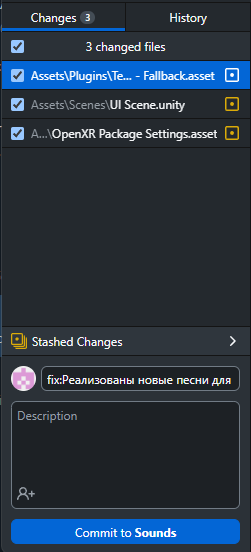


Рис.3. Пример создания коммита.

feat: - добавление чего-либо

fix: - исправление существующих объектов

После создания коммита, его необходимо выгрузить. Если все сделано правильно, то изменения в ветке сохранятся и это будет видно на GitHub.

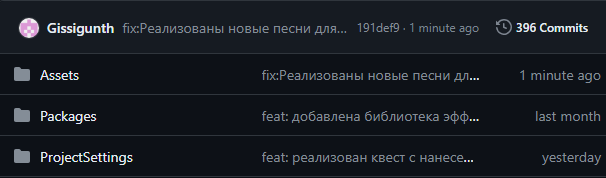


Рис.4. Пример примененного к ветке коммита.

Web-версия GitHub даёт понять другим членам команды, когда и где, были произведены изменения какой-либо ветки.

Изменения каждой ветки за время создания проекта образует дерево ведения проекта.

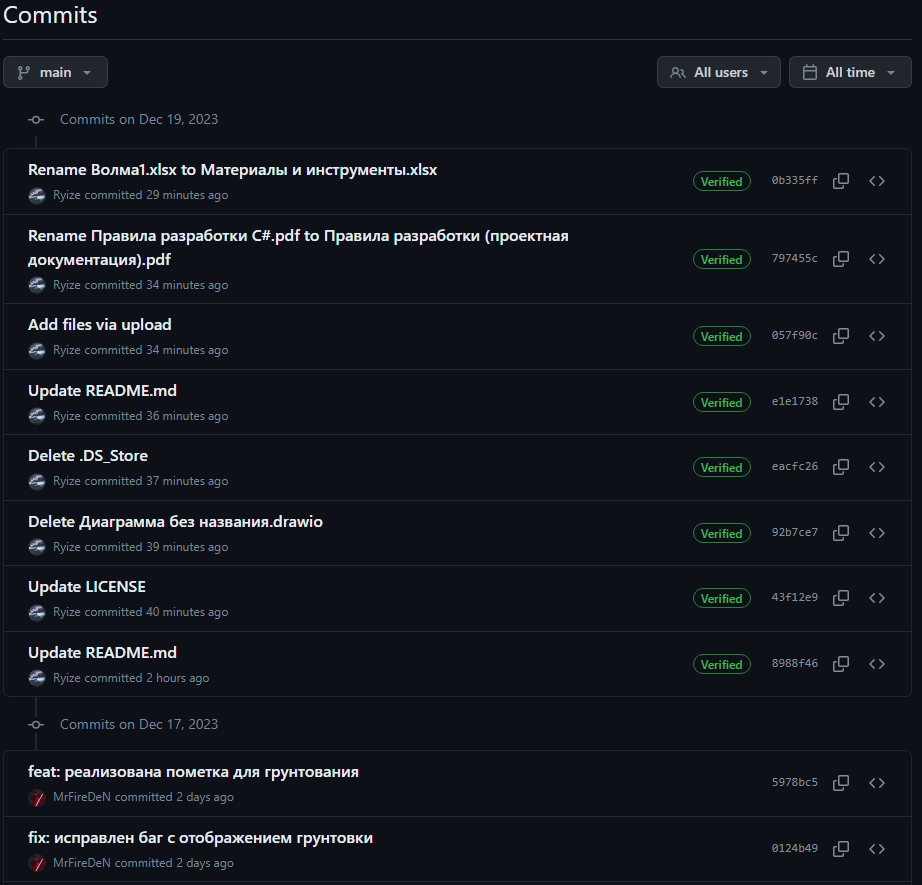


Рис.5. Дерево ведения проекта.

Rename – переименование файла

Add files via – добавление файлов

Update – изменение файла

Delete – удаление файла

В дереве ведения проекта показаны коммиты главной ветки репозитория и дата их введения.

# 3. Тестирование информационной системы

Тестирование программы является неотъемлемой частью разработки продукта, так как нужно учитывать то, что не все пользователи могут корректно пользоваться приложением и с целью избегания прекращения работы, необходимо сделать так, чтобы программный код корректно отрабатывал во всех случаях.

