



Ingeniería en computación
Universidad de La Serena
II Semestre, 2020
Versión número 1

Diseño

ALPA prototipo I

Miercoles 13 de octubre

Integrantes:

Yair Gallardo
Norton Irarrázabal
Sebastian Rojas

Docente

Guillermo Leyton.

Propósito

Este documento tiene como finalidad especificar el diseño sobre el cual se desarrollará el software Mappet, facilitando su construcción y siendo la base para su implementación.

Está dirigido a los desarrolladores del software y a las partes involucradas en la asignatura software II:

- SQA.
- Docente.

Descripción del problema

El uso de inteligencia artificial en el ámbito de los jugadores virtuales, es una **herramienta de uso investigativo y publicitario** que cuenta con larga data dentro de los últimos 30 años. En el área investigativa, se pretende encontrar el alcance de las capacidades de simulación racional con las que cuentan las máquinas; el logro de altos niveles por estos jugadores virtuales suele generar gran interés en el público.

Otra perspectiva a valorar es el amplio mercado de los videojuegos, donde la dificultad, la búsqueda de retos y experiencias satisfactorias suelen ser una de las áreas de interés para los jugadores más exigentes.

Intereses que pueden ser abordados por estos jugadores virtuales, siendo un reto y por otra una “ayuda” para jugadores menos experimentados que busquen encontrar o descubrir nuevas jugadas que faciliten su éxito mejorando su experiencia y satisfacción. Por consiguiente, es de interés para empresas desarrolladoras que buscan satisfacer a este tipo de clientes.

Si bien el software a desarrollar se desenvolverá en un ámbito concreto, juego de estrategia **ajedrez**; el enfoque es el mismo, siendo un buen prototipo aplicable en otros entornos.

Diseño

Para la implementación del software se utilizará diseño por contrato, por lo tanto, se especificarán los métodos pertenecientes a las clases con su respectiva precondición (condiciones de entrada, que se espera) y postcondición (condiciones de salida, que se garantiza).

El diseño del software se especificará mediante:

- Los paquetes.
- Las clases.
- Las clases y sus métodos.
- Métodos por contrato.
- Muestra visual.

Especificación de paquetes

ID	Nombre	Descripción
P01	Controlador	Aquí se encontrarán todas las clases necesarias para dar soporte de comunicación entre el modelo y las vistas.
P02	Modelo	En este paquete se encontrarán los objetos que conforman el software.
P03	Vista	Se encontrarán todas las vistas del software.

Especificación de clases

ID	Paquete	Clase
C01	Controlador	DrawHandler : IDisposable
C02		Historial_acciones
C03		Historial_movimiento
C04		Sprite
C05		Tiempo
C06	Modelo	Alfil
C07		Caballo
C08		Celda_tablero
C09		Jugador
C11		enum Tipo_de_jugador
C12		Movimiento
C13		enum Tipo_de_movimiento
C14		Peon
C15		Pieza
C16		enum UnColor
C17		Reina
C18		Rey
C19		Tablero
C20		Torre
C21	Vista	Recursos
C22		Ventana_bienvenida
C23		Ventana_coronacion
C24		Ventana_juego

Especificación de clases y sus métodos

ID	Clase	Métodos
M01	DrawHandler : IDisposable	DrawHandler()
M02		Dispose()
M03		Draw(precondición)
M04	Historial_acciones	Historial_acciones()
M05	Historial_movimiento	propiedades
M06	Sprite	Sprite(precondición)
M07		(Virtual) Draw(DrawHandler drawHandler)

