

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет прикладной математики, информатики и механики
Кафедра программного обеспечения
и администрирования информационных систем

**Разработка программного обеспечения на основе метода сегментации
зашумленных изображений с плавающим порогом бинаризации
для создания интерактивных неэлектронных поверхностей**

Бакалаврская работа

Направление 09.03.03. Прикладная информатика
Профиль Прикладная информатика в юриспруденции



Допущено к защите в ГЭК

05.06.2021

Зав. кафедрой _____

д. ф.-м. н., проф.

Артемов М. А.

Обучающийся Артём

Принев М. А.

Руководитель И.Е.

д. т. н., проф.

Воронина И. Е.

Воронеж 2021

Аннотация

Основная цель работы — разработать программно-аппаратный комплекс SmartWall, содержащий в себе программное обеспечение, в основе которого лежит метод сегментации зашумленных изображений с плавающим порогом бинаризации, разработанный автором в ходе исследования, и модуль удаленного доступа. Разработка позволит при помощи компьютерного зрения использовать любую поверхность в качестве интерактивного элемента, отказавшись от применения сенсорных технологий.

Использование ПАК SmartWall в образовательных учреждениях, а также в центрах реабилитации инвалидов позволит повысить уровень эффективности и доступности образования, благодаря внедрению интерактивных IT технологий.

Содержание

Введение	5
1. Постановка задачи.....	7
2. Описание базовой технологии.....	8
2.1. Основные технологические тренды	8
2.2. Описание базовой технологии.....	11
3. Требования к программному и аппаратному обеспечению	14
4. Реализация программно-алгоритмических задач	15
4.1. Разработка и оптимизация кода программы	15
4.2. Разработка обучающего и развивающего контента	18
4.3. Разработка программных средств, обеспечивающих совместную работу веб-камеры и одноплатного компьютера Raspberry Pi	19
4.4. Разработка конструктора SmartWall, позволяющего создавать шаблоны.....	21
5. Реализация аппаратно-технических задач	25
5.1. Разработка оптимальной конструкции соединения веб-камеры с одноплатным компьютером в компактном корпусе с целью обеспечения стабильной работы ПО SmartWall.....	25
6. Тестирование программных и аппаратных решений	28
6.1. Тестирование работы метода в различных условиях	28
6.2. Тестирование работы модуля удаленного доступа в реальных условиях, доработка модуля удаленного доступа по результатам тестирования.....	29
Заключение	33
Список литературы	36
Приложение 1. Фрагменты реализации метода сегментации зашумленных изображений с плавающим порогом бинаризации	39

Приложение 2. Фрагменты реализации программы, обеспечивающей совместную работу веб-камеры и одноплатного компьютера Raspberry Pi ...	43
Приложение 3. Фрагменты реализации программы «Конструктор SmartWall»	45
Приложение 4. Документы, подтверждающие интеллектуальную собственность и заинтересованность в продукции проекта	51

Введение

В настоящее время растет востребованность и увеличивается область применения программных продуктов, разработанных на принципах компьютерного зрения (определение объектов на изображениях, интерактивное управление компьютером, датчики движения, системы технического зрения и т д), а также все большую популярность приобретает использование интерактивных поверхностей [1].

Научная новизна предлагаемых в инновационном проекте решений заключается в создании на основе разработанного автором метода *сегментации зашумленных изображений с плавающим порогом бинаризации* программно-аппаратного комплекса SmartWall, с помощью которого можно использовать компьютерное зрение для применения любой поверхности в качестве интерактивного элемента, отказавшись от использования дорогостоящих сенсорных технологий.

Использование метода в программном обеспечении позволяет осуществлять сегментацию заданных областей на изображении, что делает возможным создание интерактивных областей на поверхностях, обладающих различным цветом и текстурой, при различных уровнях освещенности, а также на изображениях на экране проектора, без использования сенсорных технологий, что приведет к повышению уровня эффективности и доступности образования с использованием интерактивных IT технологий. Внедрение разработки является очень актуальным, так как позволит значительно снизить стоимость внедрения информативно-коммуникативных технологий в образовательный процесс за счет отказа от использования дорогостоящих сенсорных поверхностей. Использование неэлектронных поверхностей в качестве интерактивного элемента позволит создать бюджетную альтернативу дорогостоящим электронным приборам, например, сенсорным доскам, а также осуществлять выпуск одноразовых

неэлектронных гаджетов в тех случаях, когда их серийный выпуск является нецелесообразным.

Объектом исследования является процесс разработки и оптимизации программного обеспечения для создания интерактивных неэлектронных поверхностей и гаджетов.

Основной задачей является решение вопросов разработки и оптимизации программных и технических аспектов ПАК SmartWall, содержащего в себе модуль удаленного доступа и программное обеспечение, в которое входит как готовый, так и настраиваемый контент, а именно, тематические пакеты программ и конструктор для формирования собственных приложений и позволяющий использовать компьютерное зрение для применения любой поверхности в качестве интерактивного элемента без использования сенсорных технологий.

1. Постановка задачи

Реализовать программное обеспечение SmartWall для создания интерактивных неэлектронных поверхностей и гаджетов, обеспечив оптимальную работу метода сегментации зашумленных изображений с плавающим порогом бинаризации и аппаратную базу. Для этого требуется решить следующие задачи:

1. Программно-алгоритмические задачи.

- 1.1. Разработать программное обеспечение и оптимизировать показатели производительности.
- 1.2. Разработать развивающий и обучающий контент.
- 1.3. Разработать программные средства, обеспечивающие совместную работу веб-камеры и одноплатного компьютера Raspberry Pi.
- 1.4. Разработать *Конструктор SmartWall*, позволяющий создавать шаблоны.

2. Аппаратно-технические задачи.

- 2.1. Разработать конструкцию соединения веб-камеры с одноплатным компьютером в компактном корпусе с целью обеспечения стабильной работы ПО SmartWall.
- 2.2. Доработать конструкцию модуля удаленного доступа по результатам тестирования.

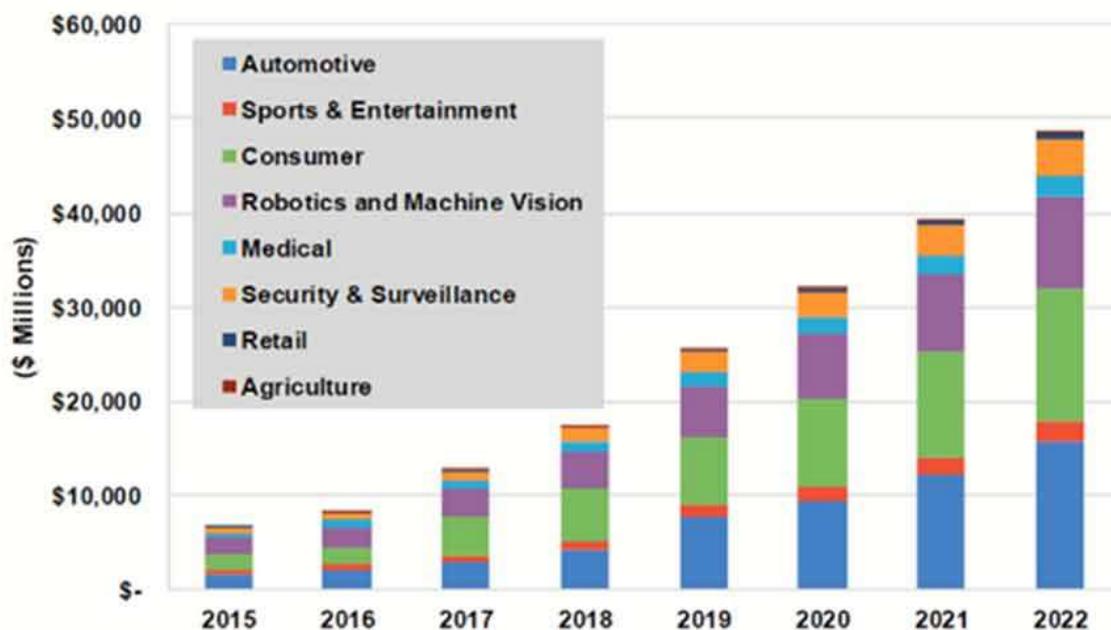
2. Описание базовой технологии

2.1. Основные технологические тренды

Компьютерное зрение (Computer Vision, CV), а также машинное зрение (Machine Vision, MV) — это автоматическая фиксация и обработка изображений, неподвижных и/или движущихся объектов при помощи компьютера. Обычно, системы CV состоят из фото- или видеокамеры, а также компьютера, на котором установлено программное обеспечение для обработки и анализа изображений. В настоящее время компьютерное зрение широко применяется во многих отраслях цифровой экономики, например, «Умный город», беспилотные летательные аппараты – дроны, автономные автомобили и системы помощи водителю (ADAS), прецизионное сельское хозяйство, система здравоохранения и другие сферы применения. CV — динамично развивающаяся область цифровых технологий, которая все чаще используется во многих сторонах повседневной жизни. Точность компьютерной видеоаналитики все время растет и использование компьютерного зрения не только позволяет улучшить качество, но и дает большую экономию средств [13].

В последние годы наблюдается значительный рост диапазона применений компьютерного зрения. В отчете по рынку компьютерного зрения компания Tractica в 2014 году выделила шесть областей применения CV. Спустя несколько лет, в новой версии отчета Tractica указывает уже восемь областей использования компьютерного зрения: все большей популярностью начинает пользоваться Retail (розничная торговля) и Agriculture (сельское хозяйство). Применение компьютерного зрения активно набирает популярность в экономике, социальной сфере и повседневной жизни. На рис. 2.1 представлен прогноз по росту объема рынка CV.

