

Этот код реализует простую игру в блэкджек на Python с использованием библиотеки `pygame`. Давайте разберем его по частям:

1. Инициализация и настройка:

- Импортируются модули `copy`, `random` и `pygame`.
- Инициализируется библиотека `pygame`.
- Определяются карты, колода и количество колод, размер окна игры, шрифты, счетчики и начальные значения игровых переменных.

2. Основные переменные:

- `cards` - список значений карт.
- `one_deck` - базовая колода, состоящая из четырех наборов значений карт.
- `decks` - количество колод (4).
- `WIDTH`, `HEIGHT` - ширина и высота окна.
- `screen` - объект экрана для отображения игры.
- `timer`, `font`, `smaller_font` - установка таймера кадров и шрифтов для текста.
- `records` - список для записи результатов: [победы, поражения, ничьи].
- `player_score`, `dealer_score` - счет игроков.
- `initial_deal`, `my_hand`, `dealer_hand`, `outcome`, `reveal_dealer`, `hand_active`, `add_score` - вспомогательные переменные для отслеживания состояния игры.

3. Функции:

- `deal_cards`: Случайным образом выбирает карту из `current_deck`, добавляет ее в `current_hand` и удаляет из колоды. Возвращает обновленную руку и колоду.
- `draw_scores`: Отображает на экране счет игрока и дилера (если карты дилера открыты).
- `draw_cards`: Рисует карты на экране для игрока и дилера. Если карты дилера скрыты, отображается текст '???'.
- `calculate_score`: Вычисляет итоговый счет для заданной руки. Учитывает особенности тузов (A) — сначала считает туз как 11, но, если счет больше 21, туз учитывается как 1.
- `draw_game`: Отображает кнопки управления игрой на экране. Также показывает кнопки для начала новой игры, хода игрока (поднять карту) или останова (пас).
- `check_endgame`: Проверяет, закончилась ли игра, и определяет исход, учитывая результат (перебор, победа игрока, победа дилера, ничья). Также обновляет счетчики побед и поражений.

4. Главный игровой цикл:

- Цикл `while run` продолжает работу, пока `run = True`.
- Если `initial_deal = True`, каждому игроку раздается по две карты.
- Если игра активна (`active = True`), происходит обновление счета игрока, отображение карт и счета на экране. Дилер автоматически берет карту, если его счет меньше 17.
- Функция `draw_game` обновляет кнопки для управления игрой.
- **Обработка событий:** Игровые события обрабатываются циклом:
 - При закрытии окна `pygame.QUIT` игра завершается.
 - Если кнопка мыши отпущена, игра проверяет, нажата ли кнопка "Начать", "Подкинуть" (новая карта) или "Пас".
- Проверка конца игры через `check_endgame`.
- Обновление экрана с помощью `pygame.display.flip()`.

5. Завершение игры:

- Когда цикл `while` завершен, вызывается `pygame.quit()`, что закрывает окно и завершает программу.

Ключевые моменты:

- Использование функции `deal_cards` для раздачи карт и `calculate_score` для расчета очков.
- Состояние игры управляется переменными (`active`, `initial_deal`, `reveal_dealer`, `hand_active`), которые регулируют видимость кнопок, ход игрока и дилера.
- `draw_game` отвечает за интерфейс, обновляя кнопки в зависимости от текущего состояния.
- `check_endgame` анализирует результаты игры, определяя исход и обновляя счет.