

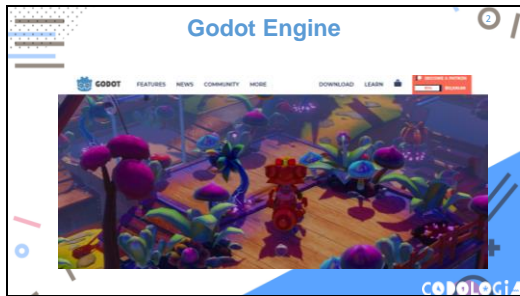
Слайд 1



Сегодня мы начнем новый курс по программе Godot. Godot Engine — открытый кроссплатформенный 2D и 3D игровой движок под лицензией MIT, который разрабатывается сообществом Godot Engine Community.

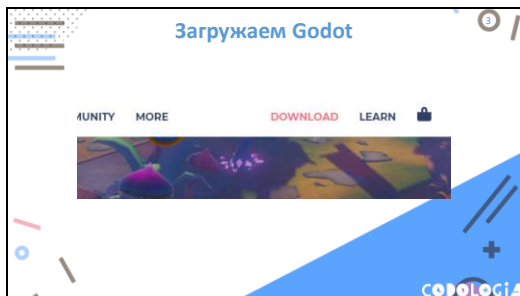
И для начала мы создадим простую 2-мерную игру, которая называется «Поймай яйцо». Есть корзина, которую мы можем перемещать с помощью мыши. Яйца будут сбрасывать сверху, мы должны поймать яйцо в корзине.

Слайд 2



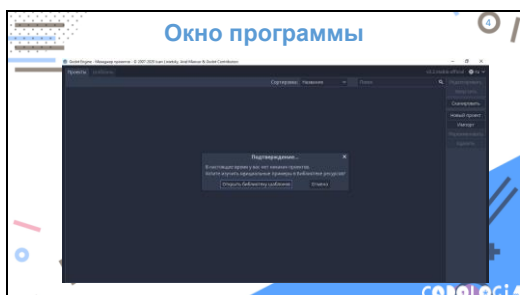
Открываем сайт <https://godotengine.org/>

Слайд 3



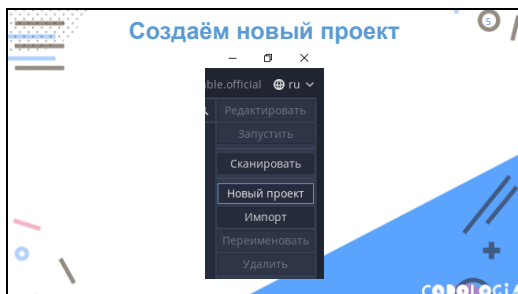
Нажимаем кнопку Download и устанавливаем программу

Слайд 4



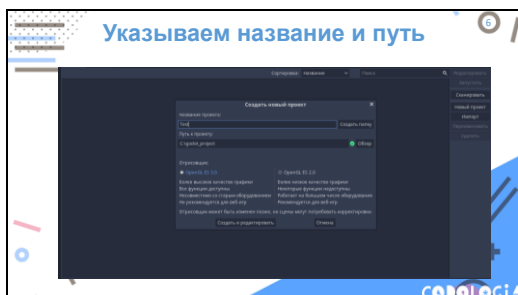
Когда вы запускаете Godot, первое окно, которое вы увидите, это Менеджер объектов. Это позволяет создавать, удалять, импортировать или играть в игровые проекты.

Слайд 5



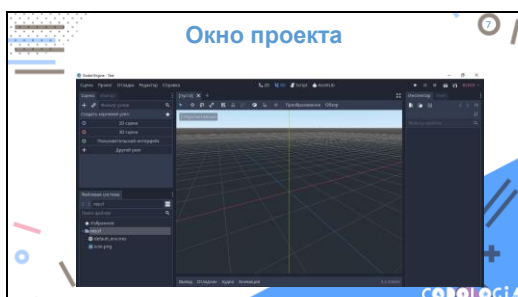
Нажимаем кнопку «новый проект»

Слайд 6



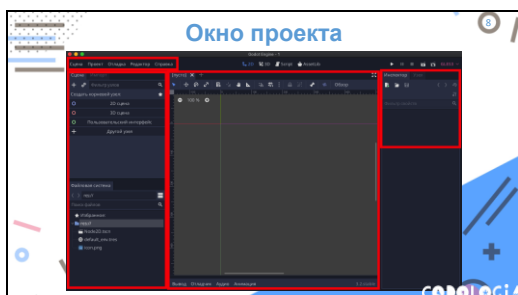
Указываем папку для сохранения проекта и его название.

Слайд 7



Когда ваш проект открыт, вы должны увидеть интерфейс редактора с трехмерным окном просмотра. Вы можете изменить текущее рабочее пространство в верхней части интерфейса. Нажмите на 2d, чтобы переключиться на 2d рабочее пространство.

Слайд 8

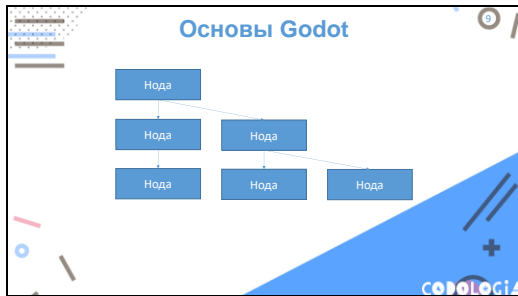


Теперь вы должны увидеть этот интерфейс с пустыми доками на правой стороне. Вверху слева направо вы можете увидеть основное меню, рабочие области и кнопки воспроизведения проекта. На левой стороне у вас есть Док-станция где вы будете управлять файлами проекта и активами. В центре вверху у вас есть Панель инструментов, где вы найдете инструменты для перемещения, масштабирования или блокировки объектов вашей сцены.

Вы можете видеть четыре кнопки рабочего пространства вверху: 2D, 3D, Script и AssetLib. Script - это полный редактор кода с отладчиком и расширенным автозаполнением.

