



Рыбное единство — День 00

Активы, GameObject, поведение, ввод, преобразование

Персонал staff@staff.42.fr

Резюме: Этот документ содержит тему дня 00 для «Рыбного единства» из [42](#)

Содержание

| | | |
|------|---|----|
| I | Стартовый комплект Piscine Unity | 2 |
| II | Основные инструкции | 3 |
| III | Конкретные инструкции дня | 5 |
| IV | Упражнение 00: Симулятор воздушного шара 2015 | 6 |
| V | Упражнение 01: Событие Quick Time | 7 |
| VI | Упражнение 02: Мини-гольф | 9 |
| VII | Упражнение 03: Flappy Bird | 10 |
| VIII | Упражнение 04: Понг ! | 11 |

Глава I

Стартовый комплект Piscine Unity

Добро пожаловать в этот первый день Piscine Unity. В качестве хорошего начала найдите ниже небольшой список советов и другие ссылки, которые окажутся очень полезными на протяжении всей рыбалки.

- [Официальная документация Unity](#), вы найдете ссылки на каждый компонент в инспекторе, который направит вас к его документации.
- [Официальная документация по C#](#).
- Тема авторитетная, не всегда доверяйте демонстрациям, которые могут содержать дополнительные необязательный контент или функции.
- Если вы создадите общий и многообразный сценарий для всего утилитарного/несвязанного направления с игровым процессом дня, вы сэкономите много времени в последующие дни.
- Когда рабочий день закончится, не храните свой проект в своем профиле, Unity имеет тенденцию создавать миллиард файлов, и ваше пользовательское пространство (а также время входа в систему) резко увеличится.
- Дни длинные, не теряйте слишком много времени на детали во время упражнений. Будет всегда можно улучшить с игрой после завершения дня.
- Если ваш MonoDevelop не запускается, проверьте тему Piscine в ветке мегаконференции Unity, где будет объявлено о всем процессе.
- Если коллеги справа и коллеги слева не имеют ни малейшего представления о технической ПРОБЛЕМЕ, вы можете сообщить об этом на форуме с тегом Piscine Unity или в slack.
- Предполагается, что README следует читать так же, как и предисловия ...

Глава II

Основные инструкции

- Учебный курс Unity должен полностью и обязательно проводиться на C#. Никакого Javascript/Unityscript, Boo и прочих ужасов.
- Использование функций или трансформов, не разрешенных явным образом в заголовке упражнения или в правилах дня, будет считаться читерством.
- Для оптимального использования Unity вам необходимо работать с ~/goinfre, который находится на локальном диске вашего компьютера. Не забудьте сделать соответствующую резервную копию с помощью локального goinfre можно очистить.
- В отличие от любых других буткепов, каждый день не требует папки ex00/, ex01/, ..., vn/. Вместо этого вам нужно будет отправить папку вашего проекта, которая будет называться как день: d00/, d01/, Однако папка проекта по умолчанию содержит бесполезную папку: подпапку «projct/Temp/». Убедитесь, что НИКОГДА не пытаетесь поместить эту папку в свой репозиторий.
- Если вам интересно, во время этого буткепа не навязывается норма 42 для C#. Вы можете использовать любой стиль, который вам нравится, без ограничений. Но помните, что код, который нельзя прочитать или понять во время экзаменационной оценки, — это код, который нельзя оценить.
- Вы должны расклассифицировать активы вашего проекта по соответствующим папкам. Каждой папке соответствует один и только один тип актива. Например: "Сценарии/", "Сцены/", "Спрайты/", "Префабы/", "Звуки/", "Модели/", ...
- Убедитесь, что каждый день тщательно тестируете представленные прототипы. Они очень помогут вам в понимании предмета, а также в том, что от вас требуется.
- Использование Unity Asset Store запрещено. Вам рекомендуется использовать ежедневно предоставляемые ресурсы (при необходимости) или искать дополнительные в Интернете, если они вам не нравятся, за исключением сценариев, очевидно, потому что вы должны создавать все, что вы отправляете (за исключением сценариев, представленных персоналом). Asset Store запрещен, потому что все, что вы будете делать, досадно там в той или иной форме.

Однако использование стандартных активов Unity разрешено и рекомендовано для некоторых упражнений.

