

Sprint 1

The game should let the player walk around in a “cave” where there are things which can be picked up. In later iterations, some things will be required to solve some puzzles or tasks. For this sprint, we’ll focus on some basic functionalities.

In short, the game have the following features:

- a room can have 0..4 directions, leading to a room
- a room has 0..n Things
- the player knows its current room
- the player shall be able to **go** in various directions
- a player can have 0..n Things in his/her inventory
- the player can **pick up** any things from a room
- the player can **drop down** any things, from its inventory, in a room
- a Thing the player has picked up is no longer present in the room, but added to the player’s inventory
- a Thing that the player has dropped down is no longer in the player’s inventory, but added to the things present in the room
- It is not possible for a player to go in a direction if the current room has no such connecting room
- when the game starts, the player is in the first room and has nothing in its inventory

We have underlined some nouns - they are candidates for classes from which we can make objects. We have made **bold** some verb phrases - they are candidates for behavior, that is methods for classes. Think of methods as messages we pass to an object of some class. If we have a Player object, we should be able to ask it to “drop” a Thing. So the Player class needs a drop(Thing) method. The result of the method call should be that the thing is moved from the player’s inventory to the room’s list of things.

In short...

The player should be able to walk around between the rooms in the “cave” and if there are any things in a room, the player should be able to pick them up (move them from the room to the player’s inventory). The player can also drop down a thing to the current room. Doing so should add the thing to the room’s list of things and remove the thing from the player’s inventory. The rooms in the cave are connected according to the data in the database (you also have some PDFs with maps - ask the teachers).

Invisible things

The game does not have any such concept. All things are visible.

Attainable things

The game does not have any such concept. All things are attainable.

Die and terminate

We will not deal with the concepts “die” and “terminate”.

Things in the game

The game shall (in this version) consist of the following objects:

- Rooms (apart from “Game Over rooms”)
- Things (have a name, e.g. “Bird”)
- Player (keeps an inventory and knows its current Room)

Rooms are connected to each other via references. A Room can have four references to other Room objects, according to the map of the cave.

Things can be used one at the time. Things that belong together (e.g. the things in a room, or the things in the Player’s inventory) can be put in some kind of list.

Excluded objects and other stuff (in this version/sprint):

- Snake
- Dragon
- All rules (room rules, thing rules)

Comments on the GUI

We provide skeleton GUIs (Swing and Android). You need to add code making the GUI react with “the game”.

We have based our GUIs on the following.

- It shall be easy to put down one or many things
- It shall be easy to pick up one or many things
- All things in a room are automatically presented to the user
- All things in the inventory are automatically presented to the user
- It shall not be possible to enter a room that does not exist