

REPORT

For adventure game “Moving through space”

The problem statement

As in real life, the player goes from home to work, restaurants, shopping malls, and universities on a daily basis. The player can relax and re-energize for a new day in the mansion, but this comes at a cost. Players can earn money at the office, but it takes some effort. Also in the market, pizzeria, brasserie, shop, and pharmacy stories, the player can spend money and gain vitality. Though the player invests time and money to acquire expertise, it doubles the player's earnings at the workplace and can only be obtained once. Through correct energy and money management, these acts enable players/users to become the strongest person.[1]

Language & Tool

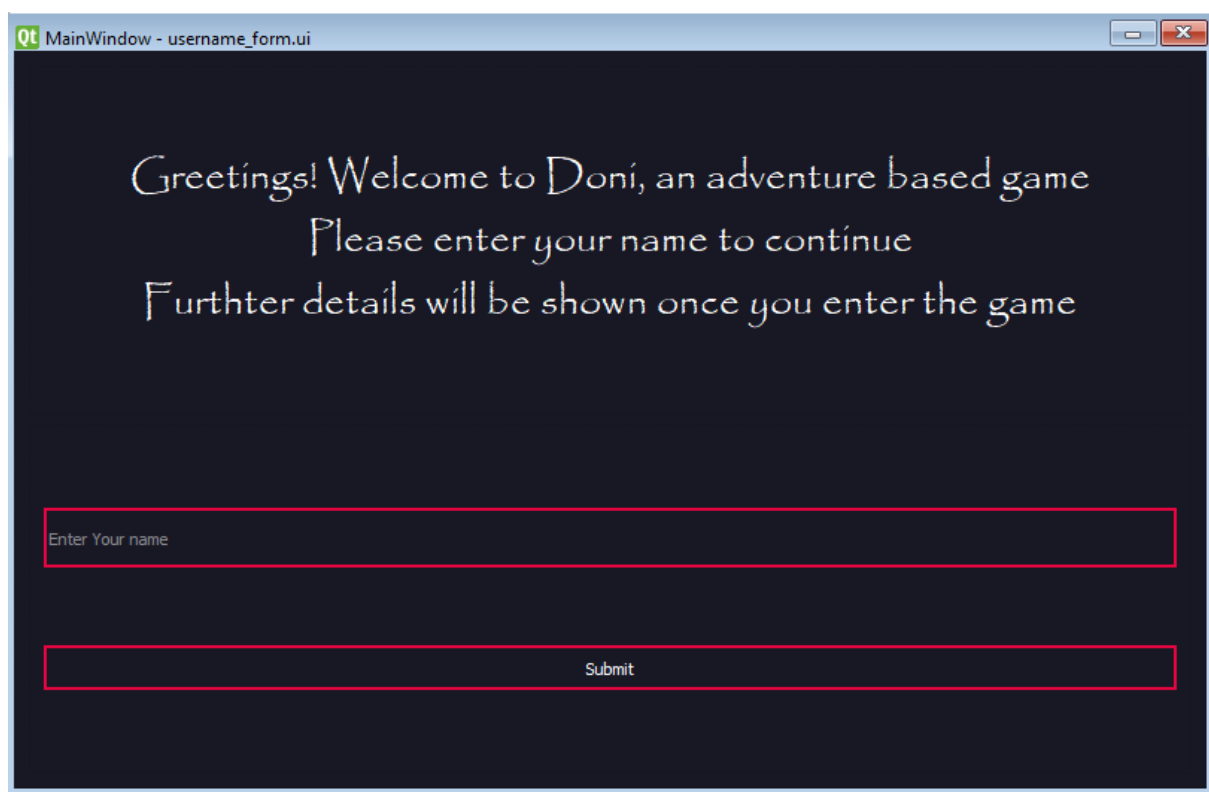
Basically the game is development in python backend and GUI both are python based. PyQt5 is based on Qt v5 and provides classes for graphical user interfaces, network communication, regular expressions, threads, SQL databases, multimedia, web surfing, and other Qt technologies. PyQt5 has a collection of Python modules that implement over a thousand of these Qt classes, all of which are housed within the PyQt5 top-level Python package. Windows, UNIX, Linux, macOS, iOS, and Android are all supported by PyQt5.

