Assignment 1 - INF-1400

Kevin Mathias Bergan

07 JAN 2022

# Introduction

In this assignment, we are supposed to make a game called *Breakout* which is an “old” (depends on how you define old), arcade video game that was developed and published by *Atari Inc* 1976. The task emphasizes us to use OOP (Object oriented program) when making breakout.

The point of the game is by moving a player (usually a rectangle) that can either go left or right inside the window. There is “enemies” or targets that the player needs to remove in order to win or procced to the next level. There is a sphere (usually called a ball) that can bounce from the screen edges, player and targets. The player has some sort of control of the ball depending on where that ball hits the player-rectangle.

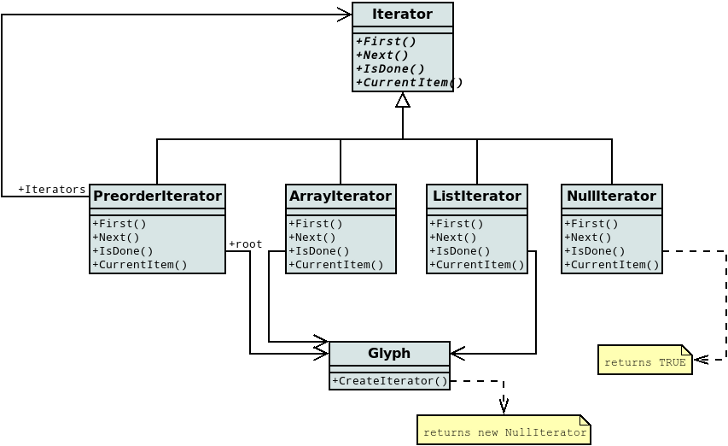
# Technical Background

I think some keywords for this game is (of course) OPP and inheritance. When thinking about this game, it makes a lot of sense to make a general type of a target with some attributes and methods, and just duplicate that type multiple times to make many targets with the same properties. And with the same reasoning, inheritance can make this easier for expanding the targets variations. By that I mean that I can generalize a target with the core mechanics and attributes, and when I want a new “type” of a target I can just inherent the core target and just add some new features that I desire.

# Design

*Give an overarching view of the structure of your solution.*

Describe how your objects fit together, a figure like figure 1 must be included, and you should refer to it in the design section. Remember to refer to figures, such as Figure 1 below, in your text.

**Figure 1**: Example UML diagr

# Implementation

*Describe implementation details, particularly those that are not obvious choices.*

For the implementation of the paddle, the visual representation on the screen is different from the internal representation used for collision detection, by representing the paddle in this way we achieve...

# Evaluation

*Examine if your submission fulfils the requirements and what shortcomings exist.*

In this solution, all requirements are fulfilled, but collision detection between the ball and paddle is inaccurate, due to differences between the visual representation and the implementation...

# Discussion

*Discuss what could be done better, problems you had, experiences etc. (we also appreciate feedback on the assignment or group sessions).*

The implementation of the paddle-ball collision could be done some other way, but due to some reason, the current implemetation is better. After spending two days trying to write the report in LATEX, I gave up, and wrote it in Word instead.

# Conclusion

*Sum up the previous sections.*

I have implemented a solution that fulfills the requirements, the implementation is moderately buggy, but does not crash too much..

# References

|  |  |
| --- | --- |
| [1] | The Dia Developers. Dia website, 2014.  URL http://dia-installer.de/shapes/UML/index.html.en. |
| [2] | Python Software Foundation. Python language reference, version 3.4, 2014.  URL http://www.python.org. |