**Министерство науки и высшего образования Российской Федерации**

Федеральное государственное автономное образовательное учреждение высшего образования «Уральский федеральный университет имени первого Президента России Б.Н. Ельцина

Институт радиоэлектроники и информационных технологий – РТФ

Центр ускоренного обучения

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №0**

**По дисциплине «Программирование»**

Студент группы РИЗ-140938у Бабенко М.А.

Преподаватель: Архипов Н.А.

**Екатеринбург, 2024 г.**

**Лабораторная работа N0**

**Знакомство с платформой «Scratch»**

Цель: Тренировка навыков программирования на абстракции с применением платформы Scratch.

**Задание 1. Игра «Лабиринт»**

Задача: сделать игру «Лабирит» с тремя уровнями сложности.

Для начала мы создаем первый уровень сложности. Для него мы рисуем сцену в редакторе. Нарисованная сцена представлена на рисунке 1.

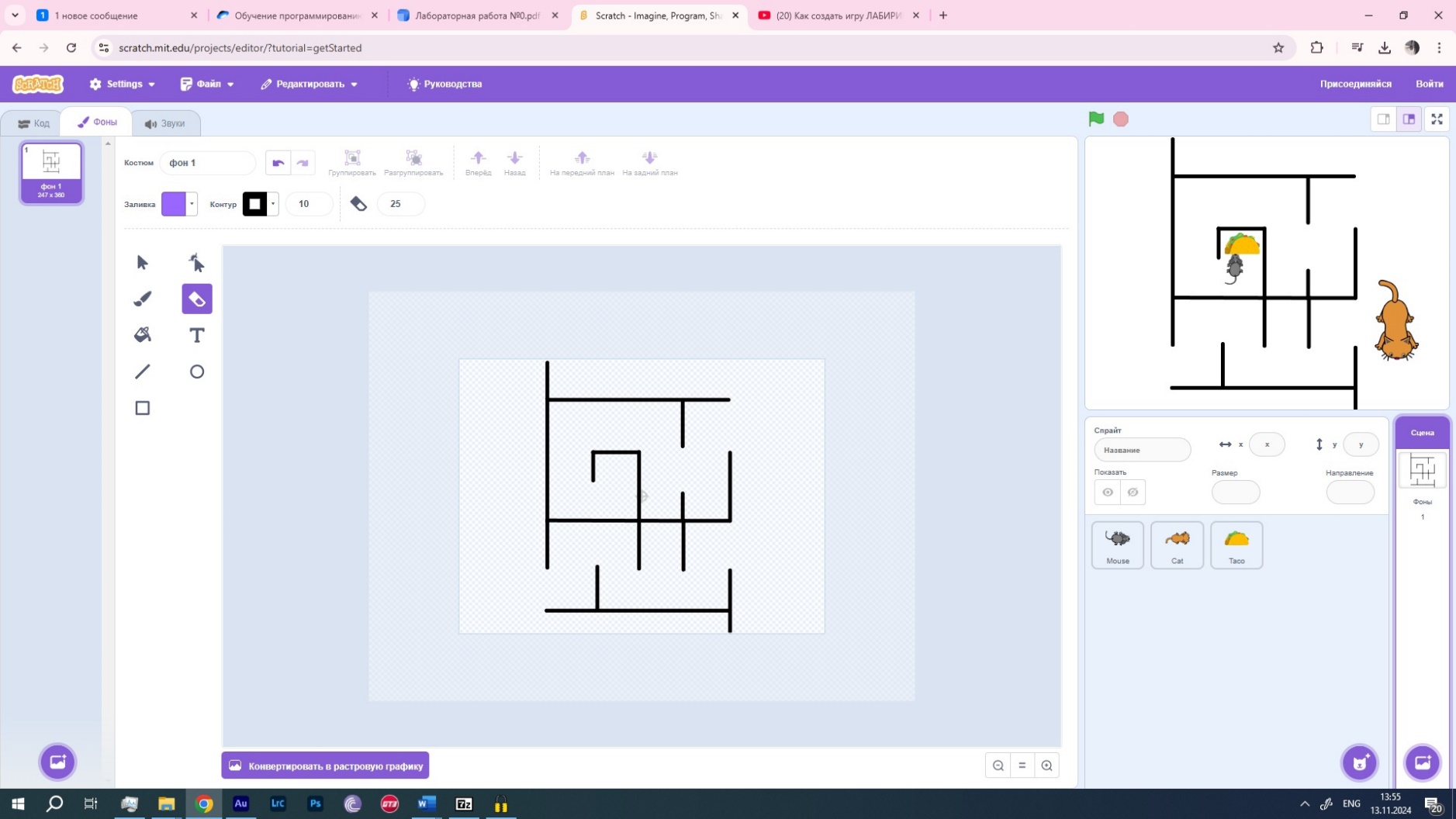


Рисунок 1 – Нарисованная сцена для уровня 1

Затем расставляем персонажей игры, даем им название и задаем их первоначальные координаты. Подготовленная сцена для уровня 1 представлена на рисунке 2.

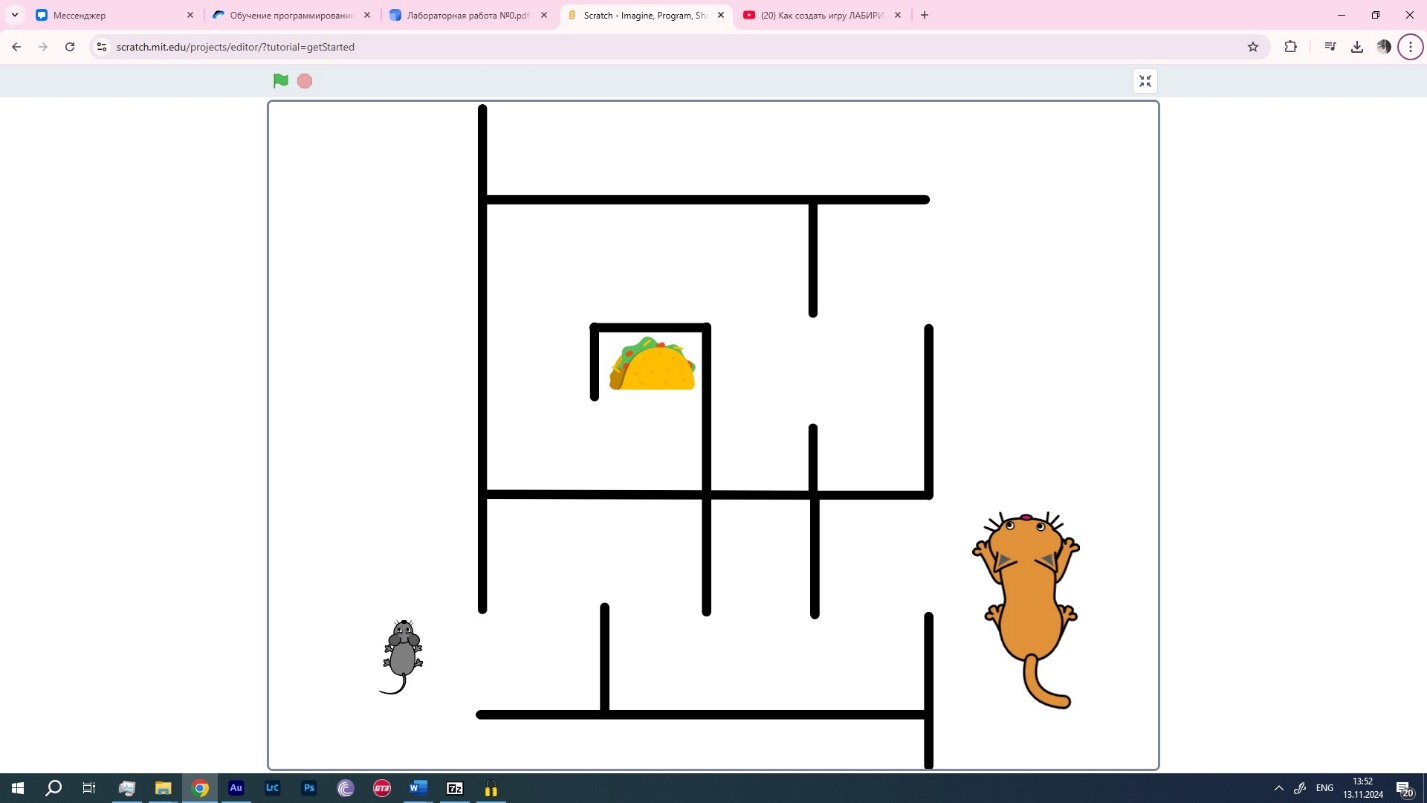


Рисунок 2 – Готовая сцена для уровня 1

Затем мы пишем код программы для кота. Задаем ему программой первоначальные координаты и его движения во время игры. Код для него представлен на рисунке 3.