Basic Working

This game contains four forms as given below.

- Login form
- Main form
- Map form
- About form

Login form GUI



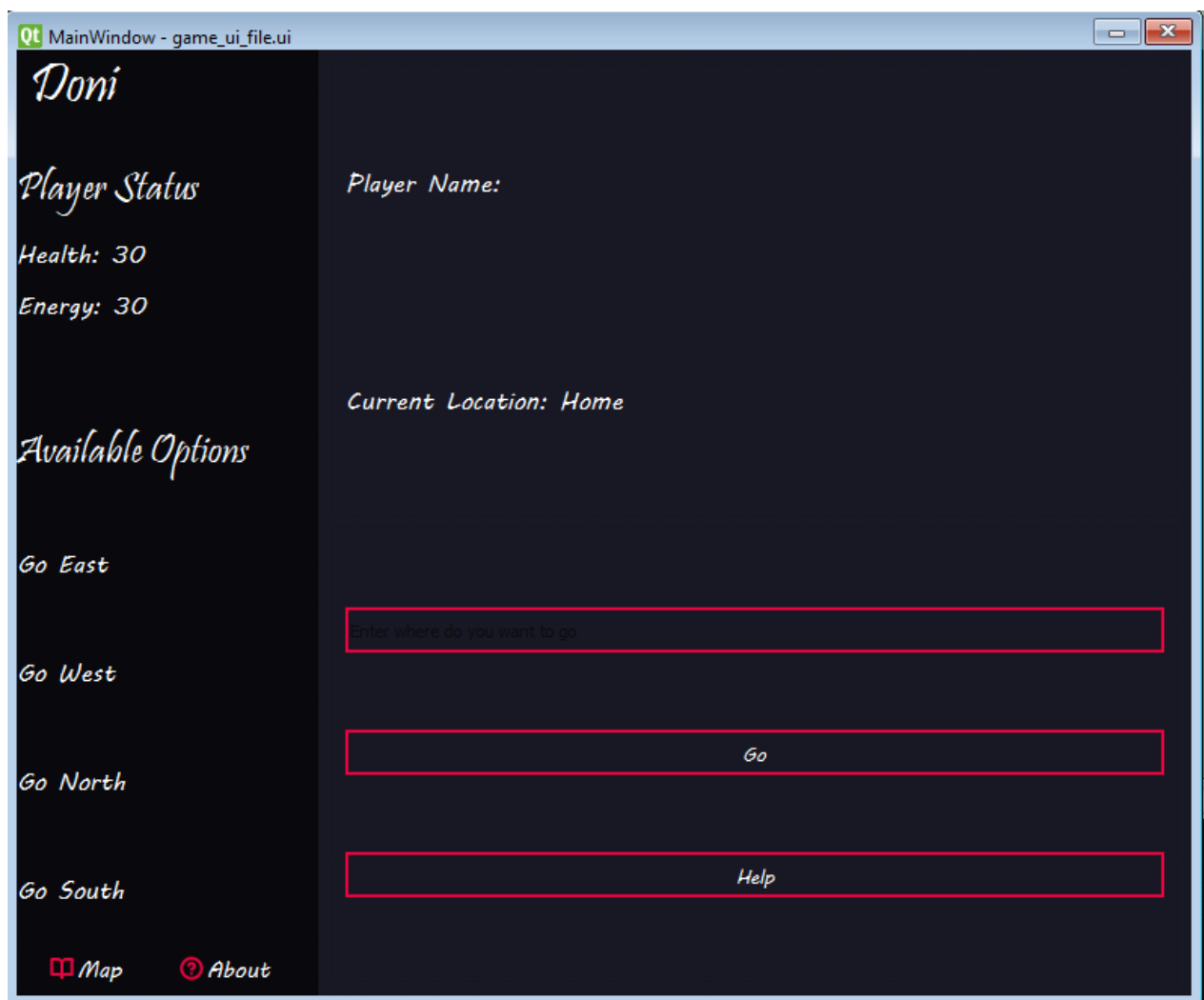
The screenshot shows a Qt window titled "MainWindow - username_form.ui". The window has a dark blue background. The text is displayed in a light green, monospaced font. The text reads: "Greetings! Welcome to Doni, an adventure based game", "Please enter your name to continue", and "Further details will be shown once you enter the game". Below the text is a text input field with the placeholder text "Enter Your name". Below the input field is a "Submit" button.

