

**ТЕСТОВОЕ ЗАДАНИЕ [Unity3D Developer]**

**Сроки:** для выполнения тестового задания мы предлагаем **две календарные недели**. Если по каким-либо причинам вам необходимо немного больше времени, заранее сообщите об этом.

Выполненное задание вышлите менеджеру по персоналу, с которым вы общались.

Команда Plarium ценит ваши время и усилия. После получения тестового задания мы детально изучим его и вышлем ответ в течение пяти рабочих дней.

**Успехов** **вам!**

**1. Основные положения**

Необходимо реализовать мини-игру согласно требованиям, представленным в данном тестовом задании, используя предоставленные графические и звуковые ассеты. Предоставляется также план разработки, которого рекомендуется придерживаться. Кроме того, в тестовом задании есть пункт „Дополнительные бонусы“. Выполнение условий этого пункта не влияет на оценку тестового задания, если базовая часть задания сделана неудачно, но учитывается во всех остальных случаях.

Тестовое задание считается готовым, если соблюдены следующие условия:

1. Есть готовый билд для PC, ОС Windows (файл \*.exe в финальной папке проекта).
2. Есть возможность собрать данный билд, используя ресурсы, представленные в финальной папке проекта.
3. Игра открывается в редакторе Unity и запускается без ошибок.
4. Соблюдены все требования.
5. Исходники лежат в открытом виде в проекте.

## **2. Требования**

1. Нельзя использовать сторонние наборы ассетов (например из Asset Store), фреймворки, библиотеки и т. д. Только стандартные средства Unity и собственноручно написанные инструменты.
2. Версия Unity — Unity 2018.4.24f1.
3. Реализация UI на нативном Unity UI.
4. Язык — C#. Разрешается использование любой стандартной версии языка. Использование сторонних библиотек не разрешается, только стандартные средства языка.
5. Необходимо отделить ООП-модель пасеки (структуры данных) от скриптов, связанных с графическим представлением Unity (MonoBehaviour).
6. В игре должны присутствовать игровые состояния, реализованные через FSM. Примеры состояний: загрузка ресурсов, главное меню, экран игры.
7. Разрешение игры — 1024х768, для всех моделей необходимо создать материалы.
8. Настройки камеры в базовом варианте — произвольные. Допускается статичная позиция камеры, но она должна охватывать текущее активное игровое пространство и позволять рассмотреть отдельных пчел достаточно подробно.
9. Должен присутствовать “мануал”, в котором кратко описаны все реализованные механизмы, а также есть информация об управлении и интерфейсах игры (как собирать/продавать мед, как смотреть статистику и т. д.).
10. Должна быть минимальная настройка игрового баланса и управления, позволяющая за адекватное количество времени посмотреть все реализованные фичи и проследить за меняющимся жизненным циклом пасеки.

## **3. Правила игры и план разработки**

Необходимо реализовать игру-песочницу „Пасека“, включающую в себя ООП-архитектуру пасеки, написанную на С#, и симулятор-визуализатор, реализованный средствами Unity.

Для общего упрощения процесса будем считать, что пчелы собирают готовый мед прямо с цветов (игнорируем механизм сборки пыльцы и вырабатывания меда пчелой).

Игра не имеет фиксированной цели либо условий победы/поражения и представляет из себя песочницу-симулятор.

**Для реализации песочницы должны быть созданы следующие объекты:**

1. Пасека — игровая зона с минимальной графикой. Она должна иметь границы, за которые нельзя выходить и на которых не могут размещаться объекты игры.
2. Ульи — объекты, генерирующие пчел и имеющие емкости для накопления меда. Выступают в роли управляющего механизма для пчел, контролируют все их действия (каждый улей контролирует своих пчел).