AssetLib - это библиотека с бесплатными дополнениями, скриптами и ресурсами для использования в ваших проектах.

Слайд 9

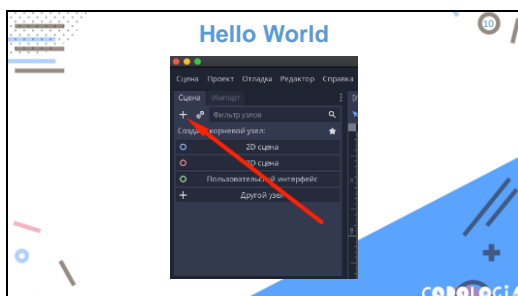


Общая архитектура движка построена вокруг концепции дерева из наследуемых «сцен». Каждый элемент сцены (нод - узел), в любой момент сам может стать полноценной сценой. Поэтому при разработке можно легко изменять полностью всю архитектуру проекта, расширять её элементы в любую сторону и работать с комплексными сценами на уровне простых абстракций. В Godot возможность упорядочивать сцены таким образом создает мощный инструмент для организации проектов. Поскольку разные сцены имеют разные функции, объединяя их можно создавать более сложные функций.

Запуск игры означает запуск сцены. Проект может содержать несколько сцен, но для запуска игры одна из них должна быть выбрана в качестве главной сцены.

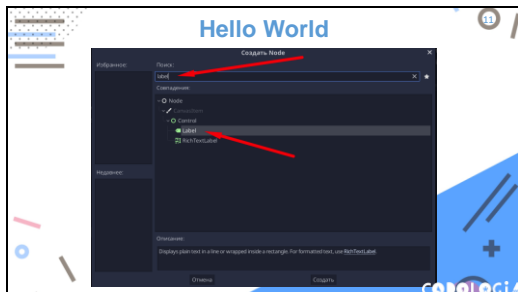
По сути, редактор Godot является редактором узлов. Он имеет множество инструментов для редактирования 2D и 3D сцен, а также пользовательские интерфейсы, но редактор основан на концепции редактирования сцены и узлов, которые ее составляют.

Слайд 10



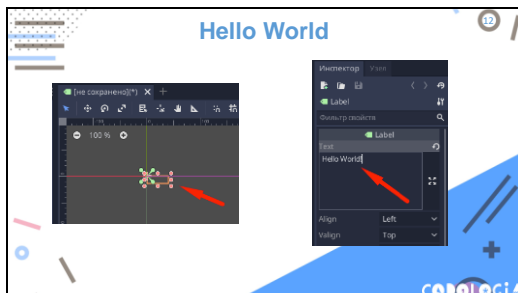
По традиции всех разработчиков, знакомство с новым языком программирования или программой мы начнем с создания программы «Hello World!». Нажмите на плюс для создания нового узла.

Слайд 11



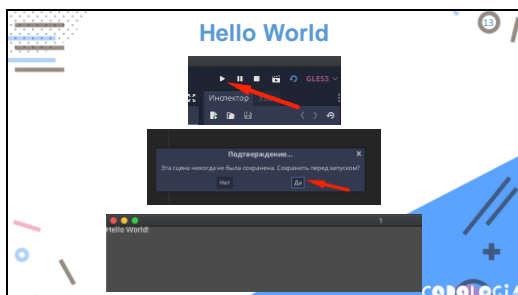
Открывается диалоговое окно «Создать узел», в котором показан длинный список узлов, которые можно создать. Наберите «Label» (метка - надпись) для более быстрого поиска.

Слайд 12



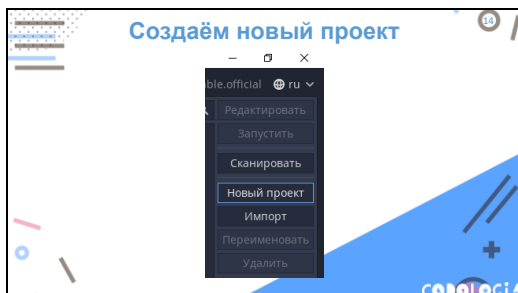
Узел появляется в редакторе дерева сцен (поле в правом верхнем углу), а свойства метки отображаются в Инспекторе (поле в правом нижнем углу). Следующим шагом измените свойство «Текст» и напишите «Hello World!»

Слайд 13



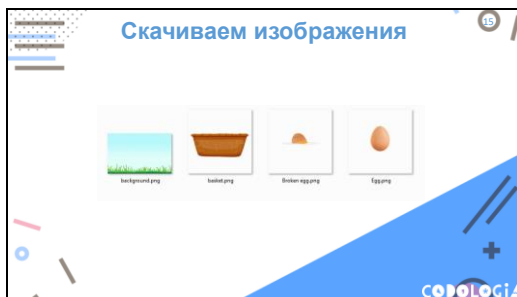
Хорошо, все готово для запуска сцены! Нажмите кнопку PLAY SCENE сверху (или F6). И ура! Вы увидите диалоговое окно с надписью «Hello World!». Это успех!

Слайд 14



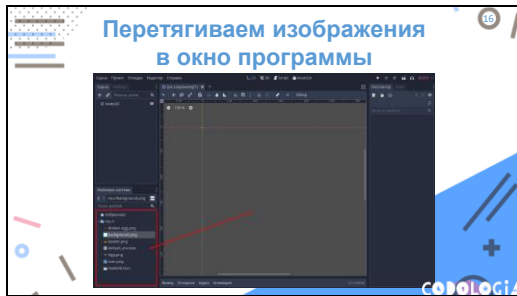
Приступим к созданию новой игры, которую мы запланировали сделать. Создаем новый проект.

