

Esta es la explicación por parte del código, del juego “Adna”.

El código contiene una clase principal llamada Partida, de la cual se desprenden 4 subclases: Mazo, Pozo, Jugador y carta.

CLASE: Partida

Empezamos con partida: Partida contiene sus atributos específicos, las cuales son fundamentales para el control del flujo de la partida, en la misma encontramos variables clave como índice (IDX) y sentido (sentido de la ronda).

Después encontramos el primer método llamado Preparación(). Preparación lo que hace es sacar la carta que esté arriba del todo del mazo, verifica si esta es numérica o no, si es numérica la adjunta al mazo del pozo para iniciar el juego, caso contrario la almacena abajo del todo del mazo y saca otra.

Sigue el método turno, el cual controla el flujo de la ronda, primero calcula el jugador actual y después el jugador siguiente. El jugador actual depende del indice, el cual es calculado a lo largo de la ejecución de este método. El jugador siguiente es calculado a partir del índice, el cual se le suma 1 y es dividido por la cantidad de jugadores, el resto de esta división será el índice del jugador siguiente.

Después obtiene el resultado de la función tirar, la cual pertenece a la clase Jugador, y en base al resultado de esta hace una acción determinada. El método tirar siempre devuelve un STRING, el cual puede ser : Toma Cuatro, Toma Dos, Salta, Reversa, Siguiente, Agarrar y Tomó Acumuladas.

Si es ‘Toma Dos’ o ‘Toma Dos’ suma a ‘cartas_a_agarrar’ 2 o 4 según corresponda, calcula el indice siguiente obteniendo el jugador siguiente en base al sentido (+1 o -1) y actualiza la variable “jugador_actual” para que prosiga el flujo del juego al jugador al cual le toque después de él.

Si es Salta, el índice será calculado de forma distinta, salteando el índice siguiente (+2 en vez de +1) evitando el turno que el jugador al que le tocaría jugar. Si es “Reversa” cambia el sentido de la ronda cambiando el valor de sentido a “-1” y calculando el índice en base a este valor, dando vuelta el orden en la cual se juega la ronda.

Tanto “Siguiente”, “Agarrar” y “Tomo Acumuladas” son condiciones de apoyo las cuales ayudan al flujo del juego, “Siguiente” hace que pase la ronda, “Agarrar” y “Tomo Acumuladas” el jugador agarra una carta, o las que le corresponden, y pasa su turno.

El método “inicio”, simplemente es un texto el cual da una introducción al juego, y además da la posibilidad darle un a los jugadores humanos.

El método “Ronda” ejecuta todas las funciones necesarias (pertenecientes a otras clases) para la correcta ejecución del juego, así como también mantiene un bucle del juego hasta que haya un ganador.

CLASE: Carta

Considero necesario explicar primero esta clase, para que la clase Mazo se entienda de mejor manera.

La clase carta contiene los atributos de las eventuales cartas, conteniendo una lista de los números los cuales las cartas pueden tener, así como una lista de colores y una lista de tipos.

CLASE: Mazo

Dentro de la clase “Mazo” encontramos 2 atributos fundamentales, las cartas generadas, las cuales son las 100 las cuales son parte del juego, así como también el mazo en si.

El método “Generacion_Mazo” recorre cada color de colores, cada número de la lista números y crea 2 copias de cada una, guardandolas en la lista de cartas generadas en este orden: índice, color, número. El bucle de abajo hace lo mismo solo que genera 2 copias cada tipo de carta, excepto las del índice 3 qué referencias las “Toma Cuatro” haciendo solo 1 copia por color y las deposita en “cartas generadas”. Todas estas cartas generadas son almacenadas en la lista Mazo.

Después encontramos los métodos que interactúan con el mazo, el primer método las mezcla usando un comando (random.shuffle), el segundo método recorre cada jugador en jugadores creando un apartado llamado “MazoJugador” en cada Jugador, almacenando 5 cartas para cada uno. Los últimos 2 métodos lo que hacen es volver la última carta carta del mazo y también la cantidad de cartas cartas que hay en el mazo.

CLASE: Pozo

La clase pozo tiene un solo atributo, el cual es “mazo_pozo” el cual es una lista que almacena todas las cartas jugadas y que pueden ser efectivamente depositadas en ella.

Dentro de esta clase encontramos 2 métodos, el primero llamado “ultima_carta_pozo” el cual devuelve la última carta jugada en el pozo, este método es crucial para el funcionamiento del juego.

El segundo método es también otra parte crucial, el método llamado “comparación” como su nombre indica compara la última carta jugada en el pozo con la carta tirada con la carta jugada por el jugador. En esta misma se comprueba primero si la última carta del pozo y la del jugador coinciden en: 1) Color o 2) Tipo o Número. Si el índice 0 coincide, significa que es del mismo color, si el índice 1 coincide significa que 1) La carta es un “tipo” valida para ser jugado en relación a la última carta del pozo. 2) es del mismo número que la última carta del mazo. En caso de ser válida comparación retorna un STRING el cual significa el tipo de jugada a realizar por el método turno de la clase Partida.

CLASE: Jugador

Esta clase sirve para almacenar a los jugadores y además de manejar el flujo de estos. Encontramos un total de 5 atributos, 4 correspondientes a jugadores, dentro de estos diccionarios de nombre correspondiente a jugadores del A al D además de su “tipo”, siendo bots o humanos.

Dentro de la clase encontramos 3 métodos, el primero es el método “tirar”. En este método se realizan múltiples acciones, como verificar si el mazo está vacío para poder ser reciclado, aportar información útil para el jugador, darle la posibilidad de tirar una carta y además de verificar si el jugador “Adna” al tener una sola carta.

Para comenzar se le aporta información al jugador, véase, a quien le toca en este turno, las cartas restantes en el mazo y la última carta jugada en el pozo. Toda esta información y la posibilidad de tirar se la da al “jugador_actual” el cual es aquel correspondiente a la lista de jugadores.

Dentro de las posibilidades del jugador esta la de tomar una carta, de “adna” en caso de corresponder y la de tirar una carta. Entre otras cosas verifica si al agarrar una carta este mismo debe agarrar mas de una si en el turno pasado se puse una carta de +2 o 4 en pozo. También verifica si el jugador dijo “Adna” o no, así como si también es válido decirlo con la cantidad de cartas que tiene.

Por último están los jugadores controlados por la máquina, los bots. Cada uno de estos tiene una estrategia distinta, el primer bot apodado “El colorista” elige su color favorito en base al color de cartas que mas posea, he intenta mantener la ronda en ese

color. El segundo bot, apodado “El estratega”, juega defensivo o agresivo dependiendo de las circunstancias de la partida, si un jugador esta cerca de ganar este bot buscará tirar un +2 o +4 a ese jugador para que no pueda ganar. Por otro lado, si nadie está cerca de ganar, jugará cartas de salto y reversa para apresurar su siguiente turno.