Федеральное государственное бюджетное образовательное учреждение высшего образования «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича».

**Факультет Информационных систем и технологий**

**Кафедра Информатики и компьютерного дизайна**

**Лабораторная работа №5**

Программирование линейных списков и связанных объектов

****

**Автор работы:**

Сухоруков А.А.

**Научный руководитель:**

Федоров Д.Ю.

Санкт-Петербург, 2020

**1. Постановка задачи.**

**1.** Подключить дополнительные расширения SDL для работы с другими форматами изображений

**2.** Научиться выводить любое количество изображений

**3.** Попробовать масштабирование изображений

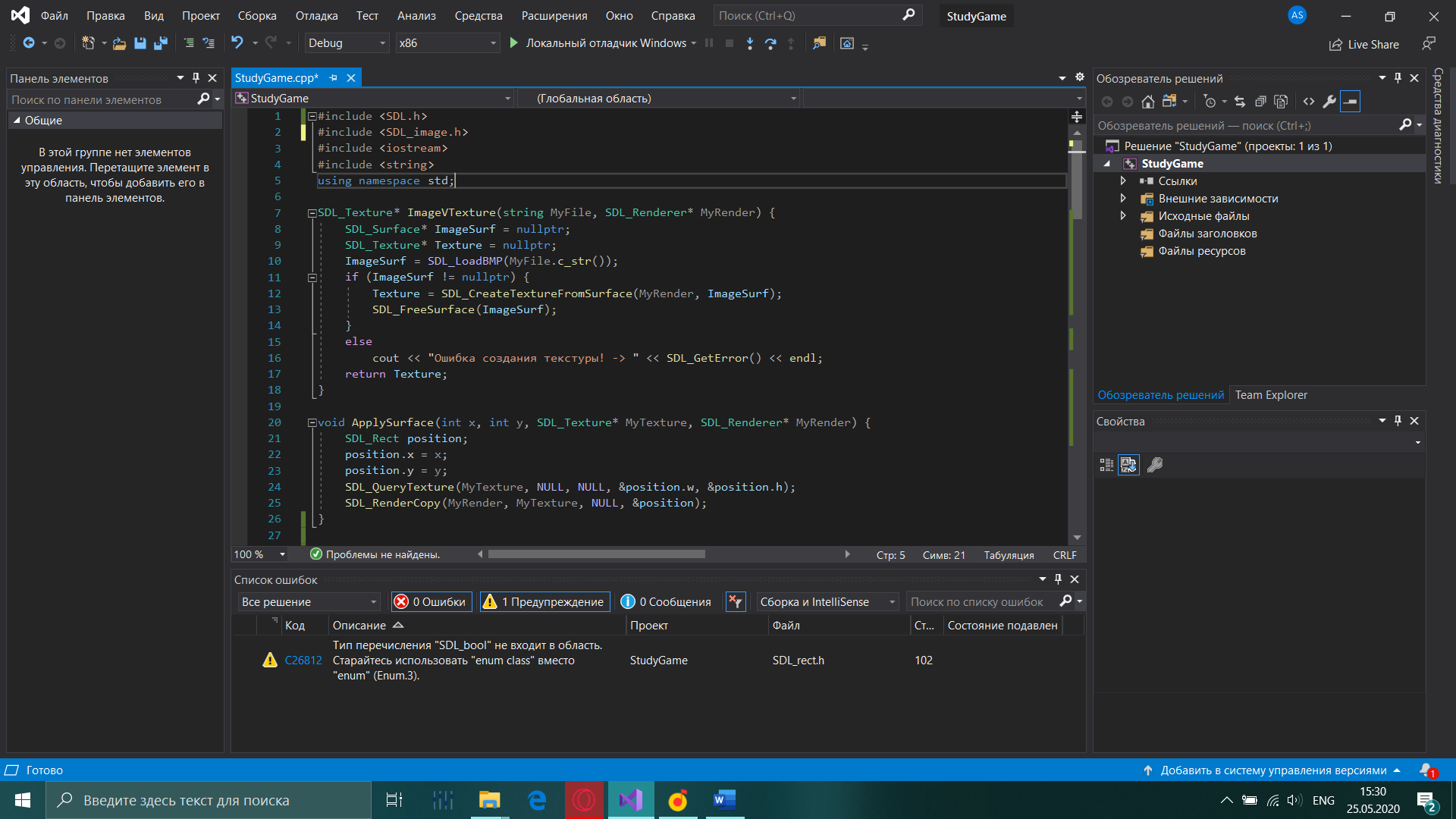
**2. Формализация задачи.**

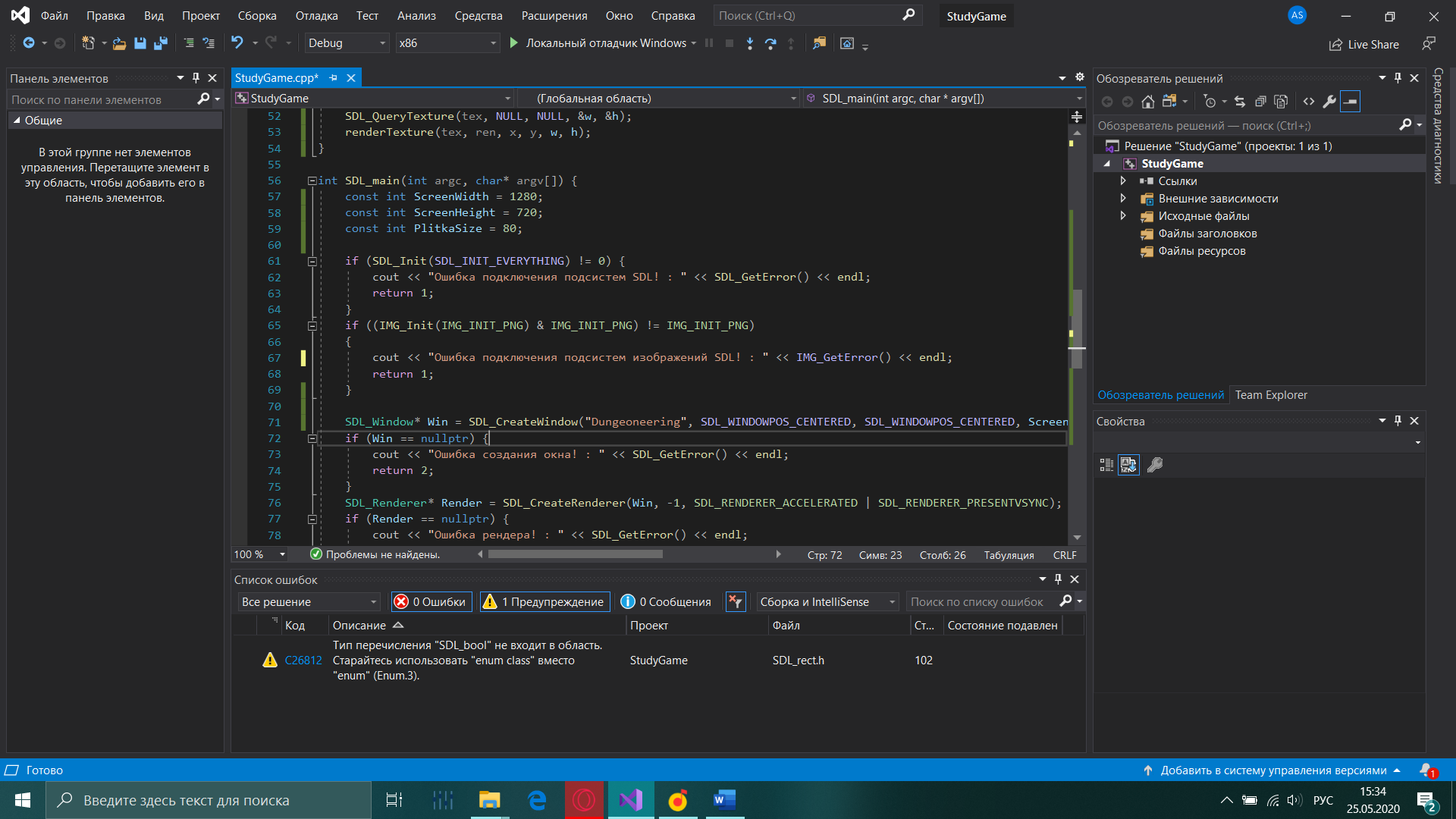