First of all after starting game the first form that is login form will appear on the screen here you have to enter the name of player and just submit the form so you can move to the main form or dashboard of the game.[2]

The user can enter their username and then click the Login button.

When you click the button, a function is executed to validate your credentials. You must write your player name to validate the username and in this function. For this example, we get the username entered by the user and print them to the console using the credential validating method.

Main form GUI

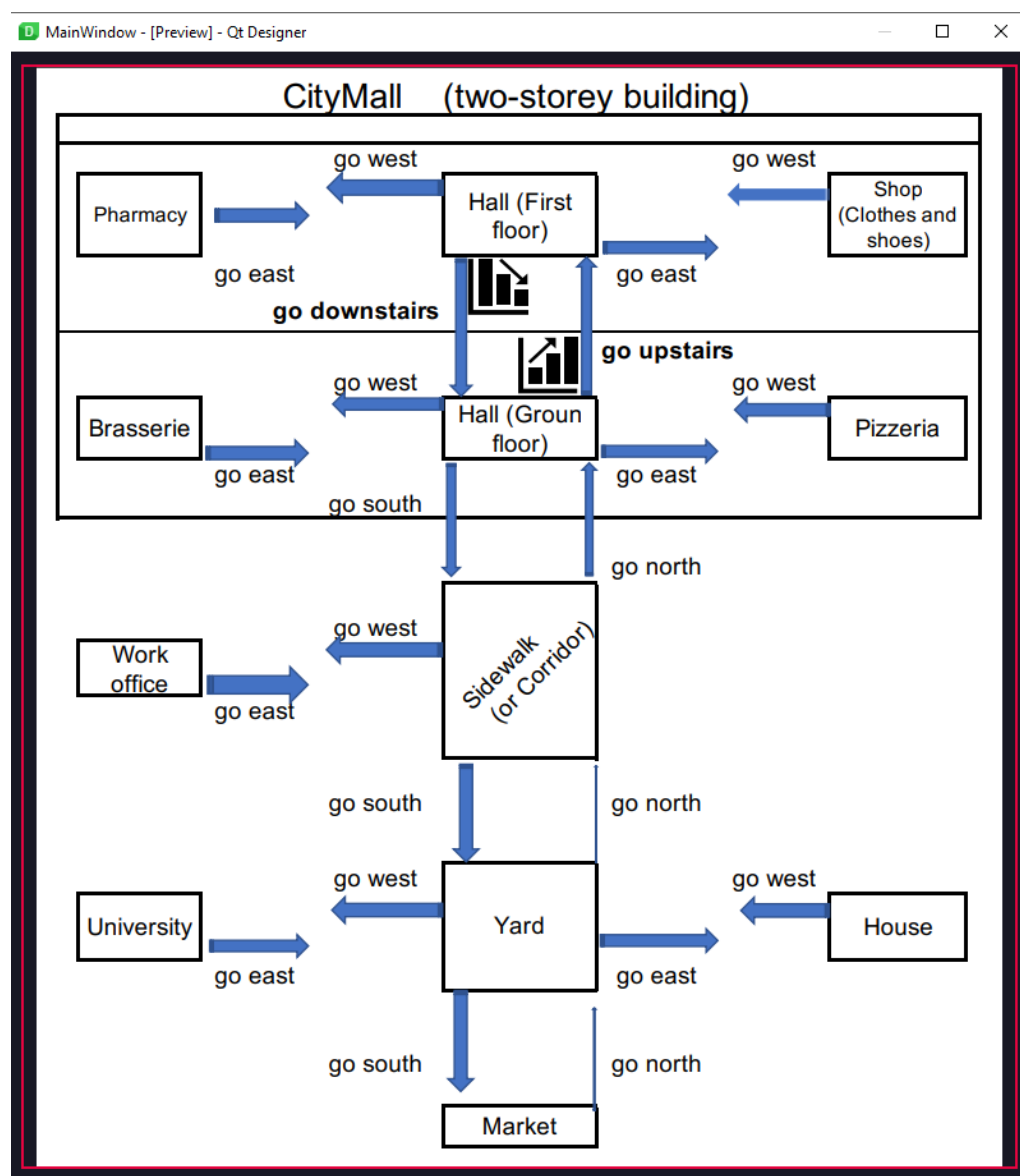


The Button widget is a common pyqt5 widget that may be used to display numerous types of buttons. A button is a widget that is intended for the user to interact with; for example, if the button is pressed with the mouse, an action may be initiated. Labels, for example, can include text and images.

