

# DESIGN PATTERNS ASSIGNMENT

AZIZ ALEXANDROS TAREK

Student id: @00571858

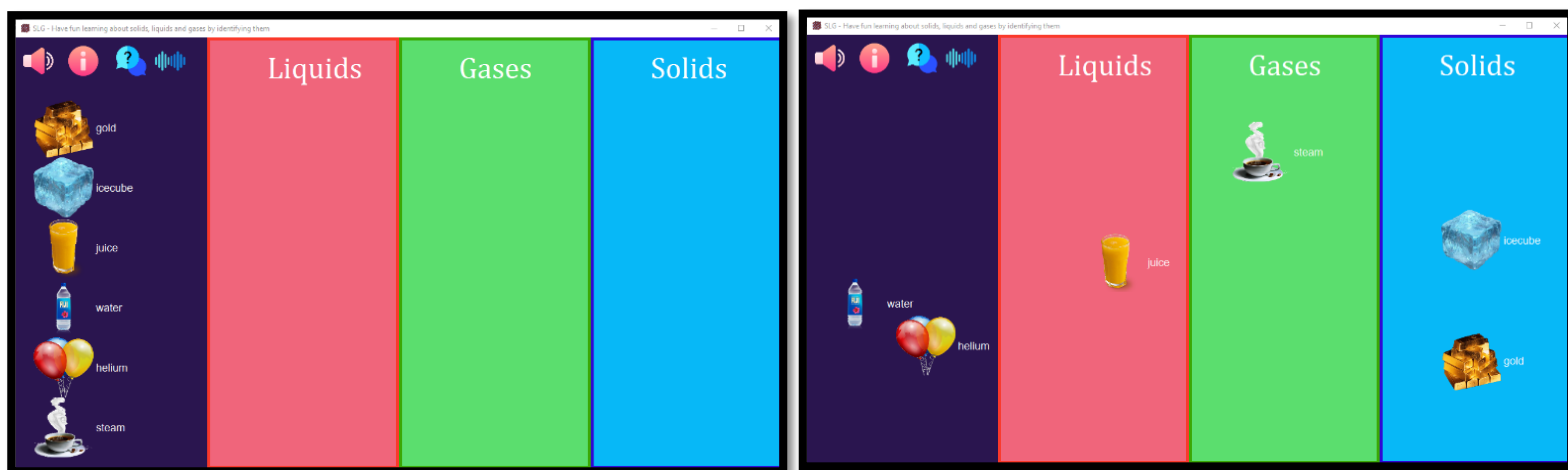
---

## CONTENTS

Page 2	App introduction and software design
Page 3	Application of design patterns & Key code elements
Page 4	Key code elements cont. & UML Diagrams
Page 5	UML Diagrams cont.
Page 6	Use case diagram & critique and evaluation of my software

## An introduction to the app and software design

The science that i chose for this assignment was chemistry, more specifically i chose to teach about the states of matter which matches the level of education of most students that will be playing this game. The application/game that i created as the solution is called SLG (the acronyms for solids ,liquids and gases plasma the fourth state of matter will not be included to avoid complexity) and teaches students to identify the state of matter of an object by dragging it to the correct pane. When an object is moved to any of the three panes a sound effect will be played to show that the player has correctly identified the object's state, otherwise a different sound effect will be played to indicate wrong identification. The design is simple and considerably colorful, it consists of four panes, one pane that has all the objects that need to be identified such as gold, water, juice etc. as well as four control buttons at the top of the pane and three panes each for a state of matter under which the respective objects must be placed. The four control buttons serve as another way of interaction with the game other than dragging objects, the first button at the top left corner of the game mutes or unmutes the background music, the second button displays information about the states of matter, the third button shows the controls of the game(keys than can trigger an action like undo/redo and more) and the fourth button is for enabling or disabling the sound effects. The sound effects are a source of feedback on whether each object has been placed on the correct pane or not and should be fairly recognizable by kids, the background music is a jazz music sample that has been placed to set a relaxed atmosphere and to make the game more compelling.



Screenshots of the game's user interface.

## Application of design patterns

The design patterns that were used for the development of this game are the factory and command pattern.