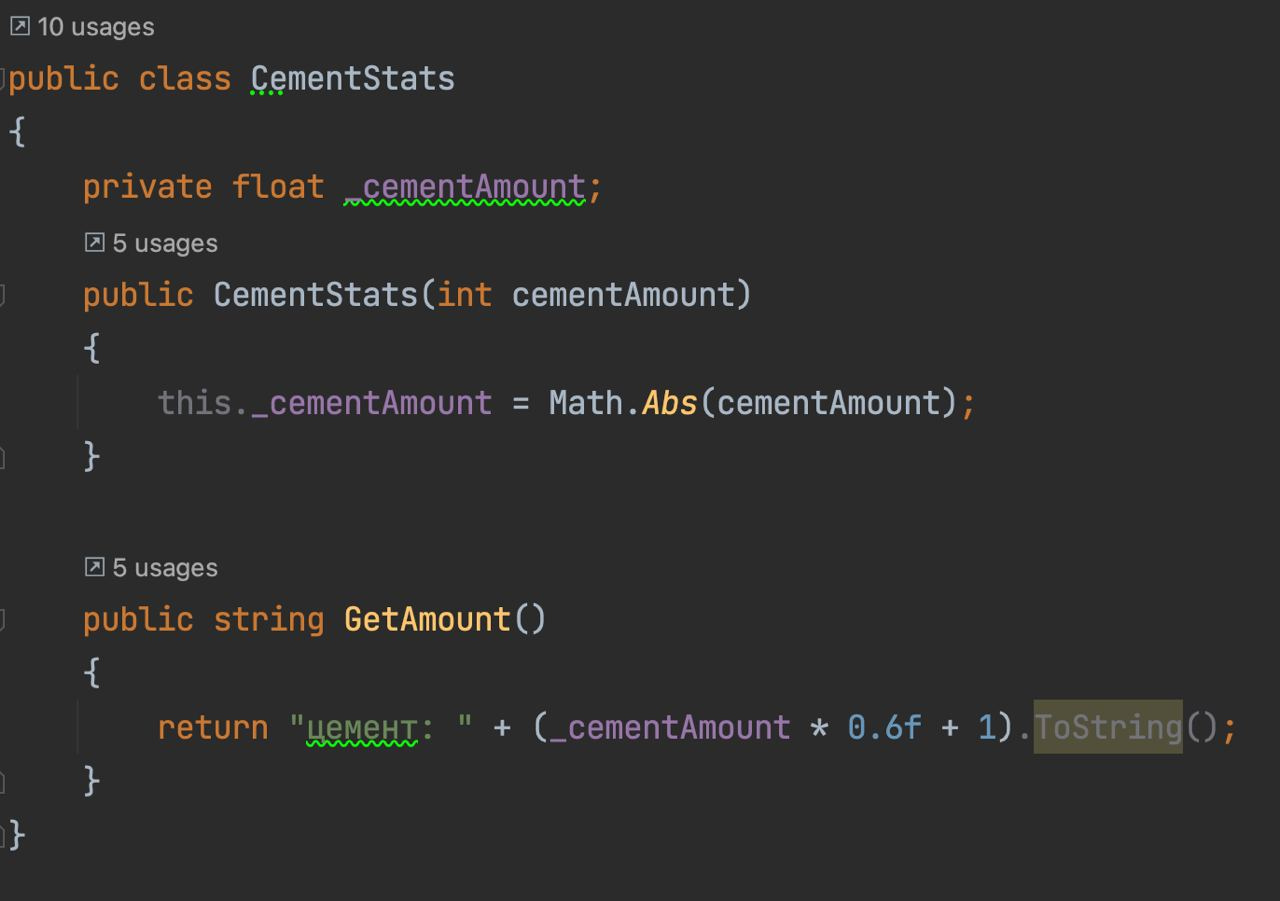


Рис. 6. Тестируемая функция

Данная функция выводит статистику использованного цемента в течение сессии.

Соответственно, масса смеси может быть только положительным числом.



Рис.7. Первая часть теста.

Первая функция теста проверяется, как отработает класс при небольших отрицательных значениях. Отрицательные значения должны браться по модулю. Вторая – как отработает класс при небольших положительных значениях.



Рис. 8. Вторая часть теста.

Первая функция теста проверяется, как отработает класс при больших отрицательных значениях. Отрицательные значения должны браться по модулю. Вторая – как отработает класс при больших положительных значениях.

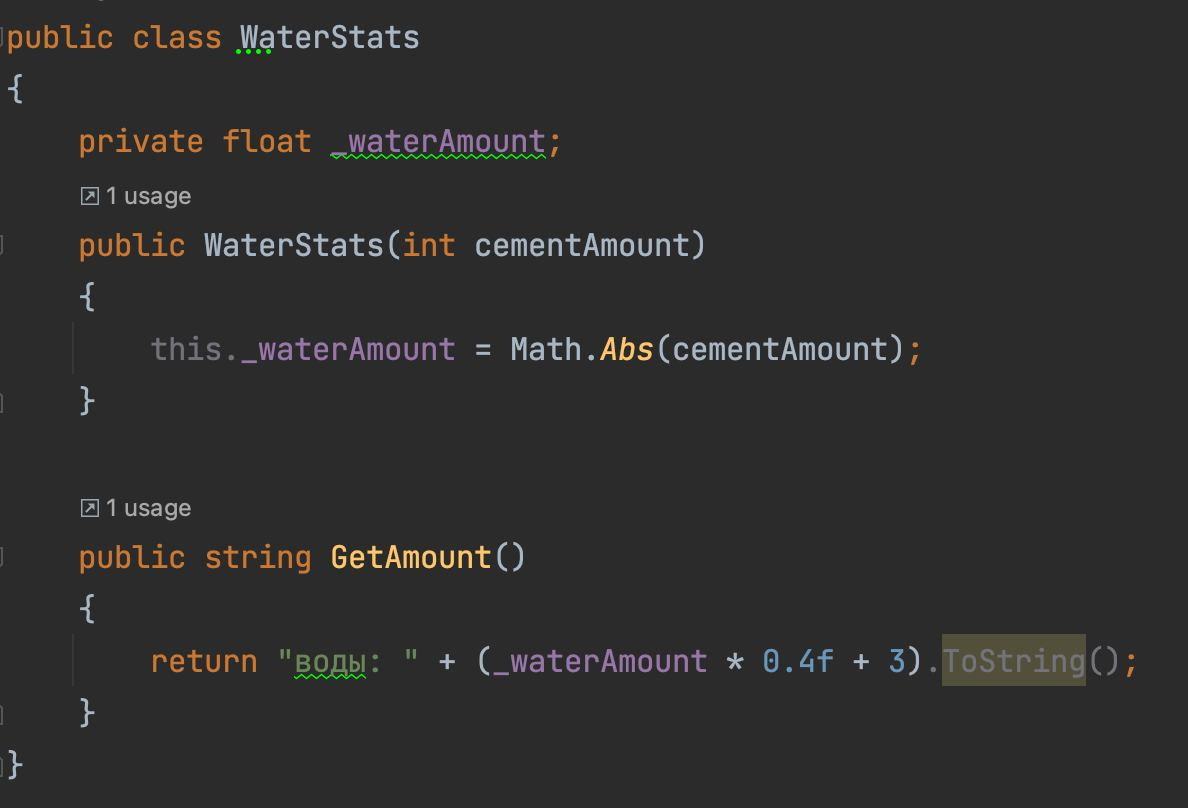


Рис. 9. Класс для статистики истраченной воды.

Данная функция в конце сессии выводит статистику о потраченной воде. Соответственно, объем воды не может быть отрицательным значением.



Рис.10. Первая часть теста второго класса.

Первая функция теста проверяется, как отработает класс при небольших отрицательных значениях. Отрицательные значения должны браться по модулю. Вторая – как отработает класс при небольших положительных значениях.



Рис.11. Вторая часть теста второй функции.

Третья функция теста проверяется, как отработает класс при больших отрицательных значениях. Отрицательные значения должны браться по модулю. Четвёртая – как отработает класс при больших положительных значениях.

# 4. Кодирование.

При разработке проекта, необходимо огласить требования к формату кода, чтобы он читался любым членом команды независимо от того, кто этот код писал.

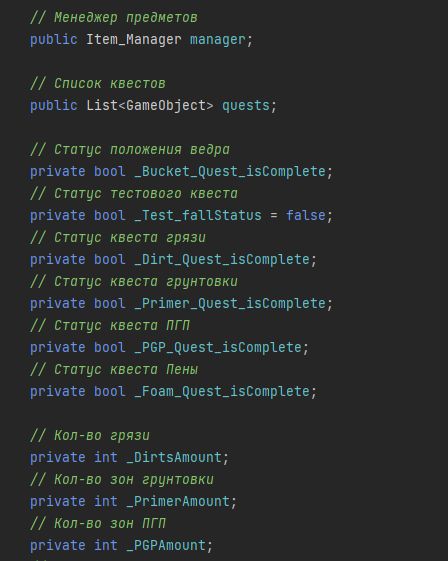


Рис.12.Пример оформления списка переменных.

Правила форматирования кода:

* В одном файле не может быть более 5 классов.
* Все названия классов и методов должны быть в PascalCase. Каждое

слово начинается с большой буквы.

* Все названия переменных/полей должны быть в camelCase.
* Если переменная/поле в одно слово, то необходимо писать с маленькой

буквы.