A button can only display text in one typeface, whereas labels can display text in several fonts. A button's wording can be longer than one line.

Here you can perform different functions related to the game you move your player in four given directions east, west, north, south also you can monitor the current location of you player you can also check health and energy of your character.[2]

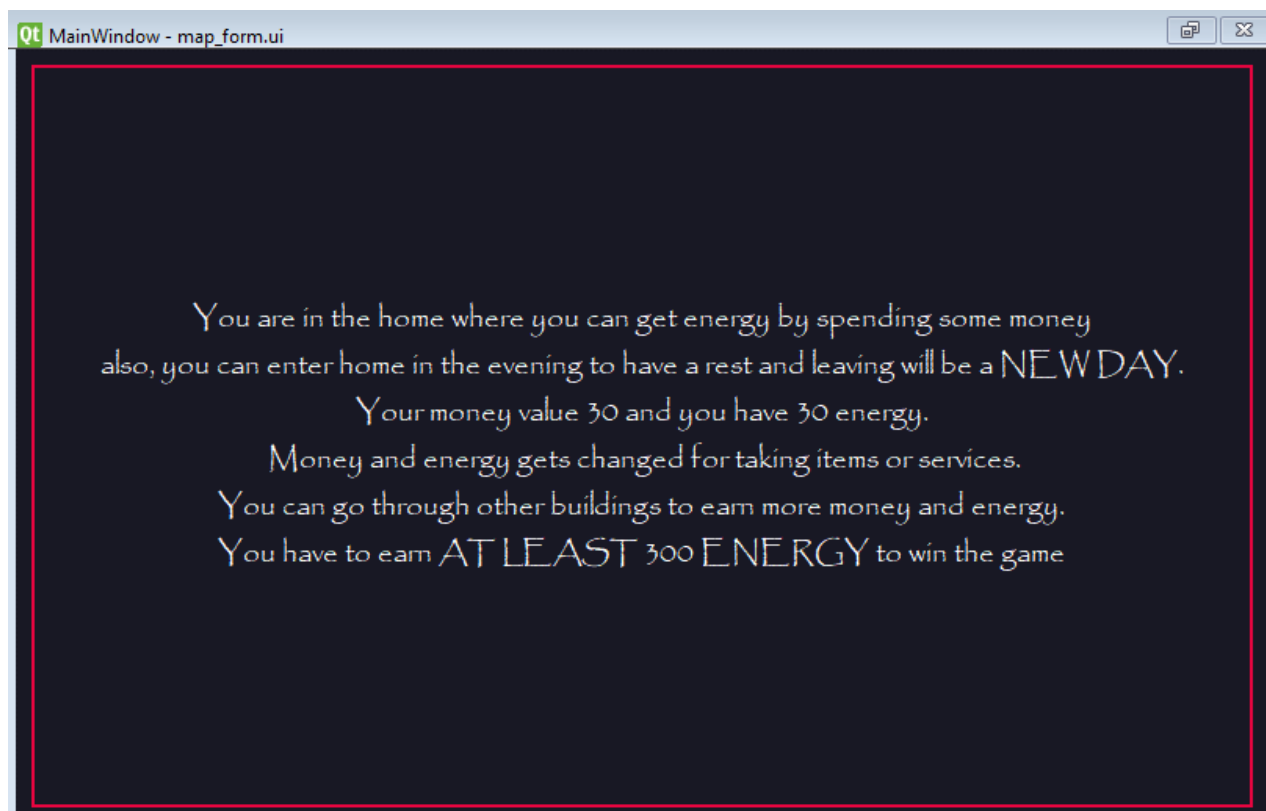
Map form GUI



In this form a game map is given through which a player can understand the location different entities like shop,mall,room etc in the game world.it is very helpful for player while playing game.

