Tic Tac Toe - FeatureIDE

Yara Diaz de Cerio Arzamendi

01/03/2022

Contenido

Inicio

Dado que el programa original era muy sencillo he tenido que añadir bastante código nuevo para que se adapte a todas las características que se pedían. Además de los cambios descritos a continuación he añadido un print a todas las constructoras para poder observar de forma sencilla que se está ejecutando en cada momento. El main se encuentra en DiazDeCerio/tictactoe/Menu.java.

Clases

Menu.java

Todo el código de las clases Menu.java es nuevo.

DiazDeCerio

Main

Inicializa el menú y cambia el icono de la aplicación.

Menu()

Ejecuta startMenu().

startMenu()

Crea el frame del menú inicial, añadiéndole solo el título “Tic Tac Toe”.

OponentComputer

startMenu()

“original()” para incluir el código de características anteriores.

Añade el botón “Play versus computer” junto con la imagen correspondiente. Al pulsarlo creará un nuevo TicTacToe con el parámetro true, este servirá para saber si el usuario ha seleccionado un modo u otro en caso de tener las dos características (VS computer, VS friend).

OponentFriend

startMenu()

“original()” para incluir el código de características anteriores.

Añade el botón “Play versus a friend” junto con la imagen correspondiente. Al pulsarlo creará un nuevo TicTacToe con el parámetro false, este servirá para saber si el usuario ha seleccionado un modo u otro en caso de tener las dos características (VS computer, VS friend).

About

startMenu()

“original()” para incluir el código de características anteriores.

Añade al menú el botón “About”, al clicarlo creará y mostrará el frame “About” (About.java), además de añadirle el icono de la aplicación.

About.java

About

About()
Constructora principal de About.java, ejecuta startAbout()

startAbout()
Crea el frame About. Añade y da formato a lo obligatorio de esta característica, es decir, el nombre de los autores del programa.

Url

startAbout()
"original()" para incluir el código de la característica anterior.

Añade al frame About la etiqueta con la URL del código fuente original.

TicTacToe.java

En esta clase gran parte del código es el original con algunas modificaciones. También se ha añadido nuevo.

DiazDeCerio

Variables nuevas/modificadas
Icono principal modificado.

Se han añadido variables para adaptarse a las imágenes de los iconos (X, O).

Nuevo panel para mostrar la etiqueta del resultado de la partida de manera más limpia, ya que al añadir la imagen del botón RESET esta queda poco visible.

TicTacToe(Boolean VSC)
Código de constructora original, se le ha añadido la asignación del booleano VSC, que indicará si el juego es contra el ordenador (VSC=true) o contra otro jugador (VCS=false).

actionPerformed(ActionEvent a)
Solo se ha mantenido el código que se ejecuta al pulsar el botón RESET (btn10). A este se le ha añadido código para eliminar los iconos (X, O).

initUI()
Se ha mantenido todo el código original y se han añadido algunos elementos como la asignación de la imagen del botón RESET.

addActionEvents()
Código original completo.

checkWinner()
Se mantiene todo el código y se le añade parte para modificar los iconos en caso de que uno de los jugadores haya ganado (la imagen se asigna en otro lugar).

Icons_BlueRed

initUI()

"original()" para incluir el código de la característica anterior.

Se asignan los iconos principales correspondientes a X y a O.

checkWinner()

Se asigna el icono del posible ganador a "imagenWinner", esta será la imagen que se modifica cuando un usuario hace tres en raya.

"original()" para incluir el código de la característica anterior.

Icons_hand

initUI()

"original()" para incluir el código de la característica anterior.

Se asignan los iconos principales correspondientes a X y a O.

checkWinner()

Se asigna el icono del posible ganador a "imagenWinner", esta será la imagen que se modifica cuando un usuario hace tres en raya.

"original()" para incluir el código de la característica anterior.

Icons_Purple

initUI()

"original()" para incluir el código de la característica anterior.

Se asignan los iconos principales correspondientes a X y a O.

checkWinner()

Se asigna el icono del posible ganador a "imagenWinner", esta será la imagen que se modifica cuando un usuario hace tres en raya.

"original()" para incluir el código de la característica anterior.

OponentComputer

actionPerformed(ActionEvent a)

"original(a)" para incluir el código de la característica anterior.

Se ha añadido una condición para saber que modo de juego se ha seleccionado (VS computer, VS friend). Primero se hace la jugada de X para la que se ha conservado el código original y se le ha añadido la parte en la que se los iconos en lugar del String "X".

Se comprueba si X ha ganado, en caso contrario se ha añadido código para realizar de manera automática la jugada de "O" y vuelve a comprobar si ha habido ganador.

La parte del RESET que es común se encuentra en otra característica.

OponentFriend

actionPerformed(ActionEvent a)

”original(a)” para incluir el código de la característica anterior.

Se le ha añadido una condición para que solo se ejecute este código habiendo seleccionado este modo de juego. Se conserva la mayor parte del código original y se añade parte para que cambie la imagen una vez seleccionada una casilla.

La parte del RESET que es común se encuentra en otra característica.

GamesCount

Variables nuevas

Se ha añadido una variable que corresponde al frame Ganador.

checkWinner()

”original()” para incluir el código de la característica anterior.

En caso de que alguno de los jugadores haya ganado se le llama a “createGanador()”

createGanador()

Crea y pone visible el nuevo frame.

TrackGames

Variables nuevas

Variables para guardar el número de partidas ganadas por X y por Y.

checkWinner()

”original()” para incluir el código de la característica anterior.

Cuando un jugador gana, modifica el número de partidas ganadas y actualiza este en el frame.

ConsecutiveGames

Variables nuevas

Nueva variable para guardar el último ganador y otra para guardar las partidas consecutivas ganadas por el mismo.

checkWinner()

Cuando un jugador gana se modifican las variables anteriores y actualiza el frame.

Ganador.java

GamesCount

Ganador(String letter)

Llama a countI pasandole como parámetro la letra del ganador.

countI(String letter)

Crea el frame del ganador y le añade la etiqueta que se mostrara en todos los casos: “Player – wins!”

TrackGames

Variables nuevas

Se han añadido nuevas variables que se utilizaran para cambiar el numero de partidas ganadas por cada jugador y sus etiquetas correspondientes.

setXWins(int x)

Asigna al frame el número de partidas ganadas por X.

setOWins(int o)

Asigna al frame el número de partidas ganadas por O.

countI(String letter)

”original()” para incluir el código de la característica anterior.

Añade al frame las etiquetas que muestran las partidas ganadas por cada jugador.

ConsecutiveGames

Variables nuevas

Se ha creado una nueva variable para guardar el numero de partidas consecutivas ganadas por el último jugador y otra para la etiqueta correspondiente.

setConsecutiveGames(int c)

Crea y asigna a la etiqueta del frame el mensaje “Player – has won – consecutive games”.

countI(String letter)

”original()” para incluir el código de la característica anterior.

Añade la etiqueta al frame.