



Packman with OOP

Console base game project



Learning Objective

Develop a console game using object oriented programming.



Let's start the pacman game with oop



How we can represent this grid in terms of OOP?

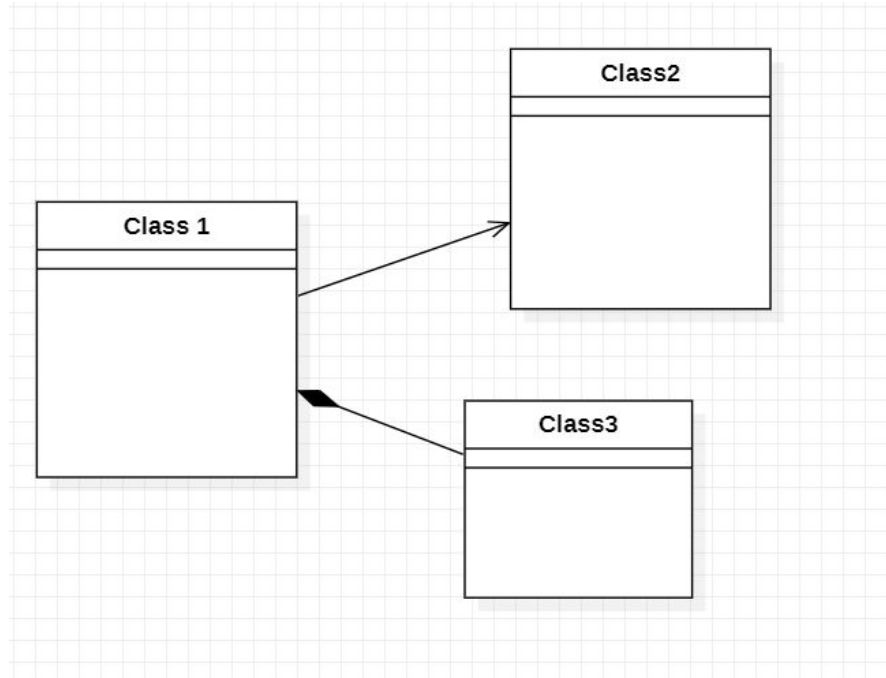


Requirements of the Pacman Game

Identify the classes for these instructions????

- Pacman game is a grid with different components.
- A grid consist of different cells.
- Each cells can contains Players, Enemies, Walls, Rewards (Bullets) or some time may be nothing.
- Also Cell display different static game objects such as Walls.
- You can think cell is smallest unit of a grid that player (Pacman) and enemies can use to navigate in any allowed direction.
- A Cell should contain the information about its adjacent cells.
- A Pacman is player who can move with the keys.

Draw the domain model diagram of your identified classes for pacman



| Let first creates set of classes that
can represent the grid, cells and
object within game