**1.** Подключить дополнительные расширения SDL для работы с другими форматами изображений  
Для работы с другими форматами изображений нам поможет SDL\_image, его установка аналогична с установкой обычного SDL, потому можно подглядеть в предыдущие лабораторные. Мы рассмотрим многие функции и преимущества, которые он дает

**2.** Научиться выводить любое количество изображений  
Мы уже выводили одно изображение, а теперь попробуем сразу несколько, причем друг на друга. Одним заполним фон нашей программы, а другую выведем по центру экрана и с прозрачным фоном, то есть мы научимся использовать уже и такие форматы изображений как PNG

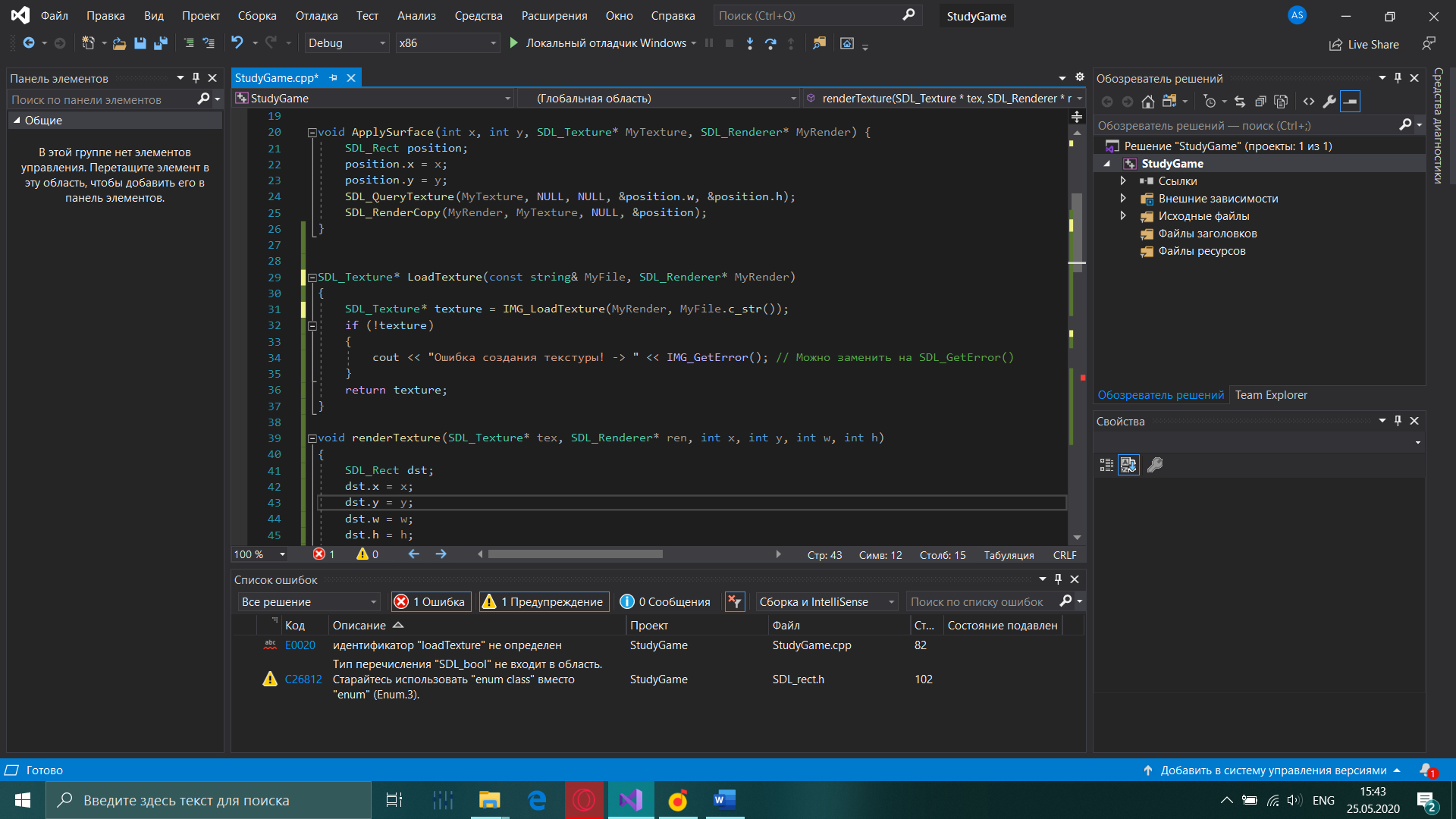
**3.** Попробовать масштабирование изображений  
SDL позволяет нам масштабировать выводимые текстуры, потому попробуем изменить размер тех изображений, что мы выводим, при этом не меняя их оригинального размера