* Не имеет смысла дописывать Method, Attribute, Class и т.д.
* Если в коде есть аббревиатуры, то вы должны дать расшифровку в

документации класса и в документации метода/функции.

* Название метода должно быть глаголом, название класса -

существительным.

* В одном файле не должно быть более 500 строк кода. В методе не более

100 строк кода. Одна строка не более 120 символов.

* Закрытые (приватные) поля помечаются нижним подчеркиванием.
* Открывающаяся фигурная скобочка должна быть на уровне кода (при

объявлении класса, метода, функции, цикла и т.д.)

* Фигурные скобочки ставятся всегда. Даже если запись в одну строку.

Структура:

1) Base

1.1) Sound

1.1.1) Notify(string, anything)

1.2) BucketQuest

1.2.1) Notify(string, anything)

1.3) … Любой скрипт, наследующийся от MonoBehaviour

1.3.1) Notify(string, anything)

2) Managers

2.1) Item\_Manager

2.1.1) NotifyBucketQuestOpenFaucet(bool)

2.2) Quest\_Manager

2.2.1) NotifyQuestComplete(string)

2.3). … Остальные менеджеры

2.3.1) Любое кол-во методов начинающихся с Notify

3) Repository

3.1) Item\_Repository

3.1.1) Поля (атрибуты) связанные с предметами

3.2) Quest\_Repository

3.2.1) Поля (атрибуты) связанные с квестами

3.3) … Любое другое хранилище данных

3.3.1) Поля (атрибуты) связанные с этим хранилищем



Рис.13. Пример директории в Unity.

Структура директорий в Unity:

1) Assets (Типы объектов)

Вложенные папки (далее тип):

1.1) Material

1.2) Script

1.3) Effect

1.4) Scene

1.5) Texture

1.6) Sound

1.7) Prefab

1.8) Plugin

Все файлы распределяются в директорию, в соответствии с их назначением.

# 5. Результат работы программы

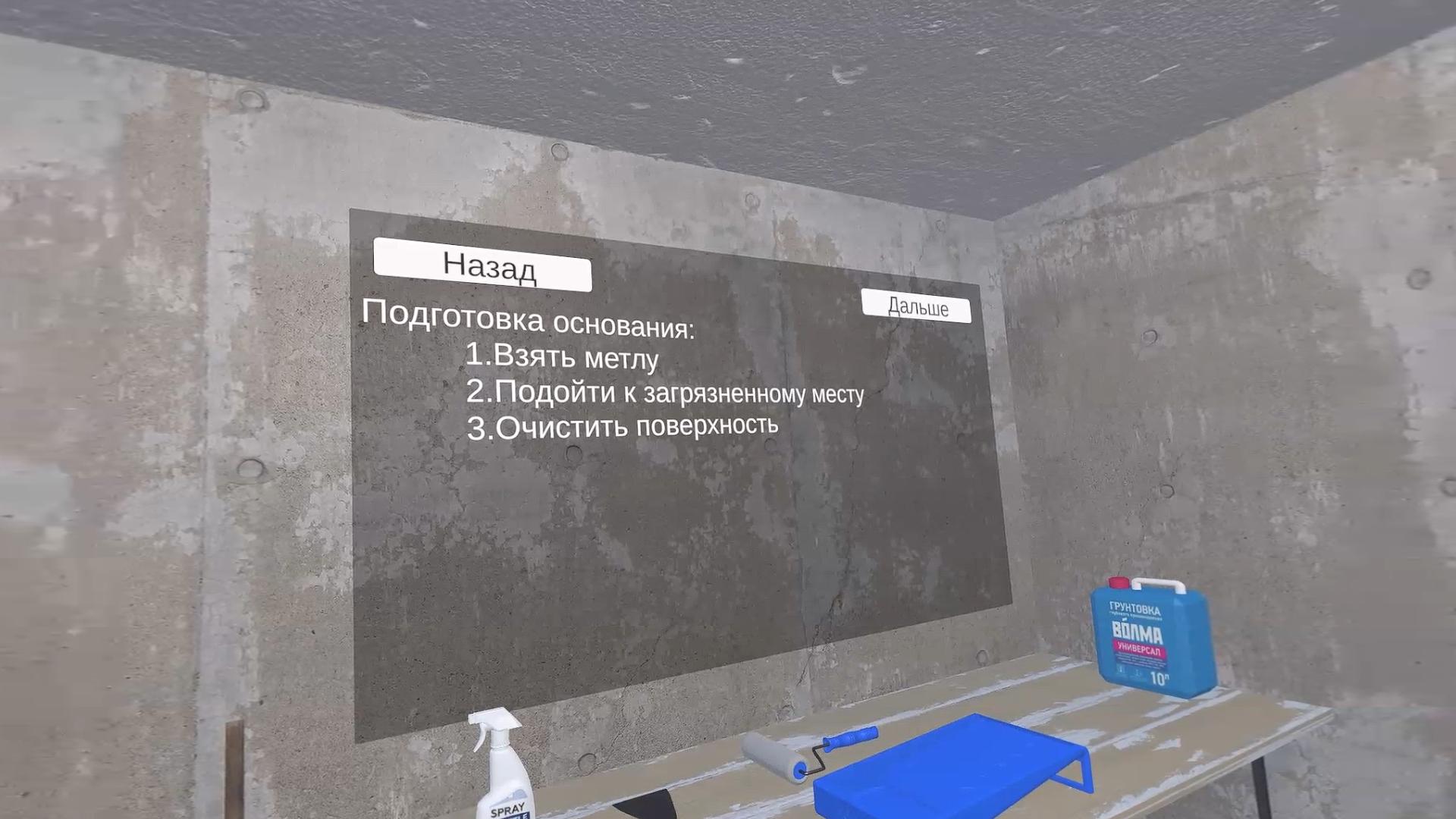
5.1. Меню игры



5.2. Радио (можно менять станцию и громкость)