Слайд 15



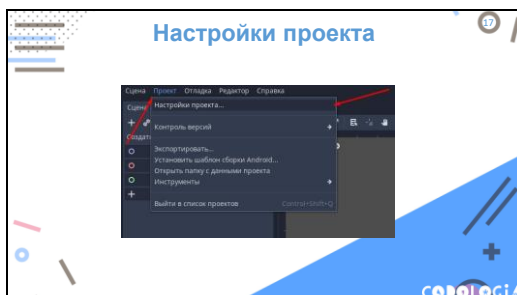
Скачиваем картинки фон, яйцо, корзину. Можно воспользоваться сайтом <https://www.freepik.com/>

Слайд 16



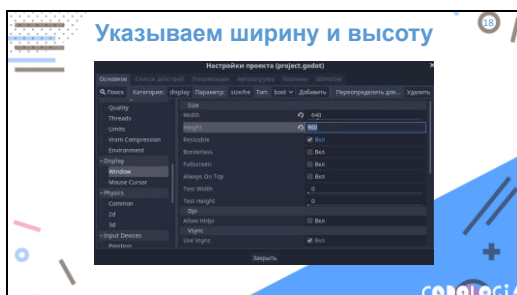
Перетягиваем файлы-изображения сюда

Слайд 17



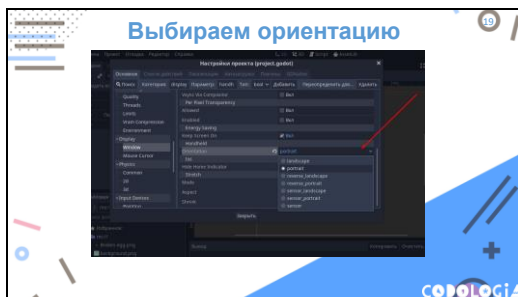
Нажимаем на настройки проекта

Слайд 18



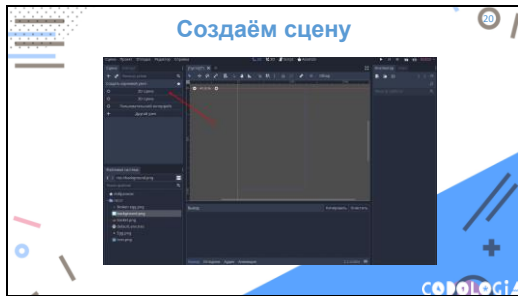
Устанавливаем параметры ширины и высоты

Слайд 19



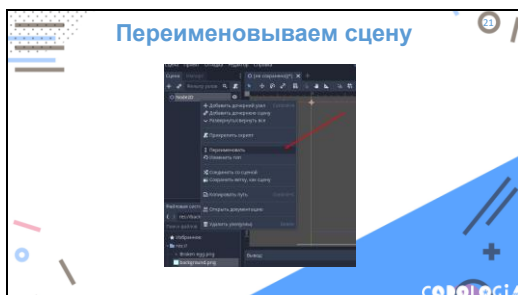
Пролистываем ниже и выбираем портретную ориентацию

Слайд 20



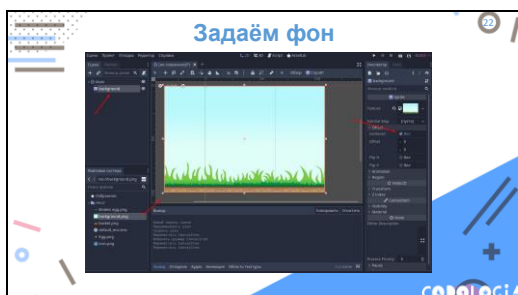
Создадим корневой узел, кликнув на кнопку «2D сцена»

Слайд 21



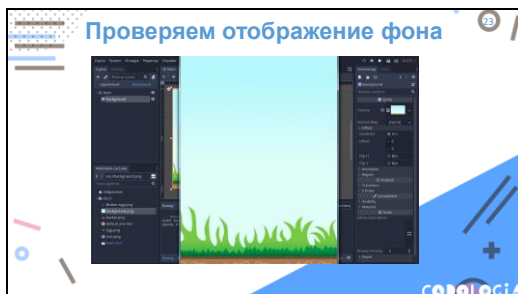
Переименуем корневую сцену и назовём её «Main»

Слайд 22



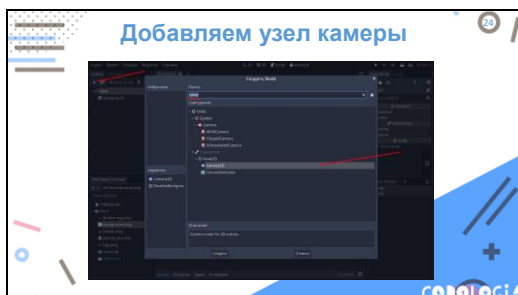
Теперь перетянем наш фон в главное окно и растянем его при необходимости так, чтобы он умещался в вертикальный прямоугольник на сцене. Также нужно нажать на него левой кнопкой мыши в списке сцен слева, чтобы появилось окно параметров справа. Выберем «Offset» и убедимся, чтобы параметр «Centered» был включен

#### Слайд 23



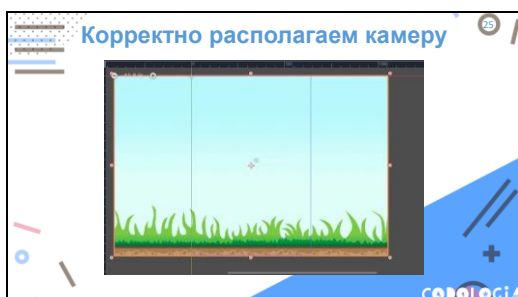
Нажмём F5, чтобы убедиться, что фон отображается корректно. Когда программа попросит нас установить главную сцену, выбираем нашу сцену «Main»

#### Слайд 24



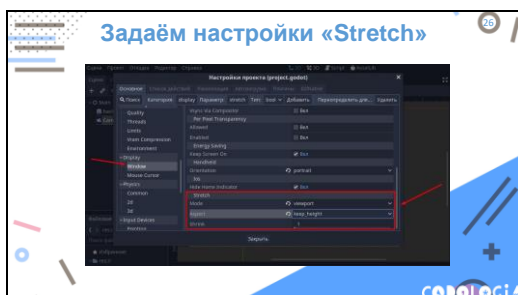
Закроем окно игры. Теперь нужно нажать на «Main» в списке сцен и создать узел «Camera2D», нажав на кнопку добавления узла. Чтобы найти этот узел, начнём писать в поиске его название