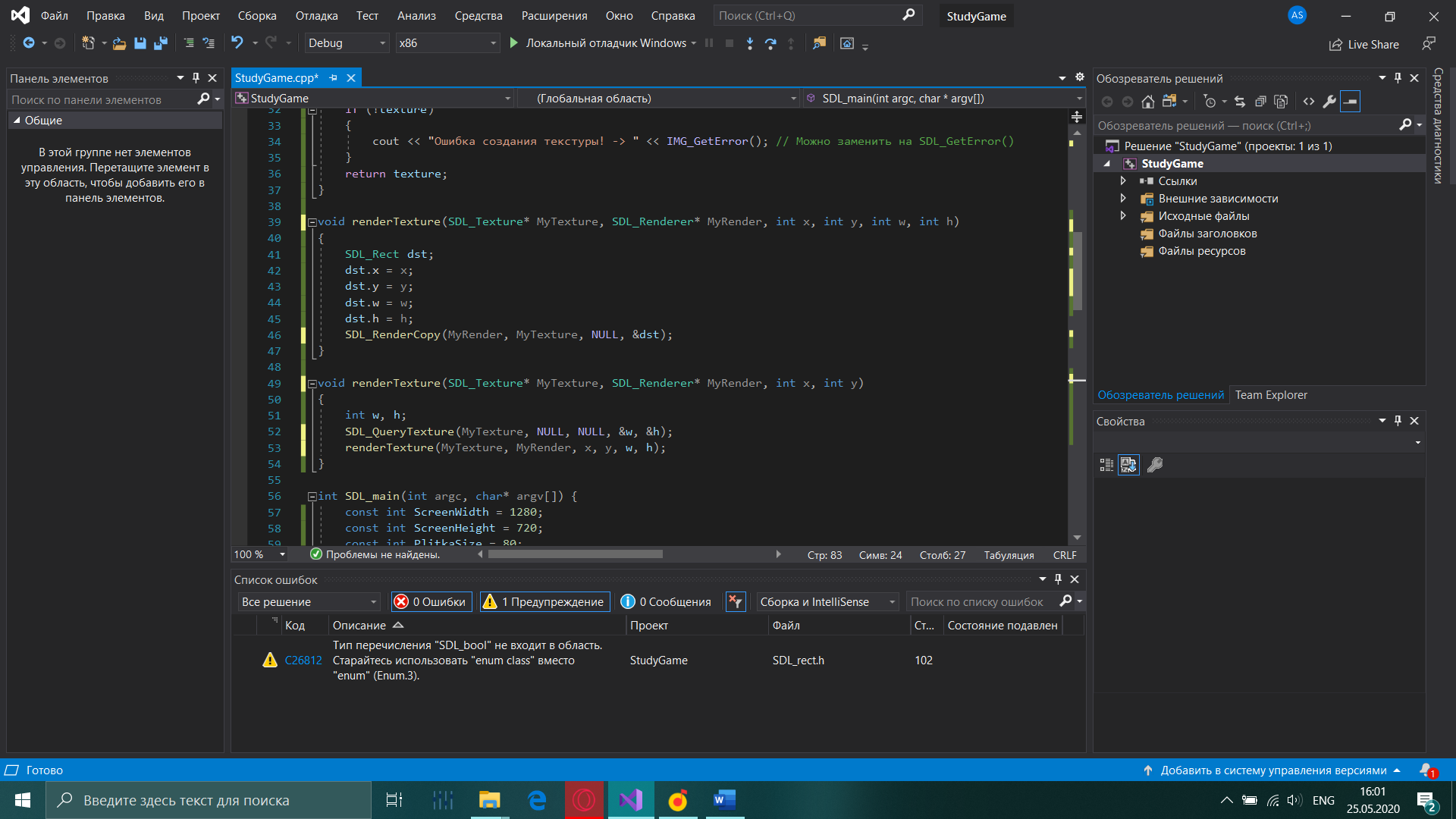
1. Подключить дополнительные расширения SDL для работы с другими форматами изображений