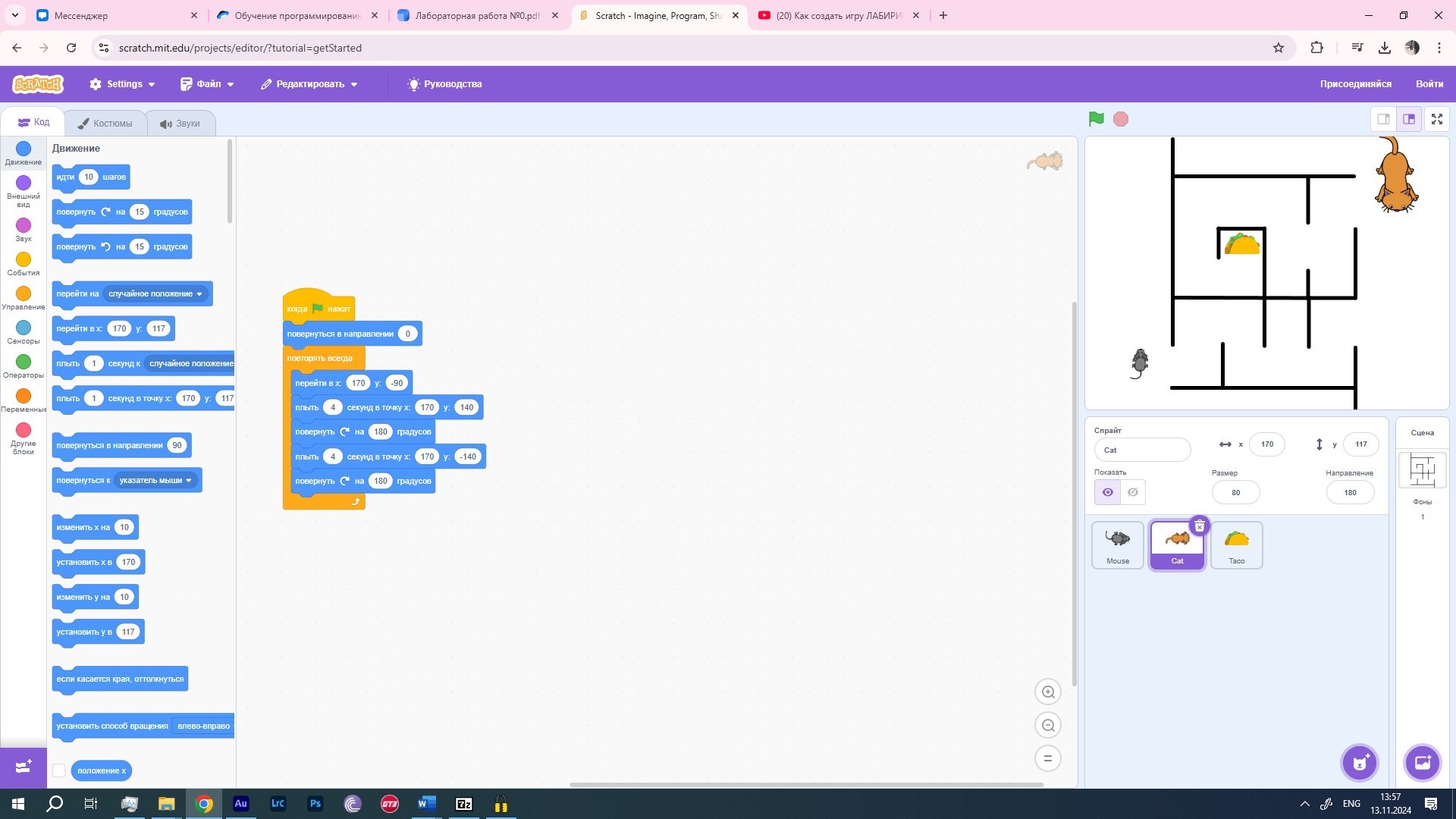


Рисунок 3 – Код, задающий движения кота в игре

Далее нам требуется написать код для мыши, которой в игре будем управлять мы. Управление назначаем на клавиши W, A, S, D; при нажатии на них персонаж сделает 5 шагов, пишем программу для его движения. Игра будет продолжаться пока мы не дойдем до тако в конце лабиринта. Когда мышь достигнет ее, персонаж «скажет» реплику «Вкусно!», и игра завершится. Если мышь столкнется со стеной или котом – игра завершится, также персонаж «скажет» реплику «Ой», и игра завершится досрочно. Код для персонажа мыши представлен на рисунке 4.

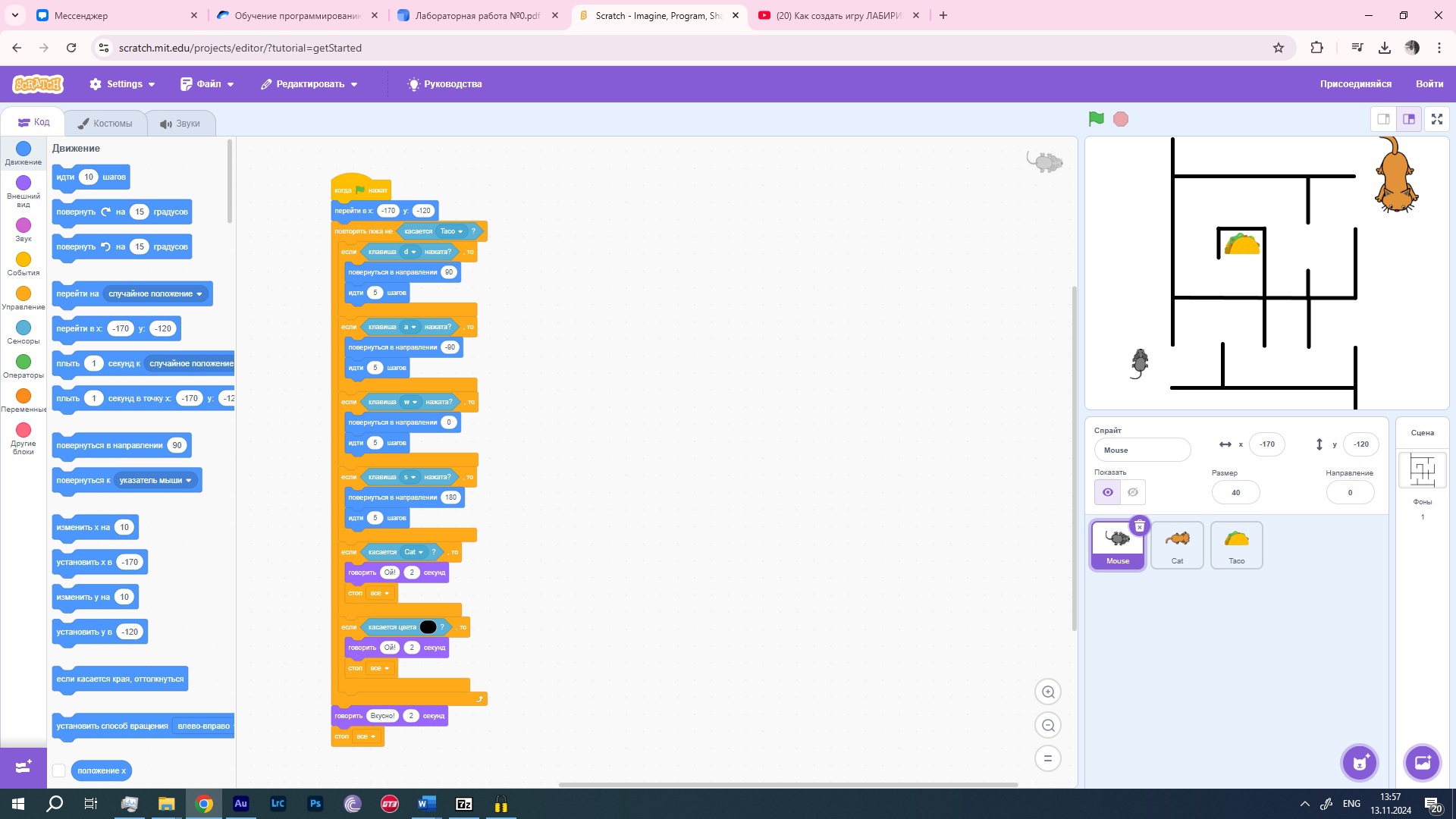


Рисунок 4 – Код для персонажа мыши

Когда мы создали и протестировали уровень 1, переходим к созданию других уровней. В редакторе создаем декорации для двух других уровней и расставляем «врагов». Также делаем экраны, иноформирующие о выигрыше и проигрыше. Локации следующих двух уровней представлены на рисунке 5 и 6.

