

2020 CI401
Introduction to programming

Week 2.03

Maze project notes

22th February 2021

Roger Evans

Module leader

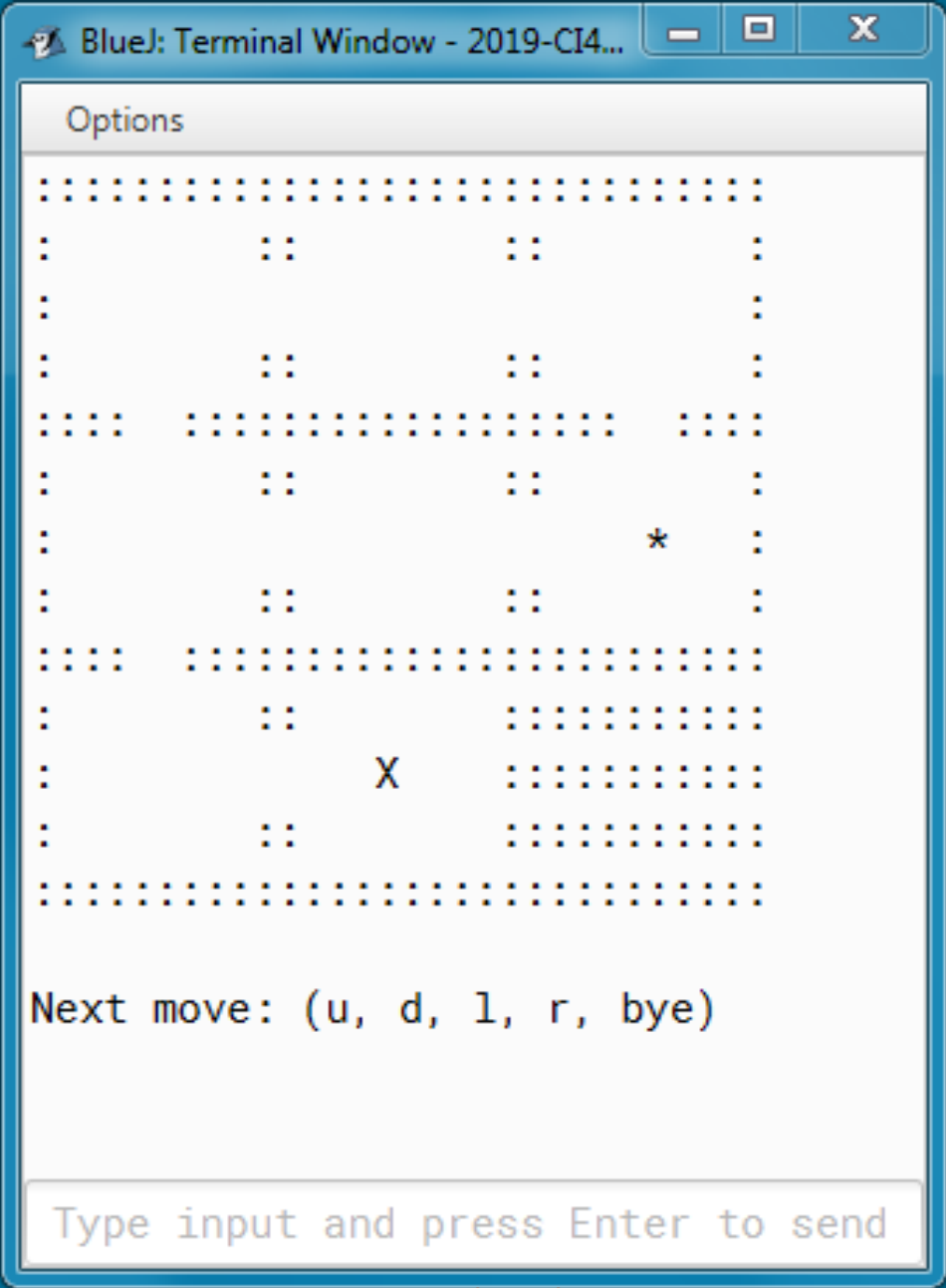
Introduction to Maze

Overview

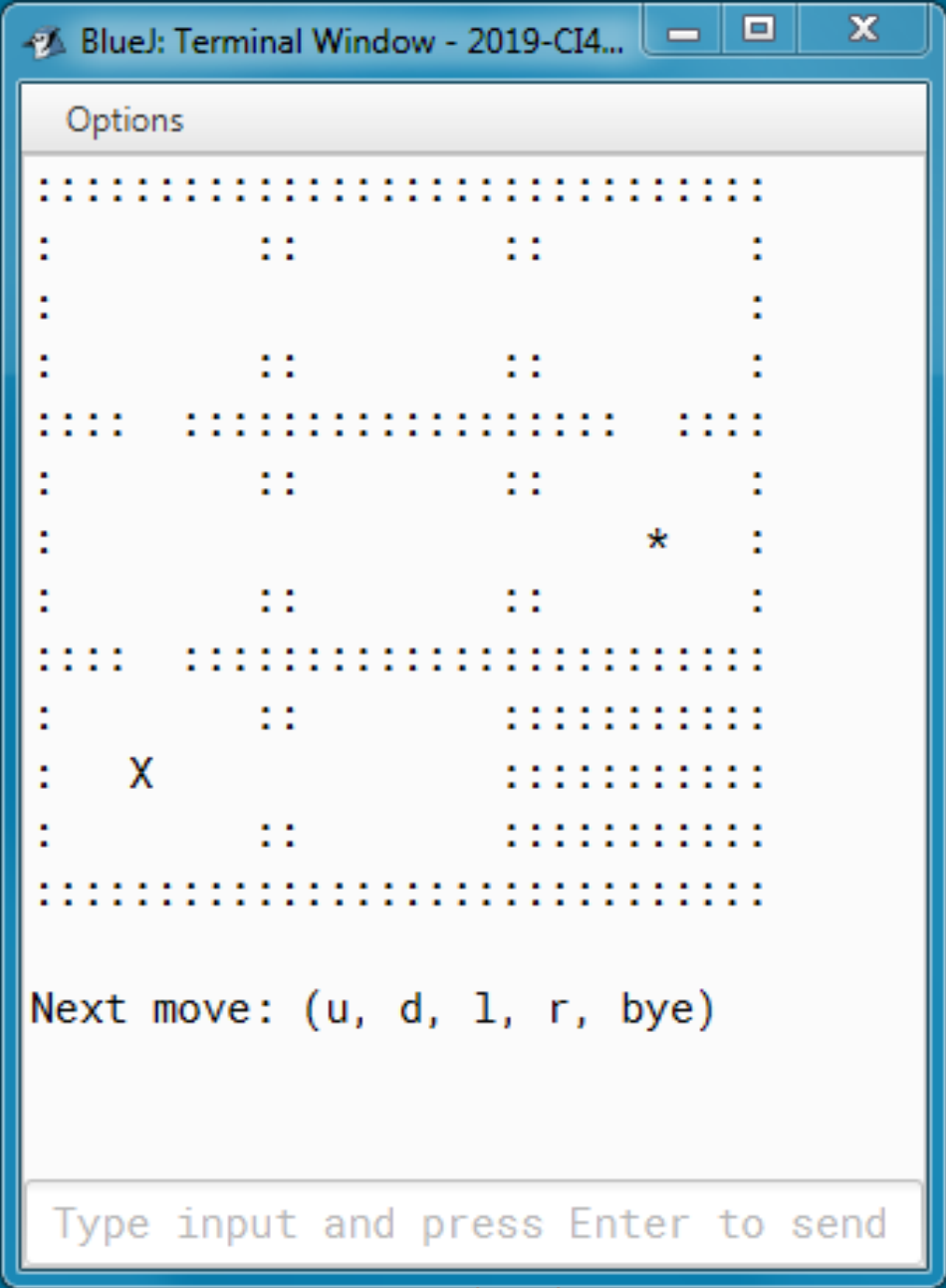
- Maze was originally a lab exercise, but was promoted (demoted?) to a project option
- It provides the basic structure for 'maze-based' games, plus three examples of simple game techniques:
 - Synchronous player-driven search for goal (treasure)
 - Asynchronous non-player-character search for goal (zombie seeking treasure)
 - Synchronous/asynchronous player searching for treasure while being chased by zombie
- Technically, it is a bit more advanced than other projects, but does not use event-driven JavaFX technology
- It provides opportunities for adding features, or enhancing the technological approach