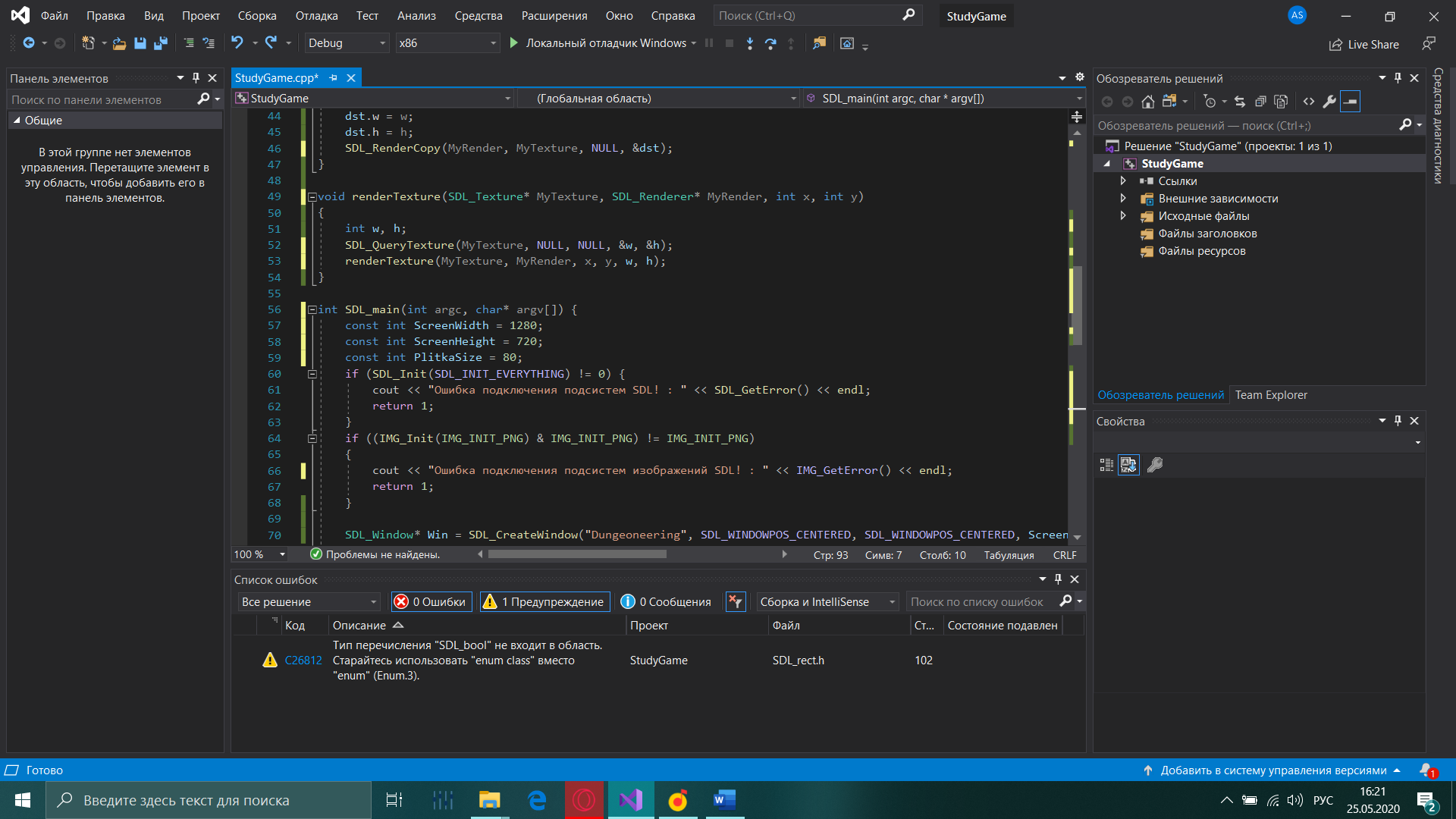
Для начала нам необходимо скачать библиотеку SDL\_image, после повторить все этапы из 3 лабораторной работы по настройке Visual Studio под использование новой библиотеки. Далее мы должны подключить SDL\_image в коде 

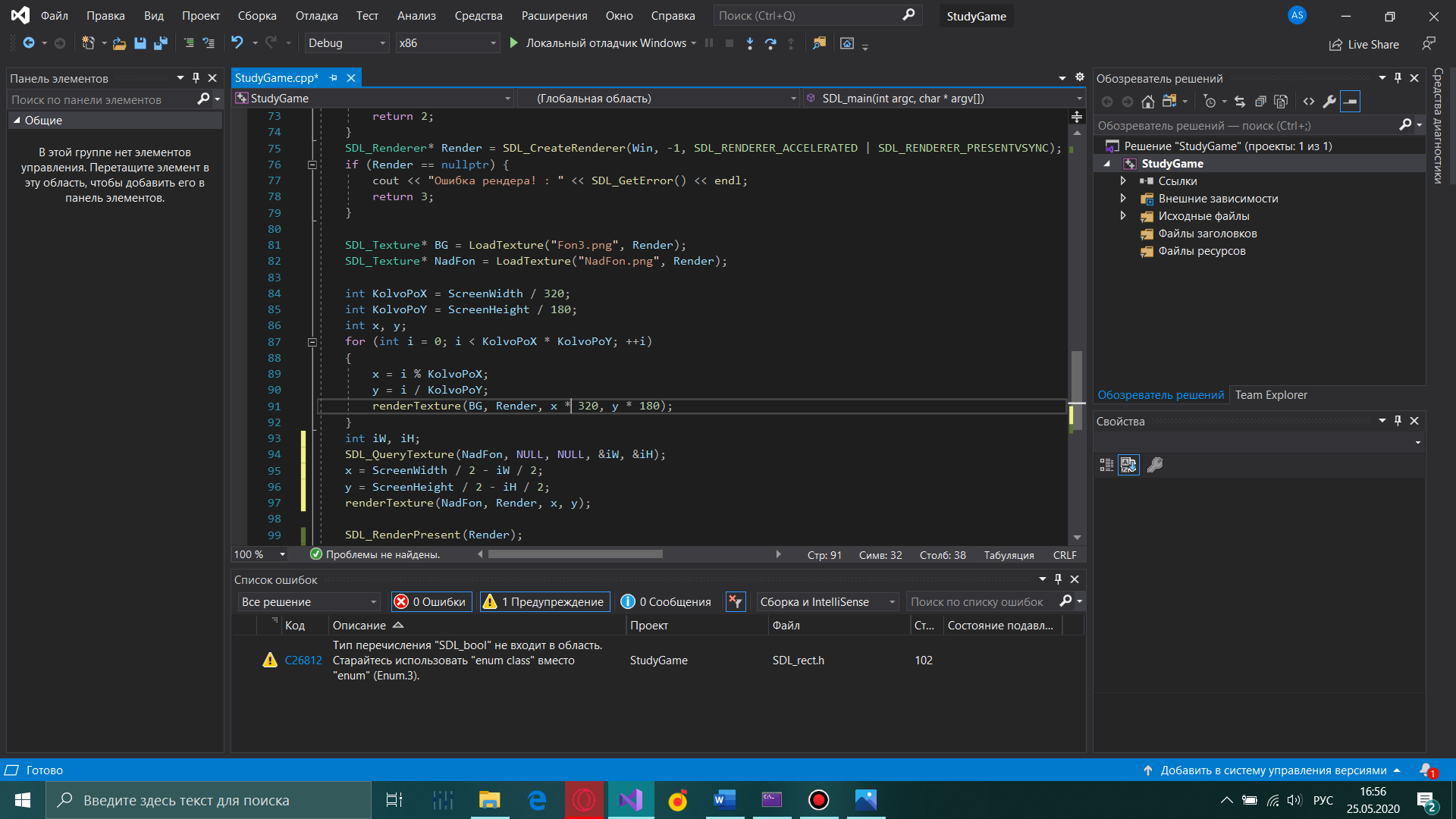
У SDL\_image также есть инициализация используемых подсистем, потому нужно также прописать IMG\_Init (но обязательно после SDL\_Init)  
В данном примере буду работать только с PNG  
Также пишу проверку на то, что инициализация прошла успешно

