Список методов, которые требуются от Arcomage.Core в Unity3d

1. Добавить игрока public void AddPlayer(TypePlayer tp, string name)
2. Начать новую игру. public void StartGame()
3. Статистика игрока
4. Статистика противника
5. Получить карты игрока (противника) для отображения. public List<Card> GetCard()
6. Имя победителя
7. Использовать карту

Информация о том, какое действие необходимо выполнять сейчас:

public enum CurrentAction

{

None, // Произошел первый запуск

StartGame, //Начало игры

GetPlayerCard, //Получить карты для игрока

WaitHumanMove, //Ожидание хода игрока

HumanUseCard, //Игрок использовал карту

HumanCanPlayAgain, //Игрок может сходить еще раз

AnimateHumanMove, //Анимация использования карты игрока

UpdateStat, //Обновление статистики игроков

EndHumanMove, //Завершение хода игрока

AIMoveAnimation, //Анимация стола противника

AIUseCardAnimation, //Анимация использование хода противника

EndAIMove, //Завершение хода противника

EndGame //Завершение игры

}

Список методов, которые требуются в Unity3dдля обработки

* + 1. Прорисовка статистики игроков
    2. Прорисовка карт игрока
    3. Прорисовка рубашек карт противника
    4. Прорисовка использования карт игроком противника
    5. Запуск /Перезапуск игры
    6. Использование карты игроком
    7. Отображать кто выиграл и кнопки управления
    8. Возвращение неиспользованной карты игрока на место
    9. Обновление возможности использования карт

Какие события в Unity3d должны обрабатываться

1. Наведение на карту
2. Клик левой кнопкой мышки
3. Клик правой кнопкой мышки
4. Нажатие на Replay
5. Нажатие на Exit

Теперь рассмотрим более подробнее действия:

* + Получить карты для игрока

1. Обратиться к ядру и получить ровно столько карт, сколько необходимо для игрока
2. Отобразить их на поле игрока
3. Затемнить те карты, которые невозможно использовать, и они становится неактивны для выбора
4. Unityне должен знать, сколько ему необходимо отобразить карт. Это обязанность ядра
   * Ожидание хода игрока
5. Никаких изменений не должно происходить на столе. Пока не произойдет событие использование карты игрока
   * Анимация использования карты игрока
     1. Карта ожидает событий. Наведение мыши на карту – отображение увеличенной копии карты. Одиночное нажатие – карта используется.
     2. Нажатие правой клавишей – карта сбрасывается. Вызывается завершение хода игрока.
     3. Если карту невозможно использовать, прерывания события.
     4. Если карту возможно использовать, должно выполниться событие – обновление статистики игроков
     5. Потом обновление возможности использование карт
     6. Далее выполняется событие – Получить карты для игрока, но уже в данном случае новой карты
     7. Добавляется событие – Завершение хода игрока
   * Обновление статистики игроков
     1. Происходит повторный запрос, и перерисовка параметров игроков
   * Завершение хода игрока
     1. Проверяется может ли игрок сходить еще раз, если да, тогда вызывается событие ожидание хода игрока и текущее событие прерывается
     2. Карты становятся недоступными для выбора
     3. Выполняется Анимация стола противника
   * Анимация стола противника
     1. Карты переворачиваются рубашкой вверх
     2. Отображается индикатор размышлений противника
     3. Ожидание события - Анимация использования хода противника
   * Анимация использование хода противника
     1. На стол вбрасывается та карту, которую противник использовал или сбросил
     2. Зависает на некоторое время, чтобы игрок успел ее разглядеть
     3. Разрушается
     4. Запускается событие – обновление статистики игроков
     5. Запускается событие – завершение хода противника
   * Завершение хода противника
     1. Карты становится доступными
     2. Выполняется – ожидание хода игрока
   * Завершение игры
     1. Все элементы на экране становятся неактивными
     2. Получаем с сервера имя победителя
     3. Отображаем его и 2 кнопки – Выход и Заново

Наглядная таблица соответствия события и методов

|  |  |
| --- | --- |
| **Событие** | **Список** **методов** |
| Запуск игры | Unity.SceneScript.Start |
| Получить карты для игрока | Core.GetCard() return List<Card>()  Unity. |