Demonstration

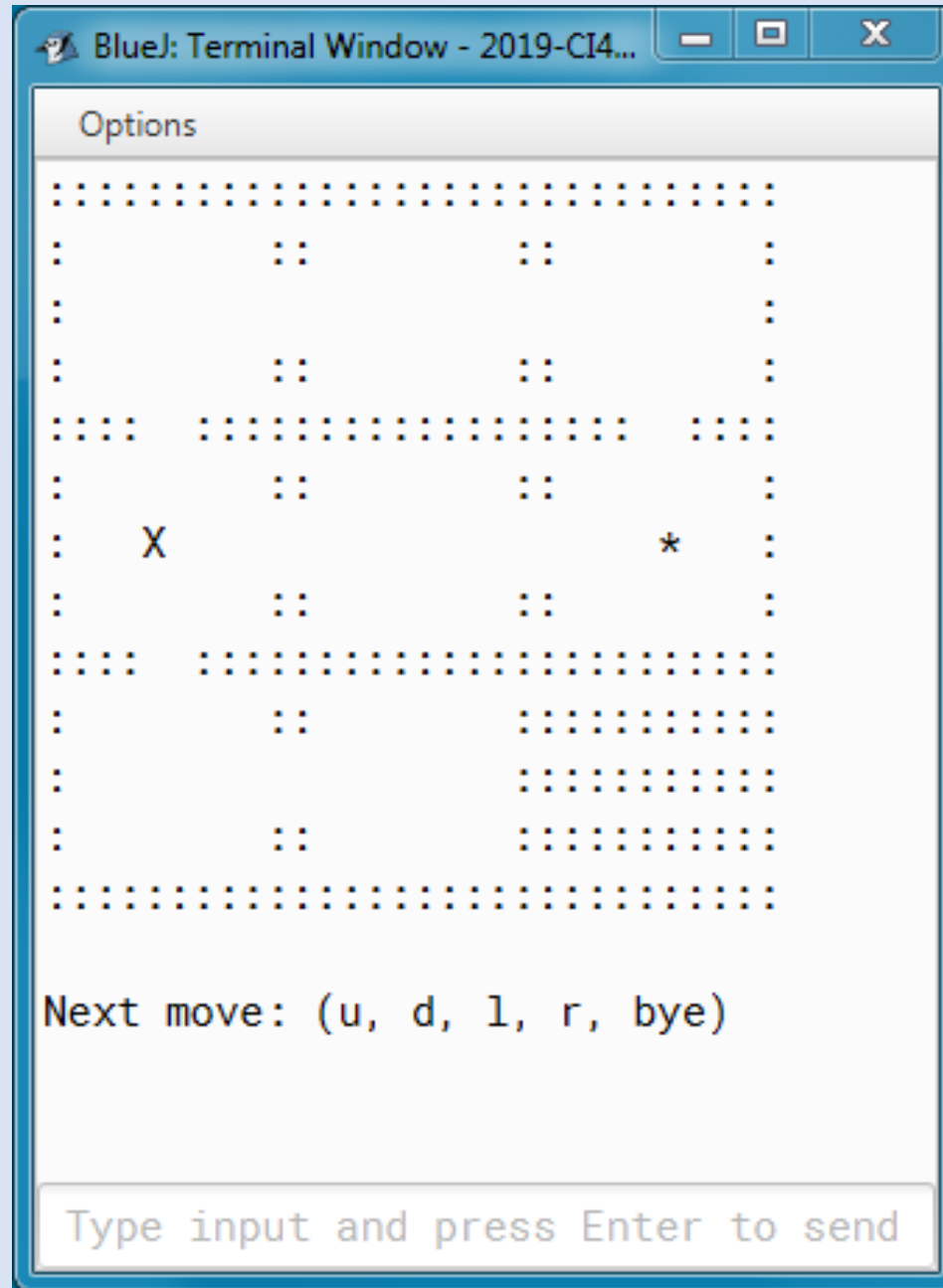
Maze game



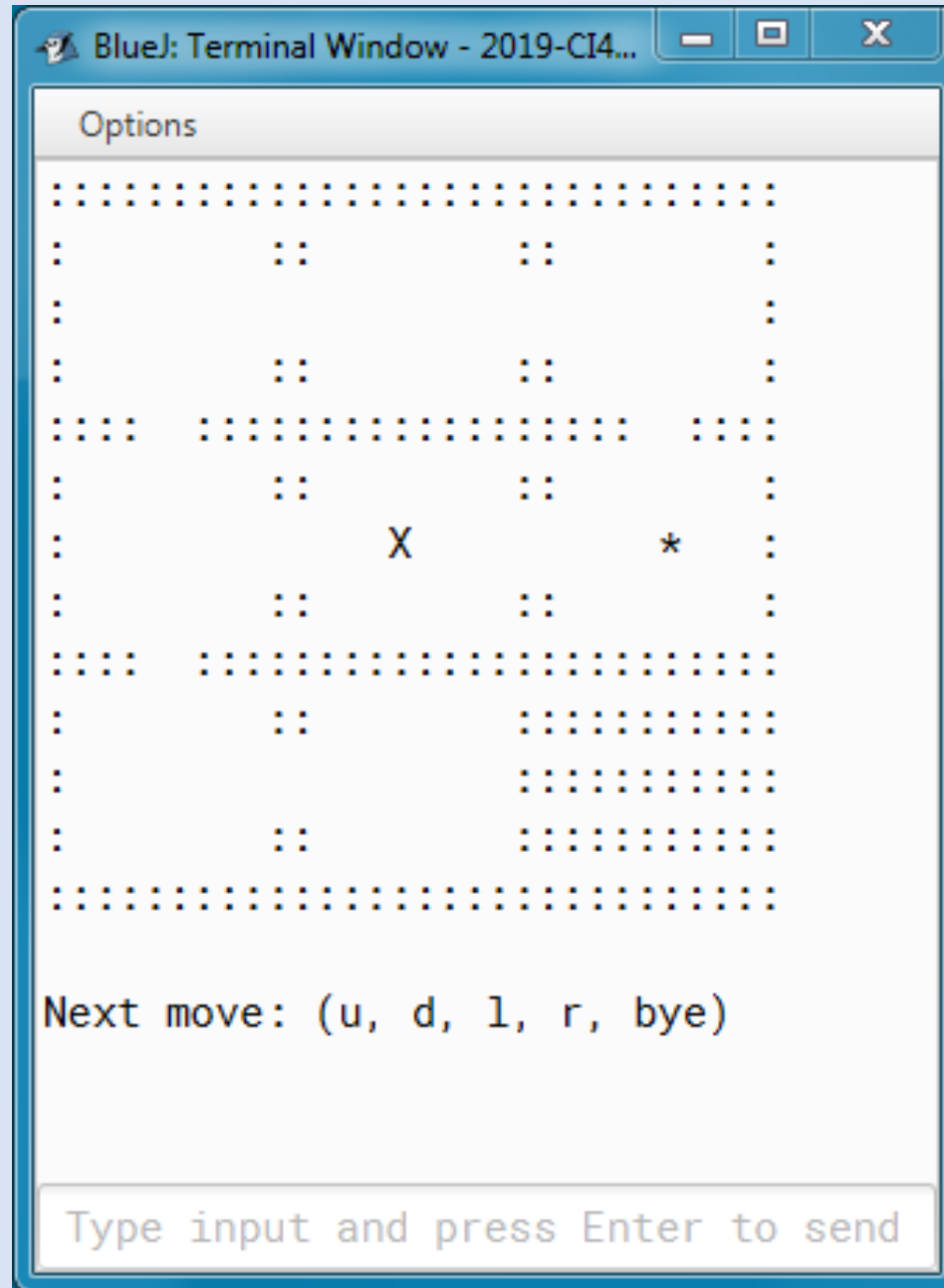
- Game initial position
 - A grid of 9 cells
 - Each cell is either a Room or a block
 - Rooms have doorways between them
 - X marks the player's position, * marks the treasure.
 - The goal is to get the treasure