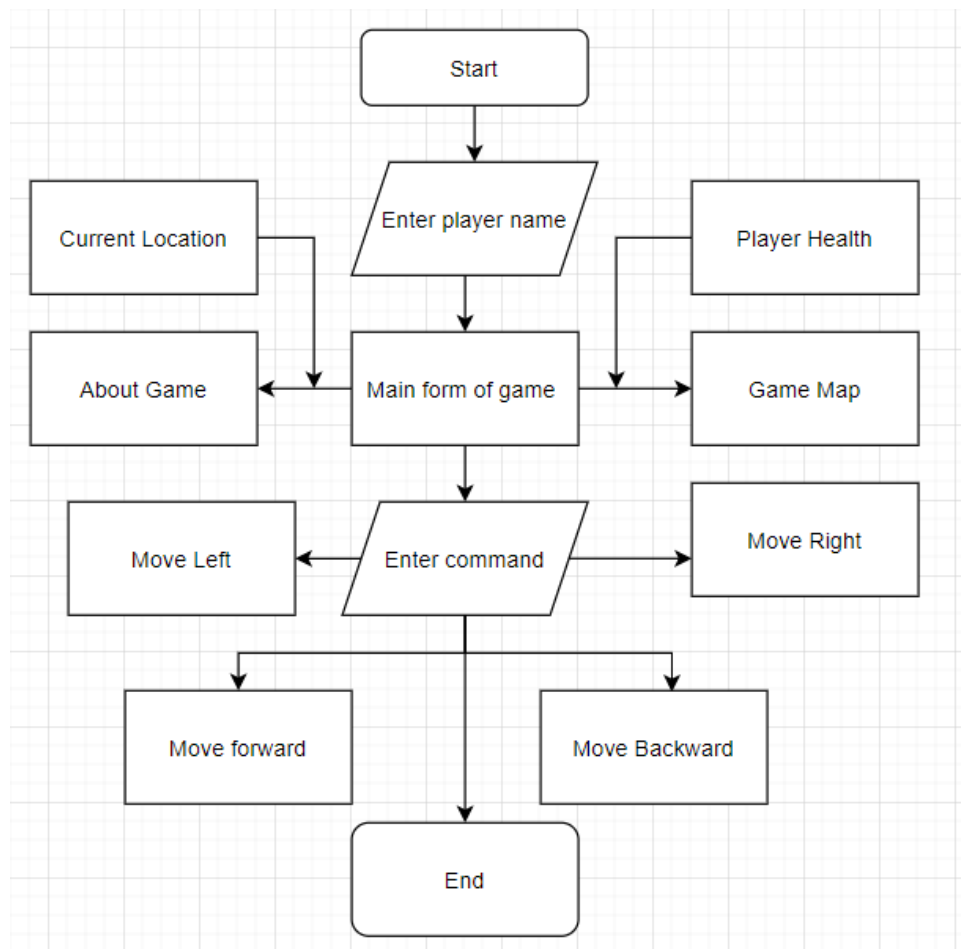
A map game is a game in which participants take turns writing down what happened within a specific time period. Essentially, map games are a user-created alternate history game in which the game's developer posts a beginning map.[3]

About form GUI



This form tells us about the game like abstract of game and basic information of the system.

