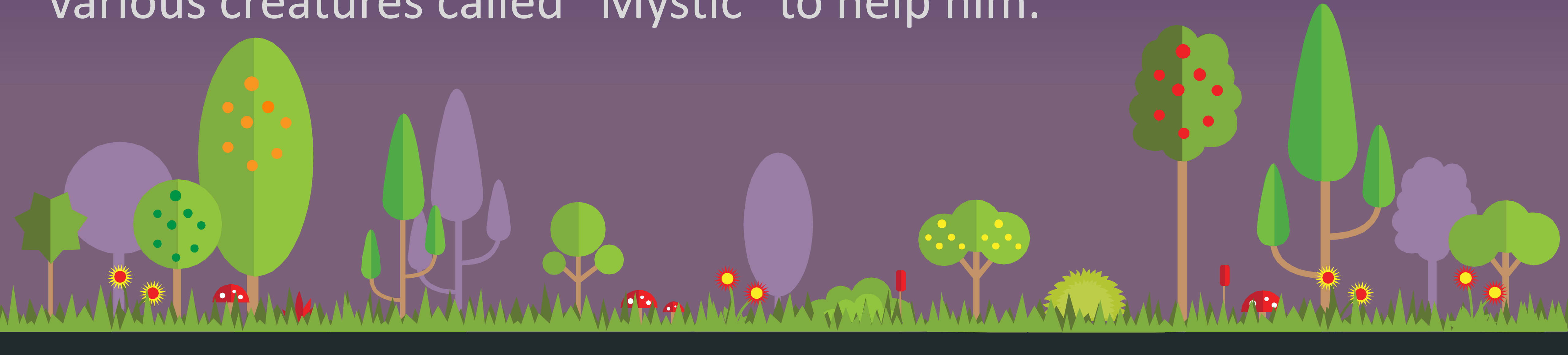
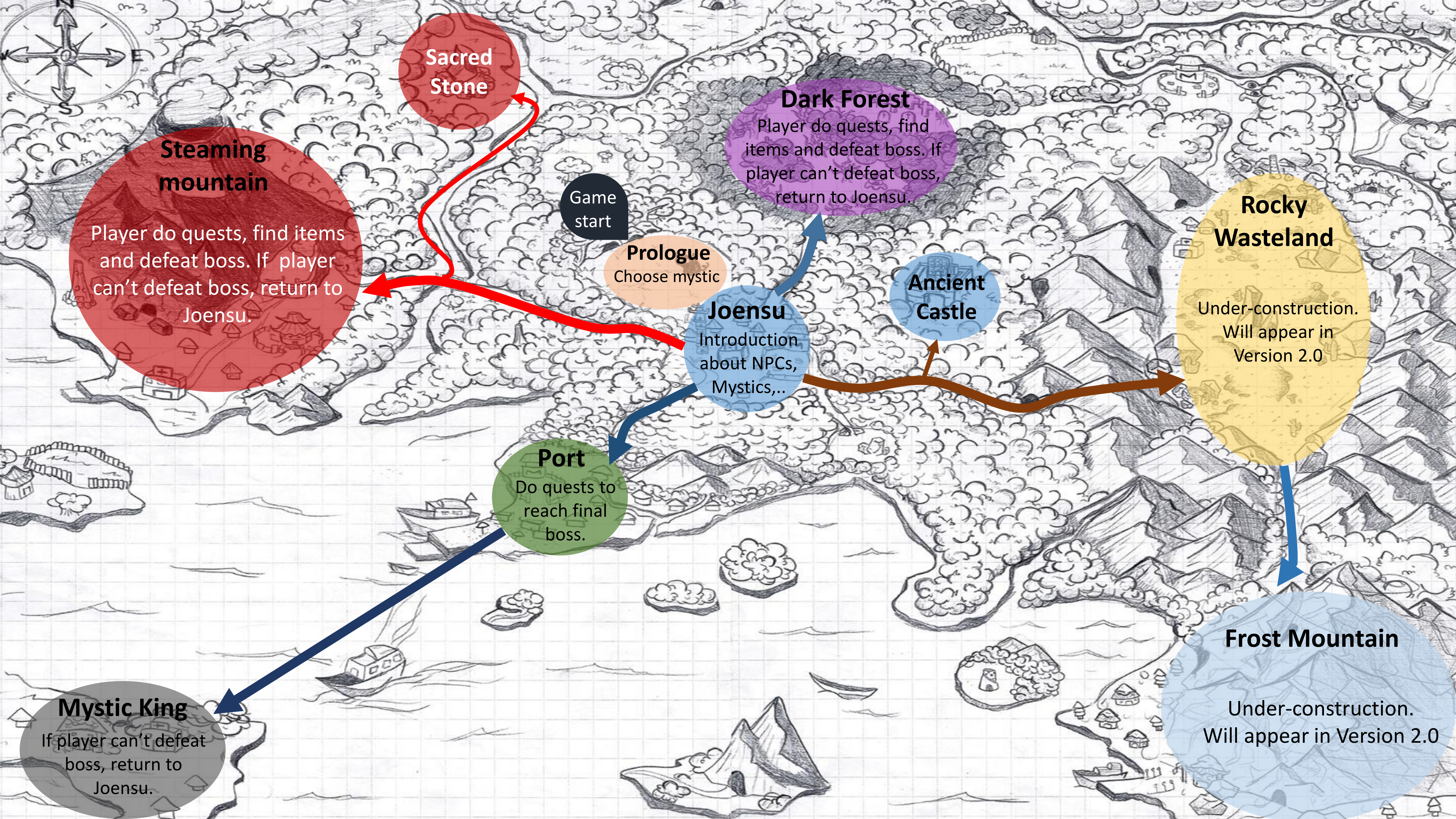
MYSTIC'S WORLD