The factory completely abstracts the client's creation and initialization of the product. This indirection allows the client to concentrate on its specific role in the application without having to worry about how the product is made[1]. As a result, even as the product's implementation evolves over time, the client remains constant and so is the ideal class for the creation of numerous specific `GameObject` instances. In this game, the factory class creates instances of `Solid`, `Liquid` and `Gas` classes by recognizing the type that was given in the constructor.

The Command class model separates application behavior from GUI objects by incorporating application behavior in subclasses of the `Command` class[2]. For instance, in this game when the application desires an object to be dragged it calls the `DragCommand` object to accomplish this action. The command pattern also allows players to undo/redo their dragging by using the keys "Z" or "Y". Players can continuously use either of these commands to undo or redo the position of an object after it was dragged as the logs of the `DragCommand` object are stored in an `ArrayList` and is the reason why it is useful as players might make a mistake and want to undo or they might want to see what lead to the mistake by redoing. The command pattern consists of the `Command Interface` (`CommandIF.java`), the `CommandPattern` class (the invoker), the `DragCommand` class that is the concrete implementation of the `Command Interface`, `GameObject` that is the receiver, and the `Client SlgApp`.

References:

[1] Purdy D. (2002). Exploring the Factory Design Pattern. p.2.

[2] Newmarch J. (2007). The Command class for the Java AWT. p.4.

## Key elements of the game's code

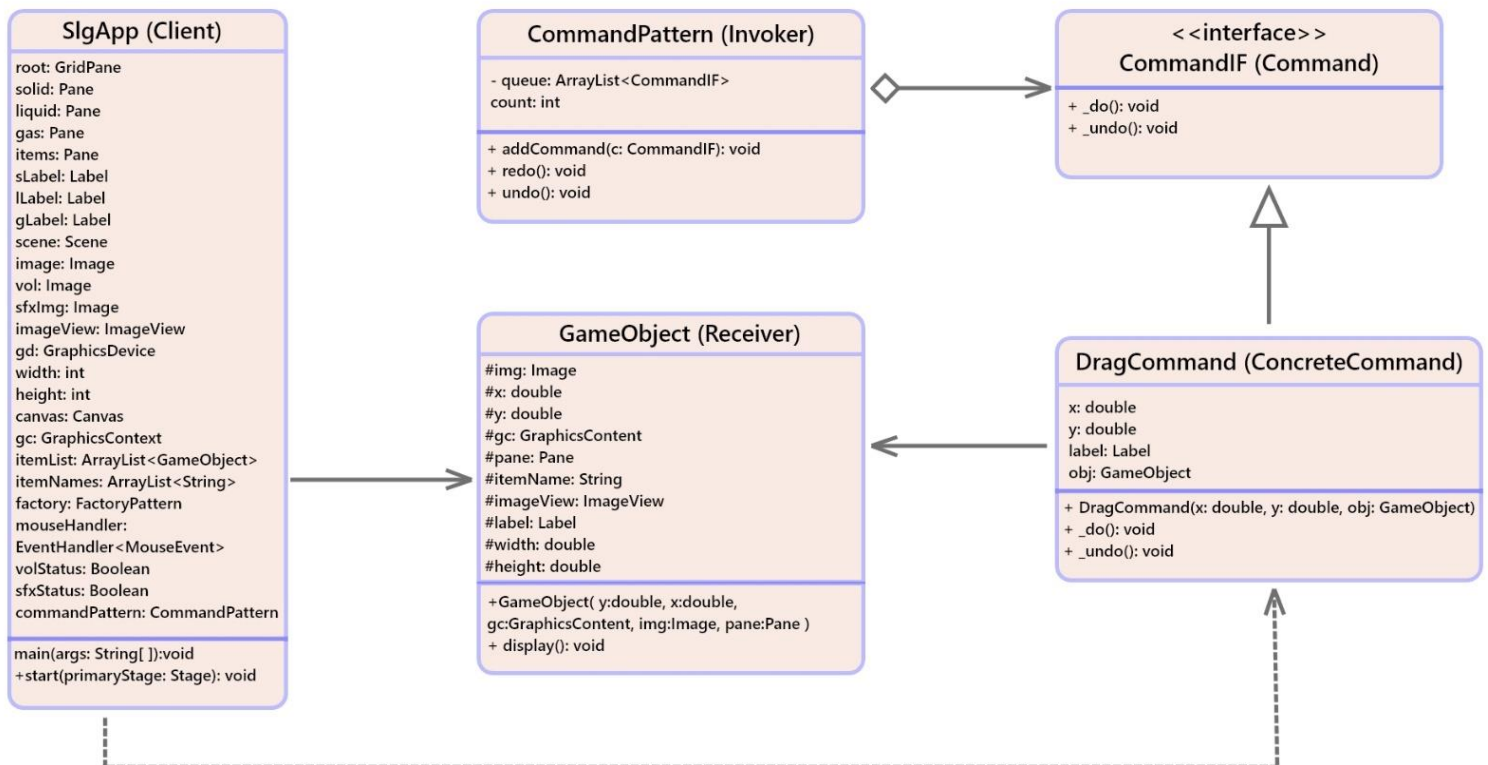
- `GridPane`: After experimenting with some built-in layout panes like the simple `Pane`, `AnchorPane`, `FlowPane`, and `GridPane` i found that the `GridPane` works best for the design of this game because it has the ability to create columns and rows as well as the ability to set the percentage of width or height each will take.
- Use of the `EventHandler` class:
  - `MouseEvent`: When the mouse interacts with whatever is on the screen like buttons, `GameObject` instances or panes, the `MouseEvent` object will check if the mouse is intersecting with one of the `GameObject` instances and only then it will allow them