Flow chart of the game

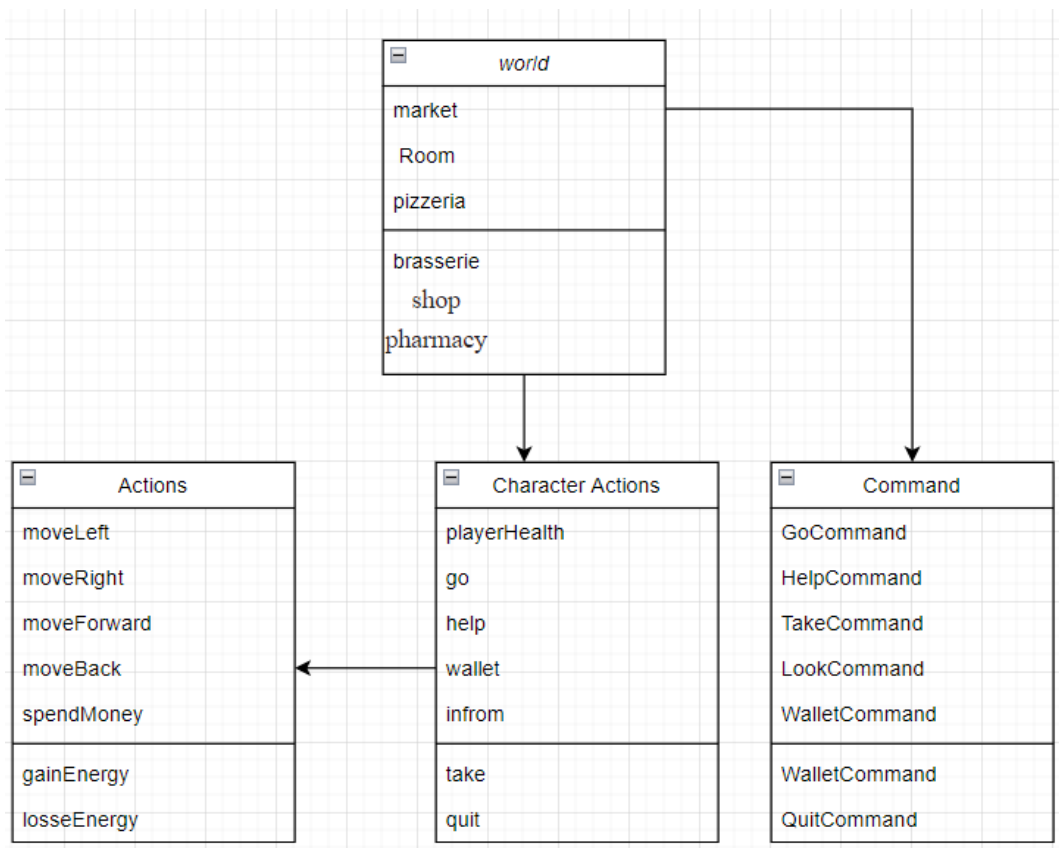


At very start of the program input box will show to input player name after doing this the control will shift to new form where you have different options, labels and buttons to control player and manage dashboard like player movement options and labels for heath and energy and current location .Game will end after the required functions will performed.

A flowchart is a graphic representation of a procedure, system, or computer algorithm. They're frequently utilised in a variety of fields to document, analyse, plan, enhance, and convey often complex processes in clear, simple diagrams. Flowcharts, also known as flow charts, are diagrams that use rectangles, ovals, diamonds, and maybe other forms to indicate the type of step, as well as linking arrows to indicate flow and sequence. They can be as simple as hand-drawn

diagrams or as complex as computer-drawn diagrams illustrating several steps and paths.[4]