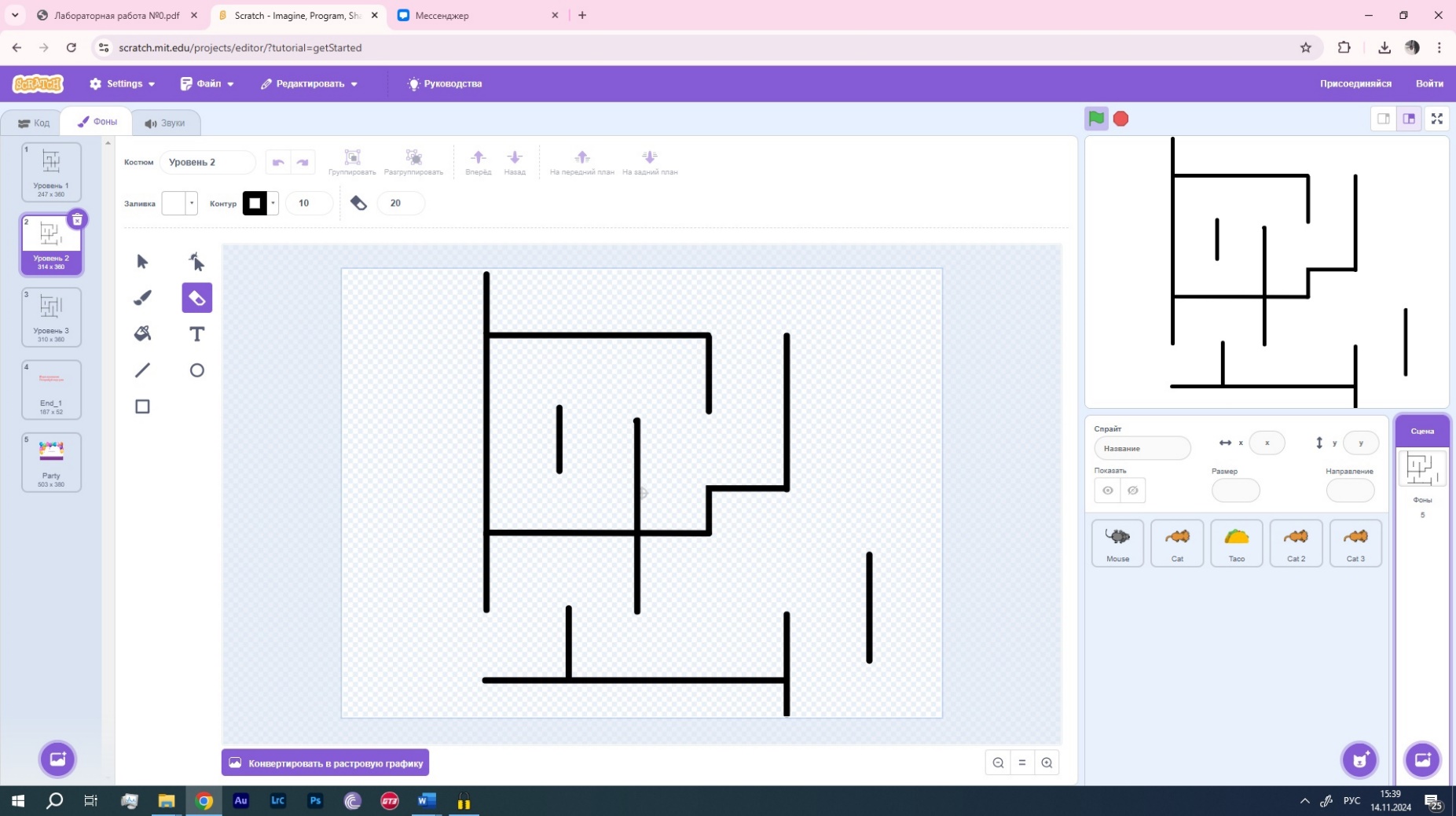


Рисунок 5 – Локация для уровня 2

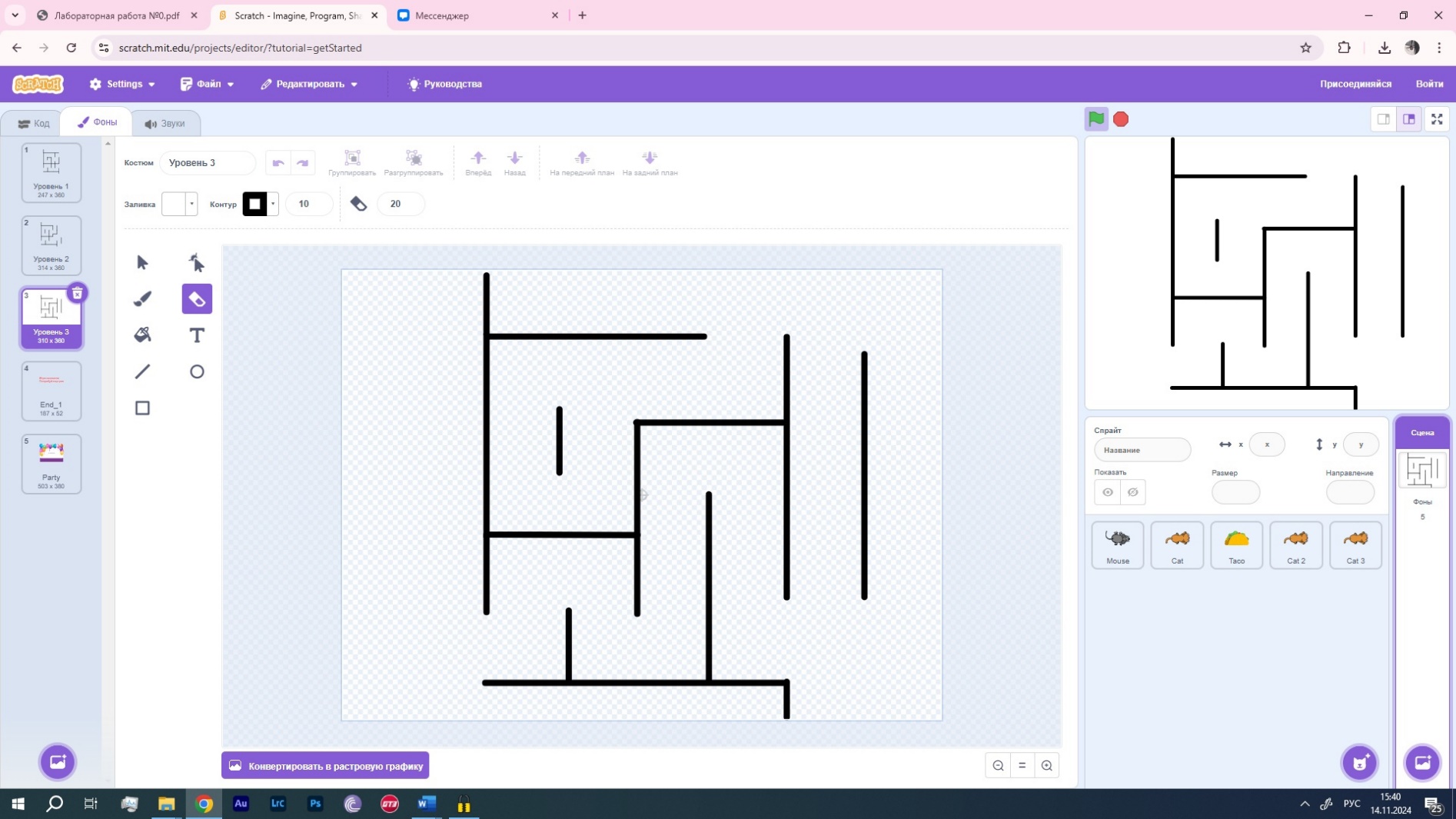


Рисунок 6 – Локация для уровня 3

Для каждого из уровней будет меняться сложность – немного меняться движения «врагов» и увеличиваться их количество. «Врага» первого мы оставляем на уровне, а остальных двух изначально скрываем и показываем при переходе на следующие уровни. Чтобы их движения при переходе на следующий уровень не накладывались друг на друга, при каждом переходе ограничиваем исполнение других скриптов спрайта. При переходе в окно победы или проигрыша мы скрываем все объекты с карты. После добавления уровней и небольшого изменения программ под них, получается следующий их код. На рисунке 7 представлен код программы для персонажа мыши.

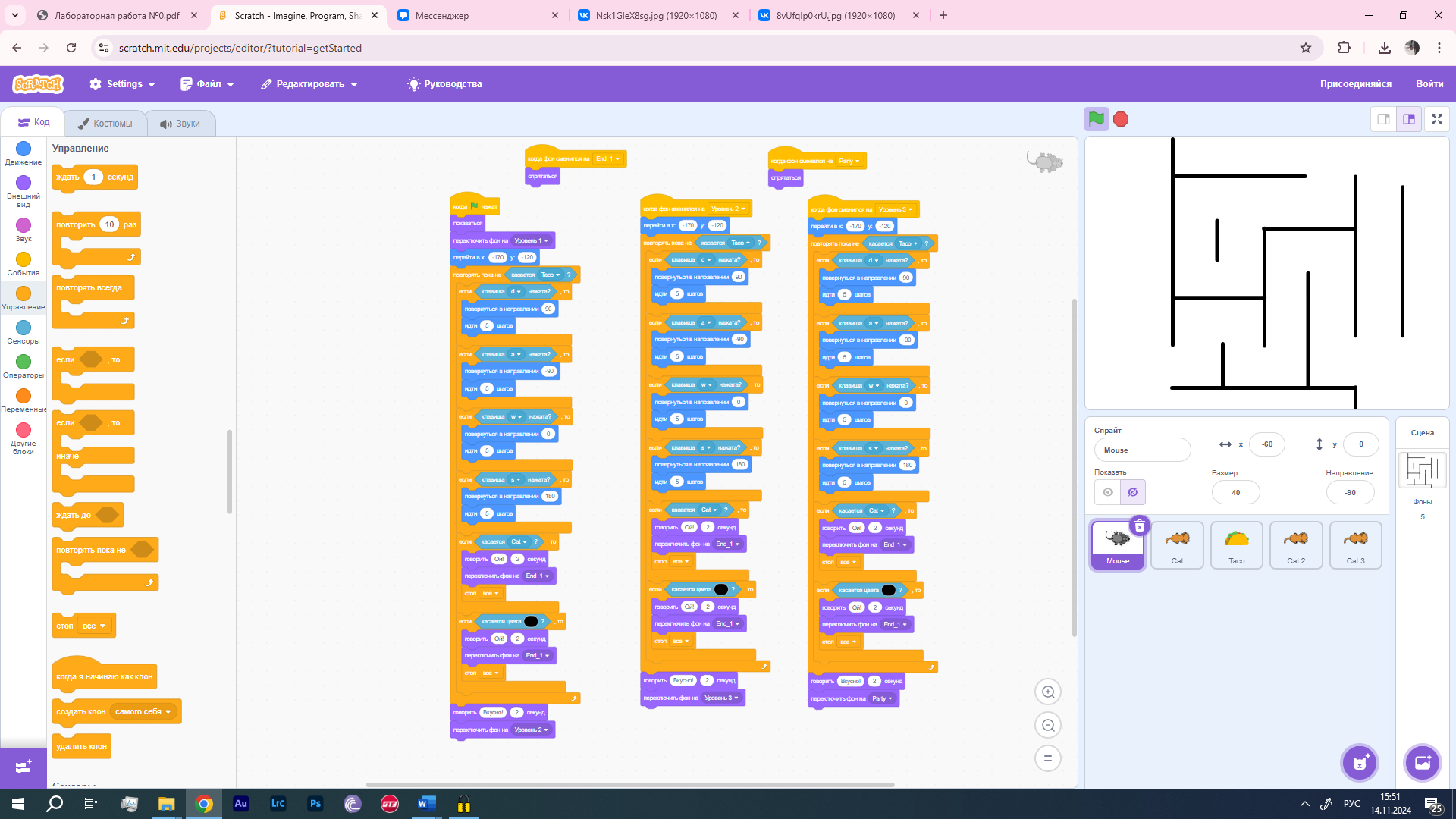


Рисунок 7 – Код программы для персонажа мыши

На рисунке 8 представлен кот программы для первого кота, появляющегося с уровня 1 по уровень 3.

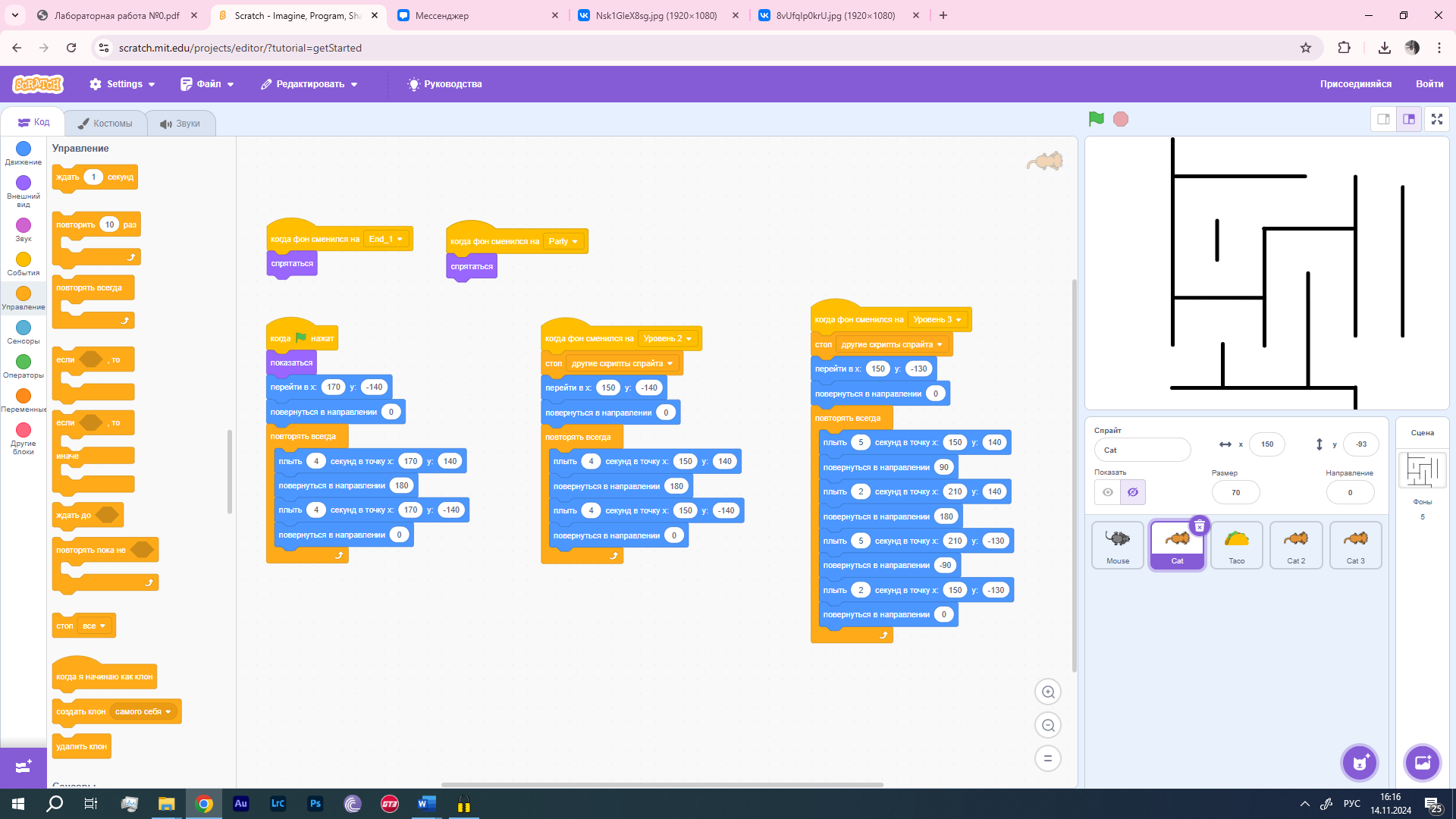


Рисунок 8 – Код программы для первого кота

На рисунке 9 представлен код программы для второго кота, появляющегося с уровня 2 по уровень 3.