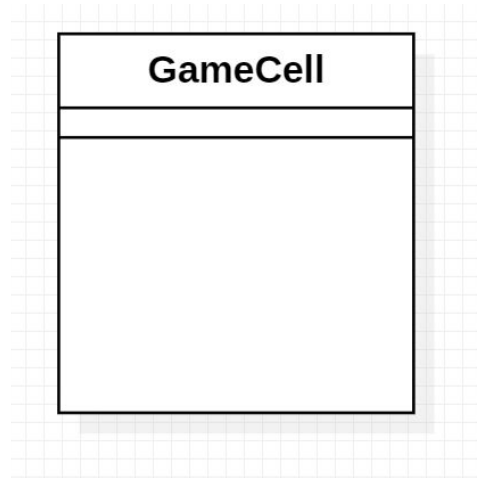
Representation of the Environment

Let start with an atomic class and that is Cell

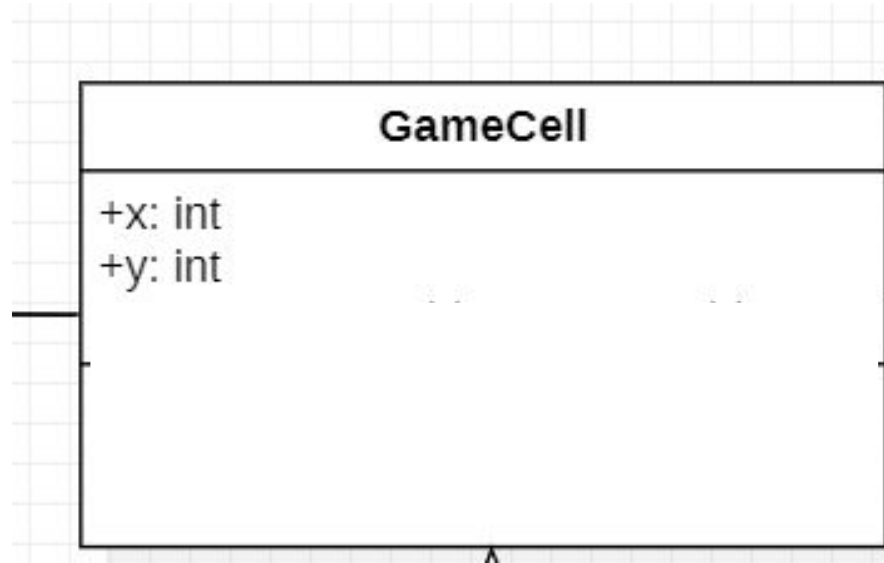
- *GameCell*

Representation of the Environment

Try to identify the attributes and functions of the class

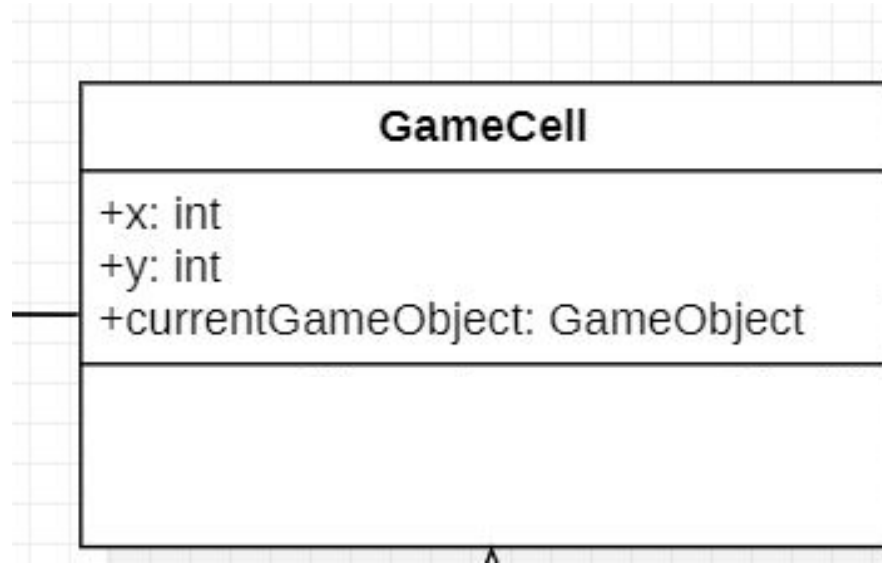


Representation of the Environment



Do Cell need to have any other information ?

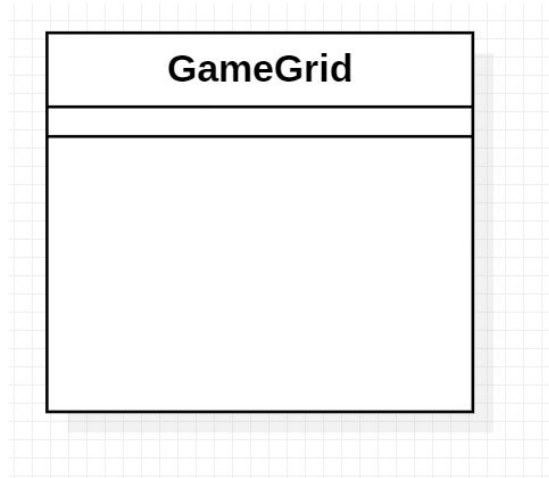
Representation of the Environment



Store the information of the *GameObject* that it is holding.