STORY

A boy gets lost in a strange world and he has to find a way back home. Here, in this new world, he encounters a lot of dangers like monsters, thieves,... that prevent the boy from finishing his journey. However, he will make a lot of friends and various creatures called “Mystic” to help him.




N
E
S
Sacred Stone
Steaming mountain
Player do quests, find items and defeat boss. If player can't defeat boss, return to Joensu.
Game start
Prologue
Choose mystic
Dark Forest
Player do quests, find items and defeat boss. If player can't defeat boss, return to Joensu.
Joensu
Introduction about NPCs, Mystics,..
Ancient Castle
Rocky Wasteland
Under-construction. Will appear in Version 2.0
Port
Do quests to reach final boss.
Frost Mountain
Under-construction. Will appear in Version 2.0
Mystic King
If player can't defeat boss, return to Joensu.

PLAYING INSTRUCTION

TYPE

COMMANDS

CHECK LOOK NORTH
EAST GO AROUND…

TO INSTRUCT YOUR CHARACTER

PLAYING INSTRUCTION

FIGHT

MONSTERS

BY ANSWERING QUESTIONS

GAIN EXPERIENCE

PLAYING INSTRUCTION

SOLVE **PUZZLES** TO UNLOCK **LOCATIONS**

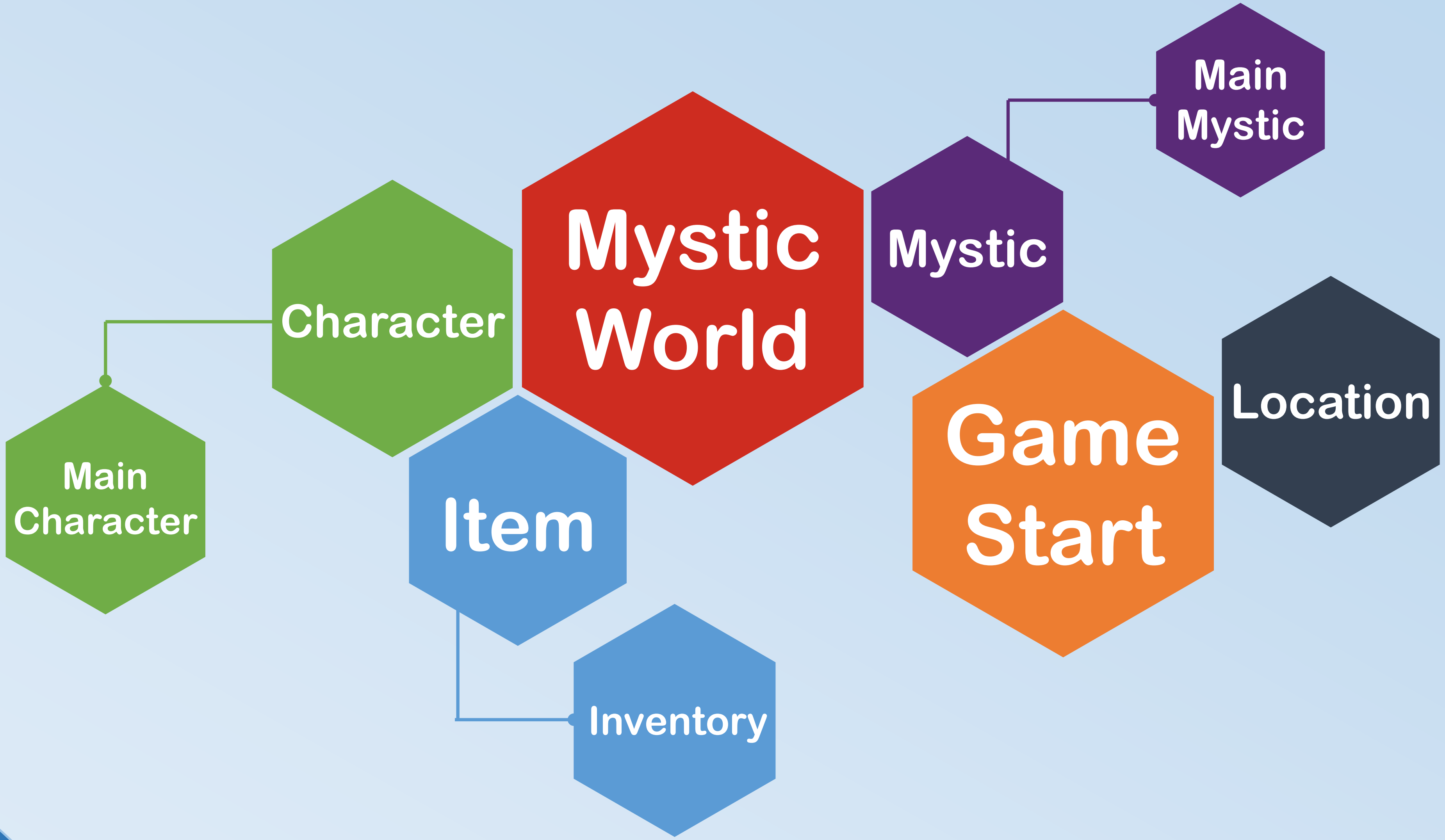
PLAYING INSTRUCTION

FIND

ITEMS

AND DEFEAT **BOSSES**

CODE


Main Mystic
Mystic
Character
Mystic World
Main Character
Item
Game Start
Location
Inventory

CODE DESCRIPTION

RANDOMNESS

Each time player encounter a monster, its name and questions are randomized from respective ArrayList.

```
public String randomMonsterName()  {
        Random random = new Random();
        int n = random.nextInt(this.listOfName.size() - 1) + 1;
        return this.listOfName.get(n);