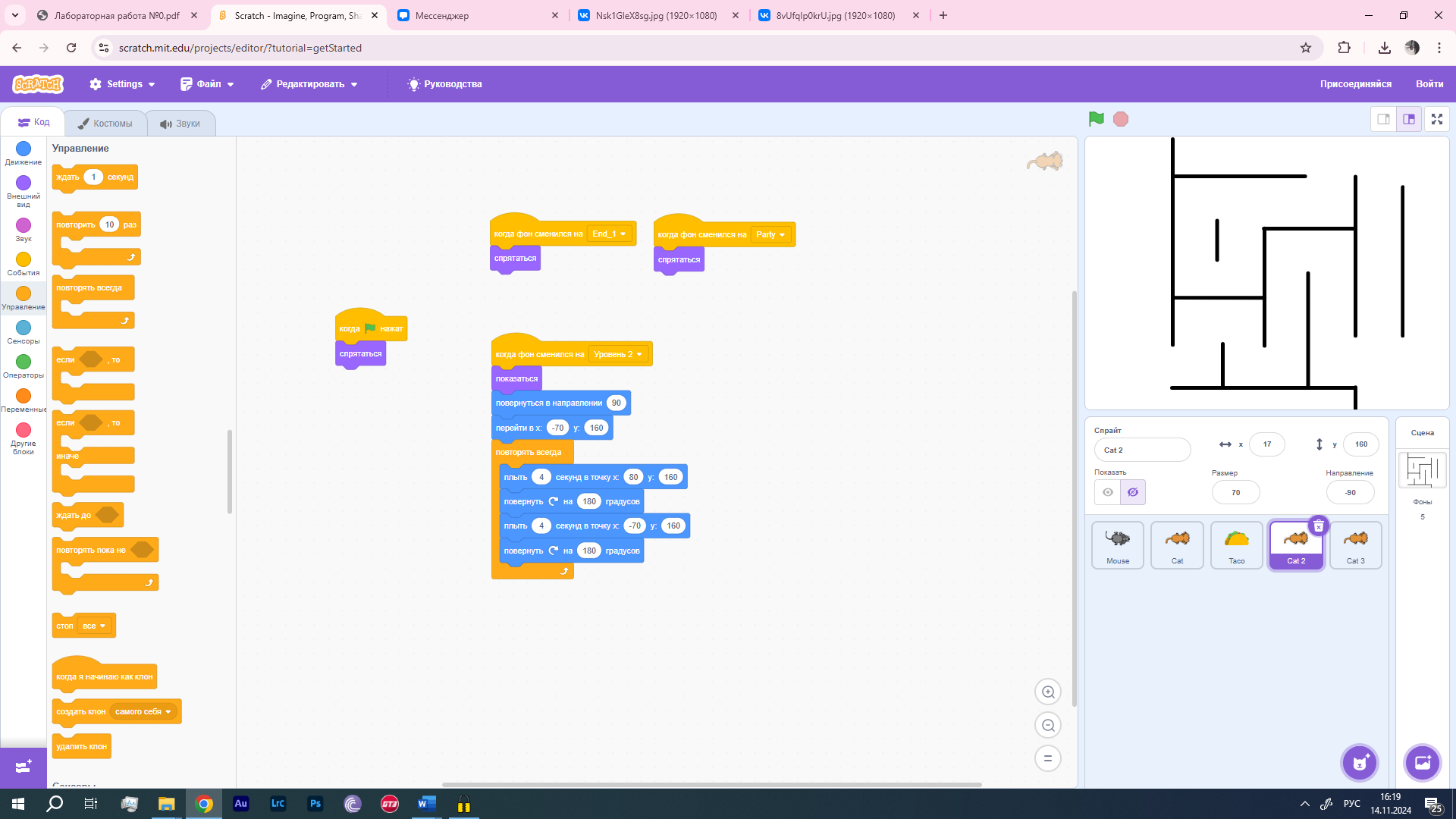


Рисунок 9 – Код программы для второго кота

На рисунке 10 представлен код программы для третьего кота, появляющегося только на уровне 3. Также на рисунке 11 представлен код для еды для мыши, которая скрывается в окнах победы или проигрыша, и появляется в начале игры, а на уровне 3 незначительно перемещается.



Рисунок 10 – Код программы для третьего кота

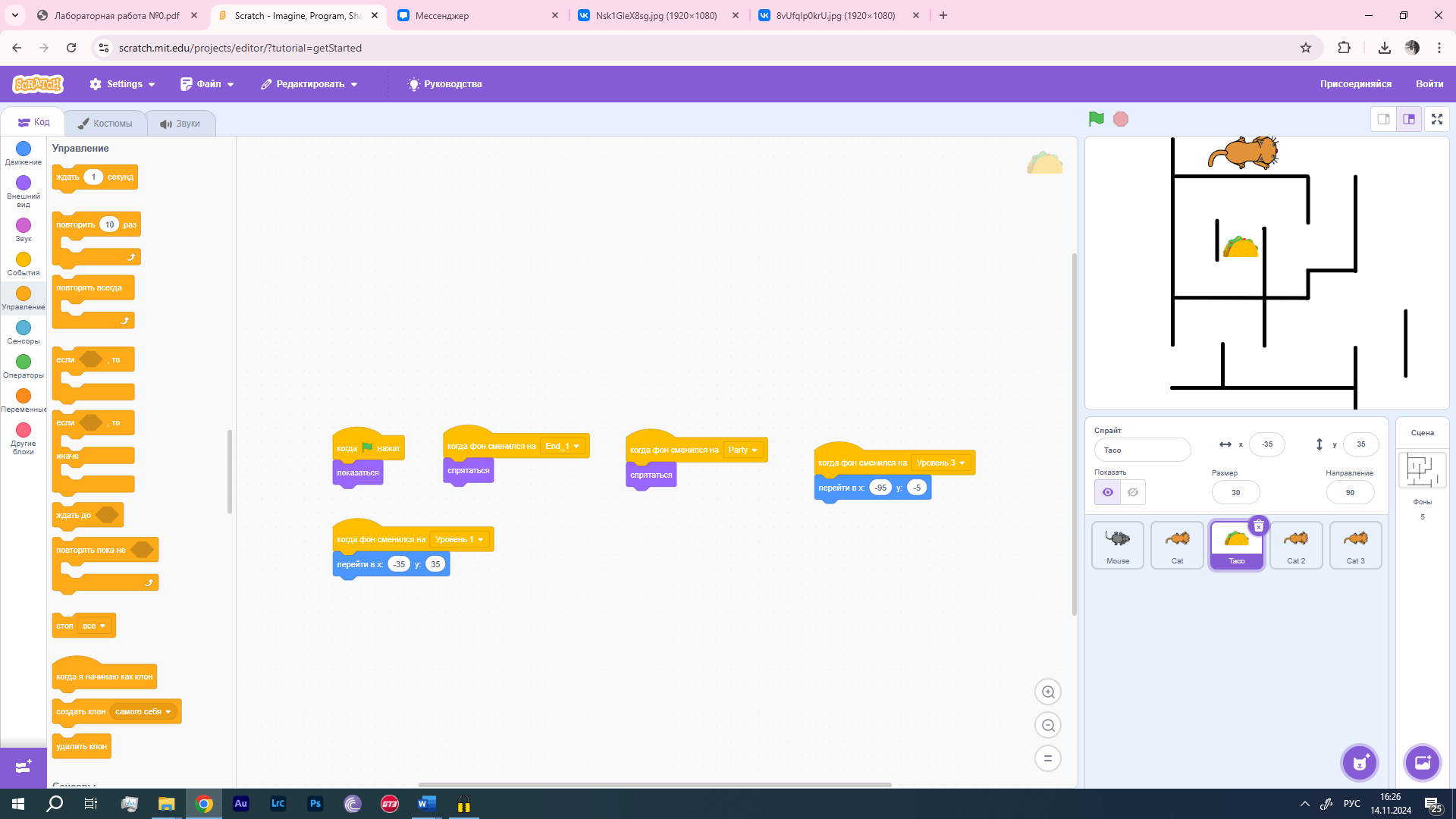


Рисунок 11 – Код программы для еды (тако)

**Задание 2. Игра «Бутерброд»**

Задача: сделать игру «Бутерброд» с проверкой правильности его формирования.

Для начала нам требуется нарисовать спрайты и локации. Делаем это при помощи встроенного графического редактора. Нарисованные спрайты и сцены представлены на рисунке 12.

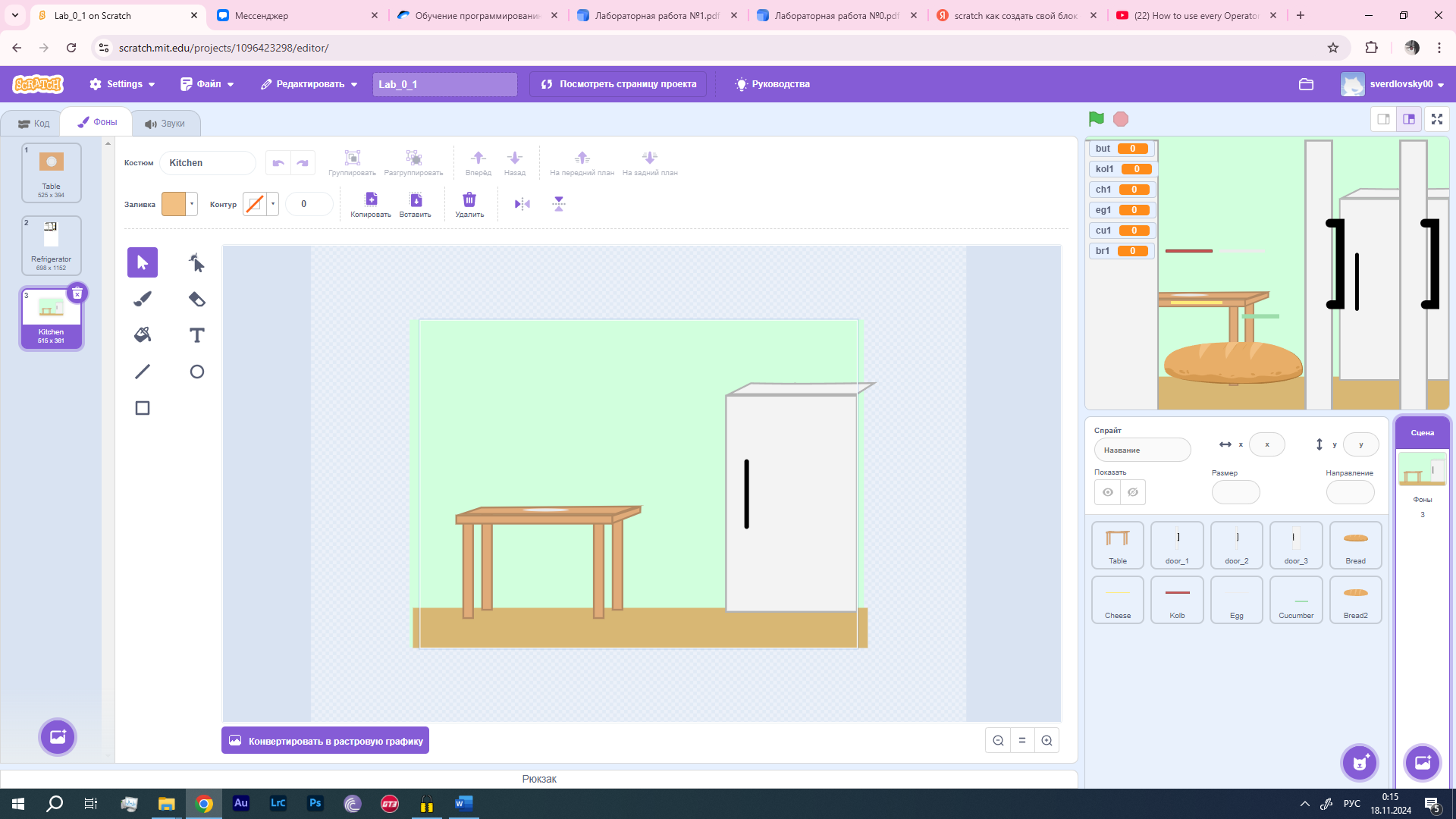


Рисунок 12 – Нарисованные для игры спрайты и сцены

Всего создаем три локации: кухня, холодильник и стол с тарелкой. Переключения между локациями производятся клавишой V. В локации с холодильником чтобы получить доступ к продуктам, требуется нажать на одну из его дверец. Код для них представлен на рисунке 13.