Representation of the Environment

Now, what shall be the attributes of *Grid Class*; the class that will hold all the cells



Solution of the pacman with OOP

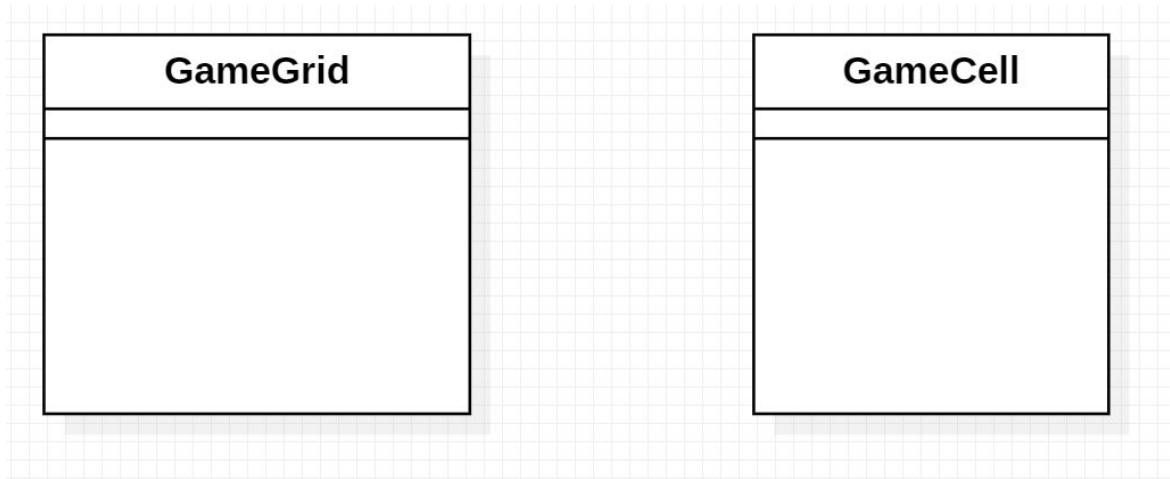
GameGrid
+GameCells[][]: GameCell +Rows: int +Cols: int
+GameGrid(string filename, int rows, int cols) -loadGrid(string filename) +getCell(int x, int y): GameCell

GameGrid contains the 2D Array of **GameCells** with two numbers of rows and columns,

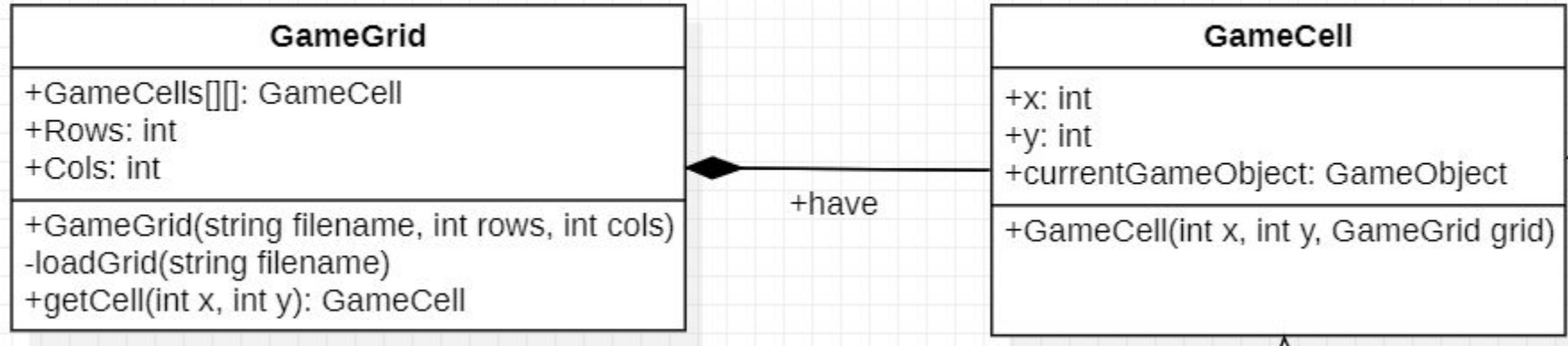
A **Constructor** to initialize the variables
a function to get any **GameCell**

Representation of the Environment

Relationship among these two classes ?



Solution of the pacman with OOP



Suppose game cell contains the **GameObject** reference and the X and Y variable that represents the Current **GameCell** position and a constructor to initialize the variables.



Question?

If we want to know from cell what are the adjacent cells of the current cell? How we can do that? What will be the function and which class will contain it?



|| Solution



There are two solutions for that.

Write that function inside the `GameGrid` class. If we do that we have to pass the cell to the `GameGrid`.