#### Слайд 25



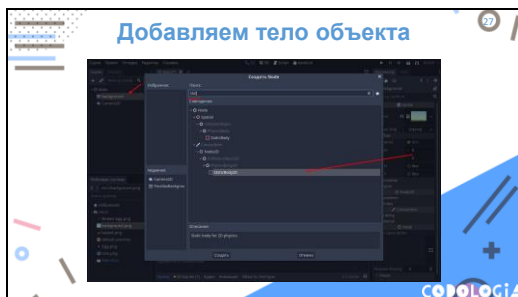
Переместим прямоугольник камеры так, чтобы он совпадал с главным прямоугольником нашей сцены

#### Слайд 26



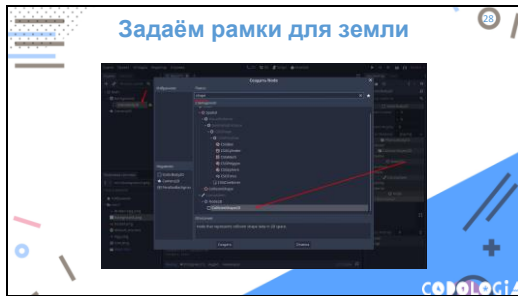
Снова зайдём в настройки проекта, категория «Display». Здесь в пункте «Stretch» выберем такие настройки

Слайд 27



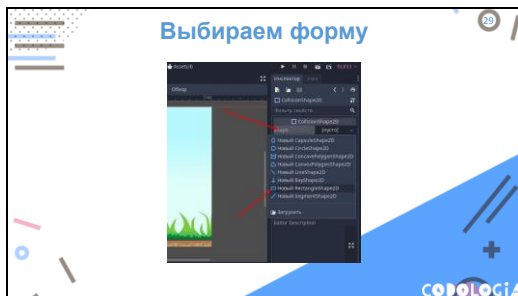
Нажмём на узел «Background» и создадим дочерний узел «StaticBody2D». Это делается для того, чтобы мы могли придать земле на нашем фоне физические свойства

Слайд 28



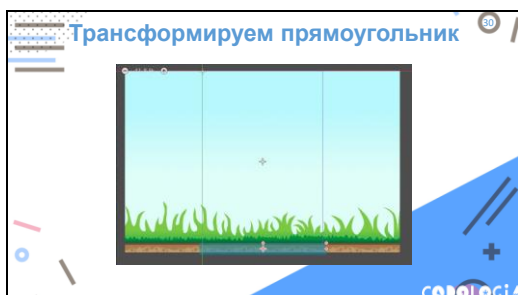
А также создадим дочерний «CollisionShape2D», чтобы задать рамки, в которых земля будет иметь физические свойства

Слайд 29



В пункте «Shape» выберем форму прямоугольника

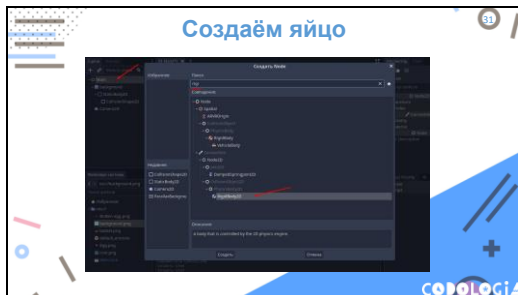
Слайд 30



Зададим прямоугольнику правильный размер

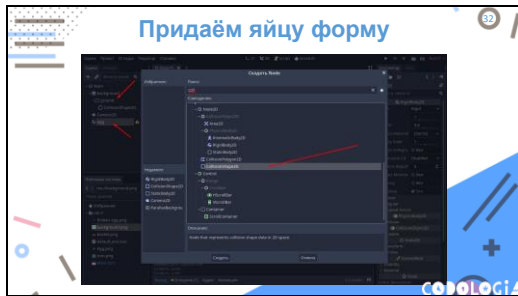


Слайд 31



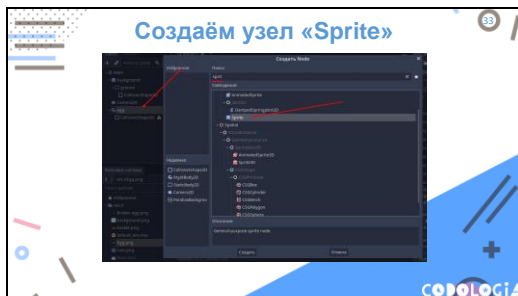
Приступим к созданию яйца. Создадим узел «RigidBody2D», как дочерний сцене «Main»

Слайд 32



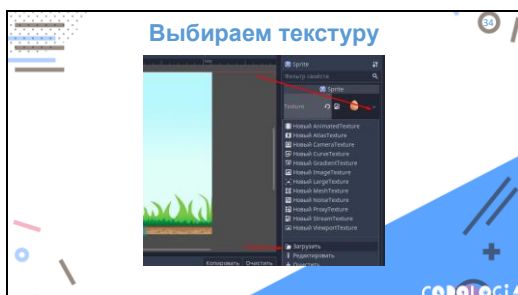
Чтобы было более понятно, давайте переименуем созданные объекты в «ground» и «egg». После чего зададим форму для яйца, создав «CollisionShape2D», как дочерний, для узла «egg»

Слайд 33



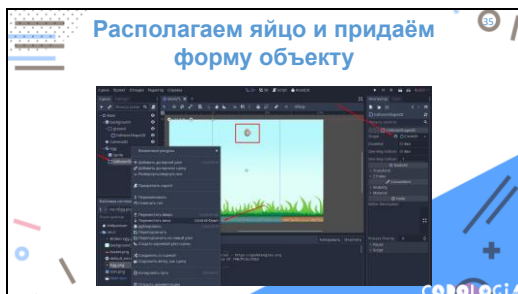
Создадим дочерний узел «Sprite» для узла «egg»