Computer Vision Revenue by Application Market, World Markets: 2015-2022



(Source: Tractica)

Рис. 2.1. Рост объёма рынка CV 2015-2022 гг. [14]

Одними из самых значимых и динамично развивающихся факторов роста использования технологий компьютерного зрения в России являются следующие:

- Развитие национальной программы цифровой экономики, где в разделе «Нейротехнологии и искусственный интеллект» компьютерное зрение прописано отдельным пунктом.
- Рост российских инновационных проектов на рынке систем автоматизации и робототехники.
- Расширение областей применения CV в розничной торговле.
- Инновационные разработки для решений «Умного города», «Безопасного города» и интеллектуальных транспортных систем.
- Цифровизация промышленной отрасли.
- Развитие Интернета Вещей (IoT) и промышленного интернета (IIoT).

Результаты опроса, проведённого в ходе исследования, показывают наиболее перспективные отрасли для развития компьютерного зрения в ближайшем будущем (рис. 2.2).

Наиболее перспективные отрасли для CV



Рис. 2.2. Наиболее перспективные отрасли для развития компьютерного зрения. [13]

ПО SmartWall является инновационным наукоемким продуктом, использующим технологию CV. Планируется использование ПО SmartWall в сегменте потребительского рынка – одной из самых перспективных и быстрорастущих отраслей рынка компьютерного зрения, а также в сфере образования. По предварительным расчетам основные экономические показатели продукта должны показать положительную динамику в ближайшие три года.

2.2. Описание базовой технологии

При использовании компьютерного зрения в программных продуктах необходимо решить вопрос выбора метода сегментации изображения. Одним из основных элементов работы системы технического зрения в автоматизированном режиме является сегментация, т.к. именно на этой стадии обработки объекты выделяются из сцены для дальнейшего распознавания и анализа [1].

Научная новизна предлагаемых в инновационном проекте решений состоит в том, что, на основе разработанного автором метода *сегментации зашумленных изображений с плавающим порогом бинаризации* был разработан программно-аппаратный комплекс SmartWall, с помощью которого можно использовать компьютерное зрение для применения любой поверхности в качестве интерактивного элемента, отказавшись от использования дорогостоящих сенсорных технологий.

Рассмотрим основные принципы работы метода. Сущность алгоритма заключается в том, что при определении порога бинаризации не ищется значение яркости пикселя, удовлетворяющее конкретным условиям, а весь процесс бинаризации происходит непрерывно при постоянном нарастании значения яркости пикселя, до тех пор, пока программа не сможет автоматически выделить, то есть сегментировать заданную область на изображении, после этого изменение порога бинаризации прекращается и фиксируется требуемое для данного изображения значение яркости пикселя, обеспечивающее при использовании его в качестве порога бинаризации надежную сегментацию нужной области на изображении. Использование метода позволяет сегментировать заданную область при изменении заднего фона, структуры материала интерактивной поверхности и уровня освещенности в помещении [5].

Типы интерактивных неэлектронных поверхностей при использовании ПО SmartWall:

- стационарные интерактивные неэлектронные поверхности (алгоритм построен на фиксации координат и характеристик каждой интерактивной области), обладают более высоким быстродействием и надежностью;
- мобильные интерактивные неэлектронные поверхности (алгоритм построен на запоминании характеристик так называемого «якоря», а координаты остальных интерактивных областей определяются опосредованно), имеют более разнообразные варианты использования и качество сегментации не зависит от случайного смещения поверхности.

В качестве интерактивного неэлектронного элемента применять поверхности любого цвета и текстуры при различных уровнях освещенности [5]. Схемы работы программного обеспечения SmartWall для различных поверхностей представлены на рис. 2.3 и рис. 2.4.



Рис. 2.3. Схема работы ПО SmartWall для стационарных интерактивных поверхностей [5]



Рис. 2.4. Схема работы ПО SmartWall для мобильных неэлектронных гаджетов [5]

3. Требования к программному и аппаратному обеспечению

Основные конструктивные и технико-эксплуатационные показатели: для использования ПО SmartWall необходим компьютер ОС Windows или ОС Linux, одноплатный компьютер Raspberry Pi, не ниже модели 3B, веб-камера (разрешение: 1280x720; максимальная частота кадров: 30 Гц). Технические параметры программных продуктов: быстродействие 2–5 с; объем памяти процессора, занимаемый приложением 60–75 МБ.

4. Реализация программно-алгоритмических задач

4.1. Разработка и оптимизация кода программы

В рамках поставленной задачи были проведены работы по оптимизации алгоритмического решения для программного обеспечения, содержащего несколько программных продуктов, в которых использовался метод *сегментации зашумленных изображений с плавающим порогом бинаризации*, позволяющий при помощи компьютерного зрения сегментировать заданные области на изображении, что позволило использовать неэлектронные поверхности в качестве интерактивного элемента без применения сенсорных технологий. Принцип работы метода представлен на рис. 4.1, 4.2.



Рис. 4.1. DFD схема. Диаграмма первого уровня



Рис. 4.2. DFD схема. Диаграмма декомпозиции первого уровня

Приложения, входящие в состав программного обеспечения для создания неэлектронных поверхностей и гаджетов (ПО SmartWall), было реализовано на языке программирования C# в среде Visual Studio 2017. Для корректной работы программ использовалась версия .NET Framework 4.5 [2, 10]. Для эксплуатации программного обеспечения необходим компьютер ОС Windows и обыкновенная веб-камера со средними показателями (Разрешение (видео): 1280x720; максимальная частота кадров: 30 Гц).

Была осуществлена оптимизация алгоритмических решений, содержащихся в коде программы SmartWall, за счет использования на этапе бинаризации изображения многоэтапного алгоритма, который позволяет осуществлять ячеистую обработку изображения при каждом проходе. Оптимизация кода программы привела к увеличению быстродействия программы. Фрагменты кода, реализующие работу метода, приведены в Приложении 1. В результате оптимизации были получены следующие

показатели, которые превосходят аналогичные характеристики первоначального варианта кода программы до 5 раз:

- быстродействие работы программы 2–5 с;
- объем памяти процессора, занимаемый приложением 60–75 МБ.

Скриншоты работы программы по бинаризации заданного количества геометрических элементов представлены на рис. 4.3.



Рис. 4.3. Контрольный пример работы программы по бинаризации заданного количества геометрических элементов

Скриншоты программы по сегментации зрачка глаза представлены на рис. 4.4.

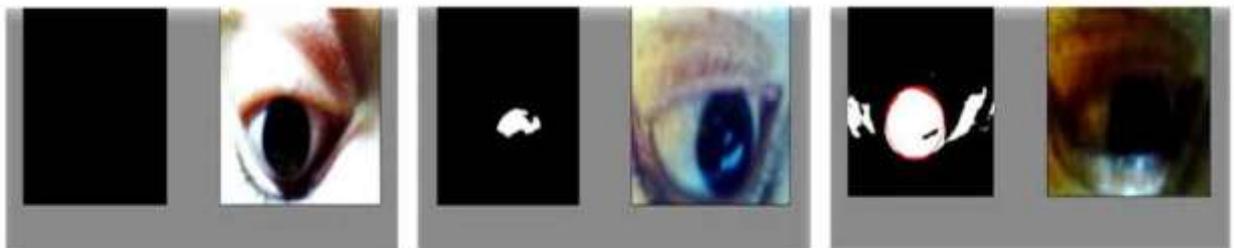


Рис. 4.4. Контрольный пример работы программы по сегментации зрачка глаза

Для корректной работы программ необходимо было найти алгоритмическое решение проблемы случайного перекрытия интерактивной области. Найденное алгоритмическое решение представляет собой два уровня защиты:

- тайминговая защита – специальные таймеры в коде программы, которые позволяют дифференцировать случайное перекрытие с нажатием

на интерактивную зону и не осуществлять предусмотренное действие при случайном перекрытии интерактивной области;

- анатомическая защита – при перекрытии нескольких интерактивных областей одновременно, предусмотренное действие совершается только по результату перекрытия одной области, которая выбирается с учетом анатомических критериев.

4.2. Разработка обучающего и развивающего контента

В настоящее время на основе предлагаемой технологии разработан развлекающий и развивающий контент, а именно:

- приложение «Гномики», позволяющее при нажатии на нарисованные кнопки вызывать движение гномиков на экране монитора, сопровождаемое звуковым эффектом;

- приложение «Домики», позволяющее нажатием на кнопки, нарисованные в два ряда, расположить животных в соответствующих домиках на экране монитора;

- приложение «Угадайка», позволяющее воспроизвести при помощи нажатия на нарисованные кнопки последовательность элементов, возникающих на экране монитора;

- концепт-версия приложения, позволяющего осуществлять сегментацию изображения с различных камер и на различных типах поверхностей, это приложение будет актуально для использования в сенсорных комнатах для детей.

Разработана программа «SmartWallBattle», предназначенная для проведения интерактивных викторин в образовательных учреждениях при помощи проектора и веб-камеры, на изображении на экране проектора выделяются интерактивные кнопки, по нажатию нарисованных кнопок определяется правильность и скорость ответа, а также количество заработанных каждой командой баллов.

Разработано приложение SmartWallBand, позволяющее при помощи кнопок, нарисованных на тканевом браслете, управлять программой Skype на экране монитора. Приложение является примером использования мобильного типа ПО SmartWall, где «якорем» является сравнительный размер или цвет нарисованных кнопок. SmartWallBand предназначен для людей с ослабленным зрением.

В рамках поставленной задачи развлекающий, развивающий и образовательный контент для детей различного возраста, разработанный на основе ПО SmartWall был протестирован в ДКШ «Перспектива» ВИБТ города Воронежа. По результатам тестирования была проведена доработка контента с учетом полученных отзывов, а именно дополнение программного продукта SmartWallBattle новыми слайдами, содержащими материалы по информатике, ориентированные на учеников 5–7 классов. Также получена информация по актуальным элементам содержания контента, которые были использованы при создании конструктора SmartWall.

Документы, подтверждающие тестирование контента в различных организациях, заинтересованных в его использовании, приведены в Приложении 4.

