Prototype

MCR - Projet

HEIG-VD

Crüll Loris, Jaquet David, Rod Julien, Rohrbasser Yoann & Selim Stephan



Table des matières

[1 Introduction 2](#_Toc11182481)

[2 Modèle de Conception Réutilisable 2](#_Toc11182482)

[2.1 Prototype 2](#_Toc11182483)

[2.2 Implémentation 2](#_Toc11182484)

[2.3 Exemple d’utilisation 2](#_Toc11182485)

[2.3.1 Situation 2](#_Toc11182486)

[2.3.2 Schéma UML 3](#_Toc11182487)

[3 Notre implémentation 3](#_Toc11182488)

[3.1 Le projet 3](#_Toc11182489)

[3.2 Schéma UML 3](#_Toc11182490)

[3.3 Bugs connus 3](#_Toc11182491)

[4 Conclusion 3](#_Toc11182492)

# Introduction

Dans le cadre du cours de Modélisation de Conception Réutilisable (MCR), nous sommes amenés à réaliser un projet inventif et original illustrant l’utilisation d’un modèle de conception. Ce modèle a été choisi en 7ème semaine du semestre et nous avons choisi le modèle de conception « Prototype ».

Nous avions cinq semaines pour nous documenter et réaliser une présentation théorique sur le modèle. Ensuite, nous avions de nouveau cinq semaines pour réaliser la partie pratique du projet. Ce rapport concerne cette dernière partie.

# Modèle de Conception Réutilisable

## Prototype

Comme dit dans l’introduction, nous avons choisi le modèle de conception « Prototype » pour faire ce projet. L’utilisation de ce modèle se fait lorsque la création d’une instance possède un grand nombre d’instructions ou prend un temps considérable. Pour faire simple, ce modèle est une sorte d’amélioration du modèle « Fabrique », mais les objets « prototypés » possèdent une méthode « clone ».

## Implémentation

Pour implémenter un « Prototype » dans un programme, nous avons besoin des éléments suivants :

* Gestionnaire de prototypes
* Classe parente à chaque élément que nous voulons prototyper
* Une classe par élément à prototyper

## Exemple d’utilisation

### Situation

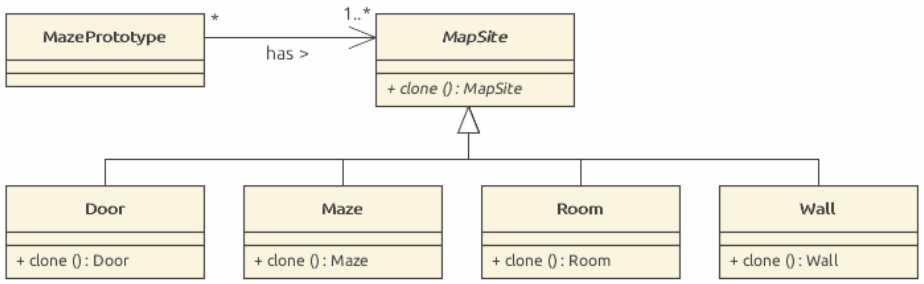
Un exemple d’utilisation de notre modèle de conception serait la création d’un labyrinthe. Pour cet exemple, nous pouvons imaginer que nous voulons une génération dynamique du labyrinthe afin d’implémenter un système de niveau (une fois un labyrinthe terminé, un nouveau labyrinthe est généré).

Dans une telle situation, on peut imaginer que chaque instance des éléments du labyrinthe (porte, pièce, mur et même labyrinthe) a des éléments communs avec les autres instances d’une même classe. En conséquence, à la place de créer plusieurs instances presque identiques, nous pouvons créer un « prototype » d’une classe que nous clonons et modifions des valeurs propres.

Pour reprendre l’exemple du labyrinthe, nous pouvons avoir un prototype d’une porte contenant des éléments qui de ne changeront pas (taille, matériaux, …) et définir les éléments pouvant être modifiés (pièces reliées à la porte, …) lors du clonage.

### Schéma UML

Voici un schéma UML de la situation décrite au point précédent. Il est important de noter que ce schéma n’est pas complet (les classes ne possèdent pas tous les attributs nécessaires, n’ont pas toutes les méthodes nécessaires, …). Néanmoins, ce schéma est suffisant pour illustrer l’utilisation du modèle de conception « Prototype ».



Voici la correspondance des éléments selon la liste fournie lors de la description de l’implémentation du modèle :

* MazePrototype : Gestionnaire de prototypes
* MapSite : Classe parente à chaque élément que nous voulons prototyper
  + Sous-classes : Elément à prototyper

# Notre implémentation

## Le projet

Explication du projet (produit fini)

## Schéma UML

Screenshot de l’UML

## Bugs connus

Explication des bugs

# Conclusion