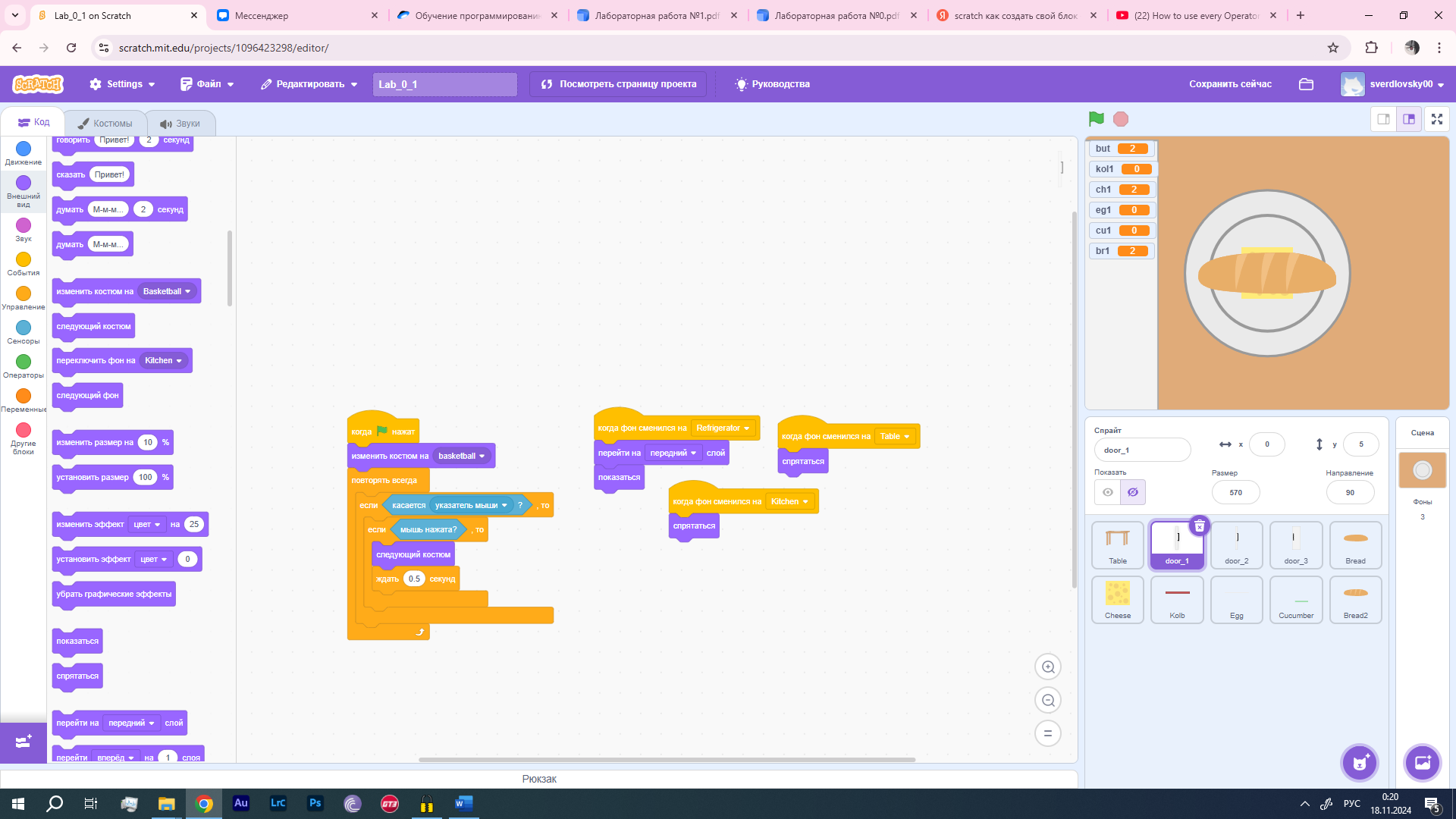


Рисунок 13 – Код для дверец локации «холодильник»

Затем мы вводим переменные, чтобы игра понимала, взяли ли мы предмет или нет. Чтобы взять предмет, требуется нажать на него, и он появится на столе. В зависимости от локации вид спрайта будет меняться. Переменная «but» выступает в роли счетчика, чтобы игра понимала, сколько предметов мы взяли. Код для спрайтов представлен на рисунке 14-17.

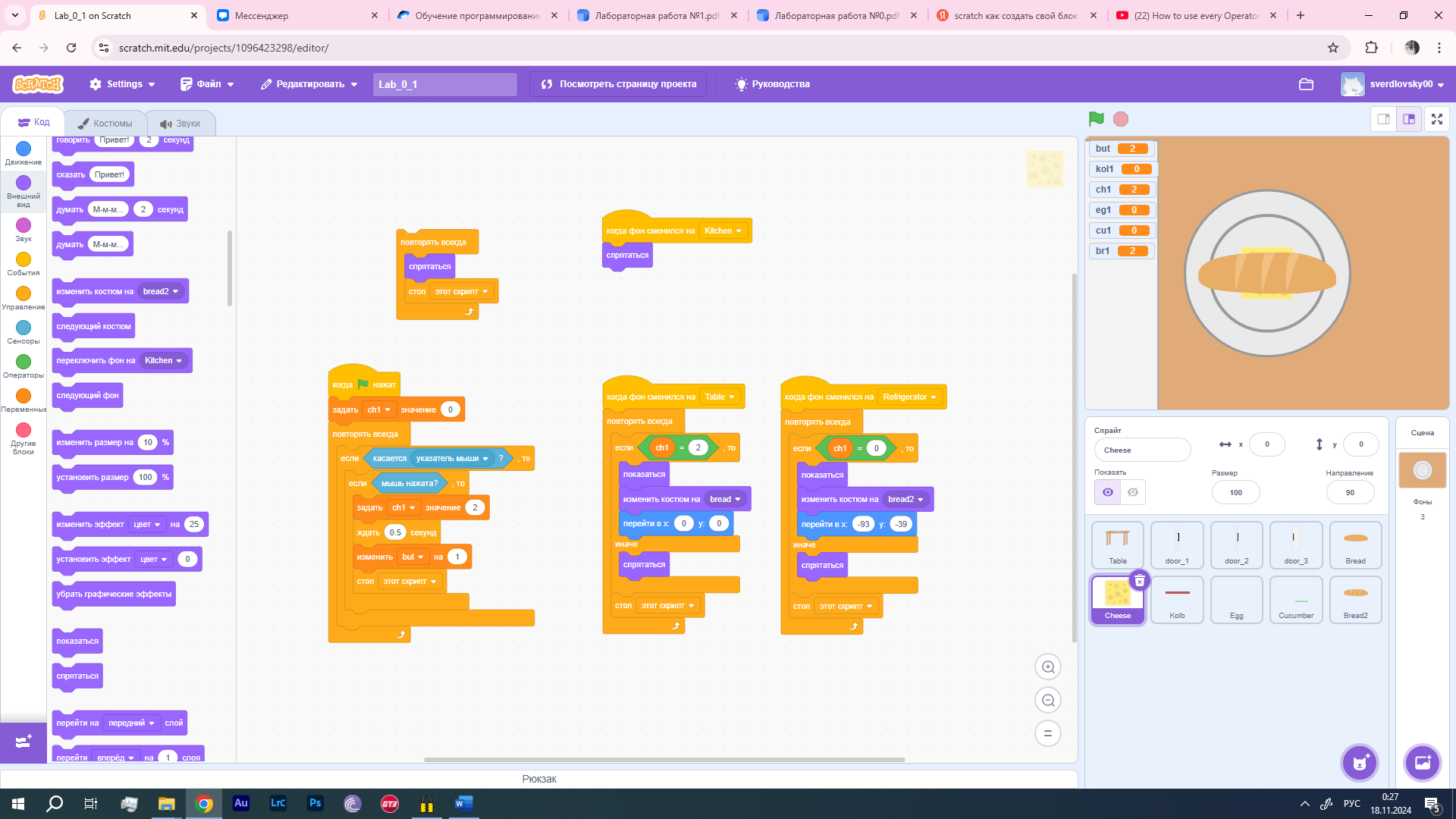


Рисунок 14 – Код для спрайта «сыр»

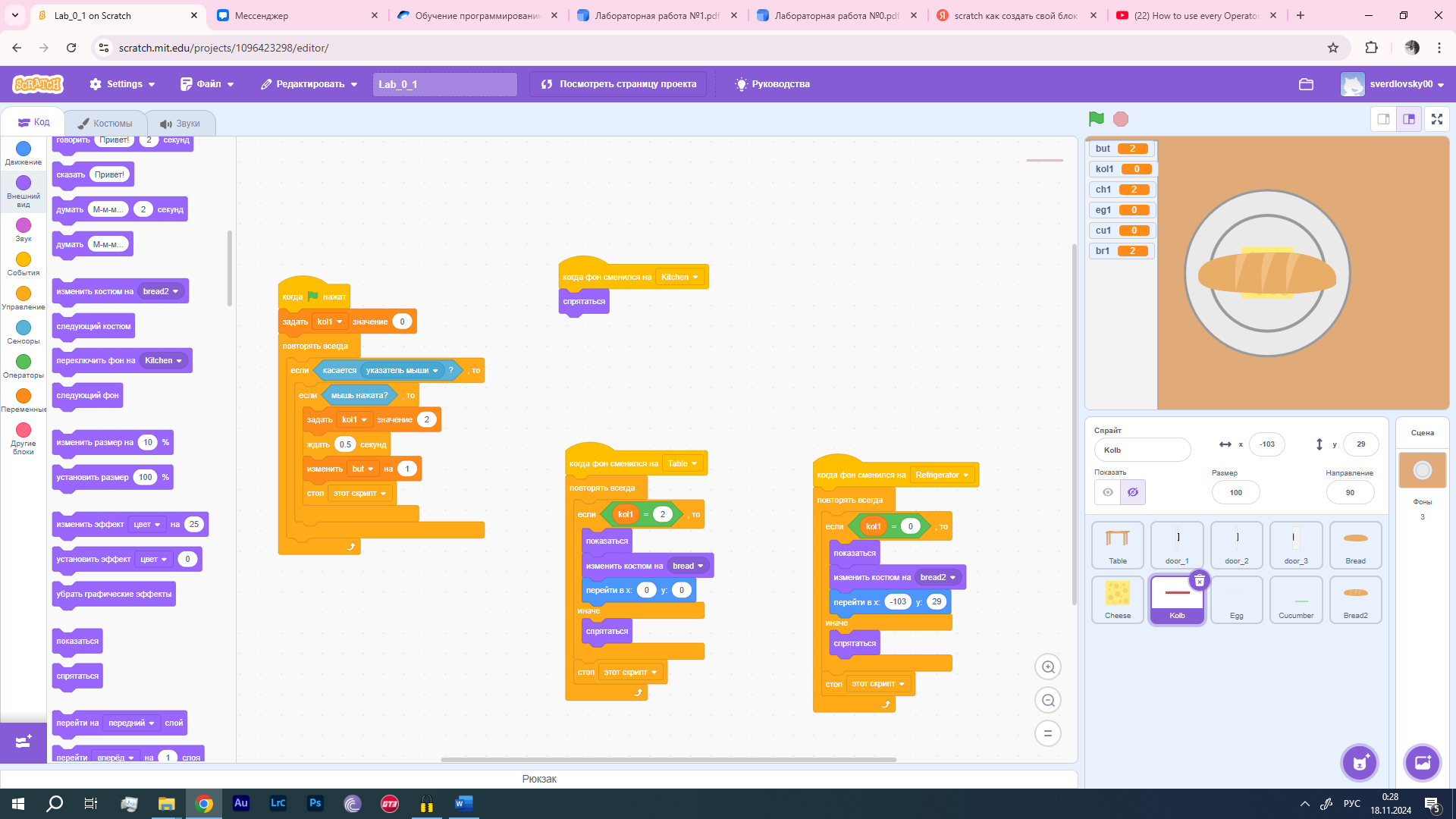


Рисунок 15 – Код для спрайта «колбаса»

