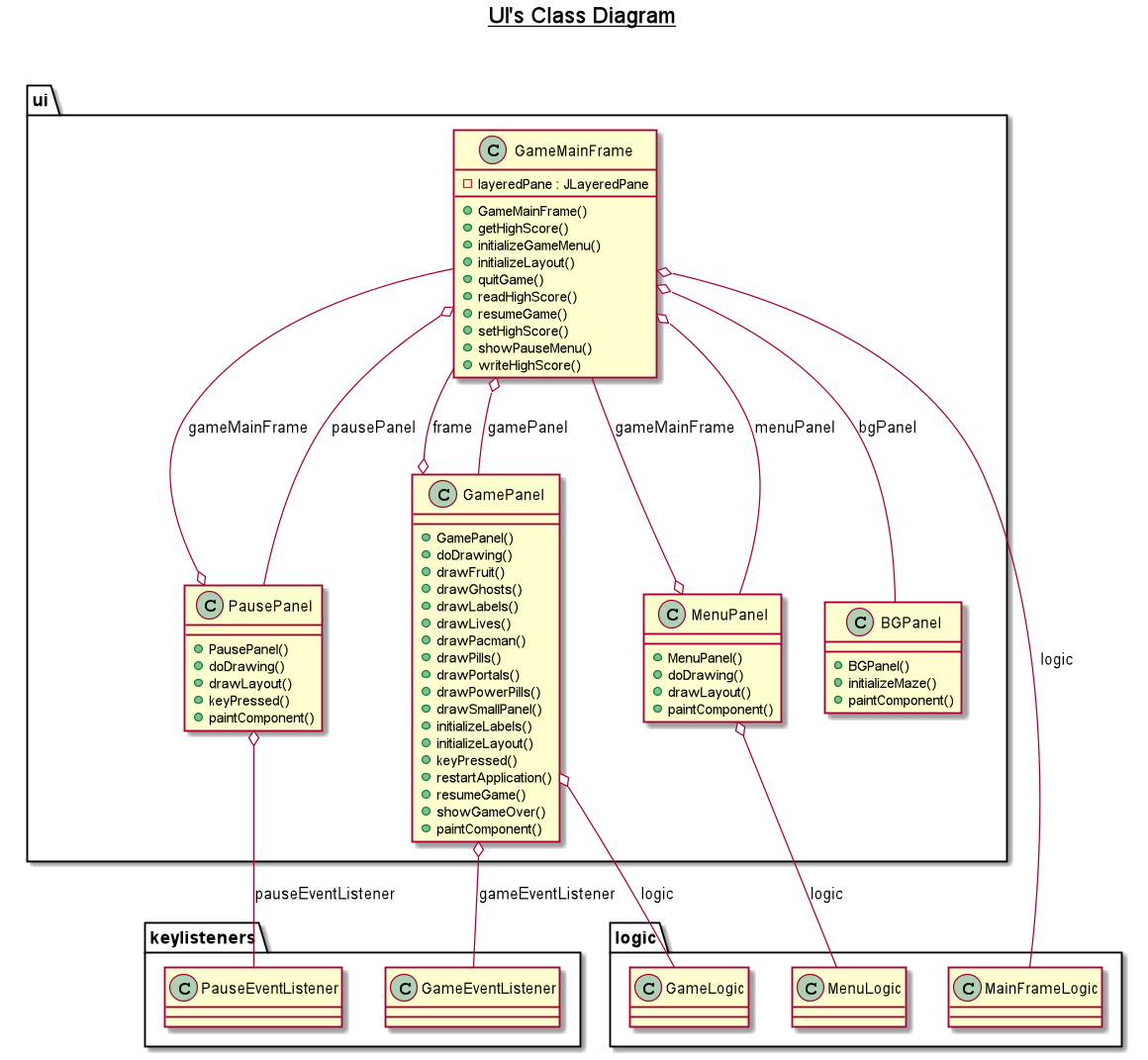
# Analisi dei package di JPacman

Nello sviluppo dell’applicazione siamo partiti con l’idea di sviluppare un gioco altamente modulare e che ci offrisse oggetti da poter riutilizzare senza troppe modifiche in altri ambiti e con funzionalità che potessero essere rimosse ed aggiunte all’occorrenza, anziché seguire pattern come quello del Model-View-Controller abbiamo cercato di inglobare il comportamento di ogni entità all’interno di singole classi a sé stanti: ogni personaggio ed elemento dell’applicativo contiene infatti non solo le informazioni sul suo stato, la sua posizione e il frame dell’animazione (ciò non dovrebbe comunque creare problemi di portabilità in quanto si tratta di un semplice riferimento ad un ImageIcon, sostituibile a seconda delle necessità), calcolato con un apposito oggeto, ma anche un Timer che gli permette di implementare un comportamento ben definito a seconda dei casi che gli vengono notificati. Le classi che si occupano di utilizzare effettivamente componenti grafiche e di stamparle su schermo sono affiancate da classi logiche che implementano l’effettivo controllo su cosa stampare e quando. Sono poi presenti classi di struttura, come il gestore dei labirinti che mette a disposizione metodi e strumenti per generare e potersi muovere dentro un qualsiasi scenario, e di supporto come il gestore dei suoni e quello delle immagini memorizzate nel File System.

Andiamo ad analizzare ogni package partendo dai più importanti:

## UI



Le classi facenti parte di questo package sono le uniche che, escludendo il riferimento alla loro immagine attuale degli sprite, operano con delle componenti grafiche e si occupano di stamparle su schermo. Eccetto GameMainFrame che estende JFrame, tutte estendono JPanel: Nel GameMainFrame infatti ci occuperemo di impilare questi quattro pannelli a seconda delle necessità: mentre l’unico presente all’avvio sarà MenuPanel che mostrerà appunto il menù principale, appena il gioco verrà avviato questo verrà sostituito da BGPanel con la struttura del labirinto con sopra GamePanel per mostrare gli sprite