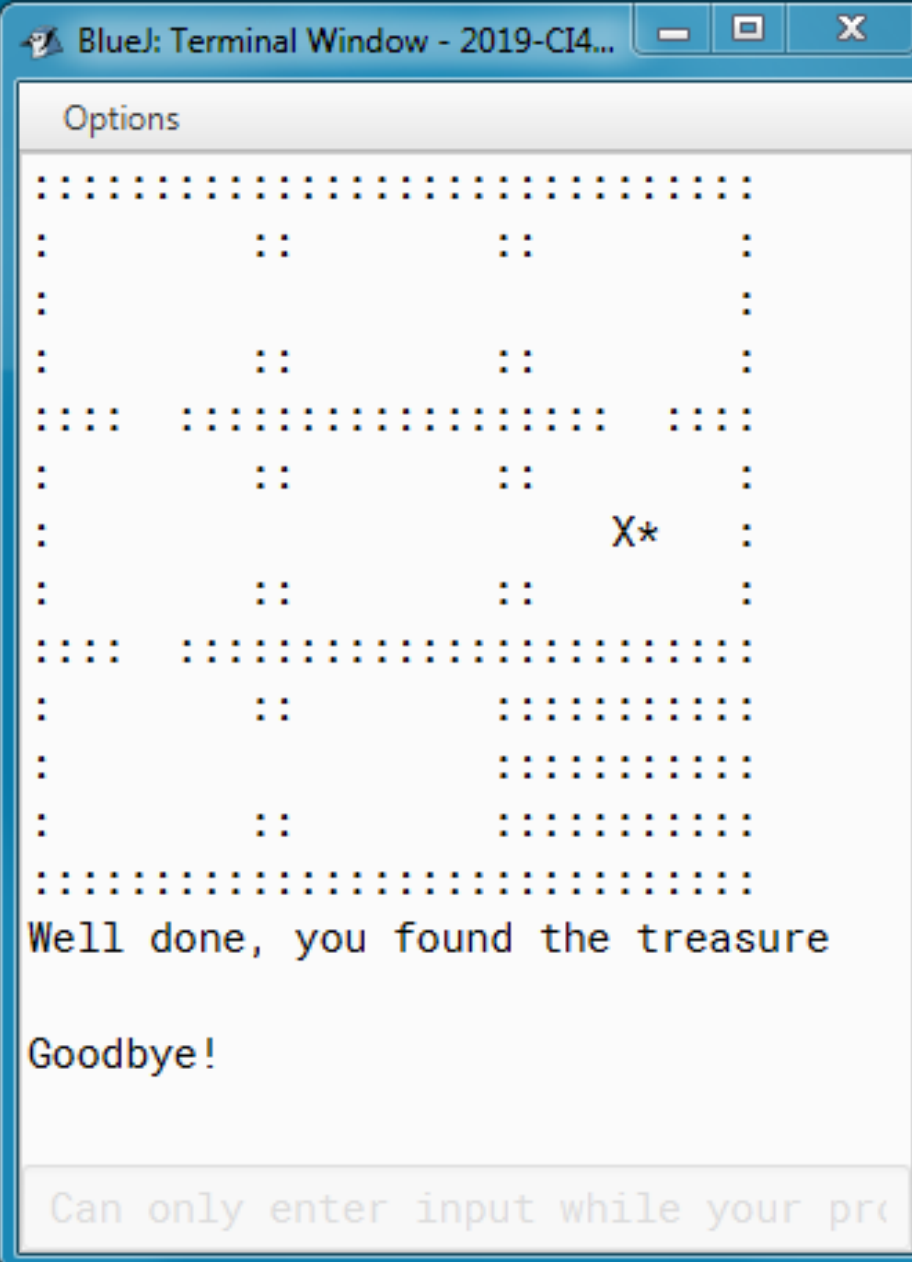
- After typing 'l' (left)
 - The grid has redisplayed
 - The player position has moved to the left



After typing 'u' (up)



After typing 'r' (right)