4.3. Разработка программных средств, обеспечивающих совместную работу веб-камеры и одноплатного компьютера Raspberry Pi

Программные средства, обеспечивающие совместную работу веб-камеры и одноплатного компьютера Raspberry Pi, были разработаны на языке программирования Java 8 [2, 11], в среде разработки IntelliJ IDEA, для ОС Linux. Фрагменты кода программы представлены в Приложении 2.

Разработанные программные средства предназначены для видеозахвата изображения с последующей обработкой по методу сегментации зашумленных изображений с плавающим порогом бинаризации,

разработанному автором проекта. Разработанные программные средства выполняют следующие функции:

- Запуск ПО SmartWall на компьютерах, оснащенных ОС Linux.
- Реализация возможности удаленной настройки ПО SmartWall, позволяющей увеличить мобильность системы.
- Обеспечение стабильности и повышение быстродействия ПО SmartWall, не более 3 с при различных уровнях освещения в диапазоне 30-1000 лк.
- Решение эргономических проблем, возникающих из-за ограничений, связанных с малой длиной провода, соединяющего веб-камеру с компьютером.

В ходе разработки экспериментальным путем было установлено, что повышение ресурсоемкости процессов на одноплатном компьютере Raspberry Pi, приводит к его нагреванию и потере до 30 % мощностных характеристик, таких как частота процессора и скорость обработки данных, что ведет к соответствующему уменьшению быстродействия ПО SmartWall. Результатом эксперимента стало принятие решения о целесообразности схемы работы программного обеспечения, представленной на рис. 4.5.



Рис. 4.5. Схема принципа работы модуля

4.4. Разработка конструктора SmartWall, позволяющего создавать шаблоны

В ходе выполнения поставленной задачи был разработан *Конструктор SmartWall*, позволяющий пользователю самостоятельно формировать контент для создания интерактивных неэлектронных поверхностей при помощи ПО SmartWall.

Конструктор SmartWall представляет собой рабочую среду для формирования шаблонов SmartWall. Интерфейс программы имеет четыре рабочие зоны:

- Зона предварительного просмотра формируемого контента;
- Зона настройки создаваемых объектов;
- Зона показа последовательности слайдов;
- Зона формирования неэлектронных кнопок.

Пользователь может формировать свой контент по разработанным готовым шаблонам: презентация и игра. При этом можно выбирать цвет фона, вставлять свой текст и изображения, формировать неэлектронные кнопки и выбирать последовательность действий.

Приложение разработано для ОС Windows на языке программирования C# в среде Visual Studio 2017, для корректной работы программ необходима версия .NET Framework 4.7 [10]. В настоящее время получено Свидетельство о государственной регистрации программы для ЭВМ *Конструктор SmartWall* [19]. Фрагменты кода программы представлены в Приложении 3.

Интерфейс и примеры использования программы представлены на рис. 4.6–4.10.