Слайд 34



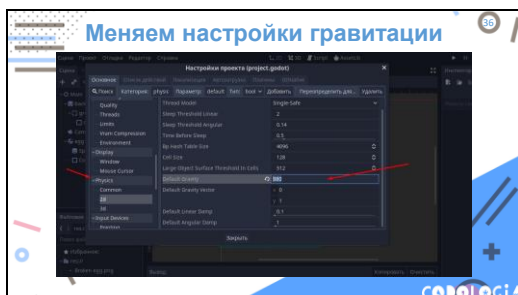
Выберем текстуру для яйца, загрузив его изображение с помощью параметра «Texture»

#### Слайд 35



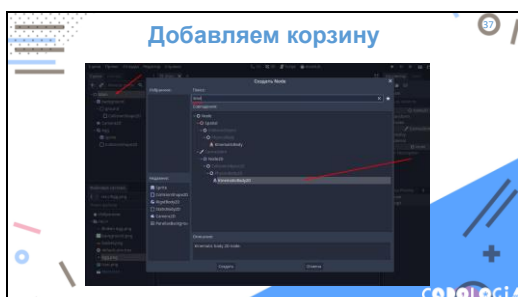
Теперь расположим наше яйцо в верхней части игры по центру, переместим вниз узел «CollisionShape2D» и зададим ему форму, назначив «Shape» в панели справа

#### Слайд 36



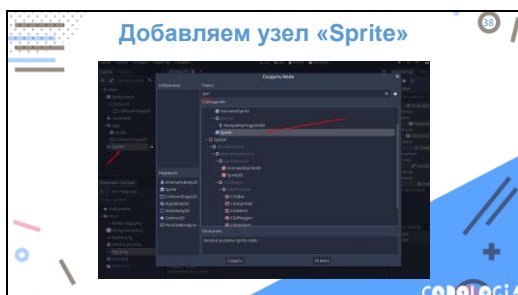
Если нажать F5, можно увидеть, что яйцо падает на землю, но слишком медленно. Чтобы это исправить, в настройках проекта зададим параметру «Default Gravity» значение 980

#### Слайд 37



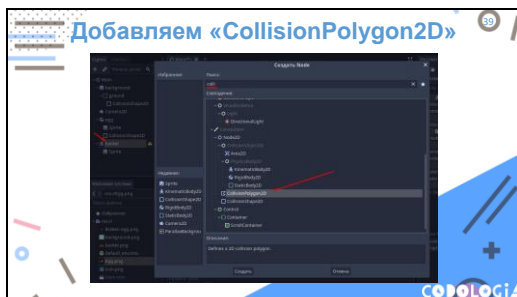
Пришло время добавить корзину. Она будет двигаться и контролироваться игроком, поэтому выбираем тип узла «KinematicBody2D» и переименовываем его в «basket». Не забываем, что этот узел должен быть дочерним для «Main»

#### Слайд 38



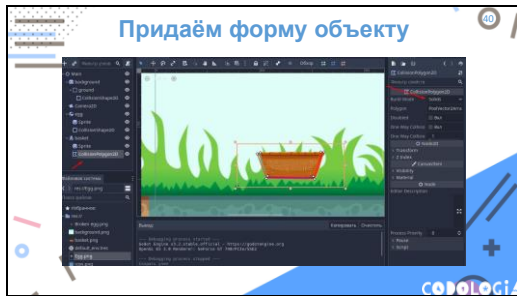
Теперь создадим для узла «Basket» дочерний узел «Sprite» и выберем текстуру нашей корзины

Слайд 39



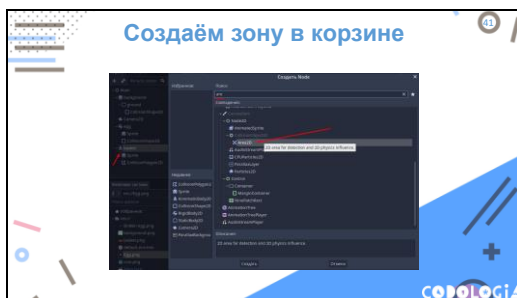
Далее нам необходимо создать «CollisionPolygon2D», дочерний узлу «basket», чтобы задать форму корзине

Слайд 40



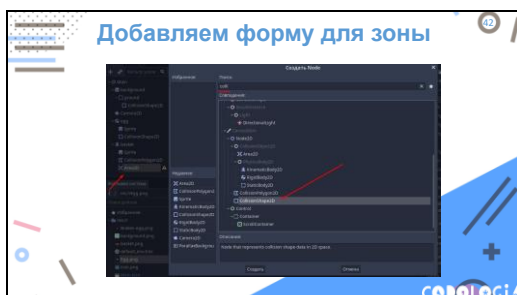
Колесиком мышки приблизим сцену и придадим корзине форму с помощью левой кнопки мыши

Слайд 41



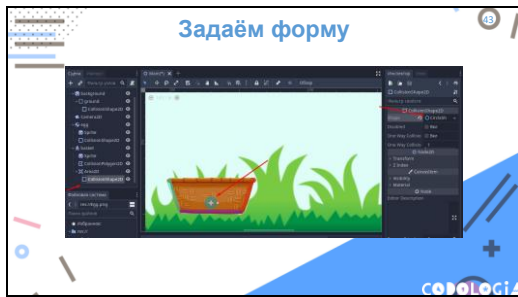
Когда яйцо падает в корзину, мы должны его удалять, поэтому создадим узел «Area2D», дочерний узлу «basket»

Слайд 42



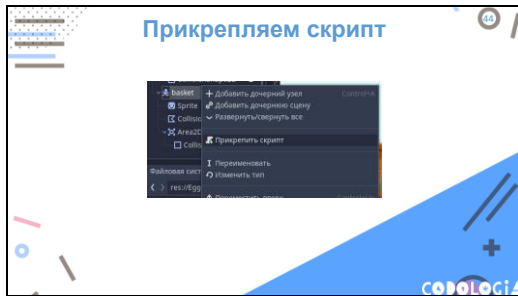
Далее создадим «CollisionShape2D», дочерний узлу «Area2D»