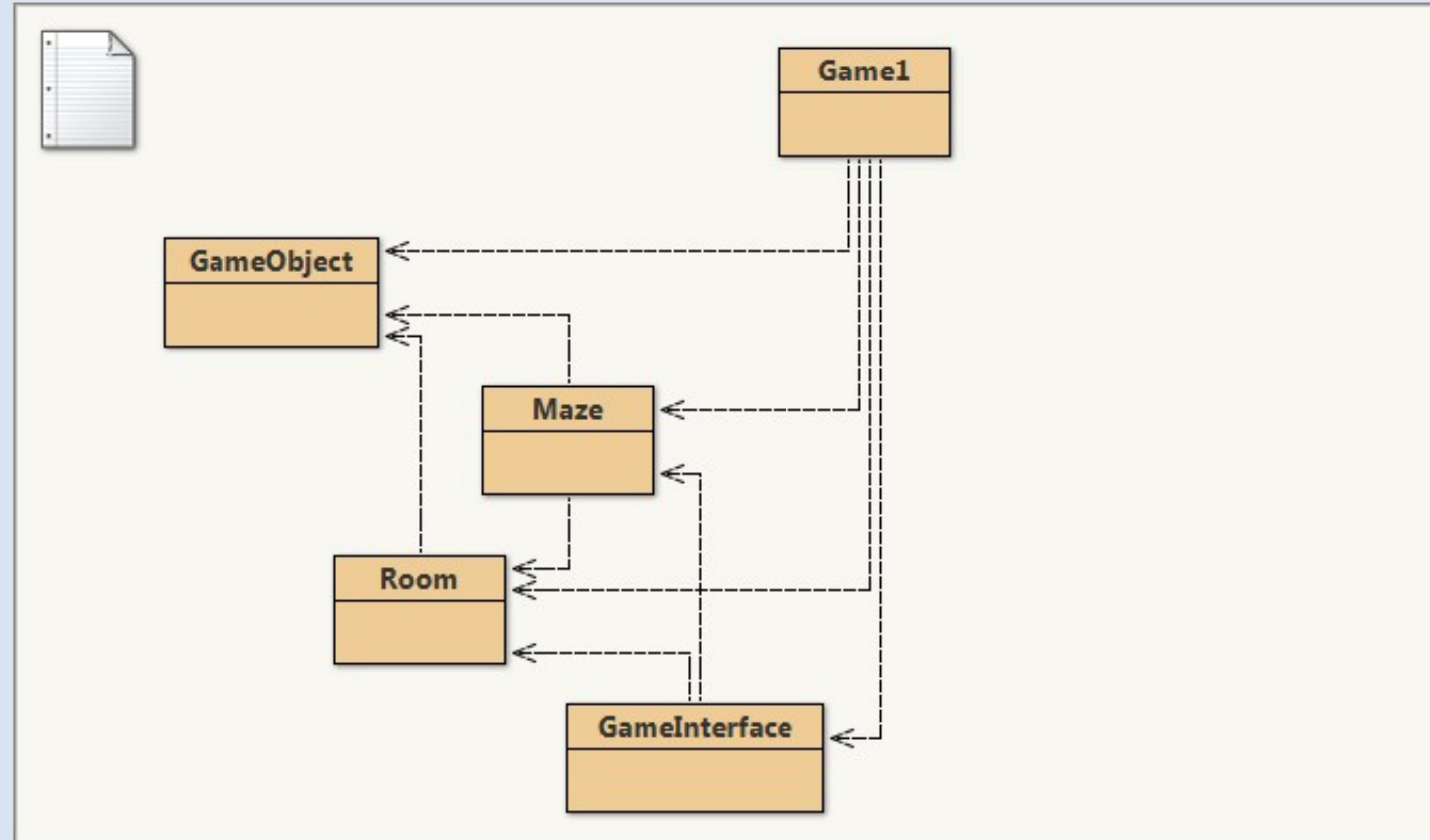


After typing 'r' (right) – we have reached the treasure and the game finishes

Maze game classes

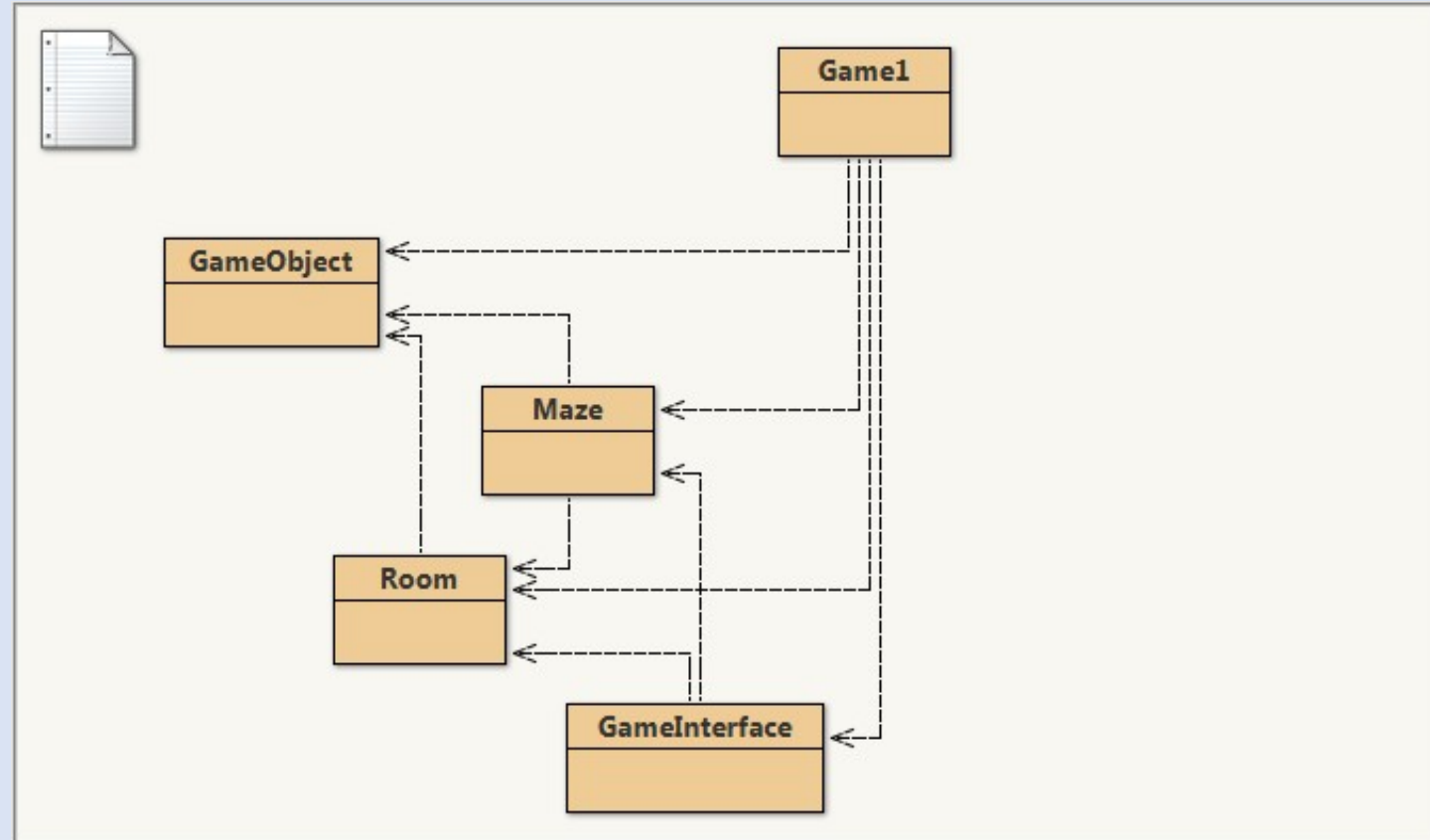
Game1

Game1 – runs the game. It sets up a **Maze** object consisting of **Rooms** and **GameObjects**, and then uses the **GameInterface** to display the maze and get user input



Maze

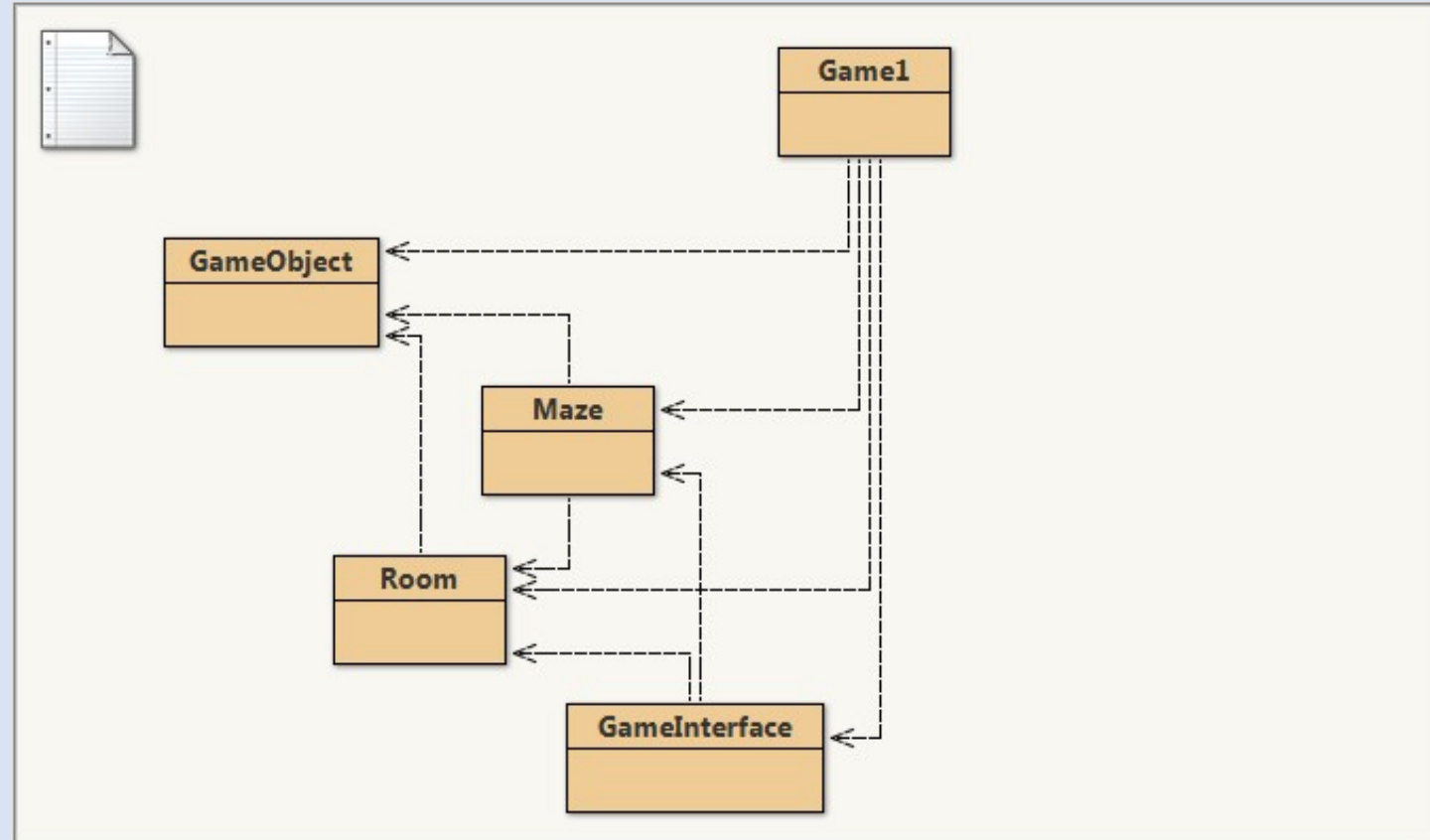
Maze is just an two dimensional array of **Room** objects. But in addition to the data it has methods for using the maze – for example a method to find out if it is possible to move from one room to another (is there a door?), and a method to move an object from one room to another



Room

Room is a data class (think Top Trumps card) with info about where it has doors (up, down, left or right) and what objects are in it (player, treasure, zombie etc).