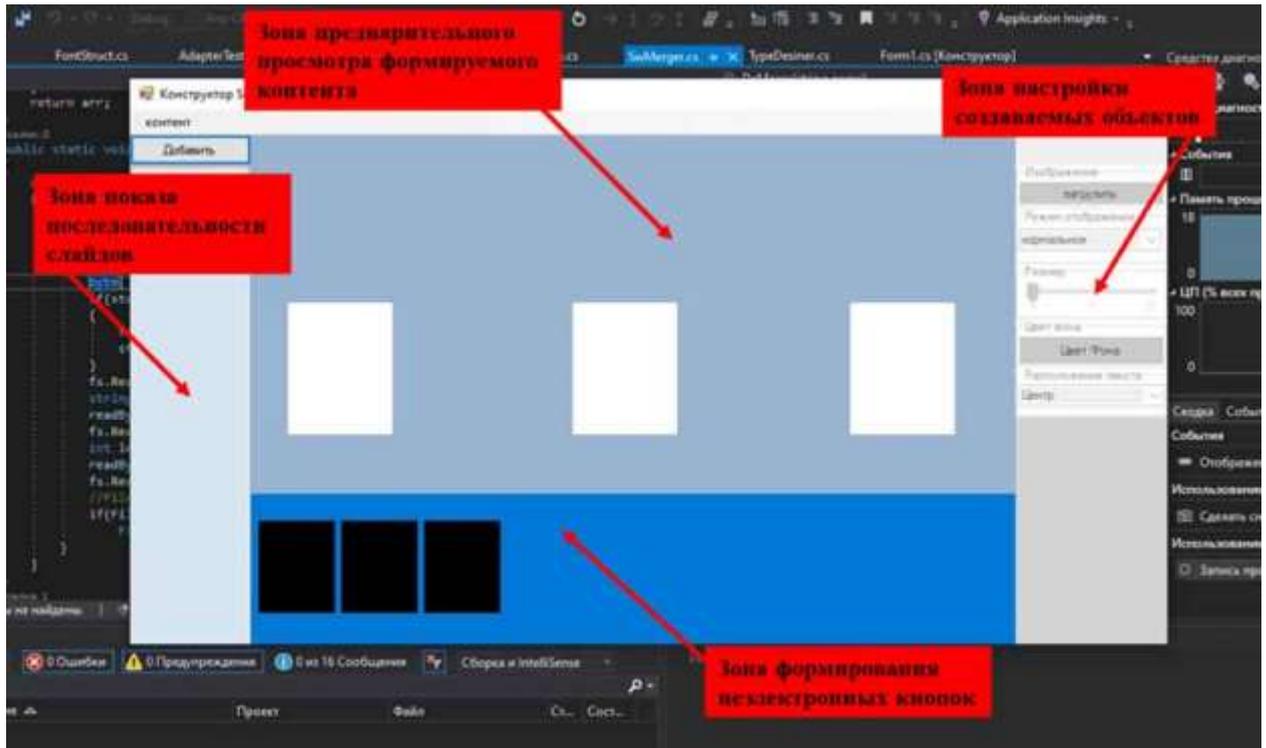


Рис. 4.6. Интерфейс программы

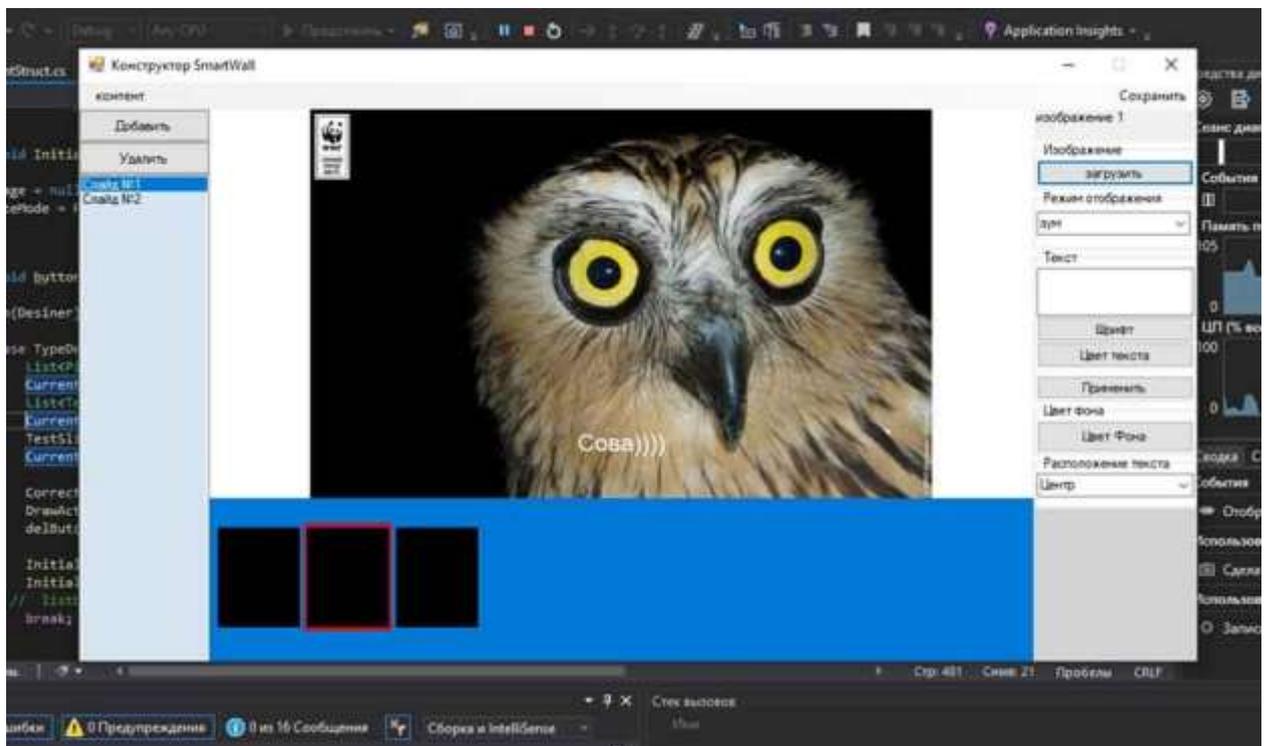


Рис. 4.7. Пример создания слайда презентации



Рис. 4.8. Пример создания шаблона игры

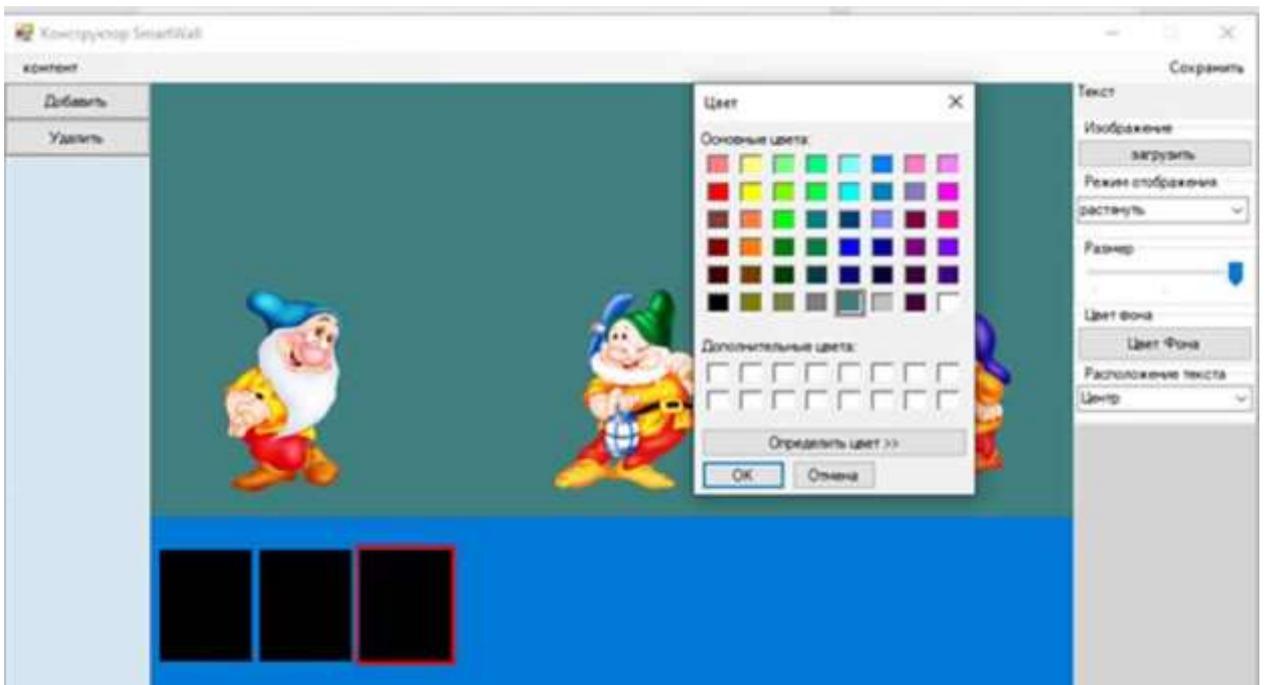


Рис. 4.9. Пример редактирования дизайна шаблона

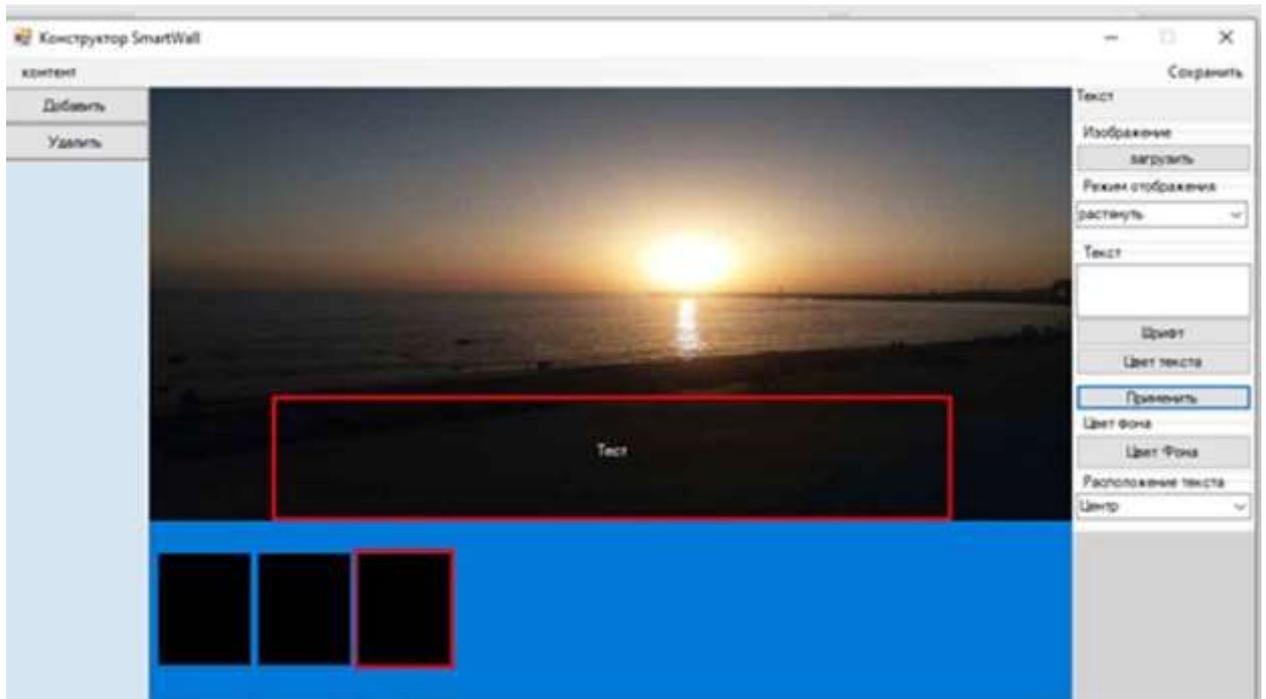


Рис. 4.10. Пример формирования неэлектронных кнопок

5. Реализация аппаратно-технических задач

5.1. Разработка оптимальной конструкции соединения веб-камеры с одноплатным компьютером в компактном корпусе с целью обеспечения стабильной работы ПО SmartWall

В ходе работы над проектом были решены конструктивные задачи соединения веб-камеры с одноплатным компьютером Raspberry Pi для создания модуля удаленного доступа, разработана оптимальная конструкция соединения веб-камеры с одноплатным компьютером в компактном корпусе с целью обеспечения стабильной работы ПО SmartWall.

Первоначальная конструкция мобильного модуля SmartWall включала в себя элементы, представленные в табл. 5.1.

Таблица 5.1. Элементы конструкции модуля SmartWall

№ п/п	Элемент конструкции	Основные характеристики	Статус элемента
1	Raspberry Pi 3 model B	Количество ядер процессора: 4, Частота процессора: 1200 МГц, Размер оперативной памяти: 1024 МБ	Приобретен
2	Блок питания, Micro USB	Выходное напряжение: 5V, Максимальный выходной ток: 2А	Приобретен
3	Camera Module v1	Габаритные размеры: 25 × 24 × 9 мм, Разрешение: 5 Megapixels	Приобретен
4	Карта памяти microSDHC	Объем памяти: 8 ГБ, Класс 10	Приобретен
5	Набор радиаторов	Материал: медь, Габаритные размеры: 15 × 15 × 1 мм, Количество: 3 шт.	Приобретен
6	Корпус	Материал: акрил, Габаритные размеры: 95 × 65 × 35 мм, Наличие посадочных и крепежных отверстий	Разработан
7	Регулируемое крепление камеры	Материал: прессованный картон, h = 3 мм, Габаритные размеры: 40 × 35 × 30 мм	Разработан

Внешний вид мобильного модуля SmartWall в сборе представлен на рис. 5.1–5.3.



Рис. 5.1. Сборка модуля

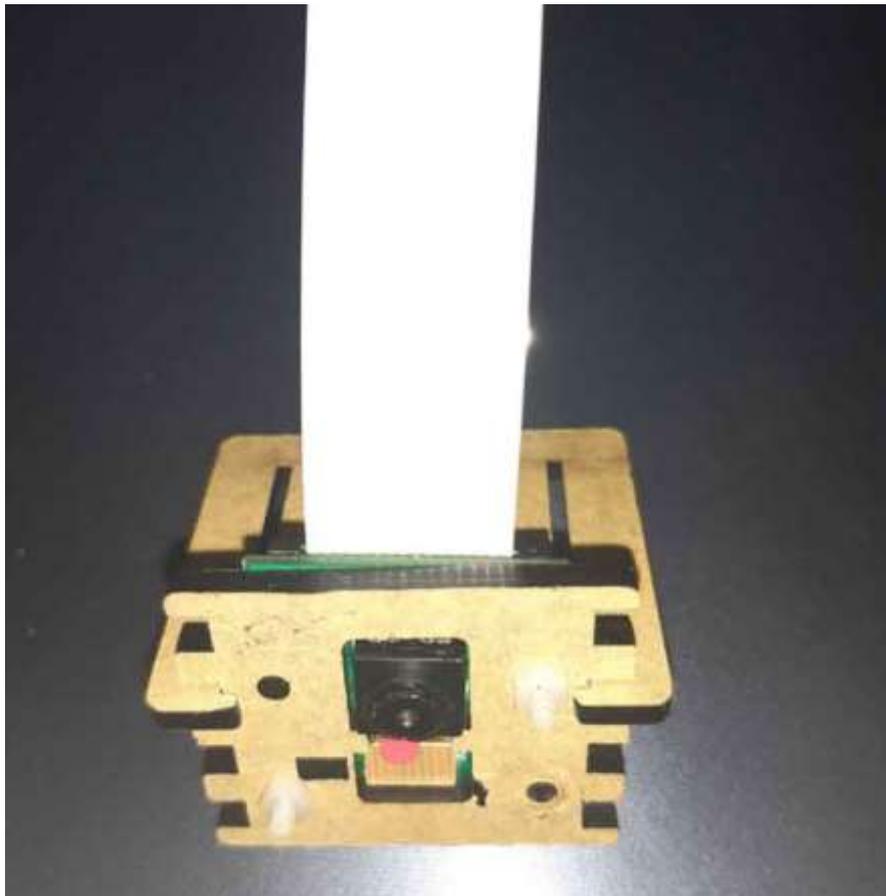


Рис. 5.2. Камера на регулируемом креплении со шлейфом



Рис. 5.3. Мобильный модуль SmartWall

6. Тестирование программных и аппаратных решений

6.1. Тестирование работы метода в различных условиях

Было проведено тестирование работы метода сегментации зашумленных изображений с плавающим порогом бинаризации в различных условиях.

Цель. Определить диапазон быстродействия работы метода и надежности сегментации изображения при различных уровнях освещенности.