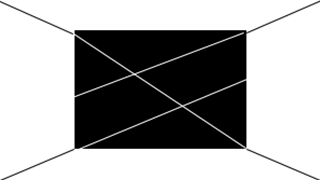
**2.** Научиться выводить любое количество изображений

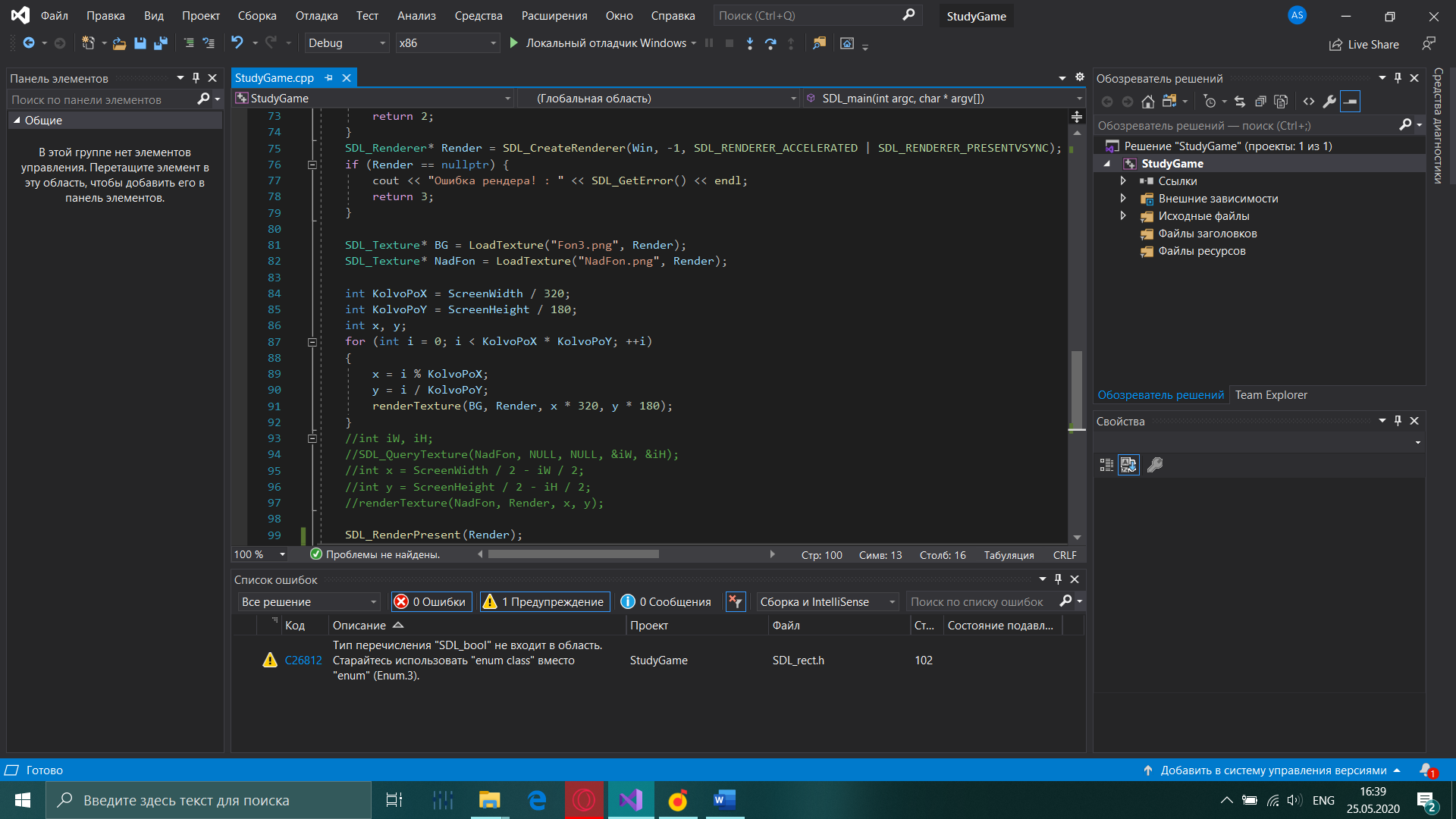
Далее мы будем использовать функцию IMG\_LoadTexture, она делает примерно то, что мы описывали в коде в прошлой лабораторной. То есть копирует изображение в ячейку памяти, преобразует ее в текстуру и возвращает нам данную текстуру.  
Покажем мы ее на примере функции LoadTexture, которая принимает путь к файлу и указатель на наш рендер. Однако данная функция нужна лишь для того, чтобы не писать проверку много раз