Слайд 43



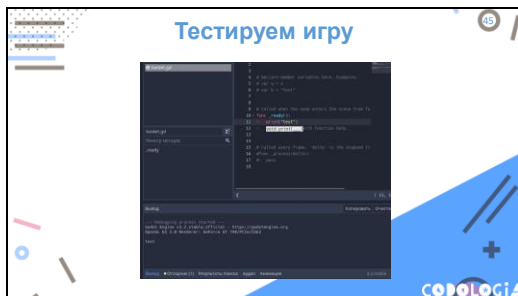
Зададим ему форму круга и поместим в корзину

Слайд 44



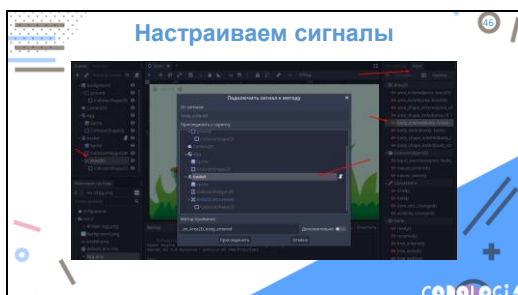
Дальше нам необходимо прикрепить скрипт к узлу «basket»

Слайд 45



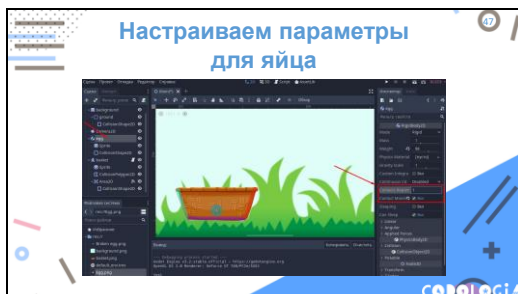
Вставим в функцию «\_ready()» строку: «print("test")», нажмём F5 и увидим, что в окне вывода появилось слово «test». Значит, всё работает корректно

Слайд 46



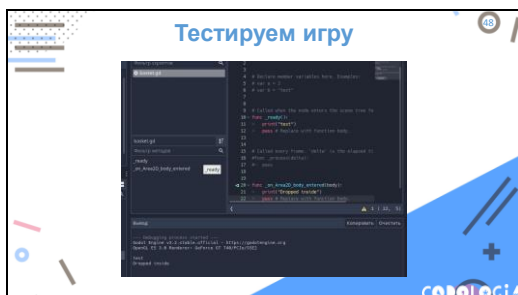
Пора приступить к настройке сигналов. Выберем узел «Area2D», справа откроем вкладку узел, найдём сигнал «body\_entered», нажмём на него правой кнопкой мыши и присоединим к узлу «basket»

Слайд 47



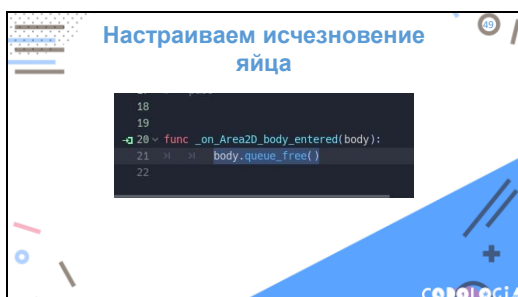
Теперь необходимо нажать на узел «egg» и в параметрах установить «Contact Monitor» на «Вкл», а «Contacts Report» на единицу

Слайд 48



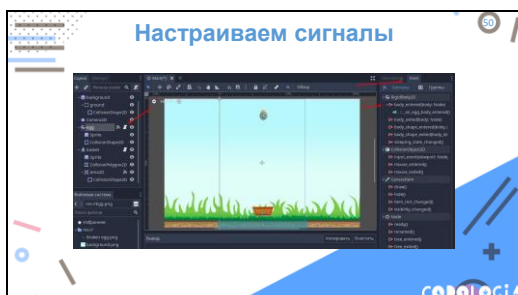
Таким образом, добавив строку «print(“Dropped inside”)» в функцию «\_on\_Area2D\_body\_entered(body)» мы можем проверить, распознаёт ли игра падение яйца в корзину. Для этого нажмём F5 и проверим окно вывода – появилась строка «Dropped inside»

Слайд 49



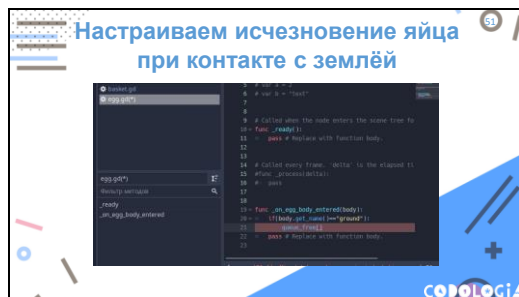
Теперь заменим содержимое функции на строку «body.queue\_free()». С помощью F5 можем увидеть, что яйцо при попадании в корзину исчезает. Это нам и нужно

Слайд 50



Теперь настроим связи так, чтобы яйцо пропадало и при падении на землю. Создадим скрипт для узла «egg» и присоединим сигнал «body\_entered» к узлу «egg»

Слайд 51



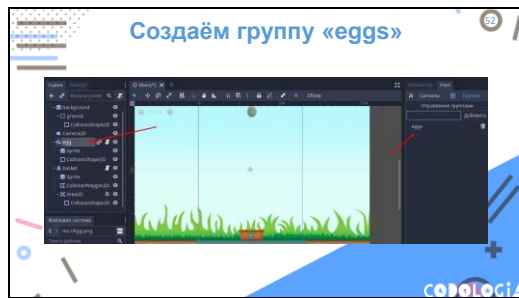
В скрипте узла «egg» вставим условие:

«if(body.get\_name()=="ground"):

queue\_free()»