Особенности ульев:

* + Имеют настраиваемый лимит пчел (ограничение на количество пчел, которые находятся одновременно под управлением улья).
  + Имеют настраиваемый лимит меда (сколько максимально меда помещается в улей).
  + Управляют поведением своих пчел. Полностью контролируют численность пчел в улье, их вылет за медом, возвращение в улей. Если лимит меда не достигнут, должен посылать пчел за медом (временные и численные параметры — произвольные). В произвольные промежутки времени происходят события, инициирующие появления новых пчел и их смерть (вместе или раздельно — условия рождения и смерти произвольные на усмотрение исполнителя).
  + Имеет модуль real-time UI-отображения текущих параметров.
  + Количество ульев и их расположение в базовом варианте фиксировано и не изменяется, единственное условие — количество ульев не должно быть слишком маленьким, предлагается делать минимум 10 ульев.

1. Пчелы — агенты, управляемые AI, собирающие мед и доставляющие его в улей.

Особенности пчел:

* + Рождаются и первоначально функционируют в ульях. Все пчелы, которые находятся вне улья, обязательно имеют свое собственное графическое отображение. Должна поддерживаться минимум одна механика смерти пчел по любому простому условию на усмотрение исполнителя.
  + При получении команды от улья на сборку меда отправляются к объектам типа „цветок“, игнорируя все препятствия на пути (перемещаются по воздуху). Цветок находят самостоятельно. При прибытии к цветку начинают продолжающийся во времени процесс сборки меда; после того как личная емкость пчелы заканчивается, переносят мед в свой улей. Если кончилась емкость цветка, ищут следующий цветок.
  + С определенным шансом вместо обычной пчелы в улье может родиться трутень. Трутень не собирает мед, он бесполезен. Если срабатывает событие смерти пчел, сначала умирают трутни, а затем, когда трутней уже не осталось, обычные пчелы. Если лимит пчел в улье достиг максимума, включается дополнительное условие для смерти трутней.

1. Цветы — источники меда. Для упрощения игнорируем сборку пыльцы.

Особенности цветов:

* + Имеют единый менеджер, управляющий появлением и исчезновением цветов на пасеке.
  + Каждый цветок имеет ограниченное количество меда. Запасы меда должны пополняться со временем. Если мед в цветке заканчивается, цветок исчезает.
  + Существует лимит на количество одновременно обслуживаемых отдельным цветком пчел.
  + Менеджер цветов обязательно должен иметь механизм задания начальных параметров количества цветов, скорости их появления, количества меда в цветах, механизм исчезновения цветов и т. д. для возможности увидеть функционирование пасеки в динамике в зависимости от заданных начальных условий. В идеале система должна поддерживать равновесие и не впадать в одну из двух крайностей: слишком много/мало пчел.
  + Все параметры должны быть настраиваемы не из кода.
  + Для наглядной оценки качества выполнения задания и работы со структурами данных должна присутствовать реализация возможности увидеть большое количество пчел на экране (несколько тысяч будет достаточно, при этом нежелательно, чтобы всегда по умолчанию в симуляции летало несколько тысяч пчел, но обязательно должна быть возможность посмотреть на такую конфигурацию) и огромное количество пчел внутри ульев (сотни тысяч и больше).
  + Игра должна иметь следующие основные окна, переключение между которыми должно быть реализовано с помощью FSM:
    1. главное меню;
    2. окно игры с HUD, в который выведена основная статистика (количество ульев, пчел, денег) и элементы управления;
    3. окно статистики с результатами общей сессии пасеки.

## **4. Дополнительные бонусы**

Данное тестовое задание подразумевает выполнение некоторого количества дополнительных заданий, что будет свидетельствовать о тяготении к определенной области в разработке игр и о более высоком уровне профессиональной подготовки в сравнении с базовой версией тестового задания. Выполнять все пункты или все группы пунктов необязательно, выполнение каждого пункта оценивается отдельно.