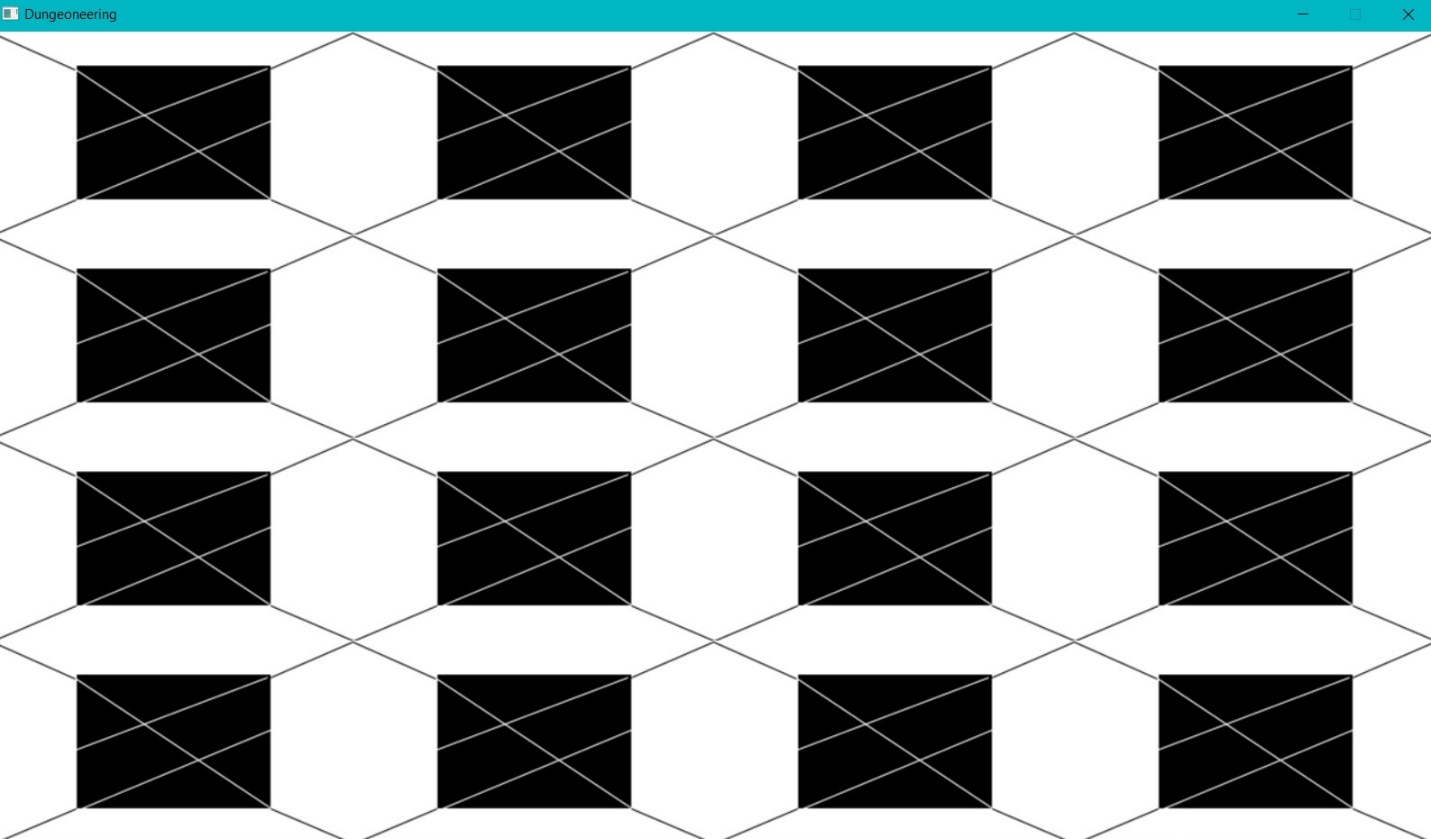
Далее создадим функцию, которая бы рисовала нашу текстуру в нужном месте и с нужным масштабированием. Однако мы же не всегда хотим изменять ее оригинальный размер, потому перегрузим данную функцию на два варианта  
1. Будет принимать текстуру, указ. рендера, координаты и нужный размер текстуры  
2. Будет принимать все кроме размера, а после масштабировать в виде 1:1 и вызывать функцию 1 , передавая ей обработанный размер  


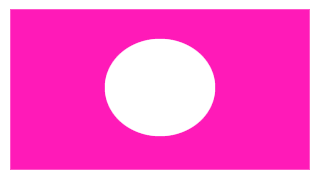
Координаты, по которым SDL рисует текстуры – это всегда левый верхний угол текстуры. Потому при расчетах координат это стоит учитывать. Также он каждое новое изображение выводит уже над предыдущими, это стоит учитывать при наложении изображений как слоев.   
Размер нашего окна я вывел через константы, в данной случае это 720p  


Также я заранее загружаю оба изображения, получая текстуры

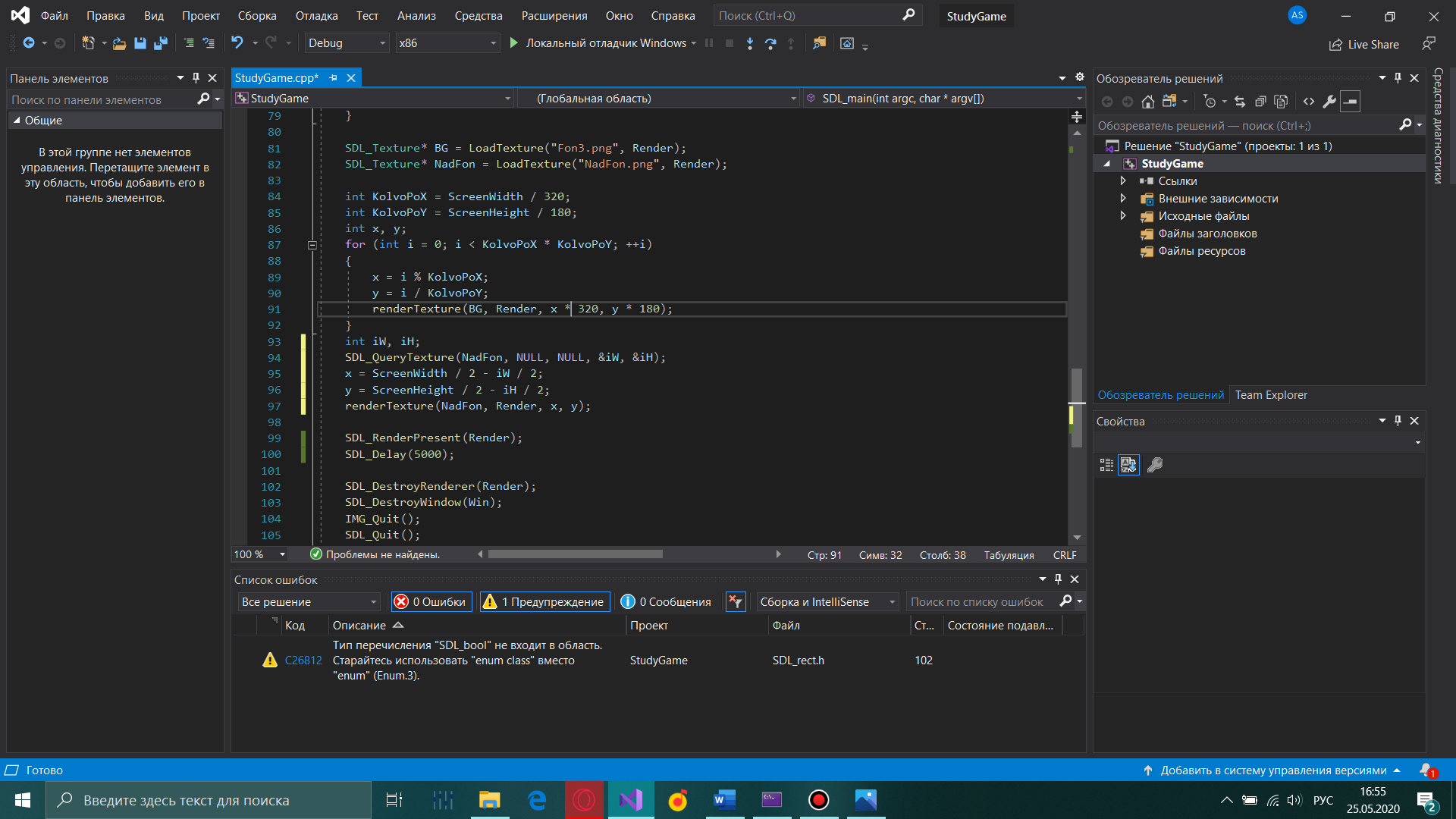
Фон я хочу сделать заполнением. Для этого в фотошопе создал PNG изображение Fon3 специально для заполнения. Оно имеет размер 320 на 180 пикселей  
  