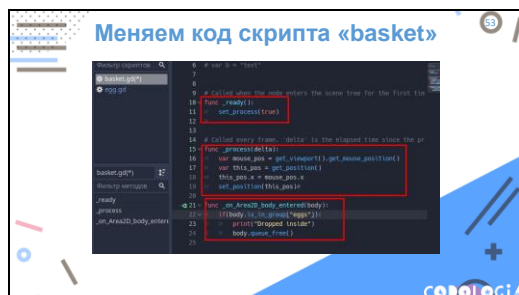
в функцию «\_on\_egg\_body\_entered(body)». Теперь яйцо исчезает при попадании на землю

Слайд 52



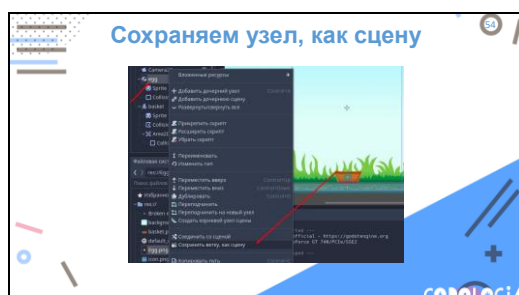
Для того, чтобы при изменении имени узла «egg» или добавлении новых узлов с этим же функционалом, яйца всё равно исчезали, добавим узел «egg» в группу «eggs»

Слайд 53

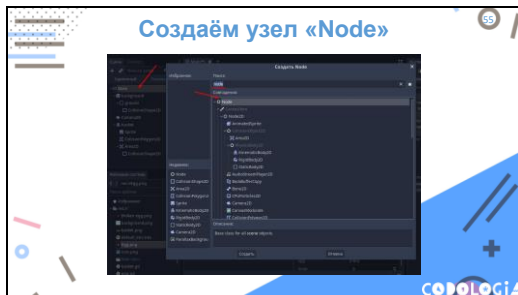


И изменим код в скрипте узла «basket» на следующий. Здесь мы добавляем отслеживание положения мыши, чтобы с помощью неё можно было передвигать корзину, а также добавляем определение объекта, который падает в корзину. Он должен принадлежать группе «eggs», которую мы создали ранее. Можно нажать F5 и убедиться, что всё работает корректно

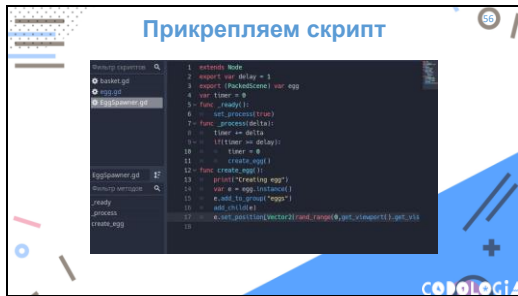
Слайд 54



Чтобы создавать несколько яиц, нам необходимо сохранить ветку «egg», как сцену, после чего можем удалить узел



Создадим новый узел «Node», как дочерний узлу «Main» и переименуем его в «EggSpawner»



Теперь нам нужно настроить появление яиц. Зададим задержку появления и настроим случайное место появления. Для этого прикрепим к узлу скрипт и вставим такой текст:

«extends Node

export var delay = 1

export (PackedScene) var egg

var timer = 0

func \_ready():

    set\_process(true)

func \_process(delta):

    timer += delta

    if(timer >= delay):

        timer = 0

        create\_egg()

func create\_egg():

    print("Creating egg")

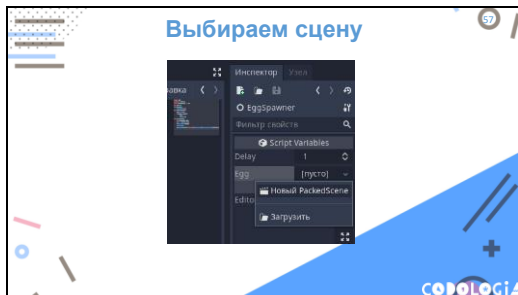
    var e = egg.instance()

    e.add\_to\_group("eggs")

    add\_child(e)

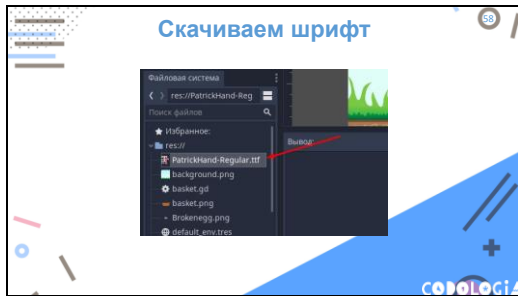
    e.set\_position(Vector2(rand\_range(0,get\_viewport().get\_visible\_rect().size.x),0))»

Слайд 57



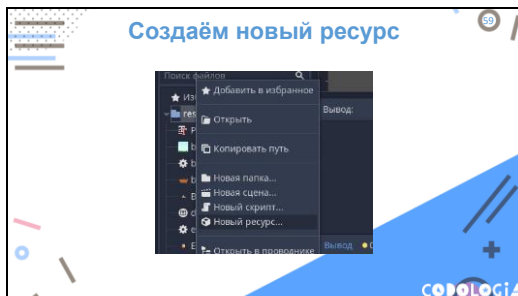
Теперь в панели справа выберем сцену «egg.tscn», нажав кнопку «Загрузить»

Слайд 58



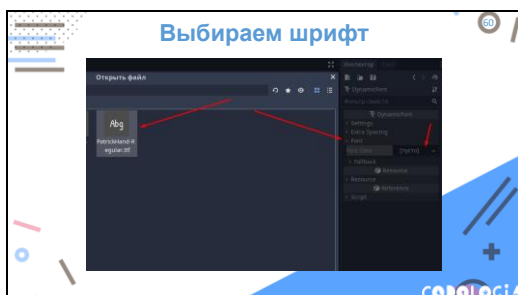
И последний штрих – добавим счётчик пойманных яиц. Для начала скачаем любой шрифт, например, этот: [https://yadi.sk/d/AFI7\\_TDYWNBCPg](https://yadi.sk/d/AFI7_TDYWNBCPg) и перенесём его в окно ресурсов (стандартному шрифту Godot нельзя менять размер)