Порядок проведения. Провести запуск программы при различных уровнях освещенности, численно оценить соответствующие рабочие параметры, оценить возможность стабильной работы ПО SmartWall в соответствии с нормами освещения учреждений общего образования, начального, среднего и высшего специального образования, детских дошкольных учреждений, а также учреждений досугового назначения.

Результаты. Было экспериментально установлено, что надежная работа метода требует освещенности не менее 50 лк, что является достаточным условием для стабильной работы ПО SmartWall в соответствии с нормами освещения учреждений общего образования, начального, среднего и высшего специального образования, детских дошкольных учреждений, а также учреждений досугового назначения в соответствии с СП 52.13330.2016, а именно не менее 75 лк [12].

Результаты тестирования работы метода сегментации зашумленных изображений с плавающим порогом бинаризации при различном уровне освещенности приведены в табл. 6.1.

Таблица 6.1. Результаты тестирования работы метода

№ п/п	Уровень освещенности, лк	Быстродействие работы метода, с	Надежность сегментации изображения, %
1	менее 30	более 5	70
2	30–50	4–5	85
3	более 50	не более 3	98

6.2. Тестирование работы модуля удаленного доступа в реальных условиях, доработка модуля удаленного доступа по результатам тестирования

Было проведено тестирование работы модуля удаленного доступа в реальных условиях с целью определить уровень влияния нагревания конструкции при работе модуля на основные показатели производительности. Для тестирования использовались программные средства, разработанные автором проекта на языке программирования Python. В процессе тестирования замерялась рабочая температура частота работы процессора при различной нагрузке и продолжительности работы модуля. Работа программы тестирования представлена на рис. 6.1.

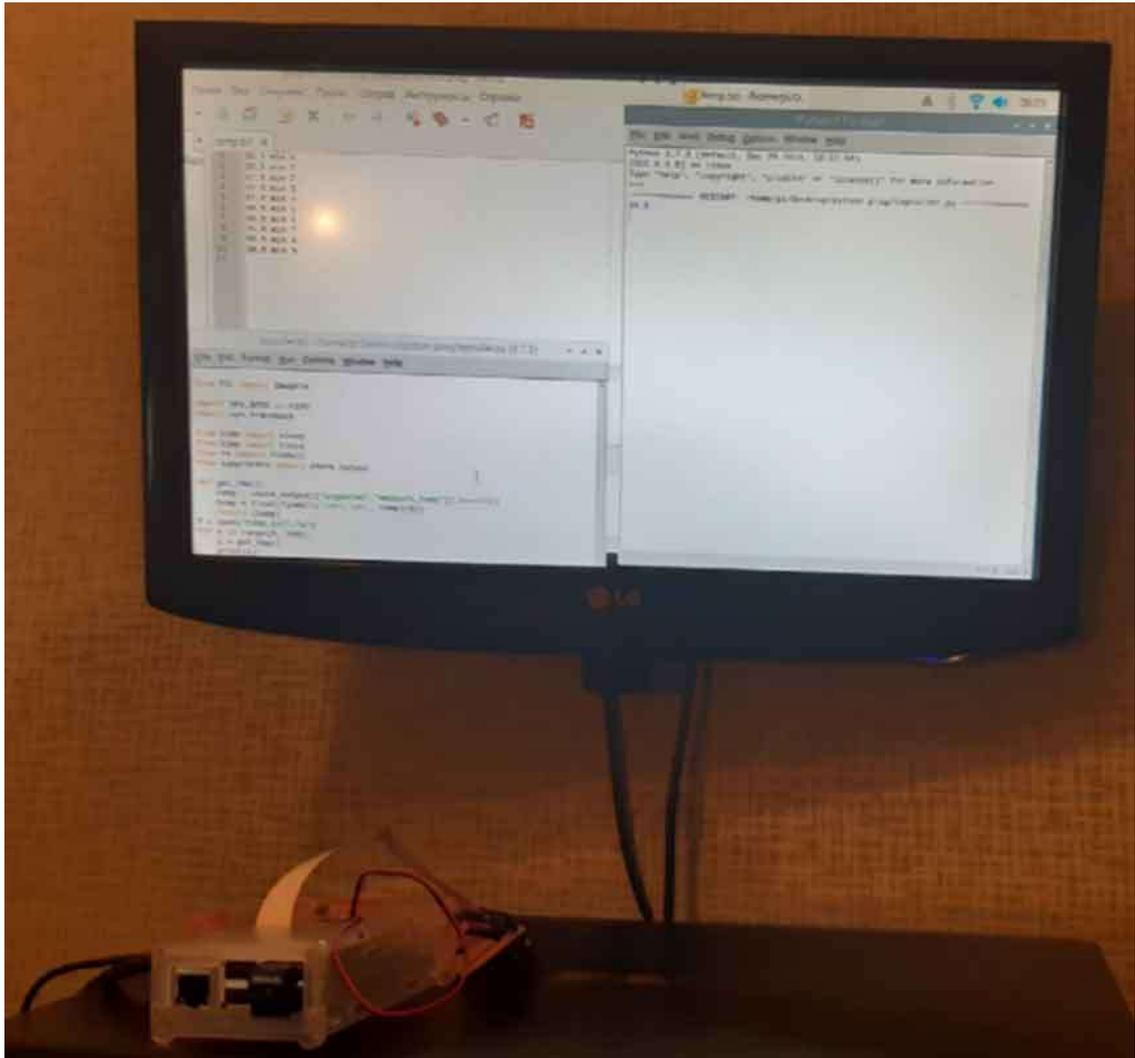


Рис. 6.1. Тестирование модуля

Результаты тестирования модуля SmartWall представлены в табл. 6.2.

Таблица 6.2. Тестирование работы модуля с пассивным охлаждением

Время работы модуля, мин	Частота работы процессора, МГц	Температура ЦПУ, °С
0	1200	38
3	1200	41
6	1100	46
9	900	55
12	850	61
15	730	63

Результаты тестирования показали, что пассивного охлаждения модуля недостаточно для его стабильной и продуктивной работы. По результатам тестирования было принято решение добавить в конструкцию модуля элемент активного охлаждения, а именно процессорный кулер для Raspberry PI 3. Была проведена коррекция конструкции модуля, заключающаяся в расчете и реализации посадочных мест на корпусе модуля в соответствии с конструкцией и крепежными габаритами кулера. После завершения работ по оптимизации конструкции модуля было проведено повторное тестирование, результаты которого представлены в табл. 6.3.

Таблица 6.3. Результаты повторного тестирования работы модуля

Время работы модуля, мин	Частота работы процессора, MHz	Температура ЦПУ, °C
0	1200	38
3	1200	38
6	1200	40
9	1200	43
12	1100	45
15	1100	46

Результаты повторного тестирования показали, что добавление в конструкцию модуля системы активного охлаждения, позволило улучшить тестируемые показатели в среднем на 30%. Конструкция мобильного модуля SmartWall, оптимизированная по результатам тестирования, представлена на рис. 6.2.

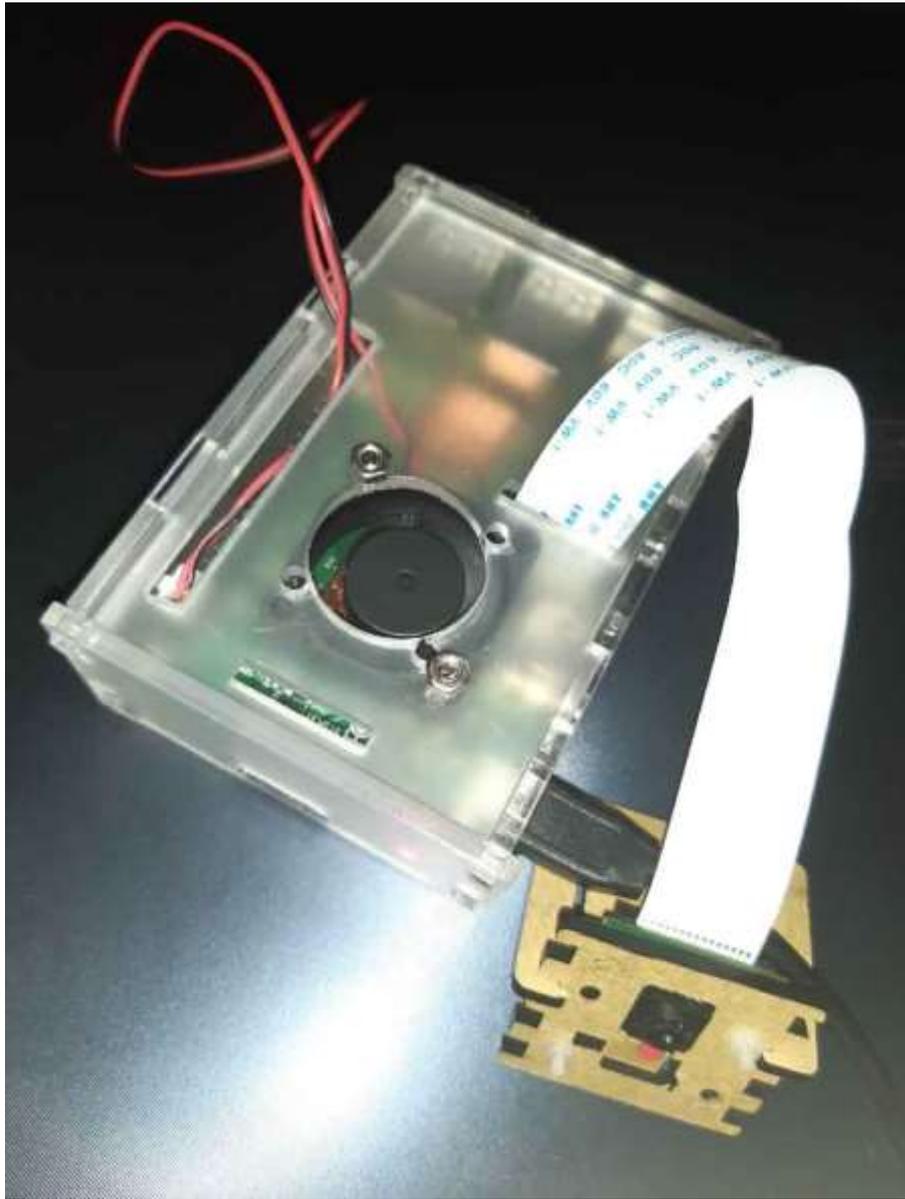


Рис. 6.2. Мобильный модуль SmartWall с системой активного охлаждения

Заключение

Проделанная работа позволила решить основную задачу – разработать программное и аппаратное обеспечение, с помощью которого возможно создавать интерактивные области на поверхностях любого цвета и текстуры, а также изображениях, полученных с помощью проектора, без использования сенсорных технологий. Разработанный автором метод *сегментации зашумленных изображений с плавающим порогом бинаризации*, лежащий в основе ПО SmartWall, позволяет осуществлять надежную сегментацию изображений при различных уровнях освещенности, обеспечивая необходимое быстродействие работы программного обеспечения. ПО SmartWall содержит в себе как готовый, так и настраиваемый контент, а именно, тематические пакеты программ и конструктор для формирования собственных приложений. Аппаратное решение позволяет осуществлять удаленный доступ при помощи одноплатного компьютера Raspberry Pi. Все полученные решения были протестированы и доработаны по результатам тестирования.