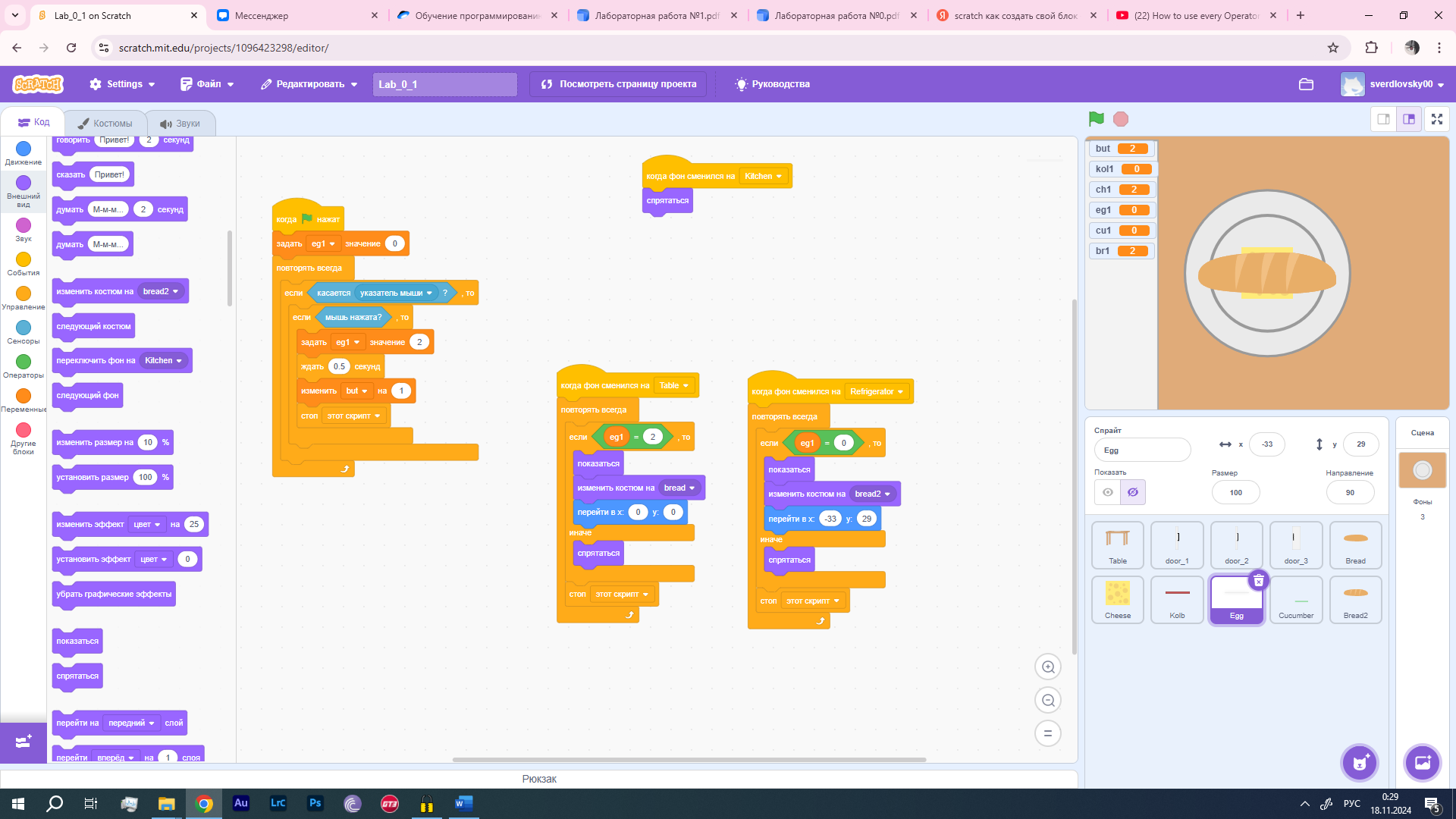


Рисунок 16 – Код для спрайта «яйцо»

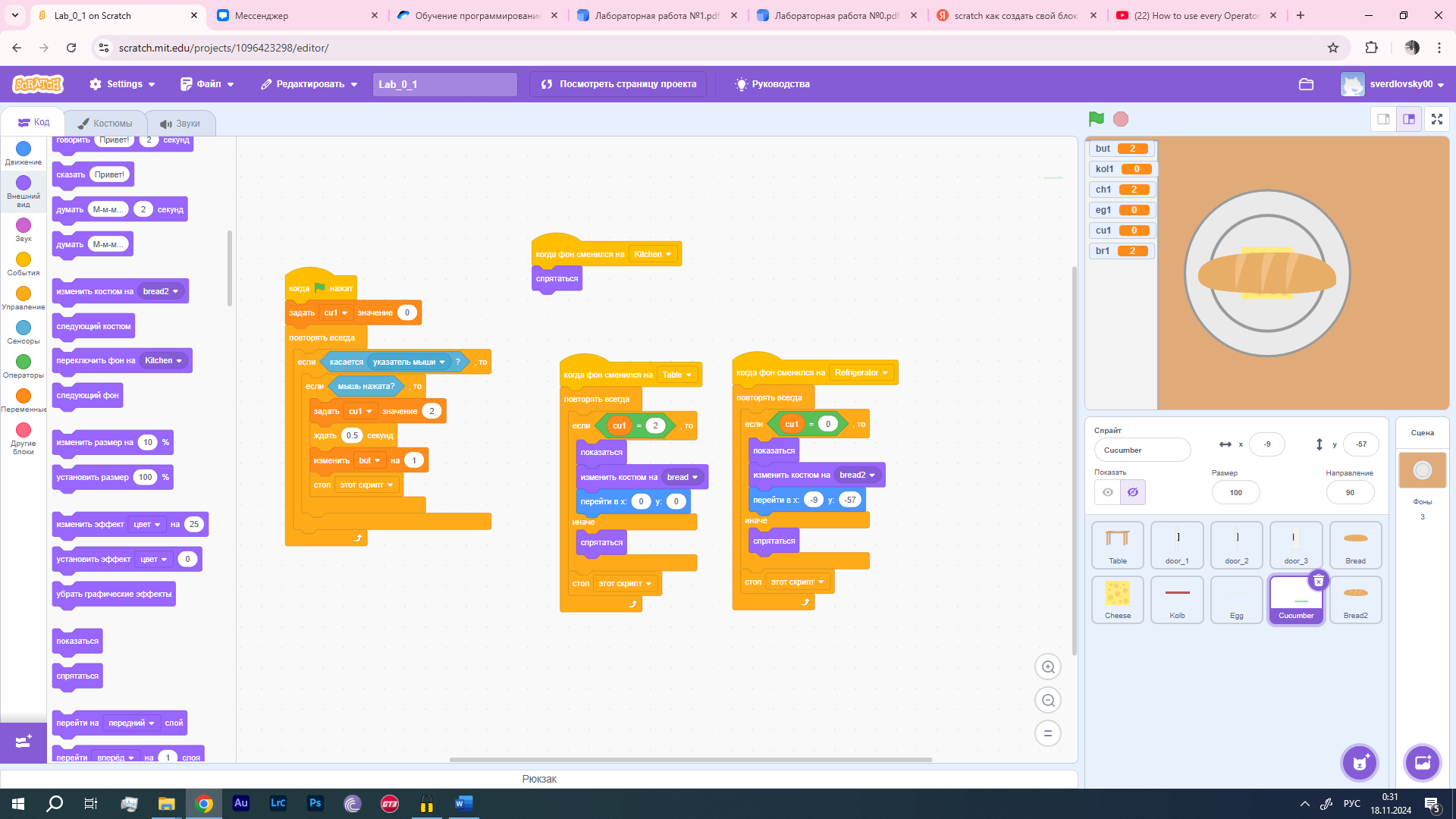


Рисунок 17 – Код для спрайта «огурец»

Для спрайта «хлеба» мы вводим проверку правильности нашего бутерброда. Если мы выбрали хотя бы два предмета (хлеб и что-то еще), сверху у нас появится еще один хлеб и игра выдаст уведомление «Бутерброд получился!». Код для спрайта хлеба представлен на рисунке 18.

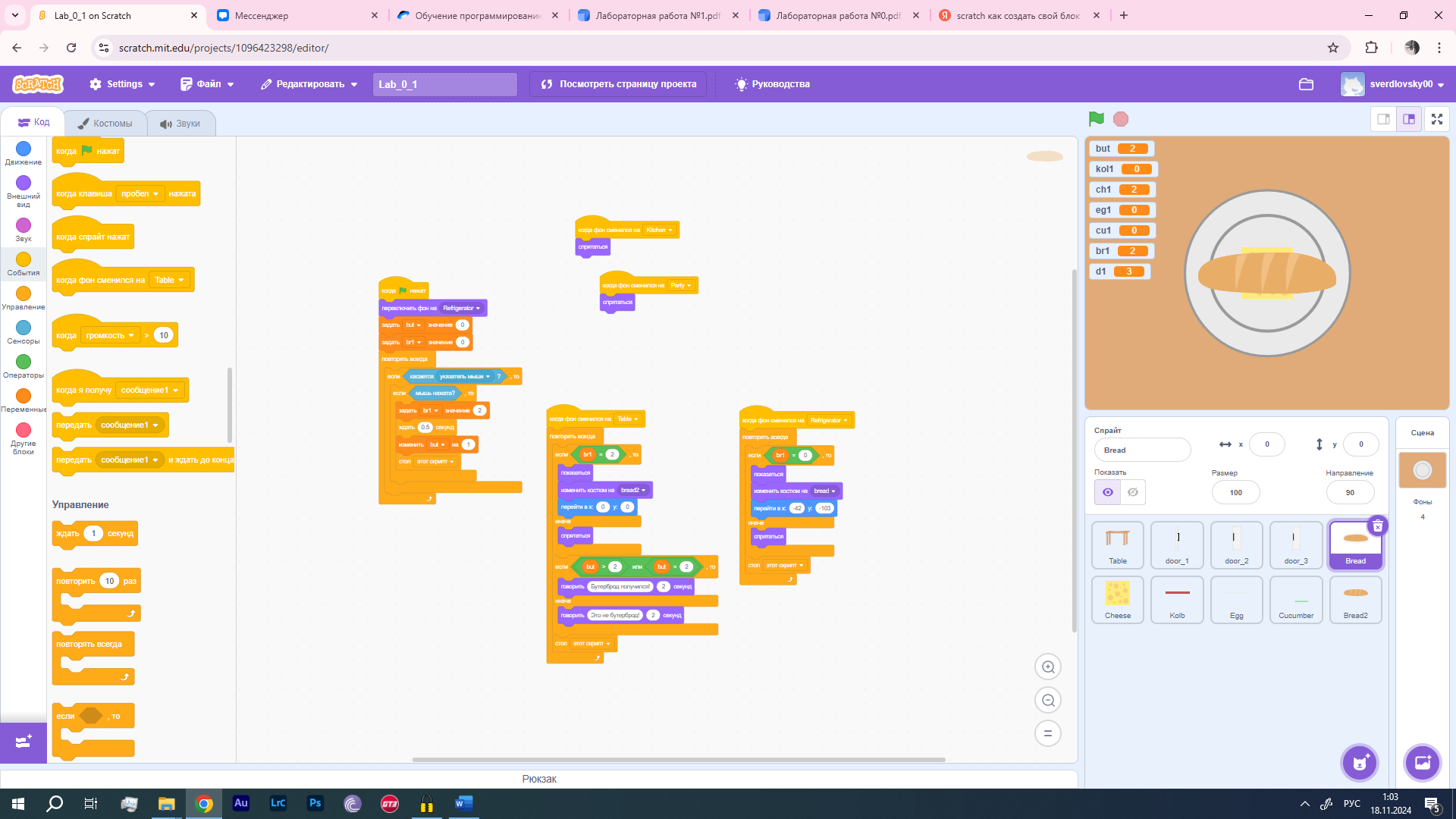


Рисунок 18 – Код для спрайта «хлеб»

Далее пишем программу для проверки, закрыты ли дверцы холодильника или нет. Если дверцы холодильника закрыты – произойдет переход к экрану победы, если этого не случится – игра будет ждать их закрытия. Программа проверки представлена на рисунке 19.