Слайд 59



Создаём новый ресурс DynamicFont и называем его Font

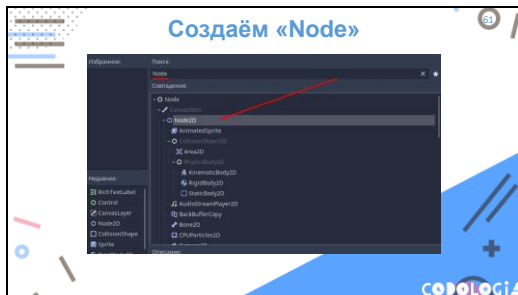
Слайд 60



Нажимаем на «Font» в окне ресурсов и во вкладке Font Data выбираем наш шрифт

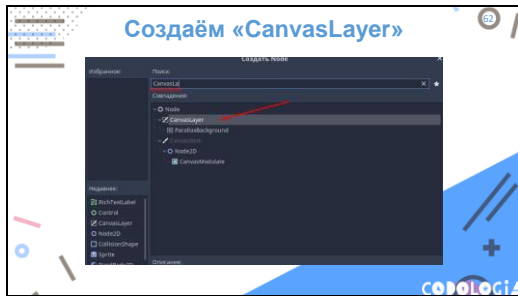


Слайд 61



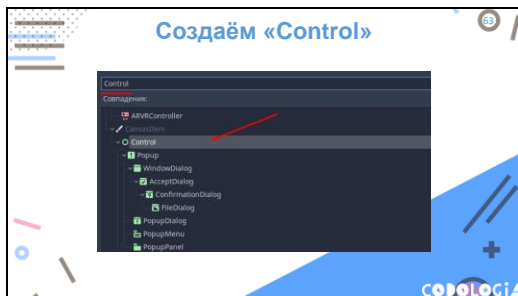
Создаём узел «Node», дочерний «Main», и называем его «ScoreCounter»

Слайд 62



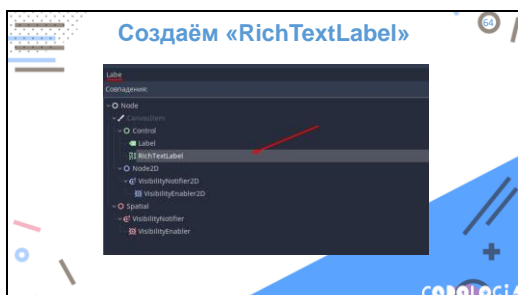
Дальше создаём дочерний ему «CanvasLayer» и называем его UI

Слайд 63



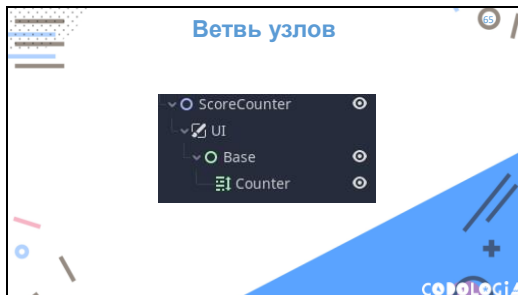
Теперь – дочерний узлу «CanvasLayer» узел «Control» и переименовываем его в «Base»

Слайд 64



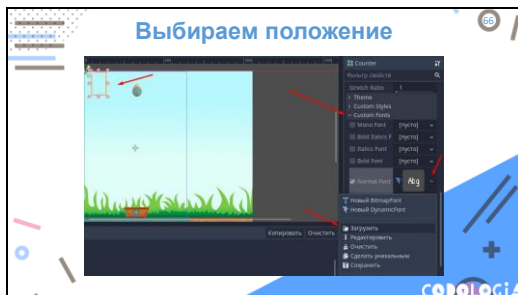
И наконец – дочерний узлу «Base» узел «RichTextLabel» и переименовываем его в Counter

Слайд 65



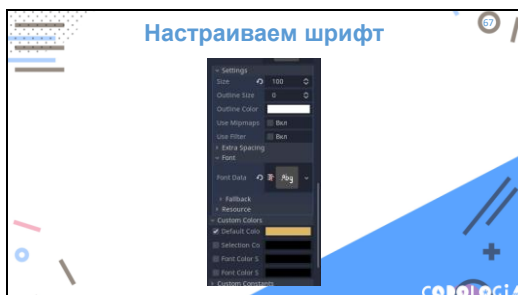
Убедимся, что наша ветвь узлов выглядит так

Слайд 66



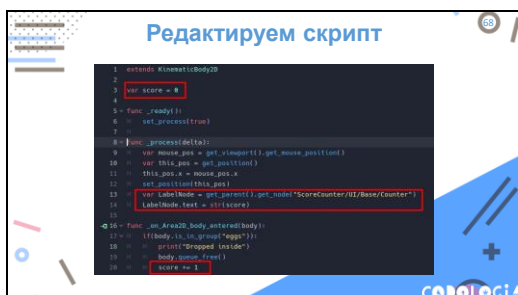
Расширим рамку и выберем положение, а также наш тип шрифта в разделе «Custom Fonts» - «Normal Font»

Слайд 67



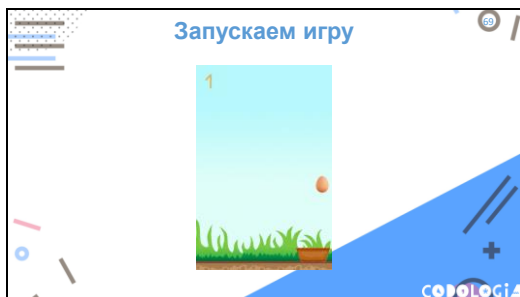
Здесь же делаем размер шрифта 100, выбираем любой цвет

Слайд 68



Теперь добавляем такие строки в скрипт «Basket». Сначала мы задаём переменную «score», которая обозначает счёт. Далее в функции «\_process(delta)» привяжем переменную к области текста, которую мы создали. И заключающий момент – пишем код, который прибавляет единицу к переменной «score» каждый раз, когда яйца попадают в корзину

Слайд 69



Нажимаем F5 и радуемся результату 😊

Слайд 70