В ходе работы были в полном объеме выполнены поставленные задачи:

1. Программно-алгоритмические задачи.
 - 1.1. Разработано программное обеспечение и оптимизированы показатели производительности.
 - 1.2. Разработан развивающий и обучающий контент.
 - 1.3. Разработаны программные средства, обеспечивающие совместную работу веб-камеры и одноплатного компьютера Raspberry Pi.
 - 1.4. Разработан конструктор SmartWall, позволяющий создавать шаблоны.
2. Аппаратно-технические задачи.
 - 2.1. Разработана конструкция соединения веб-камеры с одноплатным компьютером в компактном корпусе с целью обеспечения стабильной работы ПО SmartWall.

2.2. Доработана конструкция модуля удаленного доступа по результатам тестирования.

Проект находится в разработке несколько лет. За это время автором были созданы следующие приложения, разработанные на основе ПО SmartWall, видео работы программ можно посмотреть по ссылке: <https://cloud.mail.ru/public/85nh/1aVqs8YSB>

- Программа, позволяющая бинаризировать и сохранять зашумленное изображение как при загрузке фотографии, так и при видео-захвате с помощью веб-камеры.
- Программа, позволяющая управлять музыкальным плеером при помощи пульта, нарисованного на бумаге.
- Программа, позволяющая печатать текст при помощи клавиатуры, нарисованной на бумаге.
- Концепт-версия программы, позволяющая управлять компьютером при помощи взгляда.
- Приложение SmartWallBand, позволяющее при помощи нарисованных на тканевом браслете кнопок управлять программой Skype на экране монитора.

Получено три Свидетельства о государственной регистрации программы для ЭВМ [15, 16]:

- № 2018611550 «SmartWall» [17];
- № 2020618286 «Пакет SmartWall для различных операционных систем» [18];
- № 2020618747 «Конструктор SmartWall» [19].

Выполнен план НИР по проекту в рамках гранта от ФГБУ «Фонд содействия развитию малых форм предприятий в научно-технической сфере» (Фонд содействия инновациям), полученного автором в качестве победителя программы УМНИК.

В ходе работы над проектом было опубликовано семь статей, посвященных актуальным проблемам разработки [3–9], а также проект был представлен на различных научных форумах и конференциях, таких как Ежегодный саммит молодых ученых и инженеров «Большие вызовы для общества, государства, науки» (Сочи 2018, 2019, 2021), IV Всероссийский форум «Наука будущего — наука молодых» III Международной научной конференции «Наука будущего» (Сочи 2019), Falling Walls Lab Moscow 2019.

В настоящее время пройдена бизнес-акселерация по проекту, составлен бизнес-план, предусматривающий коммерциализацию результатов НИР по проекту, открытие МИП и выход на рынок с продукцией, основанной на ПО SmartWall, в ближайшие три года.

Список литературы

1. Тропченко А.Ю. Методы вторичной обработки изображений и распознавания объектов. Учебное пособие / А. Ю. Тропченко – СПб: СПбГУ ИТМО, 2012. – 52 с.
2. Окулов С. М. Основы программирования / С. М. Окулов. – 5-е изд., испр. – М: БИНОМ. Лаборатория знаний, 2010. – 440с. : ил.
3. Принев М.А. Программный комплекс SmartWall для создания интерактивных поверхностей и гаджетов // Информатика: проблемы, методология, технологии: сб. матер. XVIII международной научной конференции: в 7 т. / под ред. Н.А. Тюкачева, Воронеж, Воронежский государственный университет. – Воронеж: Издательство «Научно-исследовательские публикации» (ООО «Вэлборн»), 2018. – Т.4. – С. 194–198.
4. Принев М.А. Использование ПО SmartWall для решения проблем внедрения ИКТ в образовательную среду // Математика, информационные технологии, приложения: сб. науч. тр. / под ред. А.И. Шашкина; Воронежский государственный университет. – Воронеж: Издательский дом ВГУ, 2018. – С.84–89.
5. Принев М. А. Перспективы использования по SmartWall в образовательной среде / М. А. Принев // Вестник воронежского института высоких технологий. – 2019. – № 1 (28). С. 96–99.
6. Принев. М. А. Оптимизация метода сегментации зашумленных изображений с плавающим порогом бинаризации // Современные проблемы прикладной математики и информатики : сборник трудов научной сессии, Воронеж, 2019 г. / ФГБОУ ВО «Воронежский государственный университет». – Воронеж: Издательско-полиграфический центр «Научная книга», 2019. – С.81–83.
7. Принев М. А. Разработка программного обеспечения на основе метода сегментации зашумленных изображений с плавающим порогом бинаризации

для создания интерактивных неэлектронных поверхностей и гаджетов / М. А. Принев // Тезисы докладов третьей Международной научной конференции «Наука будущего» и четвертого Всероссийского молодежного научного форума «Наука будущего наука молодых» – Сочи, 2019. – С. 10.

8. Принев М.А. Детализация изображений на рентгеновских снимках с помощью плавающего порога бинаризации // Информатика: проблемы, методология, технологии: сб. матер. XXI Международной научно-методической конференции: / под ред. А.А. Зацаринного, Д.Н. Борисова Воронеж, Воронежский государственный университет. – Воронеж: Издательство «Научно-исследовательские публикации» (ООО «Вэлборн»), 2021. – С. 1626–1630.

9. Принев М.А. Разработка программно-аппаратного комплекса SmartWall // Математика, информационные технологии, приложения: сборник трудов Межвузовской научной конференции молодых ученых и студентов; Воронежский государственный университет. – Воронеж: ООО Издательско-полиграфический центр «Научная книга», 2021. – С.191–196.

10. Visual C#. – URL: <https://msdn.microsoft.com/ru-ru/library/kx37x362%28v=vs.120%29> (дата обращения: 05.05.2021).

11. Документация по Java Platform Standard Edition 8. – URL: <https://docs.oracle.com/javase/8/docs/> (дата обращения: 05.05.2021).

12. СП 52.13330.2016 Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95*. – URL: <http://docs.cntd.ru/document/456054197> (дата обращения: 05.05.2021).

13. Компьютерное зрение: технологии, рынок, перспективы. – URL: http://tadviser.ru/index.php/Статья:Компьютерное_зрение:_технологии,_рынок,_перспективы (дата обращения: 05.05.2021).

14. Computer Vision Technology Market to Reach \$33.3 Billion by 2019. – URL: <https://tractica.com/newsroom/press-releases/computer-vision-technology-market-to-reach-33-3-billion-by-2019/> (дата обращения: 05.05.2021).

15. "Гражданский кодекс Российской Федерации (часть четвертая)" от 18.12.2006 N 230-ФЗ (ред. от 18.07.2019). Статья 1225. Охраняемые результаты интеллектуальной деятельности и средства индивидуализации. –

URL:

http://www.consultant.ru/document/cons_doc_LAW_64629/2a4870fda21fdffc70bade7ef80135143050f0b1/ (дата обращения: 05.05.2021).

16. "Гражданский кодекс Российской Федерации (часть четвертая)" от 18.12.2006 N 230-ФЗ (ред. от 18.07.2019). Статья 1261. Программы для ЭВМ. – URL:

http://www.consultant.ru/document/cons_doc_LAW_64629/ce1359ed5b9bd99896d7a496c7887e7c223a2cbs/ (дата обращения: 05.05.2021).

17. Свид. 2018611550 Российская Федерация. Свидетельство об официальной регистрации программы для ЭВМ. SmartWall / М. А. Принев; заявитель и правообладатель ФГБОУ ВО «Воронежский государственный университет» (RU). – №2017662835; заявл. 11.12.17; опубл. 02.02.18, Реестр программ для ЭВМ. – 1 с.

18. Свид. 2020618286 Российская Федерация. Свидетельство об официальной регистрации программы для ЭВМ. Пакет SmartWall для различных операционных систем / М. А. Принев; заявитель и правообладатель Принев Мечислав Александрович (RU). – №2020617321; заявл. 02.07.20; опубл. 22.07.20, Реестр программ для ЭВМ. – 1 с.