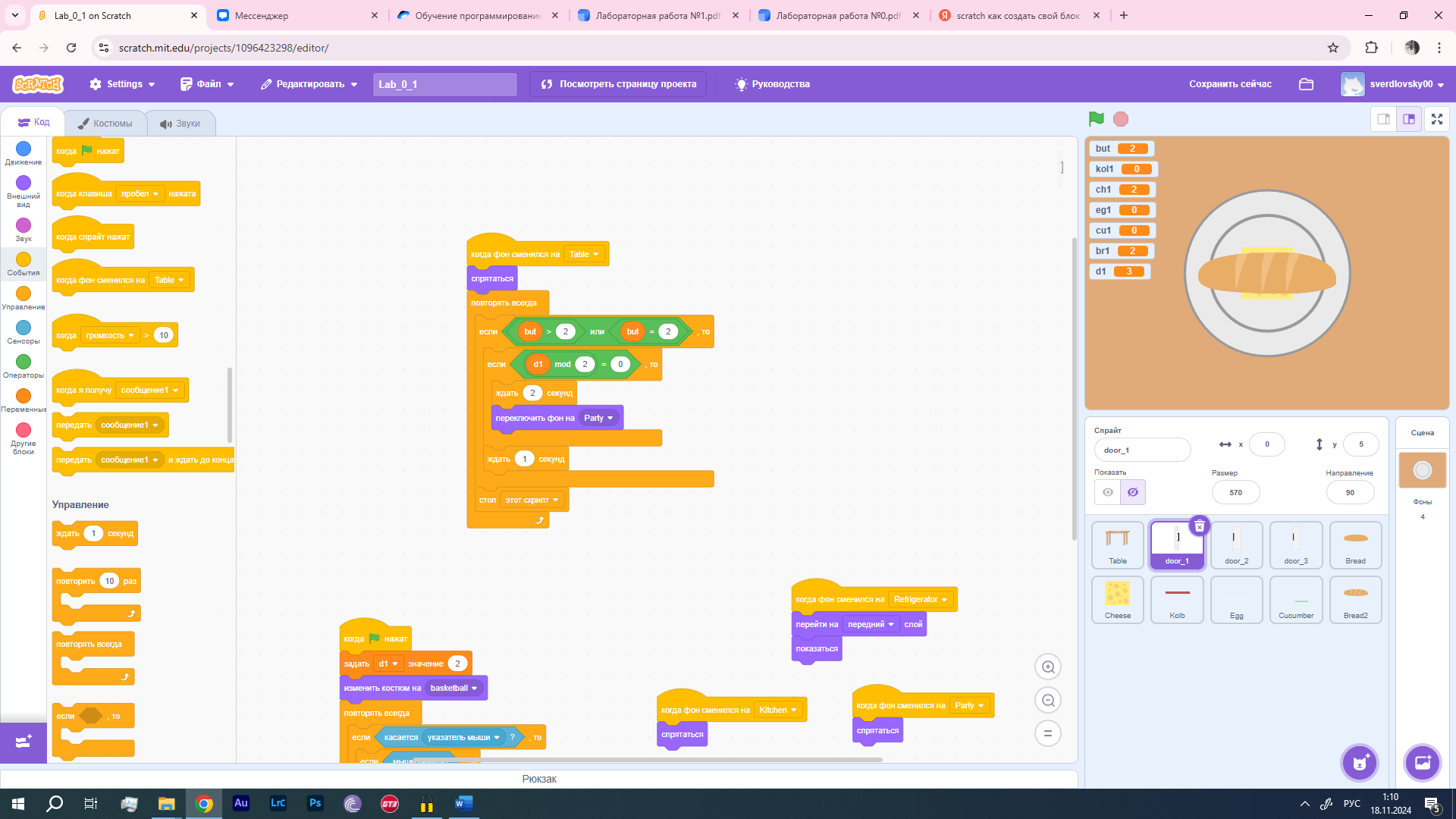


Рисунок 19 – Программа проверки дверец холодильника

**Задание 3. Своя игра**

Задача: сделать свою игру с уровнями сложности.

Для начала мы создаем сцены уровней. Нарисованные сцены представлены на рисунке 20.

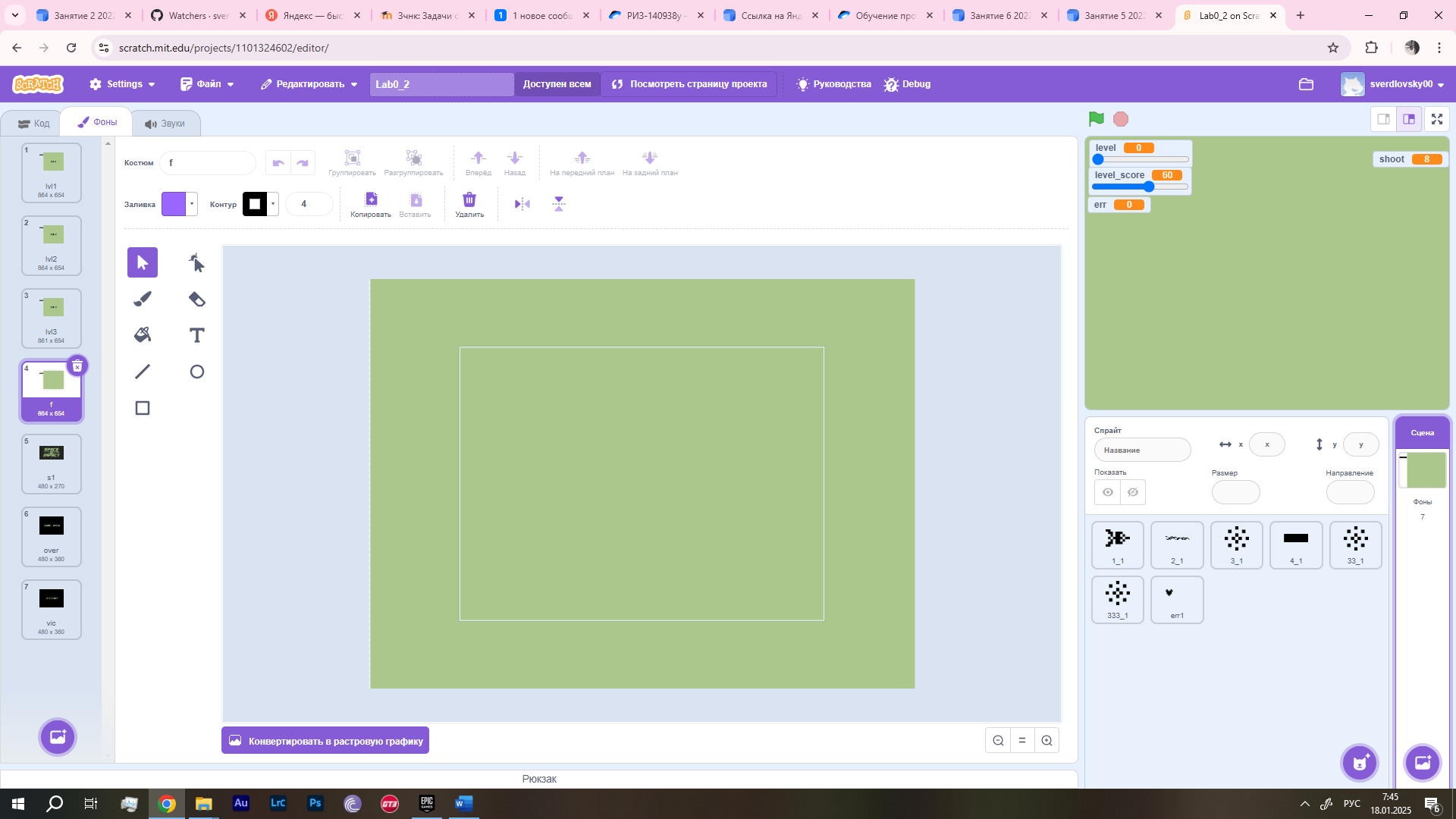


Рисунок 20 – Сцены для игры

Управляемый нами персонаж – «самолетик». Для него инициализируем управление (клавиши W,A,S,D), вносим в код условие, что если он столкнется с объектами окружения – он пропадет и после «респавна» будет мигать какое-то время. Также прописываем скрытие при смене фонов. Его код показан на рисунке 21.

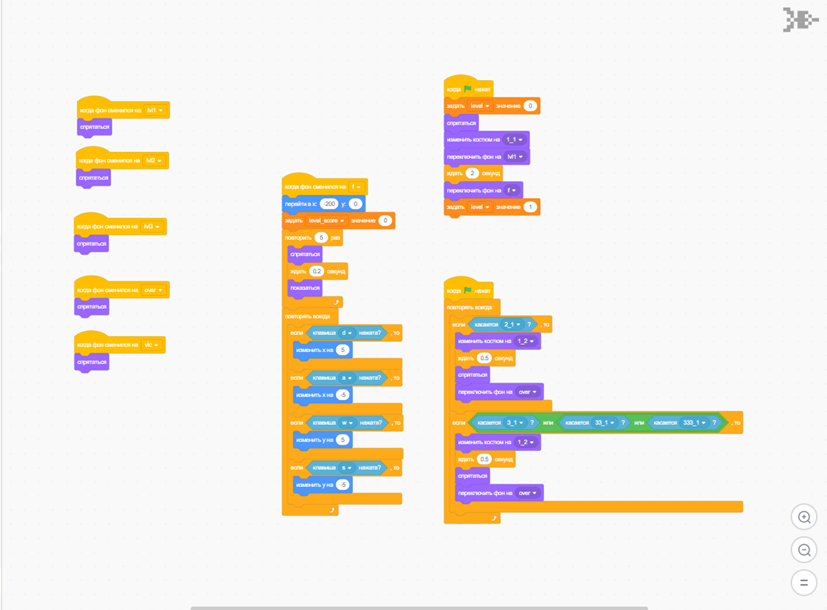


Рисунок 21 – Код для «самолетика»

Первый объект окружения – рельеф. Он имеет несколько костюмов. При полете костюмы будут меняться, создавая видимость движения этого рельефа. Также реализовано скрытие при переключении сцен. Код для него представлен на рисунке 22.