}

public void randomQuestion()  {
        System.out.println("[Question :]");
        int n = random.nextInt(33);
        System.out.println(this.listOfQuestions[0][n]);
}
```

CODE DESCRIPTION

MAPS

Entire map is created with objects. Each location is an object with “direction” value.

```
this.HomeTown = new Location("Joensu Town");
this.HomeTown.setDirection(this.DarkForest, this.PortCity,
                                            this.RockyWL, null);
```

CODE DESCRIPTION

READ FILES

Story texts are read from external text-files in order to shorten number of code lines.

```
public long readFile (String s, String a, String b, long seek) {
        long forSeek = seek;
        try {
            RandomAccessFile file = new RandomAccessFile(s,"r");
            String read, replace = "";

            file.seek(forSeek);
            ...
        }
        return forSeek;
}
```

CODE DESCRIPTION

SAVE & LOAD

Players are able to save their current state in game and load it later on.

```
public void saveGame() {
        try {
            FileOutputStream fileoutput = new FileOutputStream("save.dat");
            ObjectOutputStream objectoutput = new ObjectOutputStream
                                                                    (fileoutput);
            objectoutput.writeObject(this);
            objectoutput.flush();
            objectoutput.close();
        } catch(IOException e) {
            e.printStackTrace();
        }
}
```

CODE DESCRIPTION

THREAD

Output codes are printed line by line, not the whole at once.

Thread.sleep(millionseconds)

THE END