**Справка создана без помощи программ, т.к. код программы не комментировал, постарался описать классы и их функции.**

**Пакет классов Menu()** – включает в себя классы отвечающие за взаимодействие всех основных элементов главного меню.

**Классы TextFade()** отвечаю за весь текст, шрифт и его анимацию в игре.

Void Start – включает наш текст

Void Update – создает эффект мерцания текста

**Класс Shop BG()** - игровой магазин.

Void OnEnable() – появление нового окна магазина, отключение текста и кнопок проигрыша, запрет начала игры из магазина

Void OnDisable() - возврат всех настроек и отключение магазина

**Класс Buttons()** - отвечает за все кнопки в игре и последующие состояния системы при их нажатие.

Void Start() – Синхронизация с настройками вкл/выкл музыки

Void OnMouseDown()- увеличение кнопки меню

Void OnMouseUp()- переход кнопки к исходному размеру

Void OnMouseUpAsButton()- функции для всех кнопок по их нажатию

**Класс Buy Cube()** - отвечает за покупку куба при определенных условиях во внутриигрвоом магазине.

Void OnMousedown() – если количество игровой валюты >50, то можно купить текстуру для кубика, при удачном исходе сменить кнопку покупки на выбрать

**Класс Fall Cube()** - отвечает за падание кубика.

Void Start()- уничтожает игровой объект

Void Update()- немного перемещает его по оси У

**Класс Game Arrangement()** – класс отслеживающий все нажатия на экран приводящий к определенному состоянию системы в соответствие с местом нажатия.

25

Void Update()- Небольшое изменение освещения при старте игры

Void OnMouseDown()- запускает игру, скрывает и открывает текст, делает неактивным кнопки, делает активным рекорд и количество игровой валюты. Также делает небольшую задержку перед началом игры.

**Класс Righ Mat()** – класс отвечающий за правильность выбранной текстуры для главного кубика.

Void Start () – присвоение выбранного материала кубики

**Класс Scroll Shop()** – класс отвечающий за прокрутку кубиков в магазине.

Void Update() – если кубики сдвинуты за пределы границ, то возвращает их к определенному месту

Void OnMouseDown()-нажатие на мышку и запоминание текущей позиции, скрытие курсора

Void OnMouseDrag()- перемещение кубиков от места нажатия в сторону движения мышки и до места ее перемещения

Void OnMouseUp()- появление курсора

**Класс Select Main Cube()** – класс отвечающий за выбор текстуры в магазине для главного кубика.

Void OnMouseDown()- выбор текстуры для кубика

**Класс ScrollObjects()** –класс отвечающий за появление кнопок и их скорость появления

**Пакет классов** **Game** () – включает в себя классы отвечающие за взаимодействие всех основных элементов в процессе игры.

**Класс Collect Diamonds()** – класс отвечающий за сбор игровой валюты в процессе игры.

Void OnTriggerEnter() – класс отвечающий за взаимодействие кубика с игровой валютой при сборе в процессе игры, если они соприкоснулись, то увеличить игровую валюту и убрать ее с поля игры

26

**Класс Counts Diamonds()** – класс для подсчета количества игровой валюты и вывода ее во время игры.

Void Start() – вывод текста на экран

**Класс Cube Jump()** – класс отвечающий за прыжок кубика при нажатие и отпуске клавиши мыши в игре.

Void Awake()- обновление параметров для новой игры

Void Start() – запуск проверки на возможность прыжка

Void FixedUpdate()- запуск анимации сжатия кубика при нажатие на кнопку

Void PlayerLose()- запрет дальнейшей игры, вывод меню проигрыша

Void OnMouseUp()- расчет времени нажатия на кнопку для подсчета силы прыжка

Void OnMouseDown()- при удачном прыжке запуск анимации перемещения блоков и обнуление времени для следующего прыжка

**Класс Delet Block()** – класс для удаления пройденных блоков за выходом их из видимости с экрана.

Void OnTriggerEnter() – если платформа находится за экраном, то происходит удаление

**Класс Harder()** – класс для усложнения игры при прохождение определенного количества блоков.

Void Update() – Если пройдено 7 платформ, то происходит усложнение на один блок, а именно изменение позиции камеры в трехмерную

**Класс Main To Block()** – класс для окраски пройденных блоков в цвет главного кубика, при их взаимодействии.

OnCollisionEnter() – проверка при соприкосновении кубика с блоками, если удачно то окрасить блок в цвет кубика

**Класс Move Cubes()** – перемещение блоков при удачном прыжке.

Void Update ()- если кубик попал на следующий блок и остался там, то

27

сдвинуть блоки влево вверх

**Класс Rand BackGround()** – класс отвечающий за фон в игре.

Void Start() – выбор случайного материала для фона из заранее заданных

**Класс Rand Colors()** – класс отвечающий за расцветку кубиков в игре.

Void Start() – выбор случайного материала для кубиков в главном меню и блкоов из заранее заданных

**Класс Score()** – класс для подсчета и вывода результатов в игре.

Void Start()- обновить рекорд

Void Update() –обнулить счетчик и записывать туда достижение

**Класс Select Cube()** – выбор кубика в магазине и предоставление кнопок для покупки, выбора.

Void Start()- открыть в магазине для выбора начальный кубик

Void OnTriggerEnter()- изменить размер кубика, если он открыт показать кнопку выбрать, иначе кнопку купить

VoidOnTriggerExit() – вернуть начальный размер

**Класс Spawn Block()** – класс отвечающий за появление блоков в игре.

Void Start() – запустить генерацию блока

Void Update()- если кубик находится на другой платформе, то продолжить, если был совершен прыжок и кубик остался на той же платформе, то закончить, иначе закончить

RandScale()- выбор случайных координат из определенного диапазона для генерации блока

Void spawn ()- сгенерировать блок