**Гурина Татьяна**

**Курс "Разработчик игр на Unity"**

Игра "Питомцы БабыЙоги".

Описание и взаимодействие классов в игре.

**Описание класса:**

Название класса: **PointsMoveAnim.**

Описание: Отвечает за анимацию зверей.

**Описание методов класса:**

1. **Update():** т.к. перемещение объектов в игре реализовано методом MoveTowards, а не AddForce, поэтому velocity = 0. Отсюда принято решение брать булеву переменную isRelaxing из PointsMoving, которая и задает переключение между Idle и Going вместо скорости.

2. **AnimationGoing():** переключает анимации (Idle, Going), зависящие от скорости объекта.

3. **OnTriggerEnter(Collider2D coll):** при коллизии с предметом окружения одна из анимаций (Sleeping, Eating, Drinking, Pee) включается по срабатыванию своего триггера.

Возможно переключение из анимаций (Idle, Going) в одну из четырех (Sleeping, Eating, Drinking, Pee) по триггеру.

**Описание класса:**

Название класса: **PointsMoving.**

Описание: задает некое подобие хаотичного движения объекта по сцене.

Точки, между которыми движется объект, заданы массивом из Tranform-ов. (Передвижение задает MoveTowards, точки маршрута переключаются одна за другой по их достижению).

Рандомно возникает стояние объекта на месте (частота и продолжительность задаются в Инспекторе).

**Описание методов класса:**

1. **Update():** задает движение объекта, если тот !isRelaxing.

По достижению точки цели пути запускает метод, меняющий цель пути.

2. MoveToTarget: задает движение к цели.

3. **StandsStill():** задает остановку на месте.

4. **FindIndex():** находит индекс в массиве points[] для последующей смены цели.

5. **ChangeTarget():** меняет точку маршрута.

6. **IfRelax():** случайным образом определяет отдых со стоянием на месте, частота задается в Инспекторе.

7. **IEnumerator TimeToRelax():** задает поведение отдыха на месте с WaitForSeconds, продолжительность которого задается в Инспекторе.

8. **IEnumerator():** зацикливает корутину TimeToRelax в вечный ежесекундный цикл. Запуск из Start().

**Описание класса:**

Название класса: **BeastInterface.**

Описание: Отвечает за индивидуальный Canvas каждого зверя в игре.

**Описание методов класса:**

1. **TurnOnEatPanel():** включает значок панели "Хочу есть".

2. **TurnOnDrinkPanel():** включает значок панели "Хочу пить".

3. **TurnOnSleepPanel():** включает значок панели "Хочу спать".

4. **TurnOnWCPanel():** включает значок панели "Хочу в туалет".

5. **WhatBeastsWants():** рандомно включает один из значков "Хочу...".

6. **IEnumerator SometimesWish():** рандомно запускает желания зверей, от 8 (чтоб не слишком часто) до задаваемого в Инспекторе wishPeriodicity.

7. **OnTriggerEnter():** при попадании в триггер значок "Хочу..." выключается, добавляется монетка за выполнения желания.

8. **MinusStar():** удаляютя звезды после 5сек ожидания. На Canvas есть 3 звезды, каждая из которых постепенно исчезает (от правой к средней и к левой), если желание зверя не удовлетворено.

9. **IEnumerator BeastGoesAway():** по исчезновению всех 3х звезд появляется попап, предупреждающий о дальнейшем ичезновении зверя, затем зверь исчезает.

**Описание класса:**

Название класса: **MenuManager**.

Описание: Отвечает за поведение панелей меню, кнопок, изображений.

**Описание методов класса:**

1. **RestartLevel():** задает поведение кнопки "Заново".

2. **Pause():** задает поведение кнопки "Пауза".

3. **YouWin():** выигрыш, если заработали 30 монет (а не в наличии, поэтому goldSum, а не goldFinal).

4. **GoldCollect():** считает количесство заработанного золота.

5. **ShowGold():** показывает, сколько золота в наличии, (а подсчет доход минус расход в Update()).

6. **BuyNewBeast():** если золота достаточно для покупки, то при нажатии кнопки "Купить нового зверя" вычитает нужную сумму, появляется новый зверь.

7. **WhoIsNewBeast():** назначает следующего зверя для покупки.

8. **GameOver():** конец игры при отсутствии монет на покупку зверя и зверей на сцене.

9. **StopTimeInLearning():** если обучающий интерфейс удалился после завершения обучения, то время запускается.

**Описание класса:**

Название класса: **TeachingUI.**

Описание: Знакомит игрока с игрой, объектами и механиками.

Последовательно возникающие друг за другом попапы, организованные в массив, объясняют мир, предметы, а затем возможные дейтвия. Языком, характерным именно для мира игры.

На каждом попапе есть кнопки Продолжить и Пропустить.

**Описание методов класса:**

1. **Next():** при нажатии кнопки "Далее" текущий попап становится неактивен, следующий активизируется. По окончанию массива попапов весь обучающий канвас удаляется.

2. **Skip():** при нажатии кнопки "Пропустить" обучающий канвас удаляется.

**Описание класса:**

Название класса: **DragAndDrop.**

Описание: Дает возможноссть переносить указателем мыши зверей по экрану (до кормушки, питья, коврика, туалета).

Установлен лишь на переносимые объекты (зверей).