- Начиная с d03 для экспертной оценки вам потребуется создавать игры для их тестирования. Корректор должен будет обратиться к игре, поэтому вы всегда должны пушить проекты/исходники. Ваш проект всегда должен быть правильно настроен для сборки. Никакие изменения в последнюю минуту недопустимы.
- Предупреждение: Вы не будете исправлены программой, за исключением случаев, оговоренных в теме. Это подразумевает определенную степень свободы в том, как вы можете выполнять упражнения. Однако помните инструкции к каждому упражнению не ленитесь, вы упустите много очень интересных.
- Наличие дополнительных или бесполезных файлов в вашем репозитории не является проблемой. Вы можете разделить свой код на разные файлы вместо одного, за исключением случаев, когда в заголовке упражнения указан список файлов для отправки. Один файл должен определять одно и только одно поведение, поэтому нет пространства имен. Эти инструкции не относятся к подпапке «project/Temp/», которой не разрешено существовать в ваших репозиториях.
- Внимательно прочитайте все, прежде чем начать, действуйте, делайте это.
- Этот документ потенциально может измениться за 4 часа до отправки.
- Даже если тема упражнения короткая, лучше потратить немного времени на ее изучение. Понять, что требуется, чтобы сделать то, что лучше всего.
- Иногда вас попросят уделить особое внимание художественной стороне вашего проекта. В этом случае это будет явно упомянуто в теме. Не стесняйтесь попробовать множество разных вещей, чтобы получить представление о возможностях, предлагаемых Unity.
- Клянусь Одном, клянусь Торм! Используйте свой мозг !!!


Глава III

Конкретные инструкции дня

- Использование API Physics Unity категорически запрещено.
- Сегодня, поскольку каждое упражнение совершенно отличается от других, вам потребуются специальные каталоги для сдачи «ex0x».

Глава IV

Упражнение 00: Симулятор воздушного шара 2015

| | |
|--|---------------|
|  | Упражнение 00 |
| Упражнение 00: Симулятор воздушного шара 2015 | |
| Каталог с дачи: ex00/ Файлы для | |
| с дачи: файл сценария ex00, Balloon.cs, ресурсы, необходимые для упражнения . | |
| Разрешенные функции: Debug.Log, Mathf.RoundToInt, Input.GetKeyDown, GameObject.Destroy | |
| Примечания : н/д | |
| | |

Несмотря на то, что название предполагает это, давайте будем точными. Вам нужно будет создать сцену с воздушным шаром. Должна быть возможность надуть воздушный шар с помощью клавиши пробела. Если воздушный шар слишком большой, он взорвется, а если он недостаточно надут, игра проиграет. Воздушный шар должен заметно расти в зависимости от уровня воздуха. Если клавиша пробела больше не нажата, воздушный шар должен сдуваться.

Вы можете запустить демонстрацию прикреплённую на странице проекта, чтобы получить представление.


Вы также должны учитывать дыхание. Чем быстрее мы будем надуть, тем больше вы будете задыхаться. Достигнув определённого уровня, вы больше не сможете надуть воздушный шар, и вам придётся ждать, пока ваше дыхание восстановится.

Когда игра заканчивается, вы должны отобразить время, округлённое до единиц, в консоли следующим образом:

Срок службы воздушного шара: 120 с

Глава V

Упражнение 01: Событие Quick Time

| | |
|---|---------------|
|  | Упражнение 01 |
| Упражнение 01: Событие Quick Time | |
| Каталог с дачи: ex01/ Файлы | |
| для с дачи: файл с сцены ex01, CubeSpawner.cs, Cube.cs, необходимые ресурсы для упражнения | |
| Разрешенные функции: Debug.Log, Random.Range, GameObject.Instantiate, GameObject.Destroy, Input.GetKeyDown, Transform.Translate | |
| Примечания : н/д | |
| | |

Вероятно, вы когда-то играли в Guitar Hero или в любую другую игру, в которой вам нужно было нажать клавиши в нужное время. Для новичков это то, что называется QTE или Quick Time Event.

Вы можете запустить демонстрацию, прикрепленную на странице проекта, чтобы получить представление. Вы должны использовать A, S и D, чтобы играть в игру.

Цель этого упражнения, как вы, вероятно, поняли, состоит в том, чтобы создать QTE. Поместите полосу внизу экрана, а также одну вверх и 3 вертикальные линии от верхней до нижней. Убедитесь, что квадраты случайным образом генерируются на этих трех вертикальных линиях каждые X секунд. Эти квадраты должны соответствовать клавише (примеры вы найдете в zip-файле, прикрепленном на странице проекта). Каждый квадрат должен падать с определенной скоростью. Оказавшись на линии внизу, игрок должен нажать клавишу в нужный момент. У каждого квадрата должна быть своя линия.

Вы должны отображать точность игрока после каждого нажатия клавиши. Точность соответствует ответу расставленному между квадратом и линией, когда игрок нажимает клавишу. Он должен отображаться следующим образом:


Точность: 23,454



Квадраты должны иметь время жизни и не могут оставаться созданными после того, как покинут игровое поле.