**Дополнительные бонусы по графике:**

* + Визуальные маркеры происходящих игровых процессов (индикаторы выполнения), привязанные к объектам в мировом пространстве с фиксированным размером в экранном пространстве (например радиальный progress bar над пчелой при сборе меда пасечником). Отрисовка этих маркеров должна происходить за минимальное число вызовов отрисовки.
  + Отрисовка пчел таким образом, чтобы минимизировать число вызовов отрисовки. В идеале все пчелы должны отрисовываться за один вызов. Допускаются любые изменения изначальных ассетов пчелы (изменение или отказ от анимации, замена оригинальной модели) в рамках приемлемого визуального качества (это все еще должно быть похоже на пчелу, и в идеале — на живую пчелу).
  + Эффект outline для выбранного улья.
  + Mini-preview с 3D-ульем в окне статистики улья.
  + Эффект покачивания на ветру для цветов.
  + Отрисовка цветов за один draw call.
  + Исчезновение цветов через эффект растворения (dissolve).
  + Планарные тени от пасечника, ульев и домика пасечника.
  + Эффект смены дня и ночи — движение теней в соответствии с направлением источника освещения, видимое на объектах изменение освещения. Возможно появление самосветящихся объектов в ночное время.
  + Расширенный функционал главной камеры, смены фокусов, плавные перелеты камеры (например zoom in/out при выборе того или иного улья).