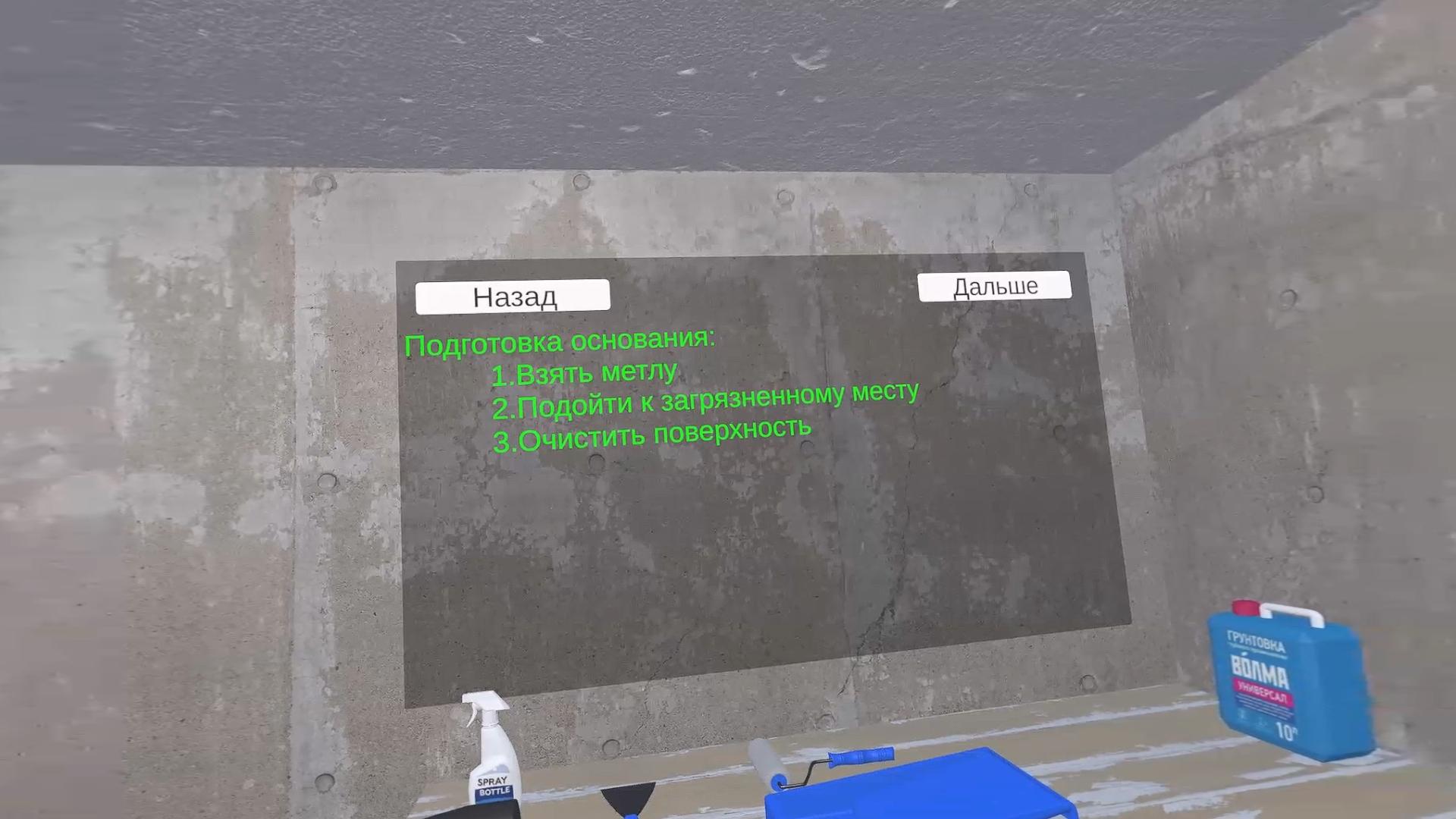
5.3. Интерфейс (показывает квесты)