  
Без масштабирования данного изображения на нашей окне поместится только 16 таких Fon3, но все равно давайте выведем их на экран

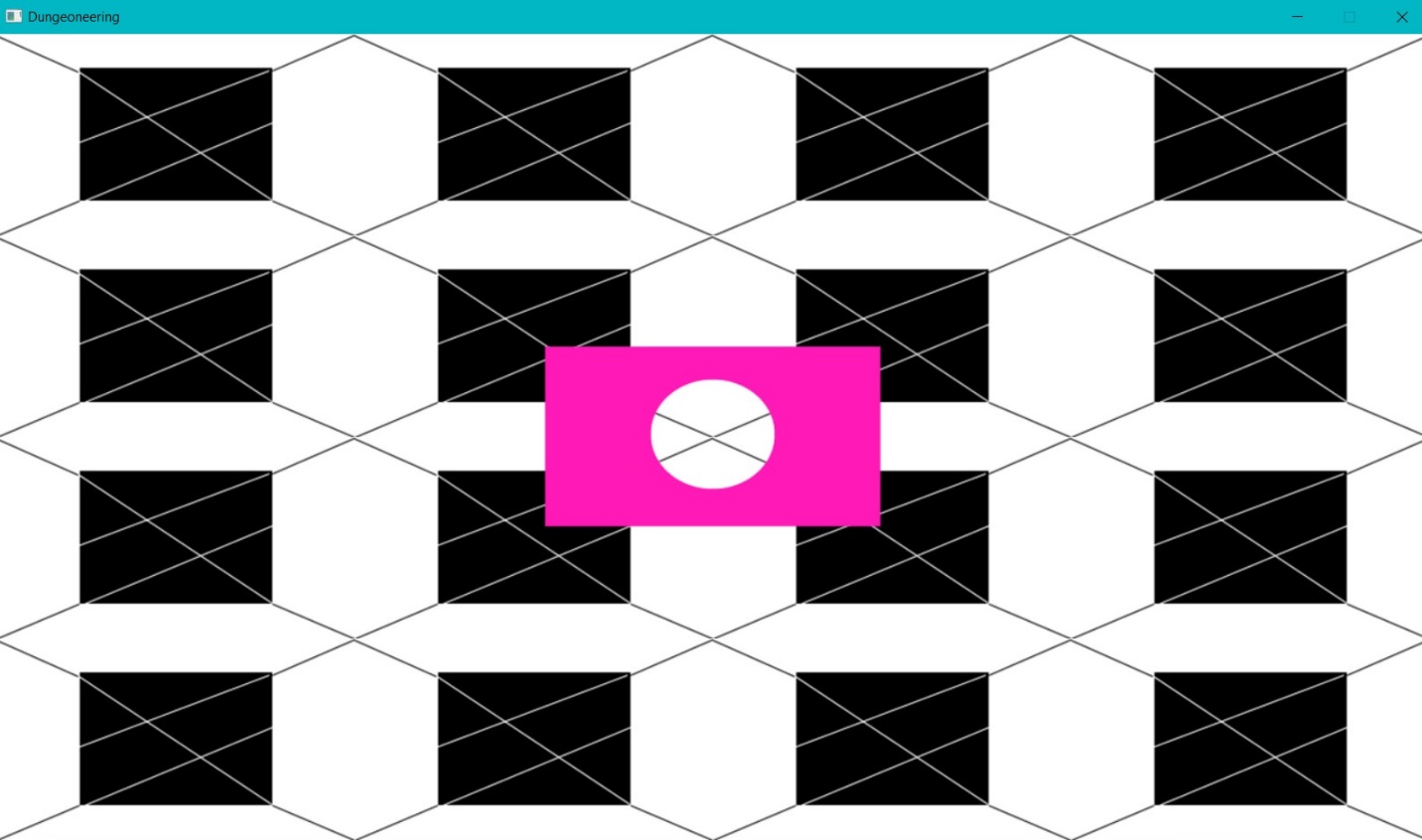
Таким вот нехитрым способом мы аккуратно заполняем фон нашими текстурами

И на выходе получаем  


Давайте же добавим еще изображений. Для этого я в фотошопе создал NadFon.png, который обладает прозрачным фоном и разрешением 320 на 180 пикселей  