19. Свид. 2020618747 Российская Федерация. Свидетельство об официальной регистрации программы для ЭВМ. Конструктор SmartWall / М. А. Принев; заявитель и правообладатель ФГБОУ ВО «Воронежский государственный университет» (RU). – №2020617762; заявл. 14.07.20; опубл. 04.08.20, Реестр программ для ЭВМ. – 1 с.


```

        int colorGray = (int)(cv.R * 0.299 +
        cv.G * 0.587 + cv.B * 0.114);
        Picture.SetPixel(x1, y1, Color.FromArgb(colorGray,
colorGray, colorGray));
    }
}

//поиск кнопок(кругов) на подготовленном изображении
public List<ISW_Button> SegmentCricleAforge(Bitmap Picture)
{
    List<ISW_Button> ListSW = new List<ISW_Button>();
    float x2 = 0, y2 = 0;
    BlobCounter blobCounter = new BlobCounter();
    blobCounter.ProcessImage(BinPicture);
    Blob[] blobs = blobCounter.GetObjectsInformation();
    Graphics g = Graphics.FromImage(BinPicture);
    Pen redPen = new Pen(Color.Red, 7);
    for (int i = 0, n = blobs.Length; i < n; i++)
    {
        List<IntPoint> edgePoints =
blobCounter.GetBlobsEdgePoints(blobs[i]);

        AForge.Point center;
        float radius;
        SimpleShapeChecker shapeChecker = new
SimpleShapeChecker();

        if (shapeChecker.IsCircle(edgePoints, out center, out
radius))
        {
            if (x2 == 0 & y2 == 0)
            {
                x2 = center.X + 25;
                y2 = center.Y + 25;
            }
            if (radius >= 5 & Math.Sqrt(Math.Pow(center.X + x2,
2) + Math.Sqrt(Math.Pow(center.Y + y2, 2))) >= 25)
            {
                g.DrawEllipse(redPen,
                    (int)(center.X - radius),
                    (int)(center.Y - radius),
                    (int)(radius * 2),
                    (int)(radius * 2));
                ListSW.Add(new
SW_CircleBut(Picture.GetPixel((int)center.X, (int)center.Y), center.X,
center.Y));
            }
        }
    }
    return ListSW;
}

//стандартная бинаризация
public void BinarizationAll(ref List<ISW_Button> Buttons,
Bitmap Picture)
{
    for (int x = 0; x < Picture.Width; x++)
    {
        for (int y = 0; y < Picture.Height; y++)

```

```

        {
            int n = Picture.GetPixel(x, y).R;
            if (n <= Threshold)
            {
                BinPicture.SetPixel(x, y, Color.Black);
            }
            else
            {
                BinPicture.SetPixel(x, y, Color.White);
            }
        }
    }

    Buttons = SegmentCricleAforge(BinPicture);
}

//быстрая бинаризация "шахматной доской"
public void FastBinarization(ref List<ISW_Button> Buttons,
    Bitmap Picture, int StepLines = 1, int StepColumns = 1)
{
    BinPicture = new Bitmap(Picture.Width, Picture.Height);
    To_Semitone(ref Picture);
    for (int x = 0; x < Picture.Width; x+= StepLines)
    {
        for (int y = 0; y < Picture.Height; y+=StepColumns)
        {
            int n = Picture.GetPixel(x, y).R;
            if (n <= Threshold)
            {
                BinPicture.SetPixel(x, y, Color.Black);
            }
            else
            {
                BinPicture.SetPixel(x, y, Color.White);
            }
        }
    }
    Buttons = SegmentCricleAforge(BinPicture);
}

//конструктор
public Binarization(int _Threshold, int _Width, int _Height)
{
    Threshold = _Threshold;
    BinPicture = new Bitmap(_Width, _Height);
}

//быстрая бинаризация сегментами
public void SegmentBinarization(ref List<ISW_Button> Buttons,
    Bitmap Picture, Tools.SegmentScreen SegmentScreen)
{
    Rectangle segment = Tools.Translation(SegmentScreen,
    Picture.Width, Picture.Height);
    Picture = Picture.Clone(segment, Picture.PixelFormat);
    BinPicture = new Bitmap(Picture.Width, Picture.Height);
    To_Semitone( ref Picture);
    for (int x = segment.X; x < segment.Width; x++)
    {
        for (int y = segment.Y; y < segment.Height; y++)
        {
            int n = Picture.GetPixel(x, y).R;
            if (n <= Threshold)

```

```
        {
            BinPicture.SetPixel(x, y, Color.Black);
        }
        else
        {
            BinPicture.SetPixel(x, y, Color.White);
        }
    }
}
Buttons = SegmentCricleAforge(Picture);
}
}
```

Приложение 2.

Фрагменты реализации программы, обеспечивающей совместную работу веб-камеры и одноплатного компьютера Raspberry Pi

```

package SmartWall.Cams;

import java.awt.image.BufferedImage;
import java.util.ArrayList;

/*
 * интерфейс для работы камеры*/

public interface IControlCam {
    BufferedImage GetFrame();
    boolean SelectCam(int index);
    ArrayList<String> GetCams();
    String GetCurrentCam();
    void Start();
    void Stop();
    void Resume();
    StatusCam Status();
}

package SmartWall.Cams;

import com.hopding.jrpcam.RPiCamera;
import com.hopding.jrpcam.enums.Exposure;
import com.hopding.jrpcam.exceptions.FailedToRunRaspistillException;

import java.awt.image.BufferedImage;
import java.io.IOException;
import java.util.ArrayList;

//класс для камеры RPi3
//наследник от интерфейса IControlCam
public class WebCamRPi implements IControlCam {

    //камера
    private RPiCamera camera;
    //кадр с камеры
    private BufferedImage frame;
    //конструктор
    public WebCamRPi() throws FailedToRunRaspistillException {
        camera = new RPiCamera("/home/pi/Pictures");
        //настройка камеры
        camera.setWidth(500);
        camera.setHeight(500);
        camera.setExposure(Exposure.AUTO);
        camera.setAddRawBayer(true);
    }

    //метод получения кадра
    @Override
    public BufferedImage GetFrame() {
        try {
            return camera.takeBufferedStill();
        } catch (IOException e) {

```

```

        e.printStackTrace();
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
    return null;
}

//метод выбора камеры
@Override
public boolean SelectCam(int index) {
    if(index == 0)
        return true;
    return false;
}

//получение списка камер
@Override
public ArrayList<String> GetCams() {
    ArrayList<String> list = new ArrayList<>();
    list.add("camera 0");
    return list;
}

//вернуть текущую камеру
@Override
public String GetCurrentCam() {
    return "camera 0";
}
private StatusCam statusCam = StatusCam.STOP;

//получить статус камеры
@Override
public StatusCam Status()
{
    return statusCam;
}

//запуск камеры
@Override
public void Start() {
    statusCam = StatusCam.START;
}

//остановка камеры
@Override
public void Stop() {
    statusCam = StatusCam.STOP;
}

@Override
public void Resume() {
    switch (statusCam) {
        case STOP:
            statusCam = StatusCam.START;
            break;
        case START:
            statusCam = StatusCam.STOP;
    }
}
}
}

```

Приложение 3.

Фрагменты реализации программы «Конструктор SmartWall»

```

using SWDesigner.Elements;
using System;
using System.Collections.Generic;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace SWDesigner
{
    [Serializable]
    public class SWPage
    {
        public TypeDesiner TypeSW { get; set; }
        public Color BackColor { get; set; }
        public ConsoleColor BgColor { get; set; }
        public List<Picture> pictures = new List<Picture>();
        public List<TestSlide> testSlides = new List<TestSlide>();
        public List<Text> texts = new List<Text>();
        public bool UpdatePictures(string pictureBoxName, int SizeMode)
        {
            foreach (Picture pic in pictures)
            {
                if(pic.picName == pictureBoxName)
                {
                    pic.SizeMode = SizeMode;
                    return true;
                }
            }
            return false;
        }
        public bool UpdatePictures(string pictureBoxName, int Width,int
Height,int Left,int Top)
        {
            foreach (Picture pic in pictures)
            {
                if (pic.picName == pictureBoxName)
                {
                    pic.Width = Width;
                    pic.Height = Height;
                    pic.Left = Left;
                    pic.Top = Top;
                    return true;
                }
            }
            return false;
        }
        public bool UpdateText(Text tx)
        {
            foreach (Text t in texts)
            {
                if (t.Equals(t,tx))
                {
                    t.LeftText = tx.LeftText;
                }
            }
        }
    }
}

```

```

        t.TextColor = tx.TextColor;
        t.TextFont = tx.TextFont;
        t.TextLabel = tx.TextLabel;
        t.TopText = tx.TopText;
        t.BackColor = tx.BackColor;
        return true;
    }
}
return false;
}
public List<Picture> ExtractPicture()
{
    List<Picture> pic = new List<Picture>();
    foreach (Picture p in pictures)
    {
        pic.Add(p);
    }
    foreach (Picture p in pic)
    {
        pictures.Remove(p);
    }
    return pic;
}
public List<Text> ExtractText()
{
    List<Text> tx = new List<Text>();
    foreach (Text t in texts)
    {
        tx.Add(t);
    }
    foreach (Text t in tx)
    {
        texts.Remove(t);
    }
    return tx;
}
public void AddTextSlide(TestSlide ts)
{
}
public void AddText(Text text)
{
    if(!UpdateText(text))
    {
        texts.Add(text);
    }
    return;
}
public string AddPicture(Picture p)
{
    foreach(Picture pic in pictures)
    {
        if(pic.picName == p.picName)
        {
            File.Delete(Environment.CurrentDirectory + "/tmp/" +
pic.FileName);
            return pic.FileName;
        }
    }
    pictures.Add(p);
    return p.FileName;
}