presenti in esso e le JLabel di stato. Alla pressione del tasto Invio sopra dei due pannelli verrà posto PausePanel che mostrerà il menù di pausa con le sue opzioni. Le funzionalità logiche delle classi sono affidate alle loro controparti nel pacchetto “logic” in quanto in “ui” tutte le operazioni sono funzionali al mostrare su schermo.

## Logic

Le classi del “logic” si occupano dell’effetiva logica con cui le classi del “ui” mostrano le componenti e le immagini, mentre MenuLogic, che popola la ComboBox per la scleta del labirinto nel menù principale, e MainFrameLogic, che registra il font nell’ambiente grafico e registra l’HighScore, sono praticamente più un supporto alle loro controparti UI, GameLogic è effettivamente una delle classi più importanti dell’applicazione, in quanto si occupa di scandire i cicli effettivi update-repaint del gioco e di inserire gli sprite all’interno dell’ambiente e di stabilire ciò che accade se avviene una collisione tra di essi. Mostriamo ora alcuni dei suoi metodi che vengono chiamati quando si inizia un partita:

1. GameLogic(): Inizializza tutte le variabili necessarie al corretto funzionamento del gioco e avvia il timer principale che richiamerà periodicamente, grazie alla classe GameLoop, il prossimo metodo;
2. doOneLoop(): Richiede il focus della tastiera su GamePanel, richiama update() e repaint sempre di GamePanel;
3. update(): Se passato il breve periodo di pausa, ordina al Pac-Man e ai fantasmi di muoversi chiamando il loro metodo move(), controlla le collisioni con checkCollision(), controlla se il livello deve terminare e sceglie quale effetto in loop riprodurre;
4. endGame(): Chiamato se Pac-Man mangia tutte le pill, resetta il labirinto e passa al livello successivo chiamando restartLevel();
5. restartLevel(): Chiamato nel caso precedente o nel caso Pac-Man perda una vita, riporta tutti i personaggi nelle loro posizioni iniziali e riavvia il timer;
6. restartApplication(): Chiamato nel caso di un Game Over o se l’utente sceglie di abbandonae la partita, salva un eventuale HighScore e fa tornare il controllo al GameMainFrame.

## Sprites