Other solution is that

We write the function inside the `GameCell`... In this case it need the `GameGrid` object.

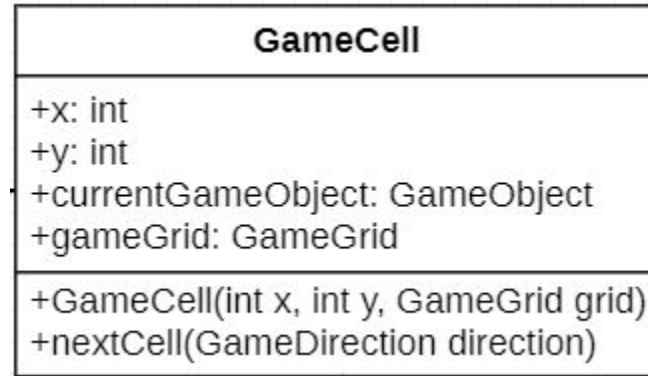
If we see the second technique



If we write the function inside the `GameCell`... In this case it need the `GameGrid` object.

In this the `GameGrid` is composing the `GameCell` and `GameCell` is composing `GameGrid`.

Is that is ok?? Or its circular relationship between the classes?



Important Instructions

- If a class contains the object of other class it does not mean its composition or aggregation everytime.
- Sometimes this is called message passing or association.
- Its is the same concepts but semantically it is different.

Important Instructions

Remember: GameGrid is containing the GameCell object this relationship is composition but GameCell contains the object of GameGrid its not the composition nor aggregation but general simple association.

We represent the association with simple line.

nextCell function with better approach

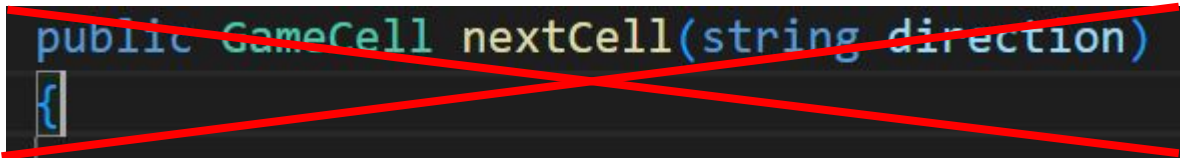
```
public GameCell nextCell(string direction)
{
```

Here is cell direction is string type which will accept **right, left, up or down** as argument.

But we are developing Framework

When we are developing framework it mean someone else like other developers will use our code to develop their own games.

In that case if someone pass the **left, right, up or down** in different format or there is spelling mistake it will not work.



```
public GameCell nextCell(string direction)
{
```

Enumeration

```
public GameCell nextCell(string direction)  
{
```

To avoid the following issue we use enumeration in C#

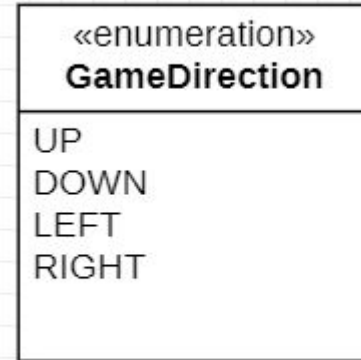
Enumeration is static class provided by the C#
that we use to avoid such issues

What is **enum** in C#?

In C#, an enum (**enumeration**) is a distinct type that represents a set of named constants. It allows you to define a list of named values, making your code more readable and self-explanatory. Enums provide a way to group related constants together and use them in your code.

Enumeration

We represent enumeration in domain model with following format



Enumeration

Just like class we make a enum and define our named constants.

```
namespace PacMan.GameGL
{
    11 references
    enum GameDirection
    {
        2 references
        Left,
        2 references
        Right,
        2 references
        Up,
        2 references
        Down
    }
}
```

Enumeration

Before

```
public GameCell nextCell(string direction)
{
```

```
nextCell("up");
```

After

2 references

```
public GameCell nextCell(GameDirection direction)
{
```

```
nextCell(GameDirection.Up);
```

The enum technique is more meaningful and error prone.