```

```

    }
}

using System;
using System.Collections;
using System.Collections.Generic;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace SWDesigner.Elements
{
    public class Picture:IEqualityComparer
    {
        public int Left { get; set; }
        public int Top { get; set; }
        public string picName { get; }
        public int Width { get; set; }
        public int Height { get; set; }
        public int SizeMode { get; set; }
        public string FileName { get; }
        private static int num = 0;
        public void UpdatePicture(Picture p)
        {
            Width = p.Width;
            Height = p.Height;
            SizeMode = p.SizeMode;
        }

        public new bool Equals(object x, object y)
        {
            Picture p1 = (Picture)x;
            Picture p2 = (Picture)y;
            return p1.picName == p2.picName;
        }

        public int GetHashCode(object obj)
        {
            throw new NotImplementedException();
        }

        public Picture()
        {
        }

        public Picture(int Width,int Heigth,int SizeMode,string Type,string
picName,int Top,int Left)
        {
            this.Width = Width;
            this.Height = Heigth;
            this.SizeMode = SizeMode;
            FileName = "P"+num.ToString()+Type;
            this.picName = picName;
            this.Left = Left;
            this.Top = Top;
            num++;
        }
    }
}

```

```

using System;
using System.Collections.Generic;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace SWDesigner.Elements
{
    public class TestSlide
    {
        public string NameSlide { get; set; }
        private int Number;
        public Picture PictureSlide { get; set; }
        public Text TextSlide { get; set; }
        private static int num = 0;
        public int CorrectAnswer { get; set; }
        public TestSlide()
        {
        }

        public TestSlide(Picture PictureSlide, Text TextSlide, int
CorrectAnswer)
        {
            this.PictureSlide = PictureSlide;
            this.TextSlide = TextSlide;
            NameSlide = "Слайд №" + num.ToString();
            Number = num;
            num++;
            this.CorrectAnswer = CorrectAnswer;
        }
    }
}

```

```

using System;
using System.Collections;
using System.Collections.Generic;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace SWDesigner.Elements
{
    public class Text: IEqualityComparer
    {
        public string LabelName { get; }
        public int LeftText { get; set; }
        public int TopText { get; set; }
        public int WText { get; set; }
        public int HText { get; set; }
        public Color BackColor { get; set; }
        public string TextLabel { get; set; }
        public Color TextColor { get; set; }
        public FontStruct TextFont { get; set; }
        public Text()
        {
        }
    }
}

```

```

        public Text(int LeftText,int TopText,Color BColor,string TextLabel,
        Color TextColor, Font TextFont,string LabelName,int WText,int HText)
        {
            this.LeftText = LeftText;
            this.TopText = TopText;
            this.BackColor = BackColor;
            this.TextLabel = TextLabel;
            this.TextColor = TextColor;
            FontStruct fs = new FontStruct();
            fs.FontFamily = TextFont.FontFamily.Name;
            fs.EmSize = TextFont.Size;
            fs.FontStyle = TextFont.Style;

            this.LabelName = LabelName;
            this.WText = WText;
            this.HText = HText;
        }

        public new bool Equals(object x, object y)
        {
            Text t1 = (Text)x;
            Text t2 = (Text)y;
            return t1.LabelName == t2.LabelName;
        }

        public int GetHashCode(object obj)
        {
            throw new NotImplementedException();
        }
    }
}

using System;
using System.Collections.Generic;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace SWDesigner.Elements
{
    public struct FontStruct
    {
        public string FontFamily { get; set; }
        public float EmSize { get; set; }
        public FontStyle FontStyle { get; set; }
    }
}

using SWDesigner.Elements;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace SWDesigner.Adapter
{
    public class AdapterTestSlide
    {
        public string Name { get; set; }
    }
}

```

```
public TestSlide testSlide { get; set; }
public AdapterTestSlide(TestSlide testSlide)
{
    this.testSlide = testSlide;
    Name = testSlide.NameSlide;
}
}
```

Приложение 4.

Документы, подтверждающие интеллектуальную собственность
и заинтересованность в продукции проекта



РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2020618286

Пакет SmartWall для различных операционных систем

Правообладатель: *Принев Мечислав Александрович (RU)*

Автор: *Принев Мечислав Александрович (RU)*

Заявка № 2020617321

Дата поступления 02 июля 2020 г.

Дата государственной регистрации
в Реестре программ для ЭВМ 22 июля 2020 г.



Руководитель Федеральной службы
по интеллектуальной собственности

Г.П. Ивлиев Г.П. Ивлиев

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2020618747

Конструктор SmartWall

Правообладатель: *федеральное государственное бюджетное образовательное учреждение высшего образования «Воронежский государственный университет» (ФГБОУ ВО «ВГУ»)* (RU)

Автор: *Принев Мечислав Александрович* (RU)

Заявка № 2020617762

Дата поступления 14 июля 2020 г.

Дата государственной регистрации

в Реестре программ для ЭВМ 04 августа 2020 г.



Руководитель Федеральной службы
по интеллектуальной собственности

Г.П. Ивлиев



В образовательном процессе Детской компьютерной школы «Перспектива» Воронежского института высоких технологий – автономной некоммерческой образовательной организации высшего образования г. Воронежа был протестирован развивающий и развлекательный контент, разработанный Приневым Мечиславом Александровичем на основе ПО SmartWall.

По результатам тестирования предоставленные программные продукты (компьютерные приложения «Гномики», «Домики», «Угадайка», SmartWallBattle), разработанные на основе ПО SmartWall рекомендованы к использованию в качестве развивающего и развлекательного контента для детей младшего и среднего школьного возраста.

Достоинствами ПО SmartWall являются небольшие вложения, необходимые для внедрения программных продуктов, а также возможность развивать у детей необходимые навыки с помощью предоставленного контента в интерактивном формате.

Автору было предложено внести дополнения в предоставленный контент, а также был утвержден план дальнейшего сотрудничества.

Руководитель ДПО ВИВТ



С.В. Орлова

Руководитель ДКШ «Перспектива» ВИВТ

О.О. Григоренко



ООО Центр Прикладных Исследований «Проектирование и разработка информационных систем»

ИНН/КПП: 3664203719/366401001 ОГРН 1153668002400 ОКПО 34023573
 р/счет 40702810513000005649 в Центральном-Черноземном банке Сбербанка РФ г. Воронеж
 к/счет 30101810600000000681 в ГУ ГРКЦ г. Воронежа БИК 042007681
 394055, г. Воронеж, ул. Легочка Демьянова 19, оф. 404
 Тел.: (473) 277-12-07, факс: (473) 271-85-90

ООО Центр Прикладных Исследований «Проектирование и разработка информационных систем» заинтересовано в программном комплексе SmartWall для создания интерактивных неэлектронных поверхностей и гаджетов, проект которого представлен автором, Приневым Мечиславом Александровичем, и готово принять участие в разработке и внедрении предлагаемого ПО. Достоинствами данного программного комплекса являются широкая сфера применения готовых программных продуктов, созданных на основе вышеуказанного ПО, их инновационность и социальная значимость для развития нашего региона, а также небольшие вложения для его разработки и внедрения, что делает участие в этих процессах привлекательным для нашей организации, так как будет способствовать ее дальнейшему развитию.

Директор ООО ЦПИ «ПРИС»



Евдокимов А.В.

« 21 » *но 28 пр* 2017 г.

**КАЗЕННОЕ УЧРЕЖДЕНИЕ ВОРОНЕЖСКОЙ ОБЛАСТИ
«ВОРОНЕЖСКИЙ ЦЕНТР РЕАБИЛИТАЦИИ ИНВАЛИДОВ»**

394062, г. Воронеж, ул. Краснозвёздная, 6 тел./факс: 8(473)270-75-14
e-mail: ogukristall2010@mail.ru

ОГРН 1133668017394 ИНН 3665093219/КПП 366501001 ОКАТО 20401000000

На базе КУ ВО "ВЦРИ" было протестировано ПО SmartWall, разработанное учеником 10 «А» класса МБОУ «Лицей №15» г. Воронежа Приневым М.А.

Данное ПО рекомендовано к использованию в качестве развивающего и развлекательного контента на занятиях для детей с ограниченными возможностями здоровья, молодых инвалидов, посещающих КУ ВО "ВЦРИ".

ПО SmartWall можно использовать как часть оборудования, необходимого для создания сенсорной комнаты для детей.

Достоинствами представленного ПО являются небольшие вложения для внедрения программ, а также возможность развивать у детей необходимые навыки на занятиях с помощью ПО SmartWall.

Автору было предложено внести дополнения в представленное ПО, а также был утвержден план по использованию данного ПО и дальнейшего сотрудничества с КУ ВО "ВЦРИ".

Заместитель директора



Л.А. Толстова



АДМИНИСТРАЦИЯ ГОРОДСКОГО ОКРУГА ГОРОД ВОРОНЕЖ
ВОРОНЕЖСКОЙ ОБЛАСТИ

Муниципальное бюджетное общеобразовательное учреждение
«Лицей № 15»

Циолковского ул., 12 а, 394029, г. Воронеж, Воронежская область, Россия
Тел. (473) 2499304

26.05.17 № 42
на № _____

СПРАВКА

Дана ученику 11 «А» класса МБОУ «Лицей №15» Принёву Мечиславу Александровичу в том, что в январе-феврале 2017 года в ходе конкурса «Учитель года» учителем Сапрыкиным Степаном Владимировичем, призером конкурса, использовалось приложение «SmartWallBattle», позволяющее при помощи проектора и веб-камеры проводить интерактивные викторины на уроках. Данное программное обеспечение получило высокую оценку жюри конкурса.

Директор МБОУ «Лицей №15»



Н.И.Яицкий



Оперативная полиграфия "В Формате"

ИНН 366301408182

г. Воронеж, Ленинский пр-т, 8/1, т.8-910-288-35-11,

сайт: vformate.pro

Утверждаю Тимошенко А.Ю.

Экспертная комиссия в составе: Тимошенко Алексей Юрьевич, Павлов Николай Александрович, Сидоров Сергей Алексеевич рассмотрела программное средство «бинаризация зашумленных изображений с плавающим порогом бинаризации» (автор Принев Мечислав Александрович) и приняла решение рекомендовать вышеуказанное программное средство к использованию в производственном процессе для решения следующих задач: выпуск рекламной и художественной продукции с использованием изображений, полученных посредством вышеуказанного программного средства.

Председатель экспертной комиссии экспертной комиссии

Директор



24.12.2015