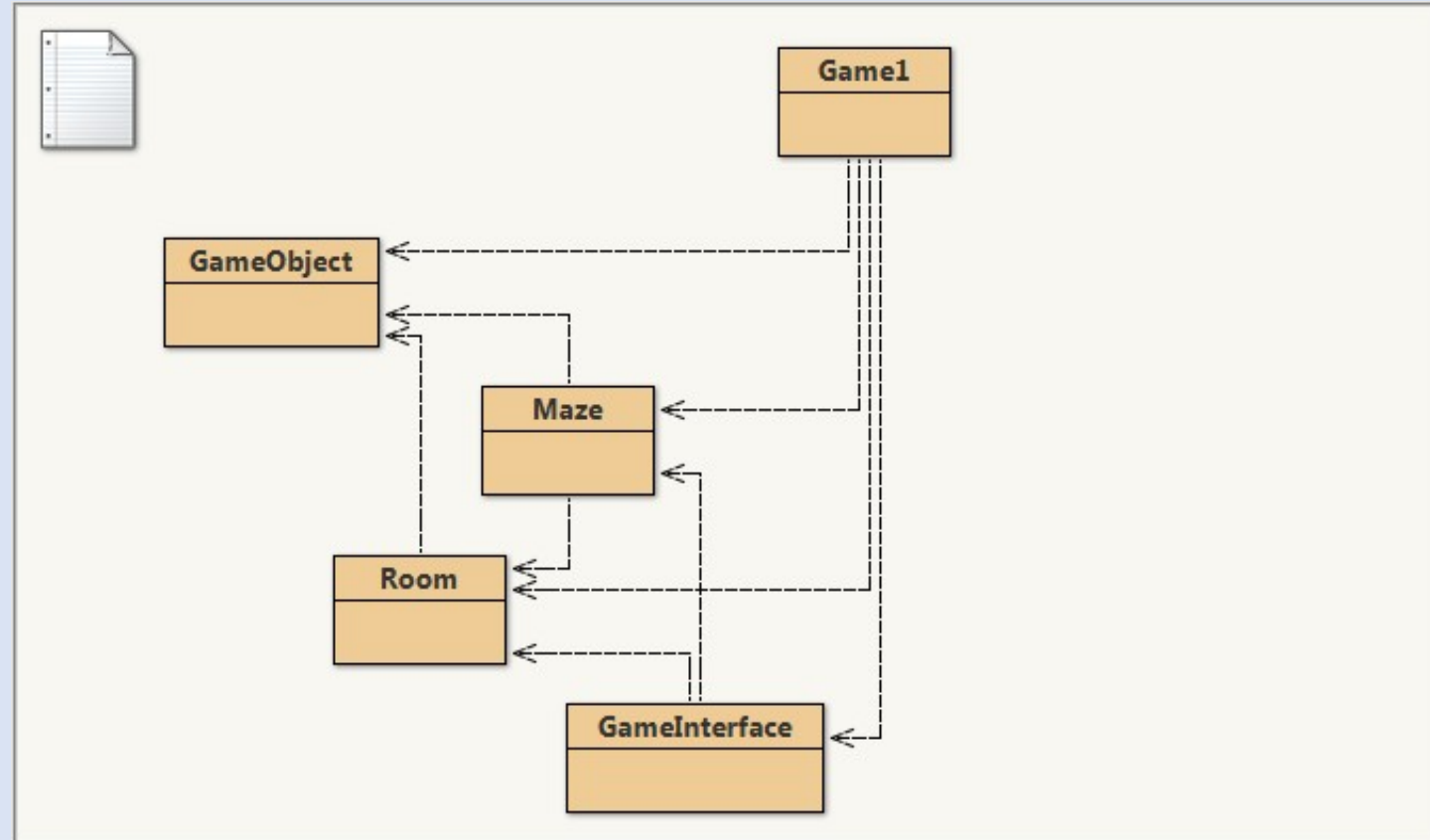
But it also has methods to add something to a room, and also to display the contents of a room.



GameObject

GameObject is another data class (think Top Trumps card) with info about individual moveable objects – players, treasure ghosts etc.