**Дополнительные бонусы по механике и логике игры:**

* Добавление в игру пасечника — агента, управляемого игроком. Теперь на все происходящее в песочнице можно влиять в рантайме с помощью этого героя. Предлагаемое к реализации управление — контекстное, передвижение и взаимодействие с объектами осуществляется с помощью мыши. Должен присутствовать механизм выбора пасечника как активного объекта и механизм взаимодействия с ульями в этом режиме. Выделение улья или домика с неактивным пасечником показывает для них информацию (текущие параметры). Использование любой другой модели управления не возбраняется. Реализованная модель управления (инпуты и само управление внутри игры) должна быть подробно описана в разделе „мануал“. Пасечник должен выполнять следующие базовые действия: сборку меда (подходит к улью, в котором есть в наличии мед, производит непрерывную, продолжающуюся во времени операцию сборки меда, с воспроизведением соответствующей анимации) и доставку меда (относит мед в специальный объект „Домик“). Собирать мед при превышении личного лимита либо при отсутствии меда в улье пасечник не может. Также он обязательно должен свободно передвигаться по пасеке и корректно обходить препятствия (ульи, домик, грузовик). Цветы обходить необязательно. Кроме того, пасечник может зайти в домик и инициировать механизм продажи меда. Домик — жилье и офис пасечника — обладает большой (но конечной и соизмеримой с емкостями ульев) вместимостью меда. Когда пасечник заходит в домик, он может продавать мед с помощью специального механизма, который должен включать в себя прибытие грузовичка для перевозки меда к домику пасечника, взаимодействие с пасечником, забор меда, начисление средств (последние две операции рекомендуется сделать атомарными, непрерывными, не продолжающимися во времени) и отбытие грузовичка с медом. Помимо этого, домик должен выполнять функцию модуля общей статистики с доступом в окошки конкретных ульев.
  + Добавление в улей параметра, влияющего на эффективность и на состояние пчел и улья (по прошествии произвольного периода времени параметр ухудшает скорость передвижения, скорость сборки меда и другие опциональные параметры у своих пчел). Является коэффициентом, принимает значение от 0 (минимум эффективности) до 1 (максимум эффективности). Чтобы поднять этот параметр, игроку с помощью агента „Пасечник“ нужно производить систематический уход за ульем. Коэффициент негативно влияет на все процессы, происходящие внутри улья (например на скорость генерации новых пчел), включая поведение пчел. Параметр также влияет на скорость взаимодействия пасечника с самим ульем: улей с низким значением данного механизма будет чиститься гораздо дольше, однако нужно предусмотреть механизм, благодаря которому операции с ульем не будут занимать критическое время (бесконечное или очень долгое). Делать это путем установки нижней границы коэффициента, например 0,5, — нельзя. Также добавление данного параметра требует добавления еще двух механизмов (для них необходимо будет самостоятельно придумать реализацию). Первый механизм предотвращает функционирование пчелы с нулевой или драматически низкой эффективностью (поскольку такая пчела будет двигаться или собирать мед с нулевой или неприемлемо малой скоростью), второй — максимально не допускает возвращение пчел в улей „пустыми“. Ситуация, когда пчела может вернуться в улей пустой, допустима (например выработан мед со всех цветков), но частота их должна быть сведена к минимуму.
  + Реализация гибкого лимита пчел. Теперь улей может превышать количество пчел, однако превышение лимита будет вести к дополнительному негативному влиянию на коэффициент эффективности улья (скорости работы пчел, скорости производства пчел, скорости взаимодействия с самим ульем).
  + Реализация классового многообразия пчел. Теперь все пчелы — это наследники одного базового класса, и каждая пчела несет особую функцию. Появляются рабочие пчелы, собирающие мед, трутни, не делающие ничего, в каждом улье появляется матка в единственном экземпляре. Выделение и создание функций базового класса пчелы — задача претендента, но в наследниках должна остаться только уникальная логика. Появляются осы — особые насекомые, не привязанные к конкретному улью и контролирующие себя сами. Осы свободно летают по пасеке, могут садиться на цветы и кусать пасечника, замедляя его текущие действия. Осы с определенным шансом продолжительно во времени убивают пчел, находящихся с ними на одном цветке. Осы могут с определенным шансом убивать пчел, если пролетают с ними на достаточно близком расстоянии. Осы могут атаковать ульи, воруя мед и убивая пчел. Теперь с определенным шансом вместо обычных пчел рождаются шершни — пчелы, охраняющие улей от ос. Они не могут собирать мед и кусают пасечника, но без них улей становится крайне уязвим для атак ос. При большом количестве ос шершни вылетают из улья, преследуя и уничтожая ос.
  + Реализация управления пчелами через особый тип пчелы — пчелу-матку. Требует выполнения предыдущего пункта. В базовой реализации классом, внутри которого происходит управлением пчелами, является улей. Теперь же пчела-матка получит функции управления. Она будет каким-либо образом получать от улья информацию о его состоянии и самостоятельно назначать другим пчелам задачи, создавать новых пчел тоже теперь будет матка, всю логику такого поведения она теперь берет на себя. Коэффициент эффективности улья учитывается точно так же.
  + Реализация механизма генерации нового улья путем разделения роя на две части. Теперь в улье, в котором количество пчел приближается к пределу, появляется маленький шанс генерации второй матки вместо обычной пчелы. Вторая матка немедленно вылетает из улья и начинает процесс поиска свободного места под новый улей, после того как поиск завершен, вторая матка строит улей и меняет принадлежность половины всех пчел со старого улья на новый. При этом те пчелы из новой половины, которые на этот момент были в старом улье, должны вылететь из старого улья и направиться в новый, а те пчелы, которые были вне улья (на сборке меда), по завершению своих задач направляются в новый улей вместо старого, включая тех, которые уже возвращались в старый улей.
  + Расширенная реализация трутней (бесполезных пчел). Отключаются старые механизмы приоритетной смерти трутней и смерти трутней при достижении лимита пчел в улье. Теперь должен быть предусмотрен отдельный продолжающийся во времени механизм изгнания, который позволяет избавлять улей от трутней, требующий фиксированного количества пчел внутри улья. Процесс может прерваться, если вдруг пчелы умрут и их количество станет недостаточным для поддержания процесса. Изгнанный трутень вылезает из улья и имеет шанс переродиться в другую пчелу одного из трех видов (обычную, шершня или осу). При успешном перерождении возвращается в свой улей в новом качестве, при неудачном — летит искать новый улей, чтобы стать трутнем там.