Enumeration

Before

```
public GameCell nextCell(string direction)
{
```

```
nextCell("up");
```

After

2 references

```
public GameCell nextCell(GameDirection direction)
{
```

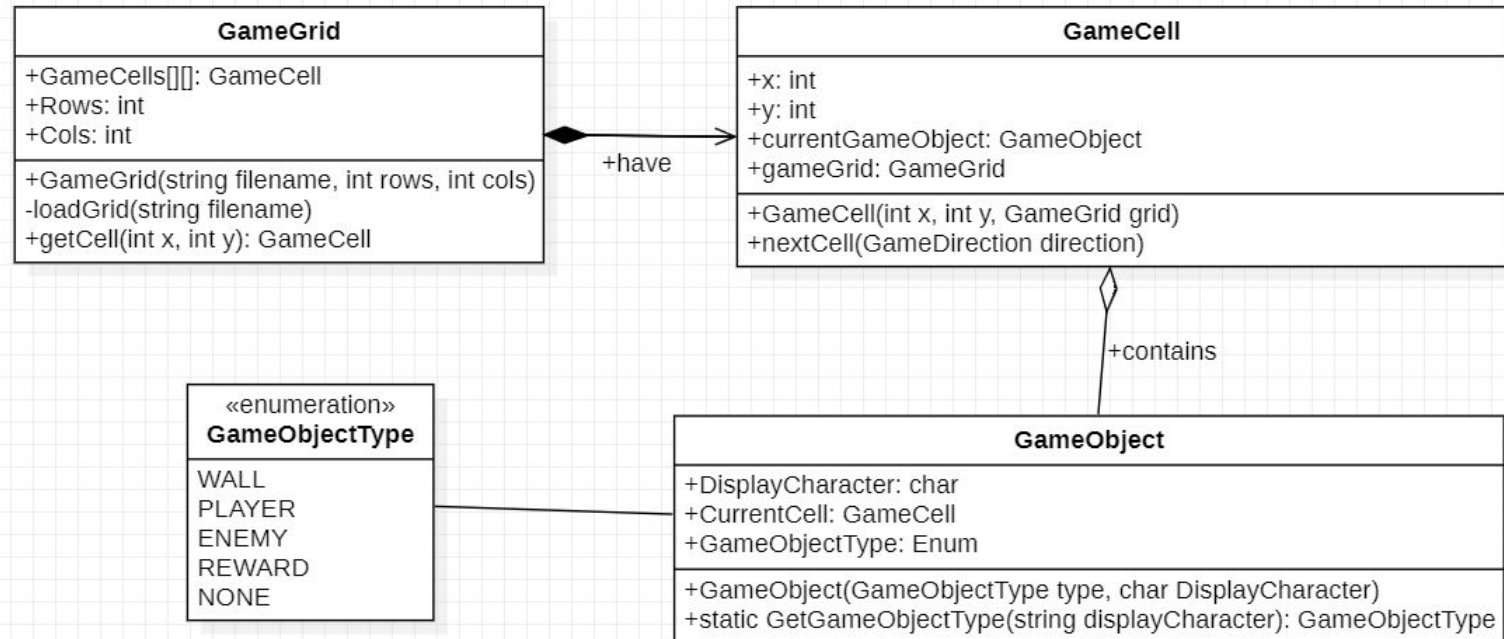
```
nextCell(GameDirection.Up);
```

The enum technique is more meaningful and error prone.

GameObject

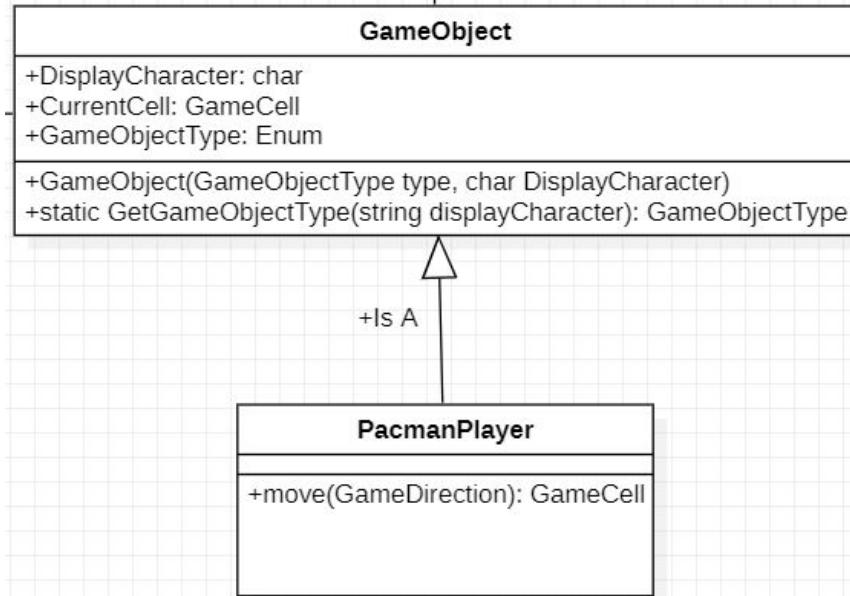
GameObject will be the made of following attributes