5.4. Очистка поверхности



5.5. Завершенный квест



5.6. Наполнение ведра



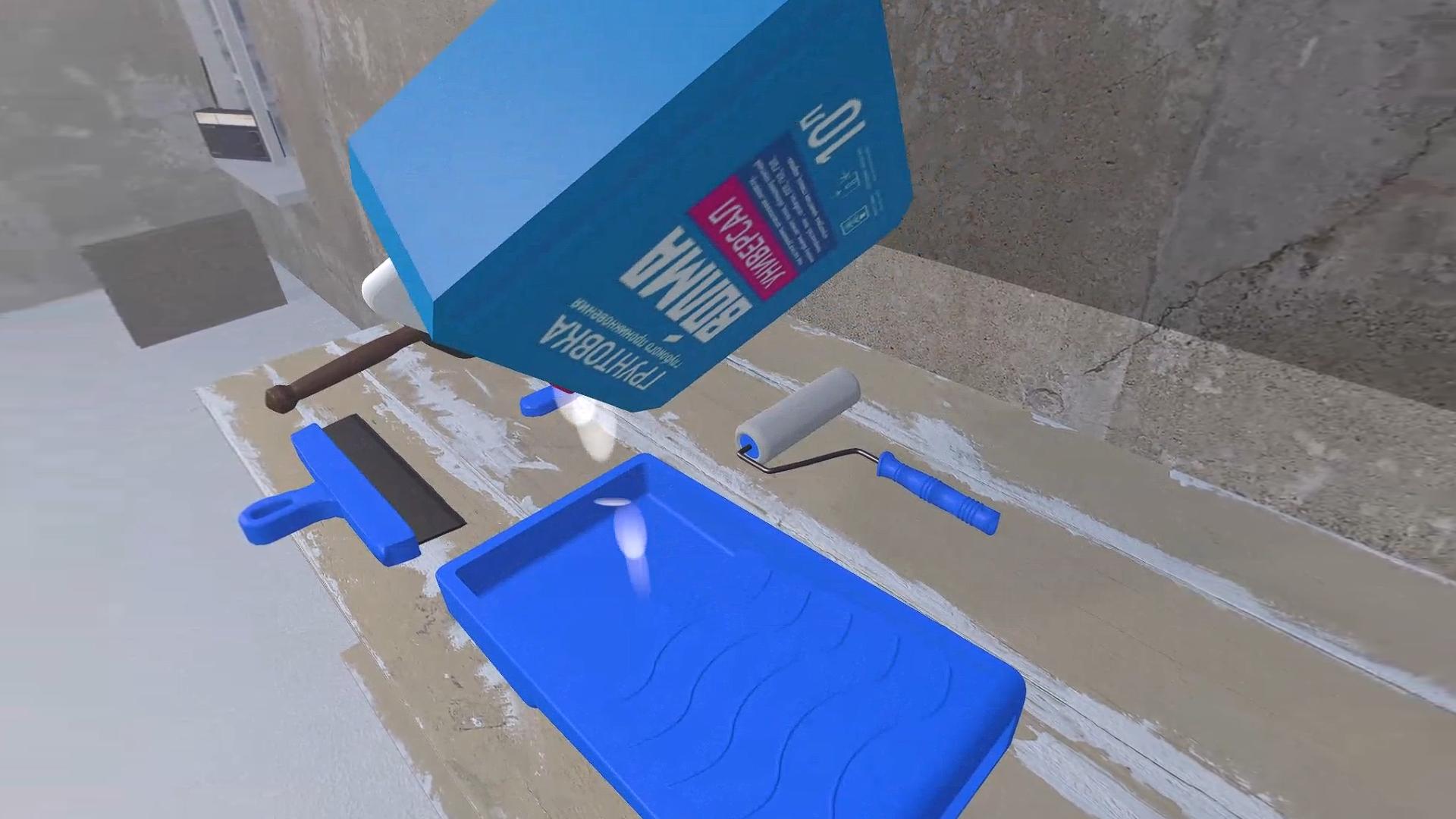
5.7. Засыпание раствора



5.8. Размещенный миксеров смесь



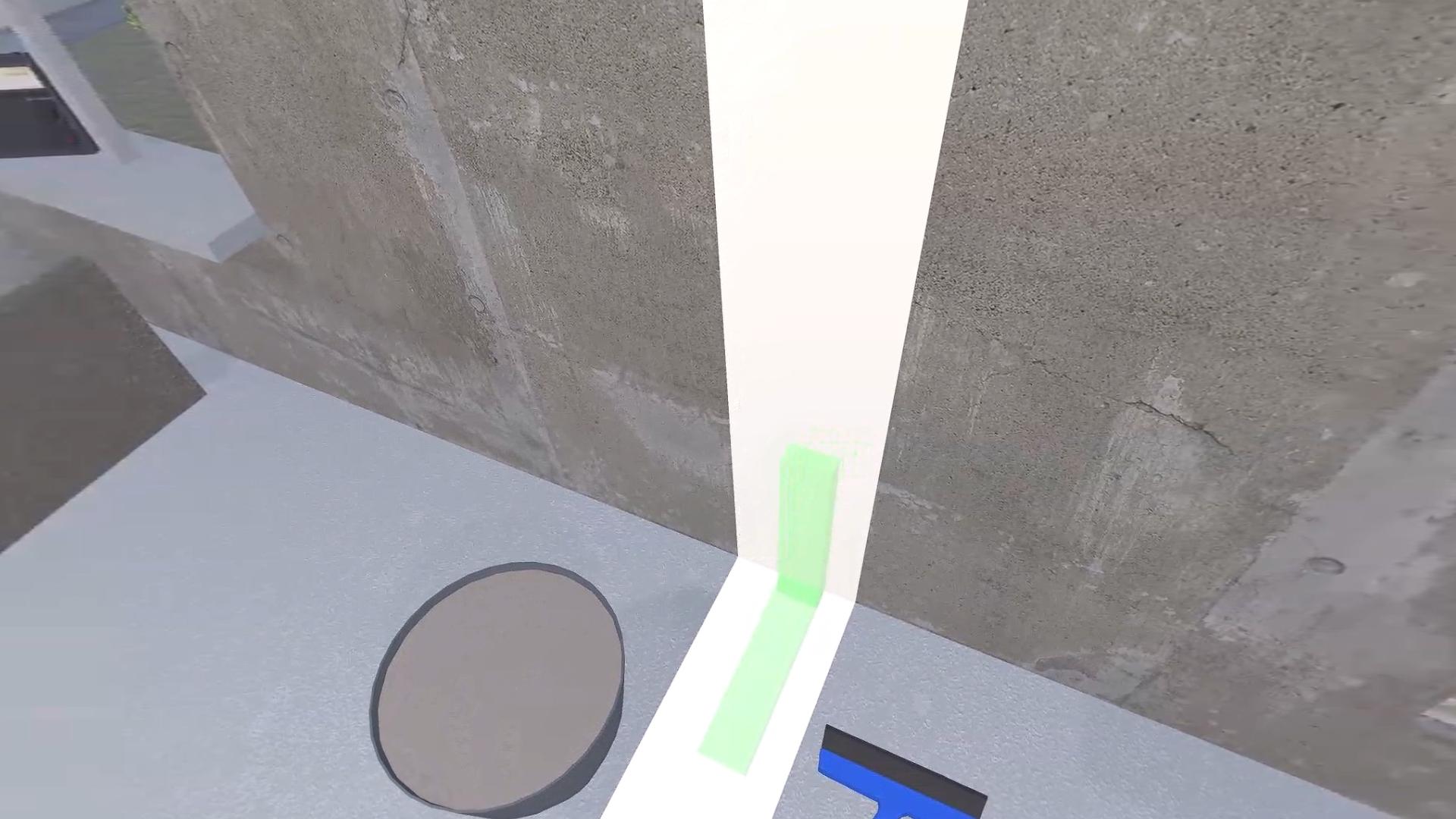