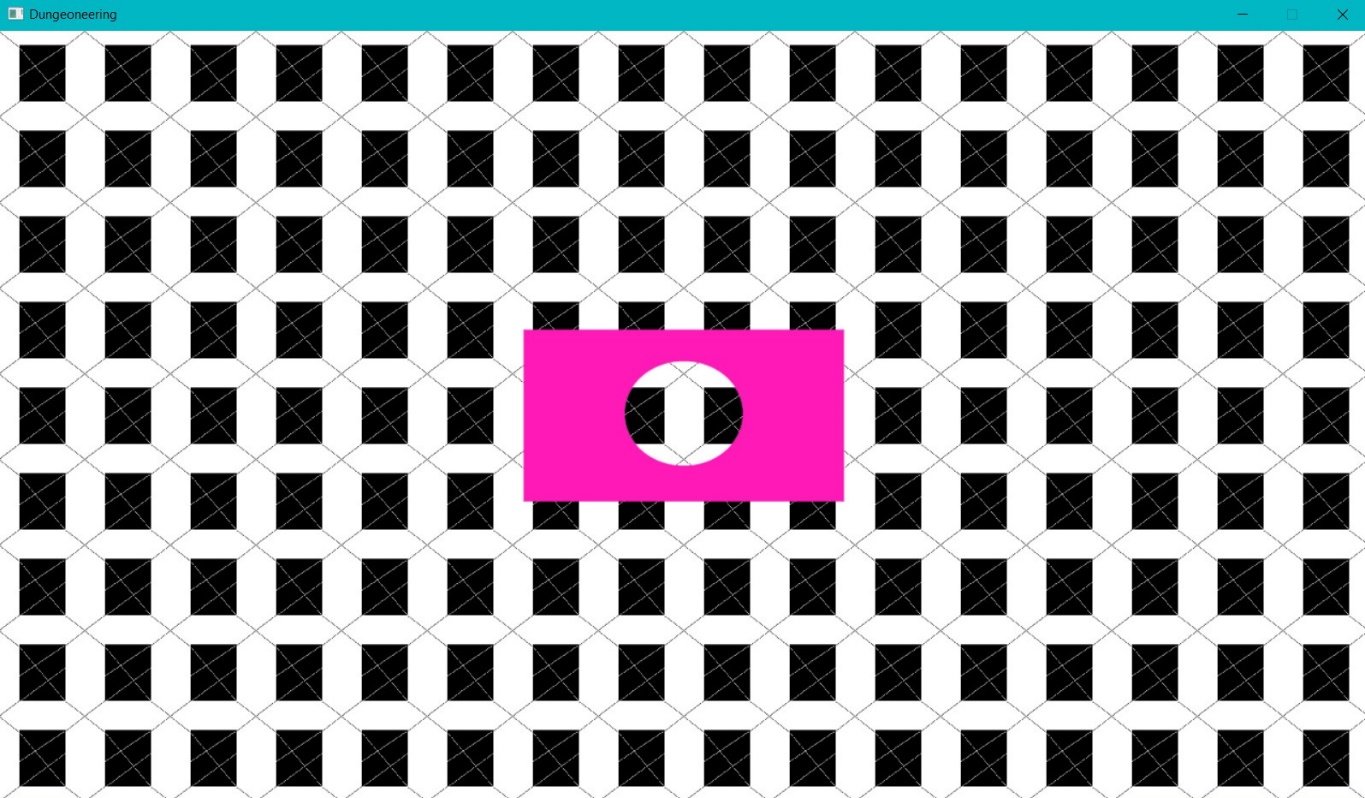
Теперь же добавим его в центр экрана и узнаем что получим на выходе рендера



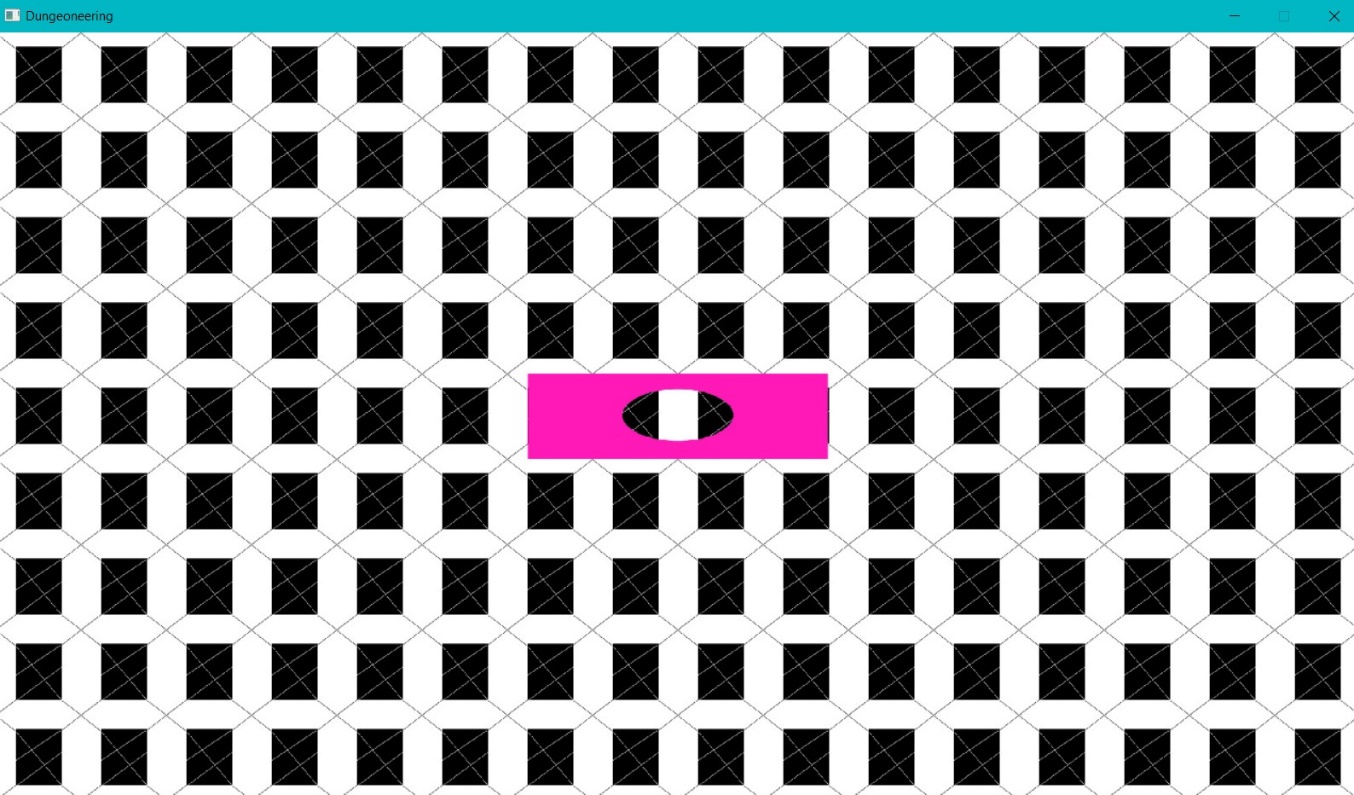


1. Попробовать масштабирование изображений

Теперь же применим масштабирование. Функцию для этого мы уже написали, потому напишем же в параметрах renderTexture() последние два параметра самостоятельно

Пока что это лишь один размер, но давайте же сделаем изображение фона размера 80 на 80 пикселей  


А изображение по центру 300 на 90 пикселей



**Проделан новый шаг к понимаю SDL!**