

## Descrizione struttura del programma e architettura:

Il programma è strutturato in modo tale che definendo una parola nel codice o passando la parola come parametro al compilatore, questo generi una diversa versione del programma.

Nella versione 1 durante la compilazione bisogna definire PROCESSES per permettere al compilatore di generare il programma.

Le funzioni sono definite per “categoria” divisi in tanti file, ognuno di questi file ha allegato un file header (estensione .h) che ne descrive per ciascuna funzione di quella categoria le sue funzioni (quello che fa).

Il programma è scritto in maniera tale che un thread scriva sul buffer 1 valore per volta, il main si occuperà di controllare la pipe e di elaborare il dato in base a delle regole impostate in esso per ciascun processo (questo significa che ci sono regole diverse per il processo fantasma e per il processo Pac-Man).

Il buffer è una variabile globale gestita con mutex, solo un thread per volta può accedere al buffer, ma questo solo usando 2 funzioni apposite un po' come per nelle pipe: `bufferWrite()` e `bufferRead()`.

Il main esegue il controllo sulla pipe (per leggere un valore) ogni tot secondi, è stato definito un numero che determina quante volte deve fare questa operazione (FAKE\_FPS). A prescindere dal fatto che sia stato letto o meno un dato dal buffer, questa attesa tra un tentativo e l'altro rimane la stessa.

Il main è stato scritto in questa maniera per dare la possibilità ad altri elementi di agire sul gioco senza bloccare l'esecuzione del processo stesso (come la scritta 1UP che lampeggia, il controllo per i fantasmi quando, prendendo una power pill, diventano vulnerabili e per rendere il movimento di Pac-Man automatico anche senza premere continuamente un tasto per la stessa direzione).

Il gioco non ha una fine preimpostata, si torna alla schermata di inizio se si perdono 3 vite, il resto viene sempre rieseguito alla stessa maniera, senza però azzerare il punteggio più alto che è stato fatto (highscore).

Gli altri thread, fantasmi, missili e Pac-Man hanno il compito di scrivere nel buffer un valore che verrà poi elaborato nel main.

Questo valore è diverso per ciascuno di loro (identico tra i fantasmi dato che utilizzano la stessa funzione), iniziando dai fantasmi, il thread di uno di questa funziona generando un numero casuale da 0 a 2 che va a specificare se il fantasma in questione vuole modificare la sua direzione corrente, nel main questo valore viene confrontato e viene provata la nuova direzione per vedere se è fattibile, nel caso lo sia, viene modificata, nel caso contrario prova l'altra direzione che non sia quella opposta a quella corrente, se fallisce anche questa continua per la direzione che aveva (come se non avesse considerato il valore nella pipe), e nel caso non potesse nemmeno andare avanti nella direzione corrente, va per la direzione opposta.

Questo complicato modo di inviare una nuova posizione ha evitato l'uso di un altro buffer (infatti il buffer è unico per tutti i thread) e la duplicazione di variabili come il labirinto su ogni thread che doveva farne utilizzo, o la creazione di un altro mutex per gestire i thread che potevano voler usare il labirinto se fosse stato globale.

I fantasmi inoltre possono decidere se muoversi o lanciare un missile, nel secondo caso viene generato un numero casuale da 0 a SHOOT\_MISSILE (#define) e se è 0, viene controllato se il missile può essere lanciato, per ogni missile se ne ha la possibilità viene generato un thread che quando scrive sul buffer, lo muove.

Se viene deciso dal thread fantasma di sparare un missile, la nuova posizione del fantasma non viene calcolata. In questo modo si ha una sola scrittura sul buffer da parte

del thread per ogni iterazione (anche i thread fantasmi, come il main, Pac-Man e i missili, hanno un ritardo di esecuzione definito nel codice tramite #define).

I thread che muovono i missili funzionano in maniera diversa, oltre al valore che li identifica (i fantasmi scrivono oltre al valore detto prima, anche un numero che identifica il fantasma in questione, che è anche la posizione che hanno nell'array in cui sono stati dichiarati) scrivono invece che un numero che determina se vogliono cambiare direzione, un numero che identifica chi li ha sparati e di quale missile si tratta. Questo numero verrà poi scomposto nel main per ritrovare questi valori e poter muovere eventualmente il missile. Questo è stato fatto per far tornare i missili al "padrone" una volta che fossero esplosi (colpiscono un muro o qualsiasi altro oggetto del labirinto, oppure hanno colpito Pac-Man o i fantasmi). Questo fa capire anche che ogni missile ha già di suo una direzione preimpostata che non può modificare.

Pac-Man invece scrive la direzione che vuole effettuare, questa direzione viene salvata dal main ad ogni iterazione e viene usata appena possibile.