5.9. Заливка грунтовки в кюветку



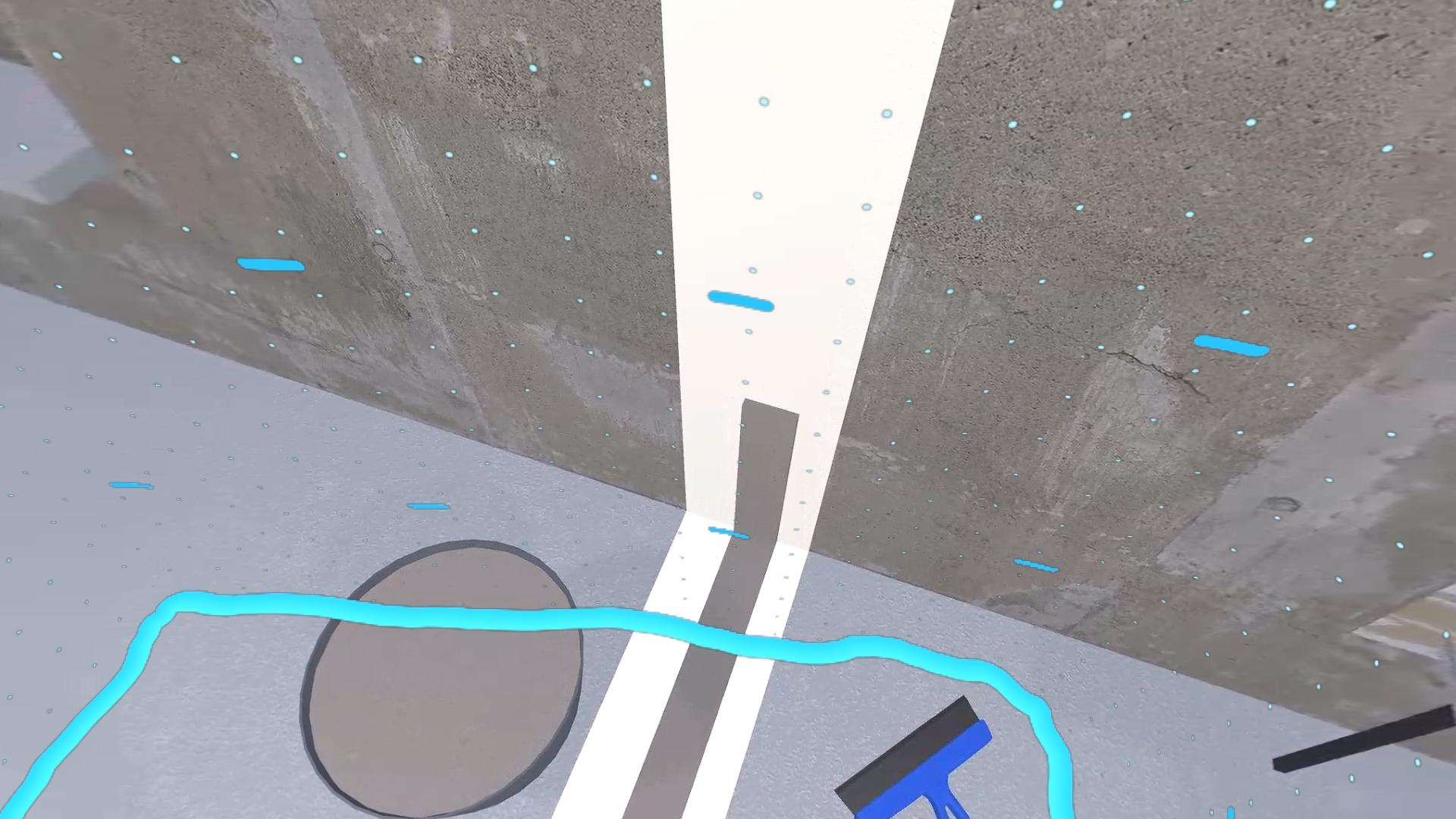
5.10. Грунтовка поверхности



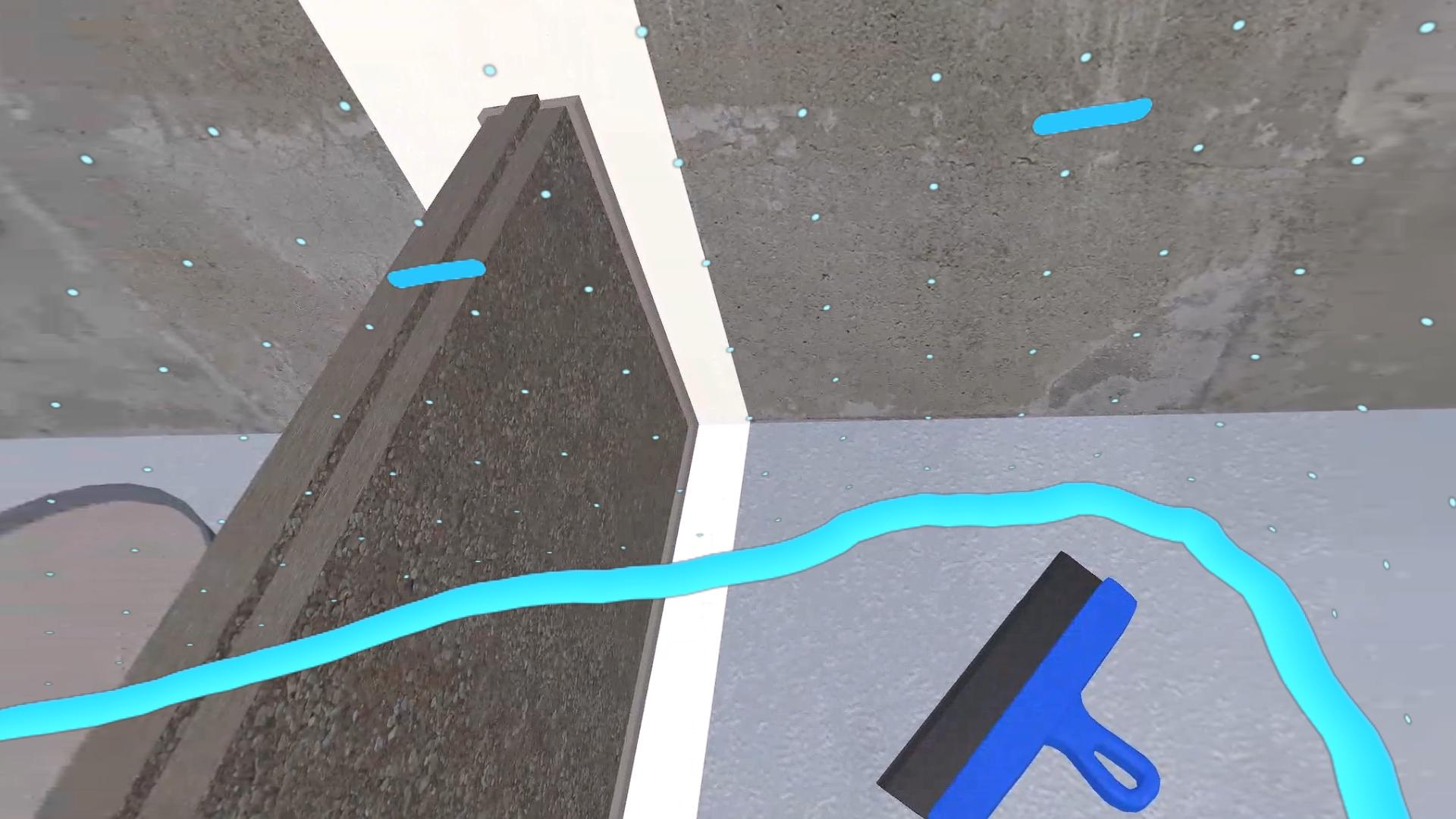
5.11. Теперь можно ставить начать монтаж плит ПГП