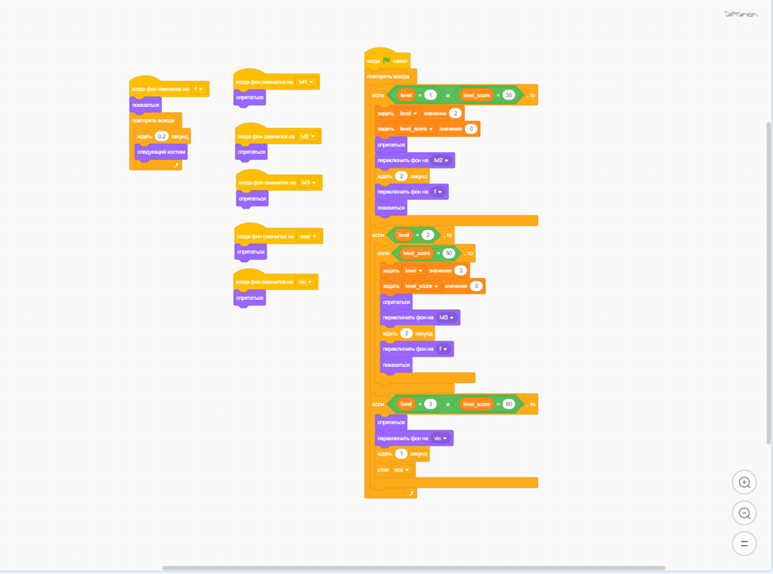


Рисунок 22 – Код для «рельефа»

Следующий объект – «самолет» противника. Они имеют также доп. костюм, который применяется чтоб показать его уничтожение. После уничтожения будет проходить какое-то время и он будет «респавниться» в случайной точке. Код для него представлен на рисунке 23.

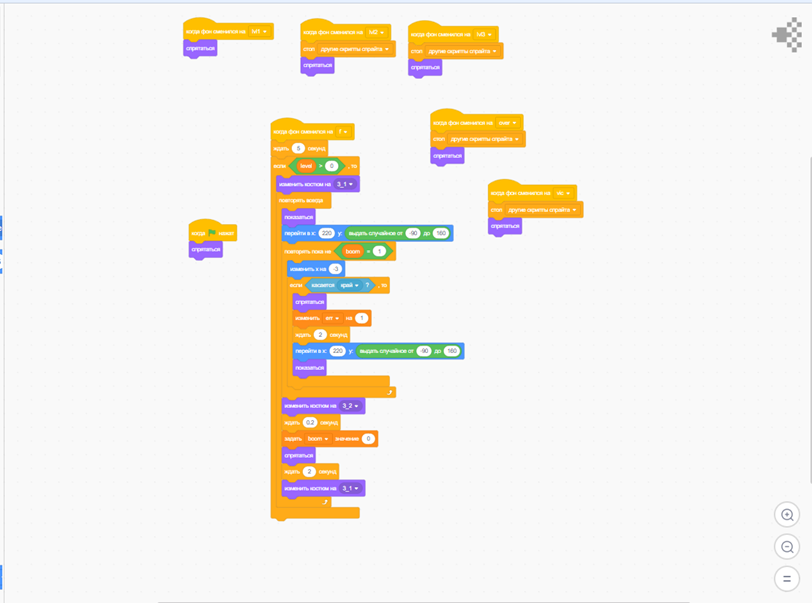


Рисунок 23 – Код для первого вражеского «самолета»

В игре имеется смена уровня сложности, поэтому реализовано изменяющееся количество «врагов». Поэтому, прошлый объект был дважды продублирован и для них написан дополнительный код, чтобы реализовать их появление и смену траекторий в зависимости от уровня. Код для «самолетов» 2 и 3 представлены на рисунке 24 и 25.

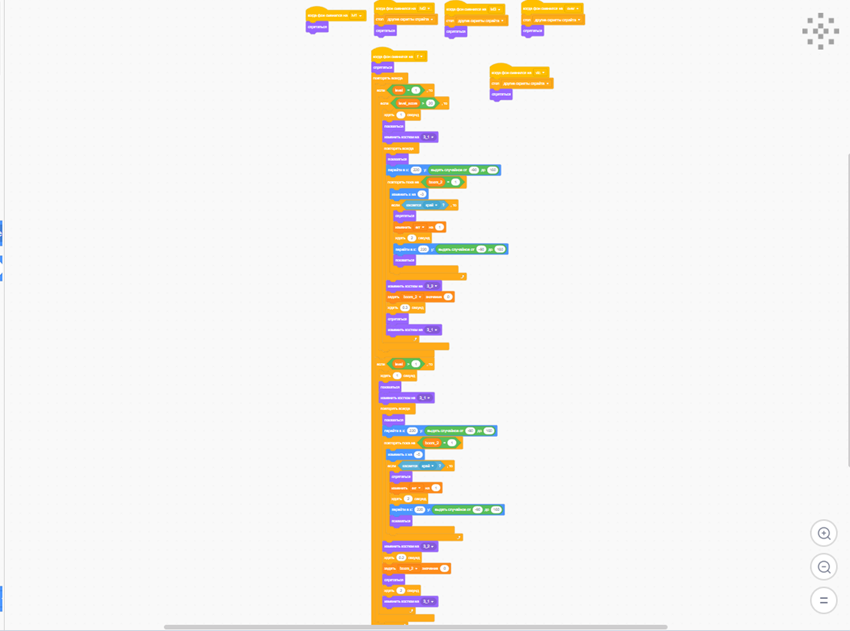


Рисунок 24 – Код для первого вражеского «самолета» 2

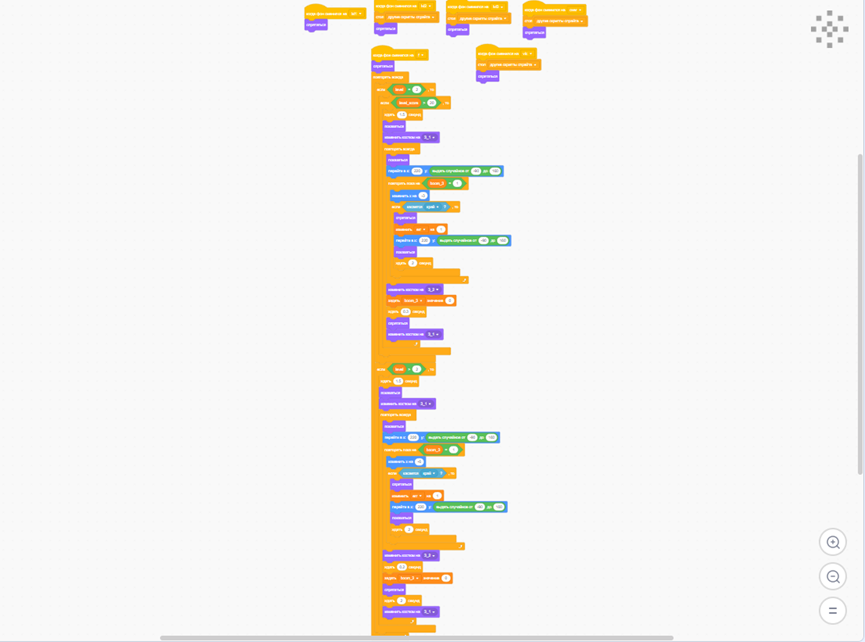


Рисунок 25 – Код для первого вражеского «самолета» 3

Также в игре реализована система жизней. Всего чтобы пройти игру дается три попытки. Спрайт будет менять количество «сердечек» показывая оставшееся количество попыток. Изначально на прохождение игры полностью дается три попытки. Код для них представлен на рисунке 26.

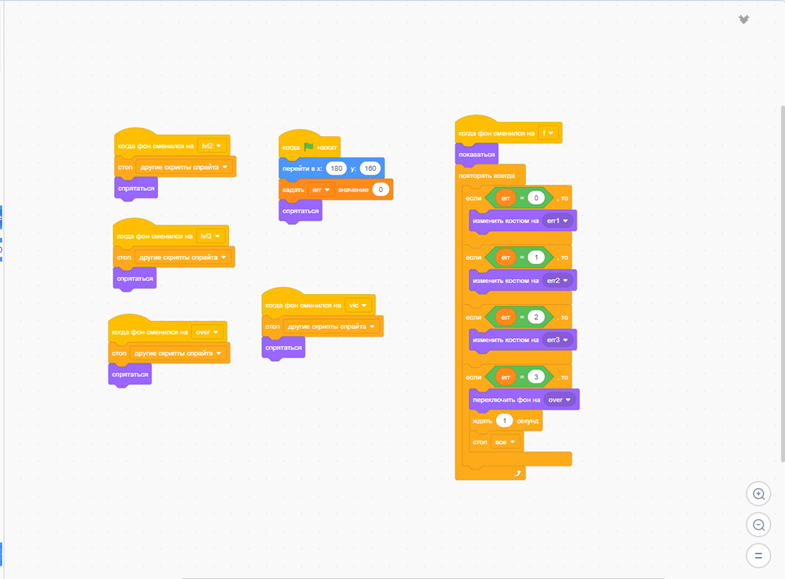


Рисунок 26 – Код для спрайта «сердечек»

Также имеется спрайт снаряда. Снаряд, при столкновении с «врагом», будет его разрушать. Это условие требуется продублировать для всех трех объектов «врагов». Код для него предсталвен на рисунке 27.

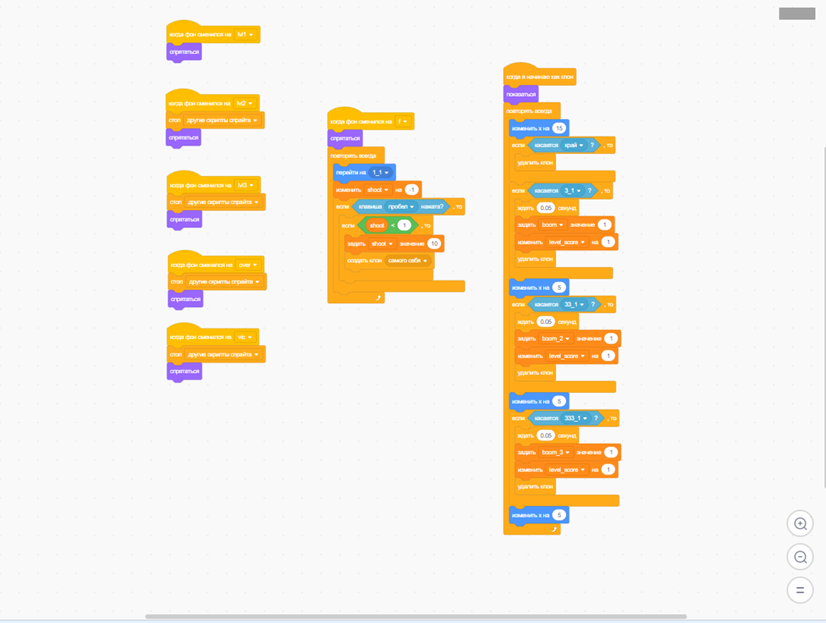


Рисунок 27 – Код для спрайта снаряда

Вывод: в результате работы были оттренированы навыки программирования на абстракции с применением платформы Scratch.