M08	Tiempo	Tiempo()
M09	Alfil : Pieza	Alfil(precondición) : base(image, color)
M10	Caballo : Pieza	Caballo(precondición) : base(image, color)
M11	Celda_tablero	propiedades
M12	Jugador	Jugador(precondición)
M13	enum Tipo_de_jugador	Clase enumerada
M14	Movimiento	Movimiento(precondición)
M15	enum Tipo_de_movimiento	Clase enumerada
M16	Peon : Pieza	Peon(precondición) : base(image, color)
M17	abstract Pieza : Sprite	Pieza(precondición) : base(image, new Point())
M18		(Override) Draw(precondición)
M19	enum UnColor	Clase enumerada
M20	Reina : Pieza	Reina(precondición) : base(image, color)
M21	Rey : Pieza	Rey(precondición) : base(image, color)
M22	Tablero : Sprite	Tablero(precondición) : base(boardImage, new Point())
M23		Desmarcar_celdas()
M24		(Override) void Draw(precondición)
M25	Torre : Pieza	Torre(precondición) : base(image, color)
M26	Recursos	propiedades
M27	partial Ventana_bienvenida : Form	Ventana_bienvenida()
M28		Timer1_Tick(precondición)
M29		Timer2_Tick precondición
M30		Ventana_bienvenida_Load_1(precondición)
M31	partial Ventana_coronacion : Form	Ventana_coronacion(precondición)
M32		Reina_click(precondición)
M33		Alfil_click(precondición)
M34		Caballo_click(precondición)
M35		Torre_click(precondición)
M36	partial Ventana_juego : Form	Ventana_juego()
M37		Boton_cerrar_Click(precondición)
M38		Boton_minimizar_Click_1(precondición)
M39		Barratitulo_MouseMove(precondición)
M40		Boton_maximizar_Click(precondición)
M41		Boton_restaurar_Click(precondición)
M42		Boton_activar_consejos(precondición)
M43		Boton_manual_usuario_click (precondición)
M44		btnStart_Click(precondición)
M45		Ventana_juego_Canvas_MouseUp (precondición)
M46		Initialize()
M47		Cargar_imagen(precondición)
M48		Comenzar_juego()
M49		Siguiente_turno(precondición)
M50		Obtener_ubicacion_pieza(precondición)
M51		Añadir_pieza(precondición)
M52		Remover_pieza(precondición)
M53		Set_pieza_seleccionada(precondición)
M54		Set_movimientos_posibles()
M55		Obtener_movimientos_posibles (precondición)

M56		Validar_movimiento(precondición)
M57		Validar_movimiento_especial (precondición)
M58		Mover_pieza(precondición)
M59		Movimiento_especial_pieza(precondición)
M60		Update(precondición)
M61		Draw(precondición)
M62	enum Estado	Clase enumerada

Especificación de métodos por contrato

Precondición	Método	Postcondición
Controlador		
Int width Int height	DrawHandler()	Asignación de propiedades: Image Imagen_base, Graphics graphics
Graphics graphics	Dispose()	Libera los recursos utilizados por Graphics graphics
Image image Point position Graphics graphics	Draw(precondición)	Dibujar imagen en pantalla
	Historial_acciones()	Asignación de propiedades: List<Historial_movimiento>, List<Pieza> Pieza_removida, Pieza Pieza_añadida
Image image Point Ubicacion	Sprite(precondición)	Asignación de propiedades Image image, Point Posicion
DrawHandler drawHandler	(virtual) Draw(DrawHandler drawHandler)	Dibuja todos los sprites en pantalla
	Tiempo()	Asignación de propiedades: DateTime Fecha_inicial, DateTime Frame_fecha_actual
Modelo		
Image image, UnColor color	Alfil(precondición) : base(image, color)	Asignación de propiedades: Image image, UnColor color, Movimiento Movimientos
Image image, UnColor color	Caballo(precondición) : base(image, color)	Asignación de propiedades: Image image, UnColor color, Movimiento Movimientos
UnColor color, Tipo_de_jugador tipo_jugador, int numero	Jugador(precondición)	Asignación de propiedades: UnColor Color, Tipo_de_jugador Tipo_jugador, int Numero
int x, int y, bool isLinear = true, Tipo_de_movimiento	Movimiento()	Asignación de propiedades: Point Direccion, Tipo_de_movimiento Tipo_de_mov, bool isLinear

tipo_de_movimiento = Tipo_de_movimiento.nor mal		
Image image, UnColor color	Peon(precondición) : base(image, color)	Asignación de propiedades: Image image, UnColor color, Movimiento Movimientos
Image image, UnColor color	Pieza(precondición) : base(image, new Point())	Asignación de propiedades: Image image, UnColor color
DrawHandler drawHandler	override void Draw(precondición)	Dibuja todos los sprites en pantalla
Image image, UnColor color	Reina(precondición) : base(image, color)	Asignación de propiedades: Image image, UnColor color, Movimiento Movimientos
Image image, UnColor color	Rey(precondición) : base(image, color)	Asignación de propiedades: Image image, UnColor color, Movimiento Movimientos
Image boardImage, Image mover_imagen	Tablero(precondición) : base(boardImage, new Point())	Asignación de propiedades: Image boardImage, Image Mover_imagen. Generar instancia con la información del tablero de juego, particularmente ubicación de cada celda.
	Desmarcar_celdas()	Desmarcar celdas que permiten movimiento
DrawHandler drawHandler	override void Draw(precondición)	Dibuja todos los sprites en pantalla
Image image, UnColor color	Torre(precondición) : base(image, color)	Asignación de propiedades: Image image, UnColor color, Movimiento Movimientos
Vista		
showDialog()	Ventana_bienvenida()	Inicializa los componentes
object sender, EventArgs e	Timer1_Tick(precondición)	Hace visible el form gradualmente según un timer
object sender, EventArgs e	Timer2_Tick precondición	Hace invisible el form gradualmente según un timer luego cierra el form
object sender, EventArgs e	Ventana_bienvenida_Load_1 (precondición)	Controla el evento de carga del formulario.
Recursos recursos, UnColor color	Ventana_coronacion(precondición)	Inicializa los componentes y propiedades
object sender, EventArgs e	Reina_click(precondición)	Guarda la selección del usuario de la coronación y cierra el form
object sender, EventArgs e	Alfil_click(precondición)	Guarda la selección del usuario de la coronación y cierra el form
object sender, EventArgs e	Caballo_click(precondición)	Guarda la selección del usuario de la coronación y cierra el form