Глава VI

Упражнение 02: Мини-г ольф

| | |
|--|---------------|
|  | Упражнение 02 |
| Упражнение 02: Мини-г ольф | |
| Каталог с дачи: ex02/ Файлы | |
| для с дачи: файл с ц ены ex02, Ball.cs, Club.cs, рес урс ы, необх одимые для упражнения . | |
| Разрешенные функц ии: Debug.Log, Mathf.Clamp, Transform.Translate, Input.GetKey Примечания : н/д | |
| | |

Нет ничего лучше с порта. Поэтому вы создадите мини-г ольф. Вы должны будете убедиться , что иг рок может брос ить мя ч с силой, определя емой клавишей пробела. Чем дольше нажатие, тем дальше полетит мя ч. Мя ч должен иметь инерц июи замедля тьс я с о временем. Ес ли мя ч кас ается ст ены, он отс какивает в противоположном направлении. Ес ли мя ч летит с лишком быст ро над лункой, он не падает.

С амое прос тое э то проверить! Запус тите де мо и немног о нас ладите с ь.


Для подс чета очков мы будем использовать упрощенную версию или правила мини-г ольфа. Иг рок начинает с -15 очков, за каждую неудачную попытку он получает 5 очков. Ес ли с чет больше 0, э то проиг рыш, но иг ра не ос танавливаетс я .

При каждом выстреле с чет должен отображаться в консоли с ледующим образом:

Оценка -5

Глава VII

Упражнение 03: Flappy Bird

| | |
|---|---------------|
|  | Упражнение 03 |
| Упражнение 03: Flappy Bird | |
| Каталог с дачи: ex03/ Файлы | |
| для с дачи: файл с ц ены ex03, Bird.cs, Pipe.cs, рес урс ы, необх одимые для упражнение | |
| Разрешенные функц ии: Debug.Log, Input.GetKeyDown, Transform.Rotate, Transform.Translate, Mathf.RoundToInt Примечания : н/д | |
| | |

Для э тог о упражнения давайте сделаем упрощенную версию Flappy Bird. Помес тите птиц у в ц ентр с ц ены. Он должен падать, пока иг рок не нажмет пробел, чтобы он с нова подня лс я . Запус тите демонс трац ию прикрепленную на страниц е проекта, чтобы получить предс тавление.


Давайте быст ро рас с мотрим некоторые тех ничес кие с оображения . Птиц а не может двиг атьс я по ос и х. Трубы должны быть на одной выс оте, потому что мы еще не видели, как э то сделать правильно. Не может быть более 2 э кземпля ров каналов, поэ тому вы должны ис пользовать их повто рно. С корос ть должна увеличива тьс я по мере тог о, как иг рок прох одит вс е больше и больше труб.

Вы также должны внедрить с истему подс чета очков и времени. Каждая труба прошла наг рады 5 баллов. В конц е каждой иг ры вы должны отобразить с чет в конс оли с ледующим образом:

```
Оценка: 15
Время : 25 с
```

Глава VIII

Упражнение 04: Понг !

| | |
|--|---------------|
|  | Упражнение 04 |
| Упражнение 04: Понг ! | |
| Каталог с дачи: ex04/ Файлы | |
| для с дачи: файл с ц ены ex04, PongBall.cs, Player.cs, необх одимые рес урс ы для упражнения | |
| Разрешенные функц ии: Debug.Log, Random.Range, Input.GetKey, Transform.Translate | |
| Примечания : нет данных | |

З авершить конц епц иякоторая доказала с воюц еннос ть и я вля ется относ ительно универс альной. Вам нужно будет с оздать Pong. Ес ли вы не знаете, чтоэ тотакое, запус тите демо. Иг рок с лева перемещет с вою ракетку, ис пользуя W и S, а иг рок с права, ис пользуя стрелки вверх и стрелки вниз.

Убедитес ь, что 2 пря моуг ольника нах одя тся на каждой стороне иг ры. Квадрат должен отс кочить между 2, ег онаправление меня ется на противоположное пос ле каждог оотс кока, в том чис ле по вертикали.

Оба пря моуг ольника должны управля тся отдельно от друг ог о по вертикали с помощью клавиш клавиатуры, имея в качестве ог раничения стенки с верх у и с низу. Ес ли квадрат прох одит через ракетку, он возвращетс я в ц ентр, и иг рок с оперника получает очко.

Оц енка должна отображатьс я пос ле каждой точки в конс оли с леду ющим образом:

Иг рок 1:0 | Иг рок 2: 0