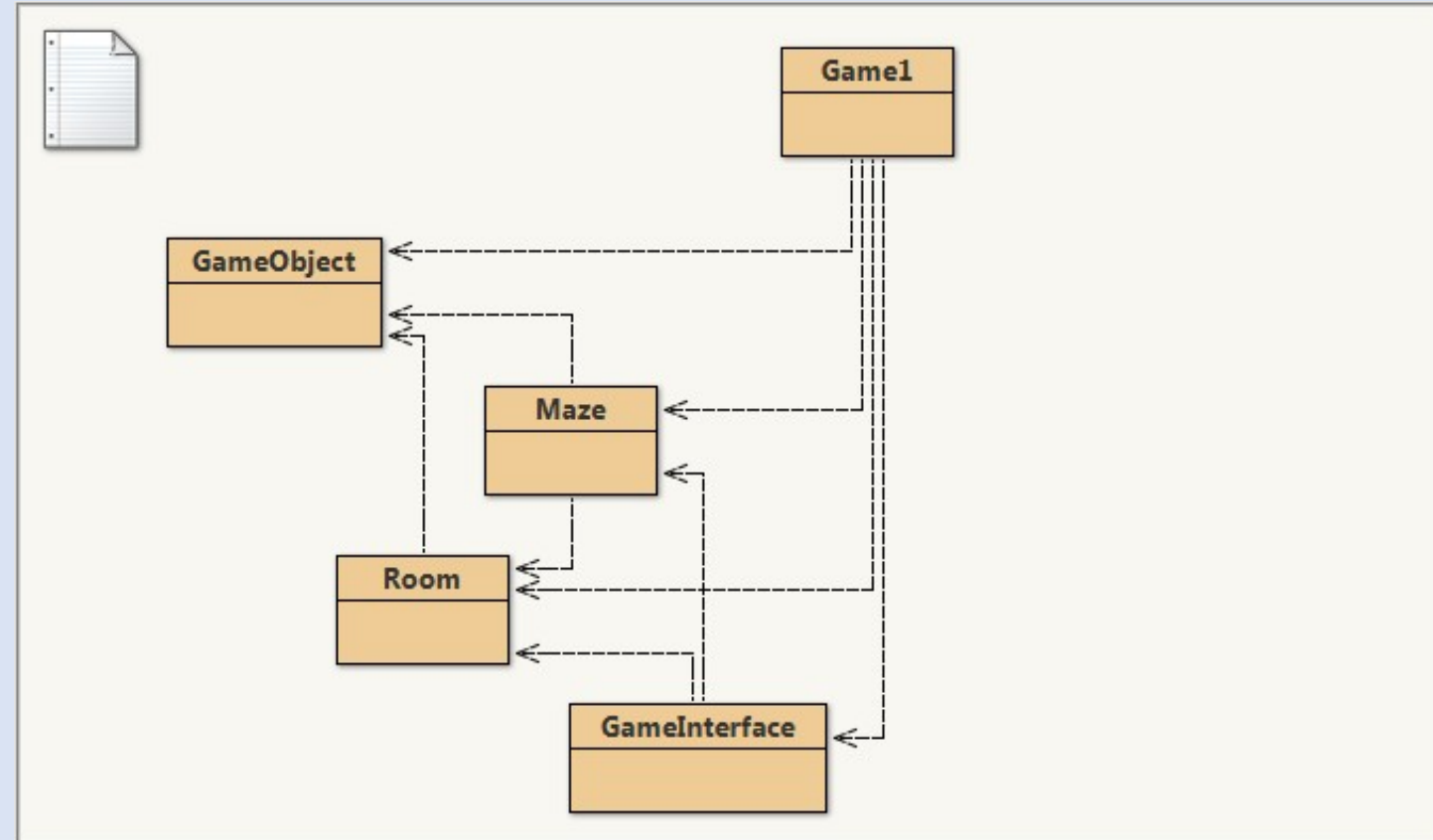
It mainly remember what type they are, where they are (in the maze) and how they are displayed



Game interface

GameInterface manages the screen and keyboard. The output is very basic (because we only know how to print strings!), and the input uses Scanner.

Game 1 repeatedly updates the screen by printing the interface every half second, to give an impression of movement.



Feature additions

Exercises

- change the character for the player to be "P" instead of "X"
- change the starting position of the player and the treasure
- design your own grid - make it bigger and more interesting (not always as easy as it seems - draw it on paper first, and then work out where all the solid walls are)

Treasure

- the method `updateTreasure` is called every time we go round the game loop, but doesn't do anything because the treasure never changes.
- Add a new instance variable to the game which is the lifetime of the treasure, and modify `updateTreasure` so the lifetime reduces by one and when it gets to zero, the treasure disappears
- NB: `Game3` has a solution for this!

Non-player characters

- Add one or more GameObjects for non-player characters (NPC) you may meet in different rooms (eg pets or people)
- You will need to update the Room class to have another contents type - a new instance variable and an update to the contents() method – and work out how to display it.
HINT: copy the code for the treasure in Room and Game1 as the basis for this.
- Get these people to interact with you - say a greeting, or ask a question
- Add objects you can collect, and maybe use (eg weapons)

General game playing

- Zombie/monster characters – smarter search strategies, ability to fight them etc
- Add a second player, who also types in a commands on different keys after the first one, and is racing the first player to get to the treasure
- Use the Random class to see how to introduce random start positions for players and treasure etc.

Technical bits

Drawing the maze

- All just printing simple strings
- Print on terminal every half second, clearing it between updates
- Triple depth loop is a bit complex:
 - Two loops for X Y dimensions of map
 - One because each 'row' is actually 5 lines of text
 - 'interleaving' of the column and line drawing
- Would be much easier to 'draw' onto an array of characters and then just write out the resulting array
- Convert to JavaFX?

Main game loop

- Uses Thread technology (like Breakout), but running once every half second
- Call explicit 'update' methods for each GameObject
- Includes some use of random numbers to create more interesting behaviour
- Switch to a proper event-driven/event handler model?
- MVC?
- JavaFX, CSS etc?

Synchronous input

- Can't sit and wait (block) for user input
- Uses technology to ask System.in whether the user has typed anything before trying to read it
- Completely different from JavaFx approach (but not dissimilar to how the 'secret' GUI loop actually works!!)