object sender, EventArgs e	Torre_click(precondición)	Guarda la selección del usuario de la coronación y cierra el form
Application.Run	Ventana_juego()	Inicializa los componentes
object sender, EventArgs e	Boton_cerrar_Click(precondición)	Cierra el form
object sender, EventArgs e	Boton_minimizar_Click_1(precondición)	Minimiza el form
object sender, MouseEventArgs e	Barratitulo_MouseMove(precondición)	Mover el form
object sender, EventArgs e	Boton_maximizar_Click(precondición)	Maximizar el form
object sender, EventArgs e	Boton_restaurar_Click(precondición)	Restaura el form
object sender, EventArgs e	Boton_activar_consejos(precondición)	Permite activar los consejos del agente y modifica el color del botón
object sender, EventArgs e	Boton_manual_usuario_click (precondición)	Permite apertura del manual y modifica el color del botón
object sender, EventArgs e	btnStart_Click(precondición)	Permite comenzar la partida desde 0.
object sender, MouseEventArgs e	Ventana_juego_Canvas_MouseUp (precondición)	Muestra los movimientos disponibles para la pieza seleccionada en el tablero
	Initialize()	Permite cargar los recursos necesarios para mostrar el tablero particularmente imágenes.
string path	Cargar_imagen(precondición)	Devuelve una variable de tipo Image
	Comenzar_juego()	Asignación de propiedades: List<Historial_acciones>ActionLog List<Pieza> Piezas Tablero (Image,Image) Board Estado GameState, Añade todas las piezas al tablero y llama al siguiente turno.
bool firstTurn	Siguiente_turno(precondición)	Finaliza el turno actual y recalcula los movimientos para cada pieza del tablero, Recalcula los movimientos disponibles, Revisa si el rey se encuentra en jaque, Asigna el estado al match,
Point location	Obtener_ubicacion_pieza(precondición)	Devuelve coordenadas (modificadas) de la pieza acordes al tablero (Point).
Pieza piece, Historial_acciones log = null	Añadir_pieza(precondición)	Permite añadir una pieza al listado de piezas
Pieza piece, Historial_acciones log	Remover_pieza(precondición)	Elimina una pieza del listado de piezas
Point cell_Location	Set_pieza_seleccionada(precondición)	Desmarca las celdas del tablero, Deselecciona todas las piezas,

		Colorea las celdas donde tiene movimiento la pieza seleccionada.
List<Pieza> Piezas	Set_movimientos_posibles()	Obtener los movimientos habilitados para cada pieza.
Pieza piece, List<Pieza> boardPiezas	Obtener_movimientos_posibles (precondición)	Envía un arreglo con las coordenadas de los movimientos a la cual puede moverse una pieza
Pieza piece, Point newLocation	Validar_movimiento(precondición)	Devuelve un booleano true si el movimiento es válido false en caso contrario, esto es para evitar que quede en jaque al mover una pieza.
Pieza piece, Movimiento move, Pieza rivalPiece	Validar_movimiento_especial (precondición)	Devuelve un booleano que revisa si puede desplazarse la pieza seleccionada utilizando un movimiento especial, true si puede false en caso contrario
Point cell_Location	Mover_pieza(precondición)	Devuelve un booleano que determina si es posible mover la pieza a la ubicación especificada, además pinta en el tablero la pieza desplazada en su nueva ubicación y llama al siguiente turno.
Pieza piece, Point targetLocation, Historial_acciones log	Movimiento_especial_pieza(precondición)	Ejecuta un movimiento especial, por ejemplo, coronación, comer al paso, enroque corto y largo.
Tiempo gameTime	Update(precondición)	Actualiza la posición de todas las piezas en el tablero
DrawHandler drawHandler,	Draw(precondición)	Dibuja todos los sprites en pantalla

Muestra

Ventana de bienvenida



Ventana del juego



Ventana cuando ocurre selección de pieza



Ventana cuando se requiere coronación de peón.



Nota Prototipo 1: Se encuentra desarrollado el aspecto visual del software y también el aspecto lógico de todas las reglas del juego incluyendo movimientos especiales:

- Al paso.
- Enroque.
- Coronación.