Class diagram



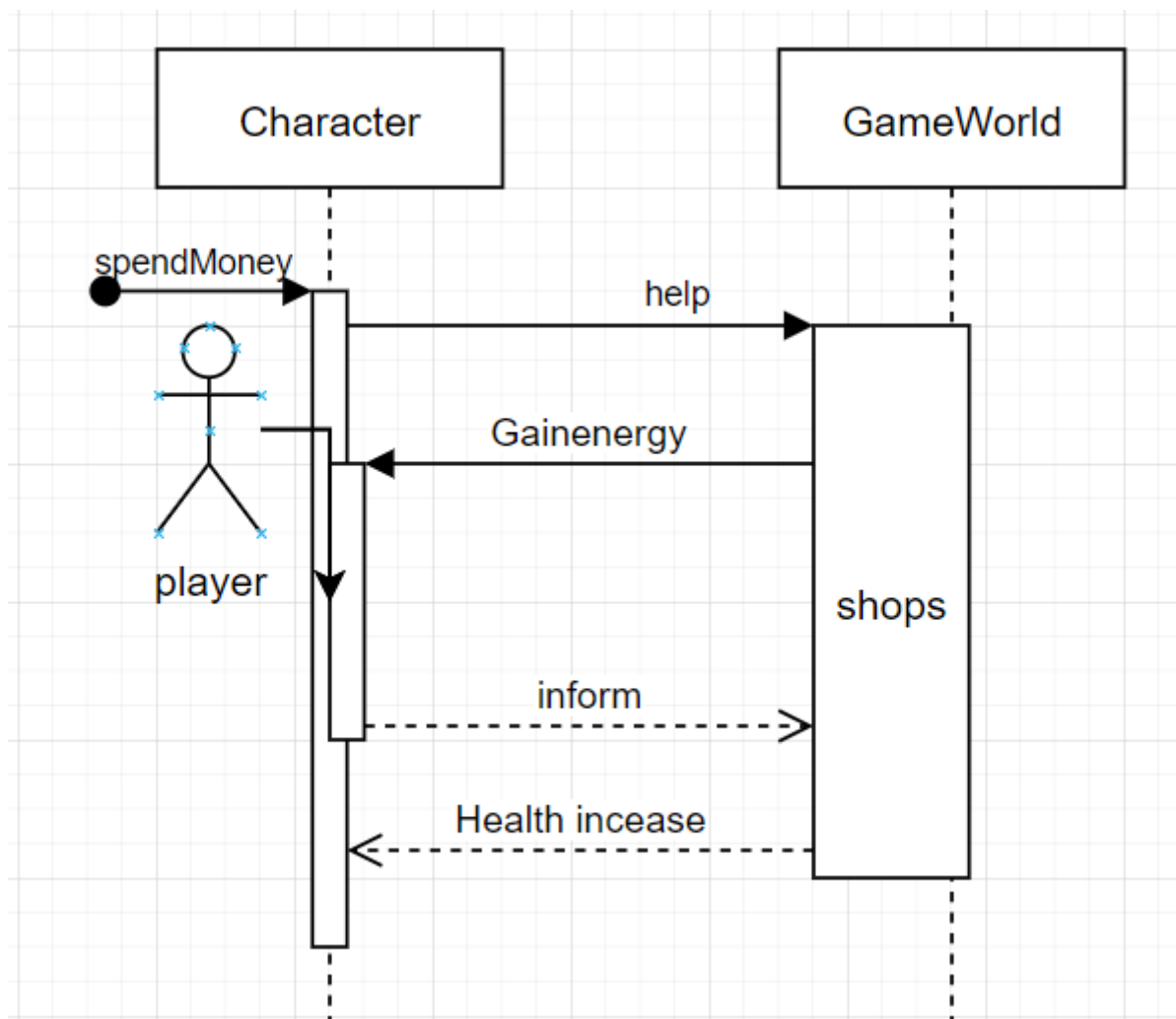
A static diagram is a class diagram. It depicts an application's static view. A class diagram is used not only for visualising, describing, and documenting many parts of a system, but also for creating executable code for a software programme.

A class diagram depicts a class's attributes and operations, as well as the system's limitations. Because class diagrams are the only UML diagrams that can be directly mapped with object-oriented languages, they are frequently utilised in the modelling of object-oriented systems.

There are three basic classes which defines the whole working of functions and the relations with each other and subclasses .Name of classes are given below.[5]

- World
- Character Actions
- Command
- Actions (subclass of Character Actions)

Sequence Diagram



Here the player is interacting with the game world like he is asking for help that means he want energy he send some money to buy food and after taking meal the health of the player will increase and the entities of game world will inform the player that its health is increased.[6]

1. Joselli, M., et al. *An architecture for game interaction using mobile*. in *2012 IEEE International Games Innovation Conference*. 2012. IEEE.
2. Stuckner, J., et al., *AQUAMI: An open source Python package and GUI for the automatic quantitative analysis of morphologically complex multiphase materials*. 2017. **139**: p. 320-329.
3. Sun, C., et al. *Facilitating monkey test by detecting operable regions in rendered gui of mobile game apps*. in *2016 IEEE International Conference on Software Quality, Reliability and Security (QRS)*. 2016. IEEE.
4. Duke, R.D.J.S. and games, *A paradigm for game design*. 1980. **11**(3): p. 364-377.
5. Jonsson, J., *Game Development: Using UML Class Diagram*. 2013.
6. Greenyer, J. and J. Rieke. *Applying advanced TGG concepts for a complex transformation of sequence diagram specifications to timed game automata*. in *International Symposium on Applications of Graph Transformations with Industrial Relevance*. 2011. Springer.