Here the **GameObjectType** is enumeration

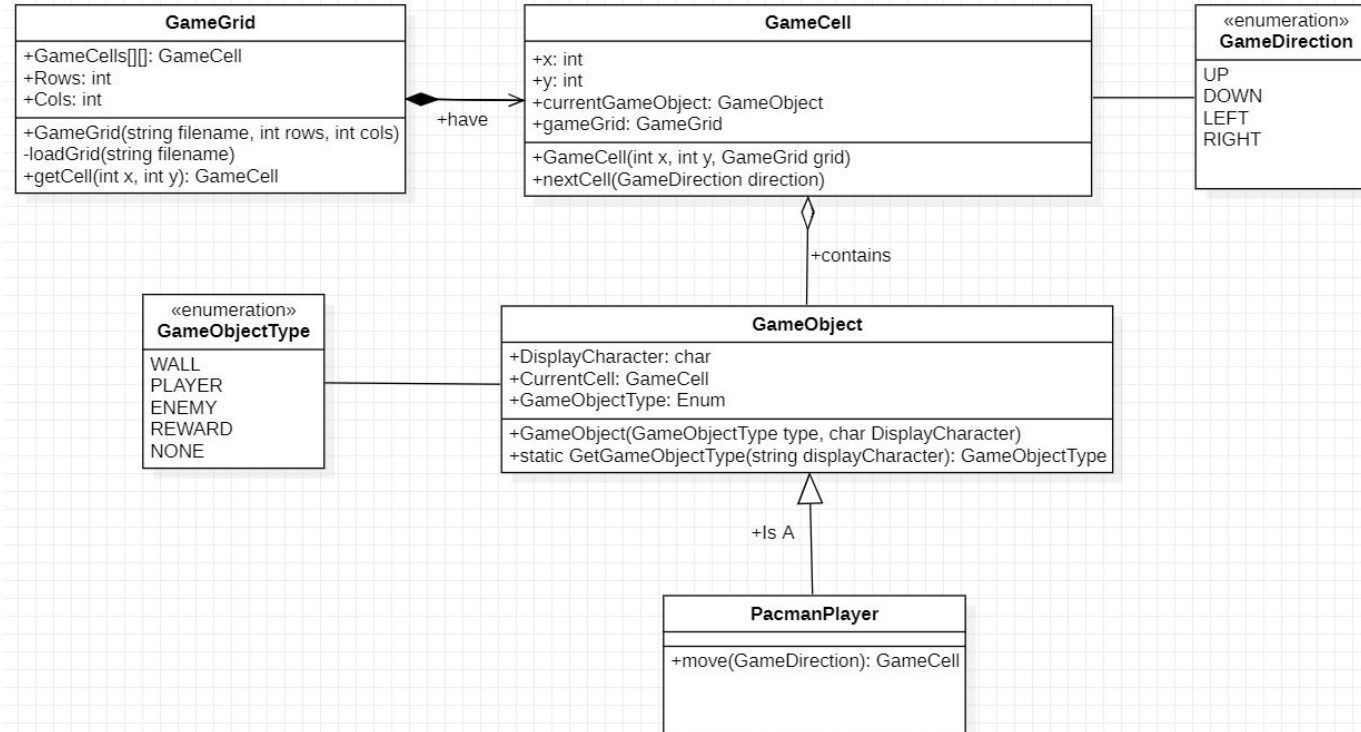


Pacman Class

Make pacman separate class and it will extends the GameObject



Complete Domain Model



Driver Program

Get the **Program.cs** code from google classroom and use our class diagram and make your game

Framework Task 01

Make the complete game with above discussed
method