to be dragged using the mouse. In addition, there is another MouseEvent instance that is triggered when an object is dragged into one of the other three panes, when the object intersects with the correct pane a sound effect will be played, and likewise for when it intersects with the wrong pane.

- Key Event: Players can press a selection of keys that will trigger an event like turning on or off the music or the sound effects, they can also undo or redo their actions .
- Use of the media class:
  - MediaPlayer: It takes in a media file, an audio file in this case and auto plays it continuously after it ends.
  - MediaView: The MediaPlayer object is imported into it , and then the MediaView object can be imported into the GridPane.
- DialogPane: It is a great way of showing information only when needed without taking that much space from the panes, a DialogPane instance can be triggered when a button is clicked which in turn will show information or instructions on how to play the game.

## UML Diagrams

Diagram of the CommandPattern.



## Diagram of the FactoryPattern (Abstract Factory).

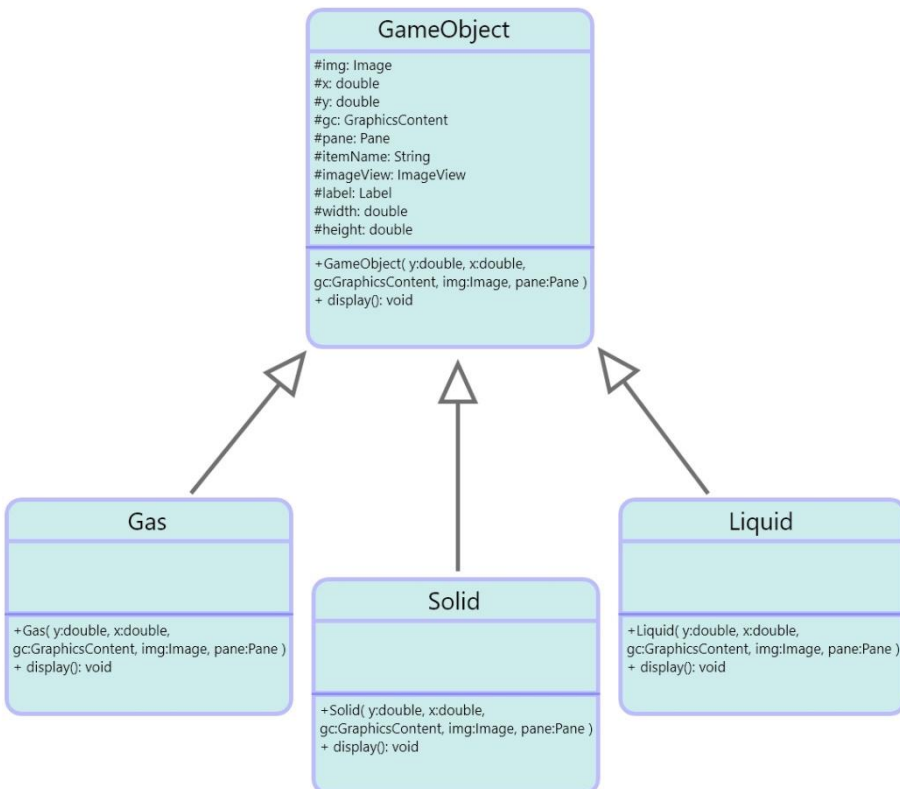
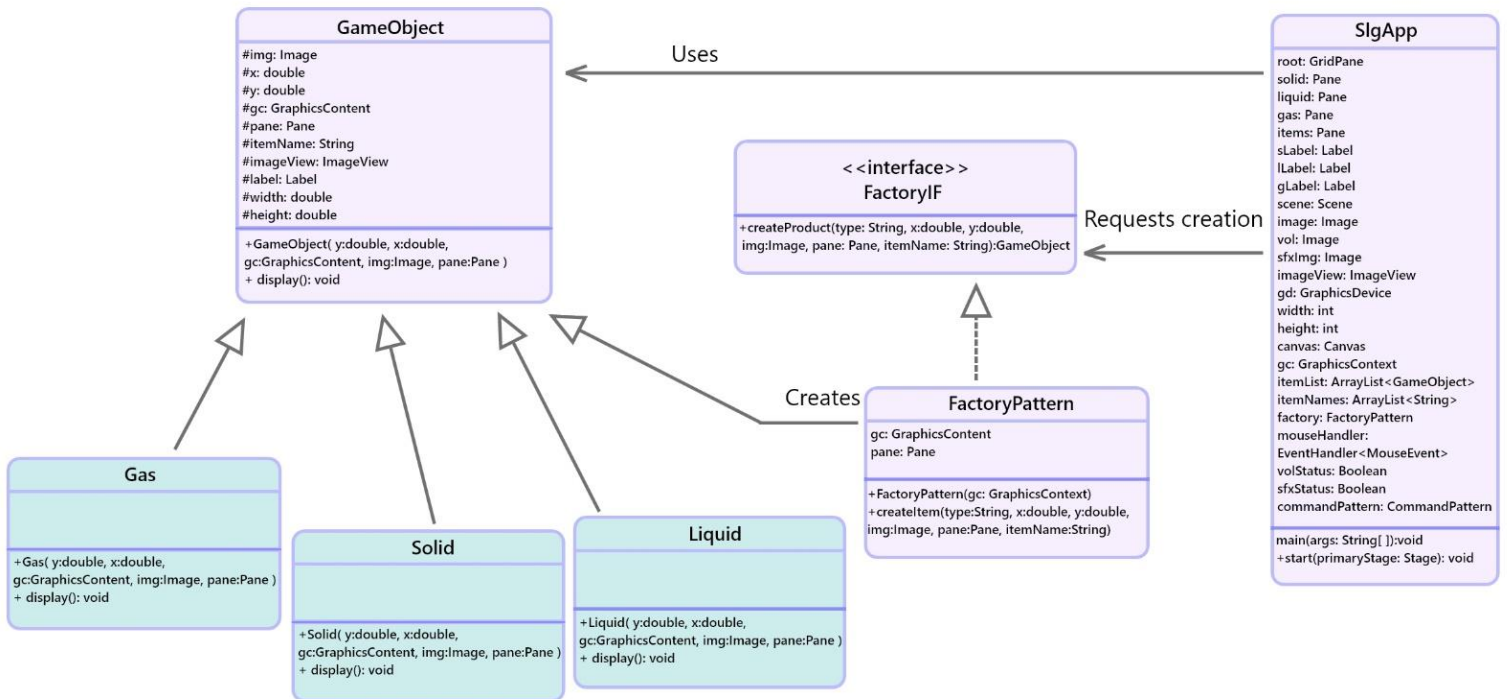
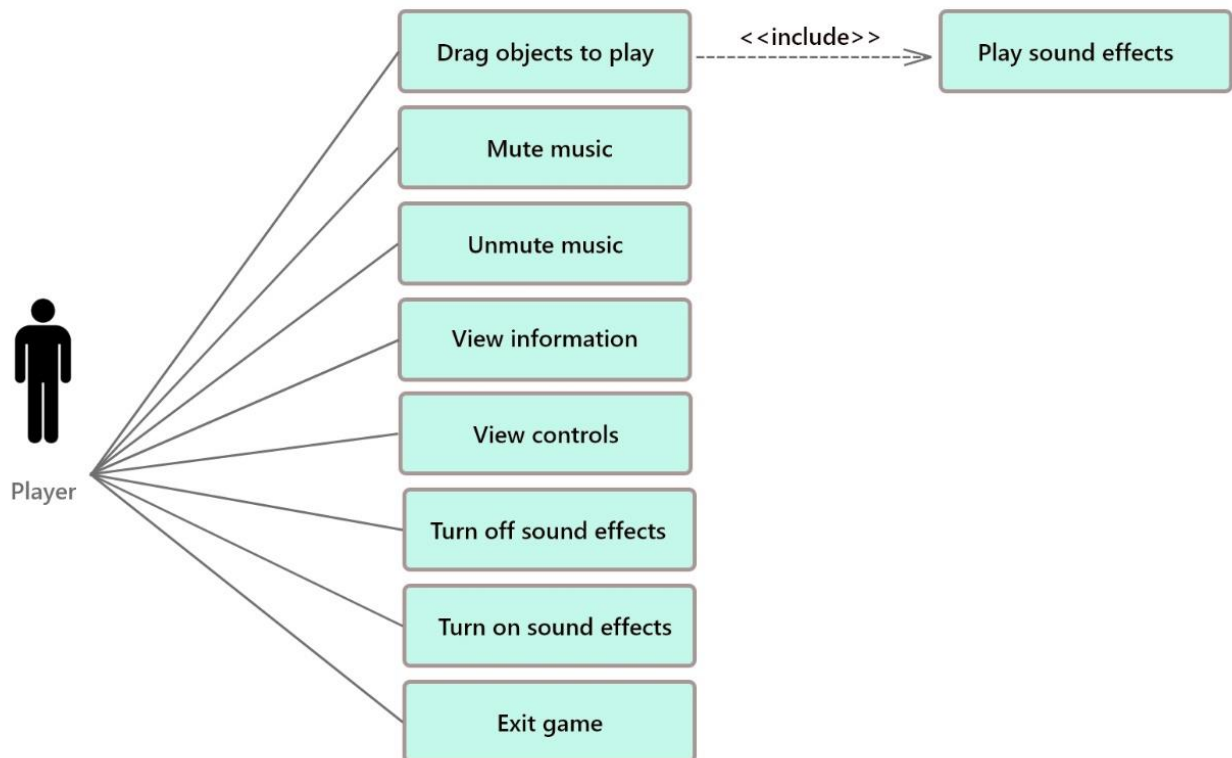


Diagram of the GameObject(SuperClass) and its subclasses.

## Use case diagram



## Critique and Evaluation of my software

The game has been made to teach an aspect of science and it aims on looking engaging to young audience, as a variety of colors have been used to make it appealing, as well as some relaxing music plays in the background while playing. The directions to play this game are quite easy to understand, although they can only be found after the clicking its respective button, although some information could have been showed at all times to guide the players. Since, the target audience is very young children it is important to give feedback in an obvious way when teaching , and so sound effects are a good way of doing that , but visual effects could also be useful if some children have a certain disability or preference. The rate of information that is being displayed is minimal and does not overwhelm the players. The skill level to play this game is low therefore no prior gaming experience is needed. It is easy to navigate through it only requires a click to toggle the audio or to view information. To sum up, the accessibility factor might be lacking a bit as the game does not provide any visual feedback for when the answer is wrong or right, the usability could also be better as the dragging speed does not always match the pointer speed , the user experience is expected to be moderate to high depending on whether they liked the science that was chosen and the game as a whole. My personal overall marking would be 7.5/10.