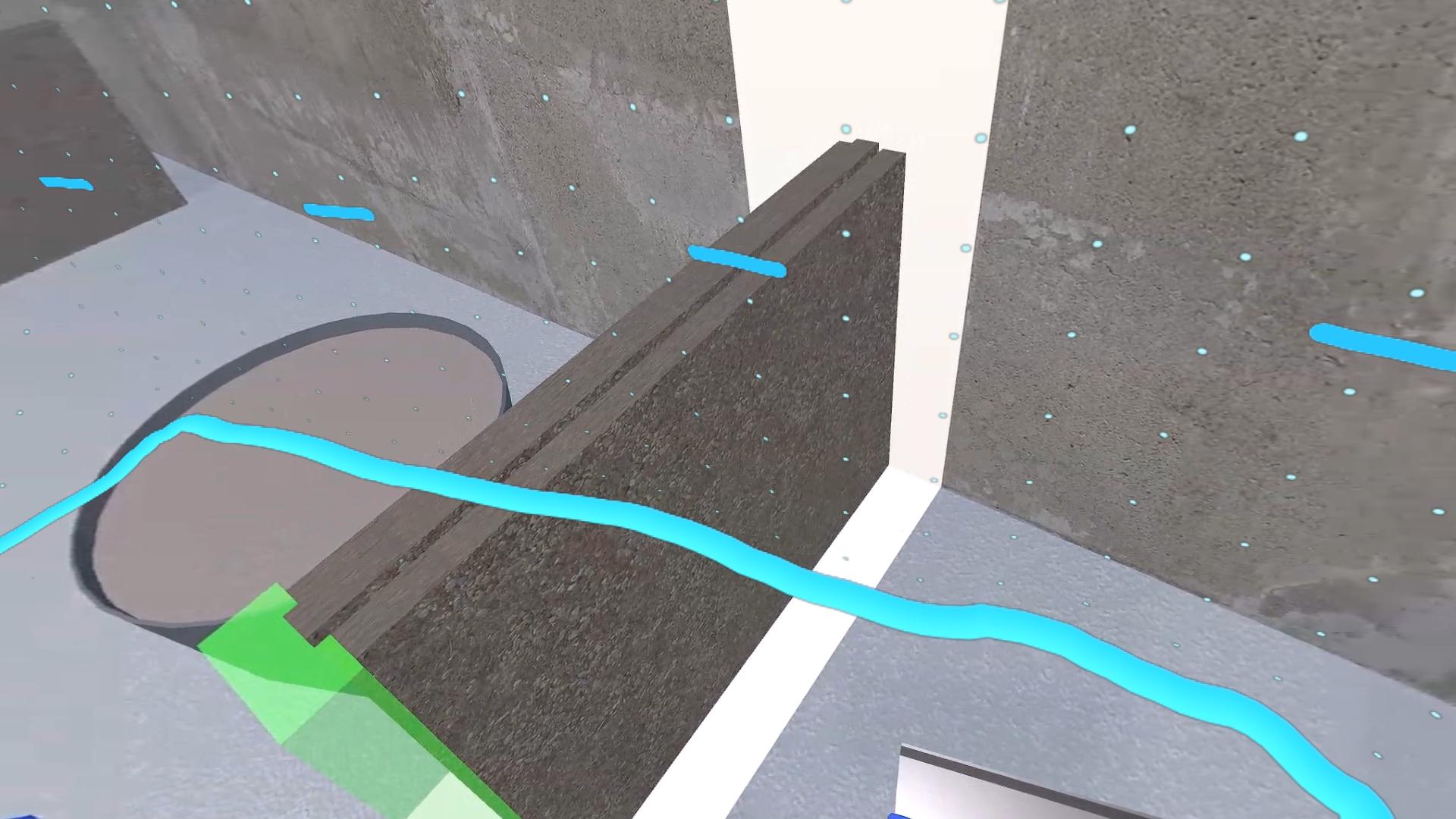
5.12. Клей на поверхности



5.13. Плита установлена



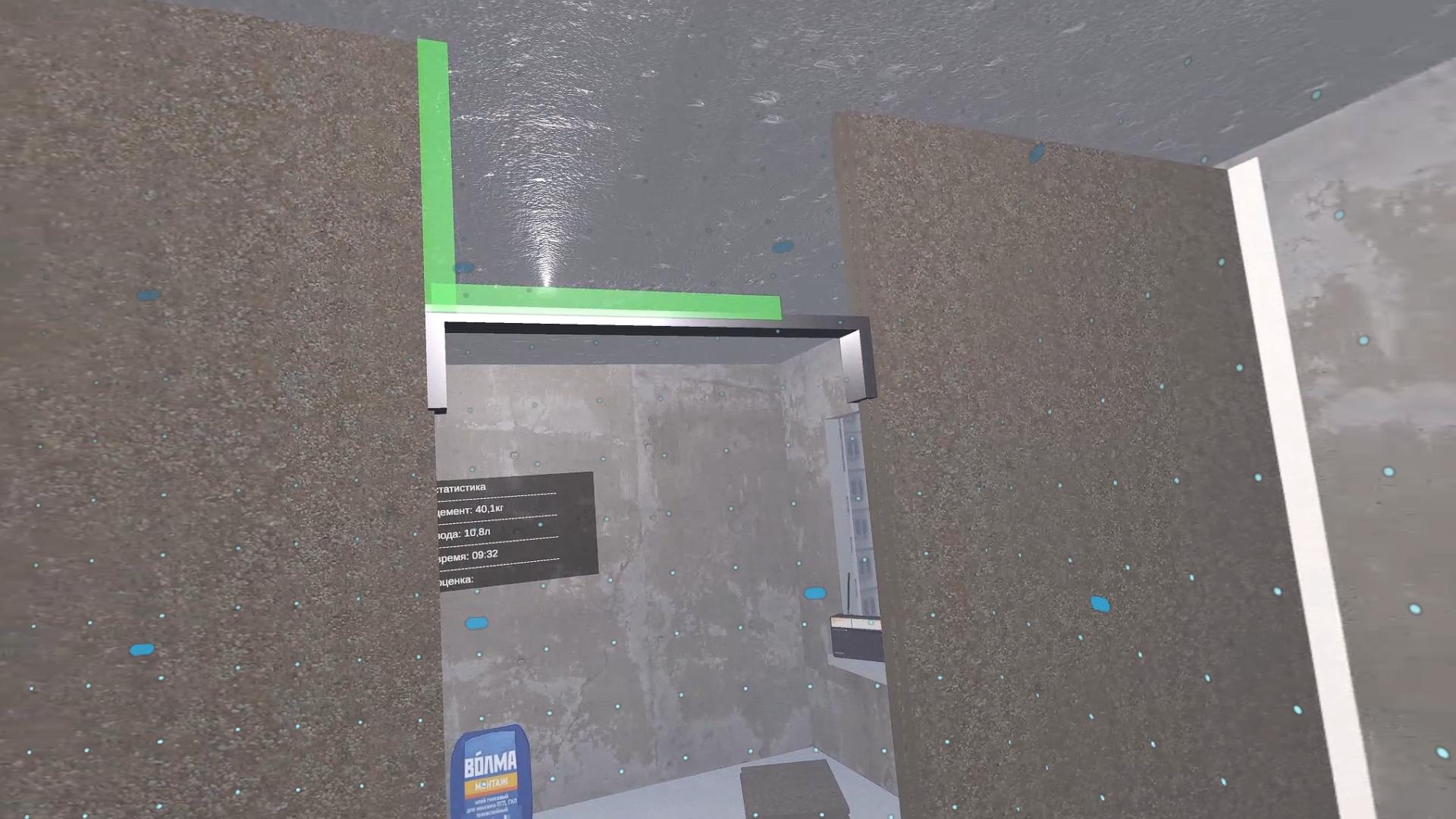
5.14. Убран лишний клей (можно начать монтаж следующей плиты)



5.15. Нужно поставить распорку над перегородкой



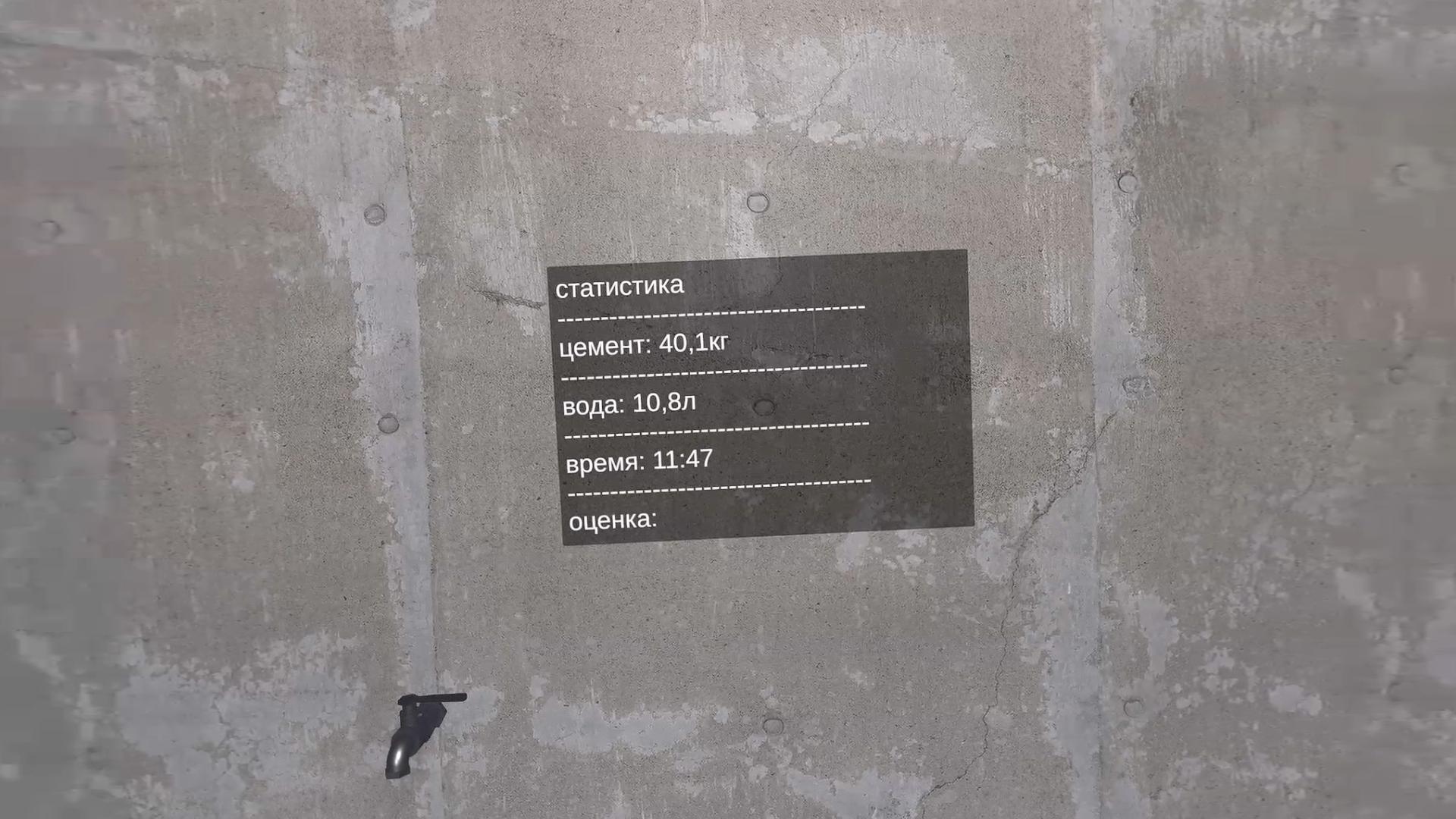
5.16. Распорка установлена



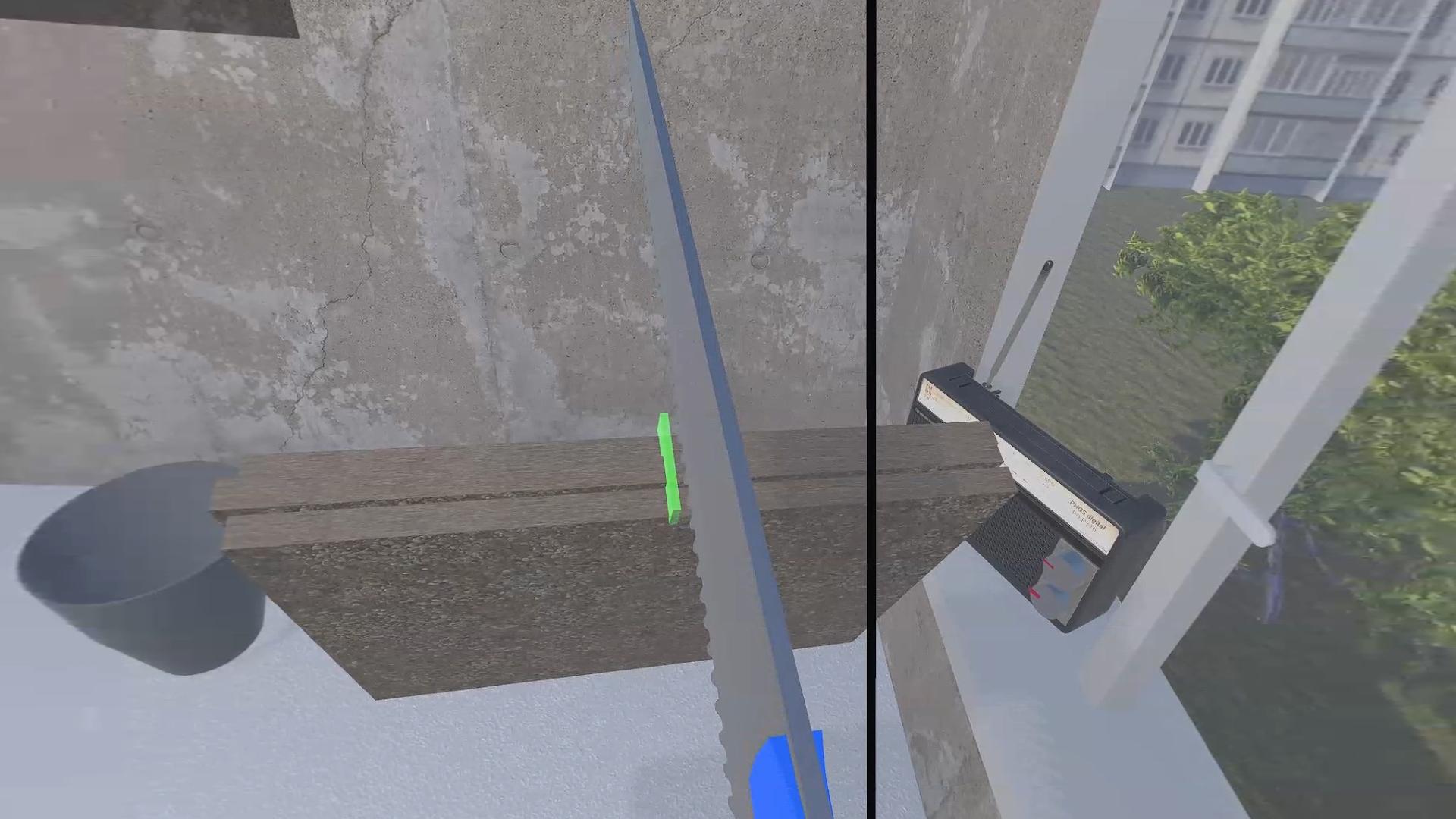
5.17. Стена полностью готова



5.18. Итоговая статистика



5.19. Распил плиты (показывается маркер, где будет надрез)



5.20. Разрезанная на две части плита (пока не имеет функционала)

