SAMEGAME

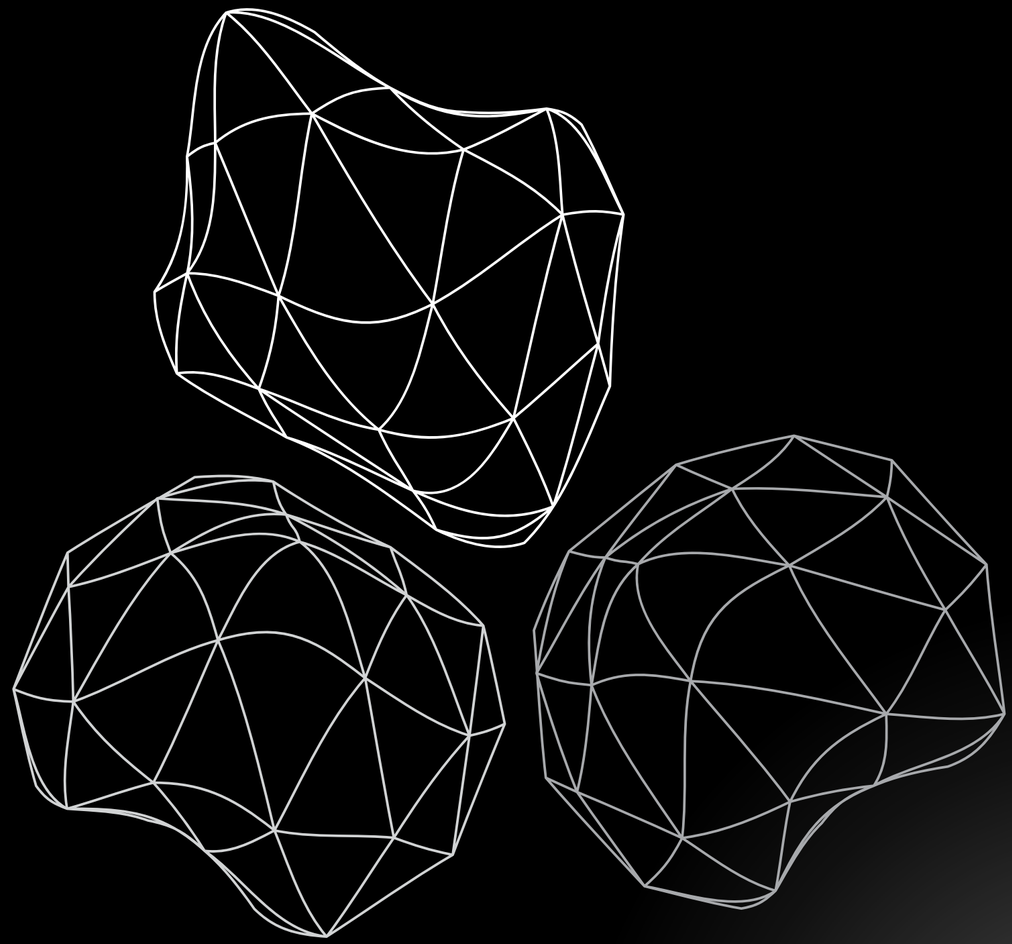
PRESENTATION

MEMBRES

- Alli Rayane
- Ait Ali Ilyas

LIEN GITHUB

https://github.com/MWB21223/SomeGame



INTRODUCTION À SAMEGAME

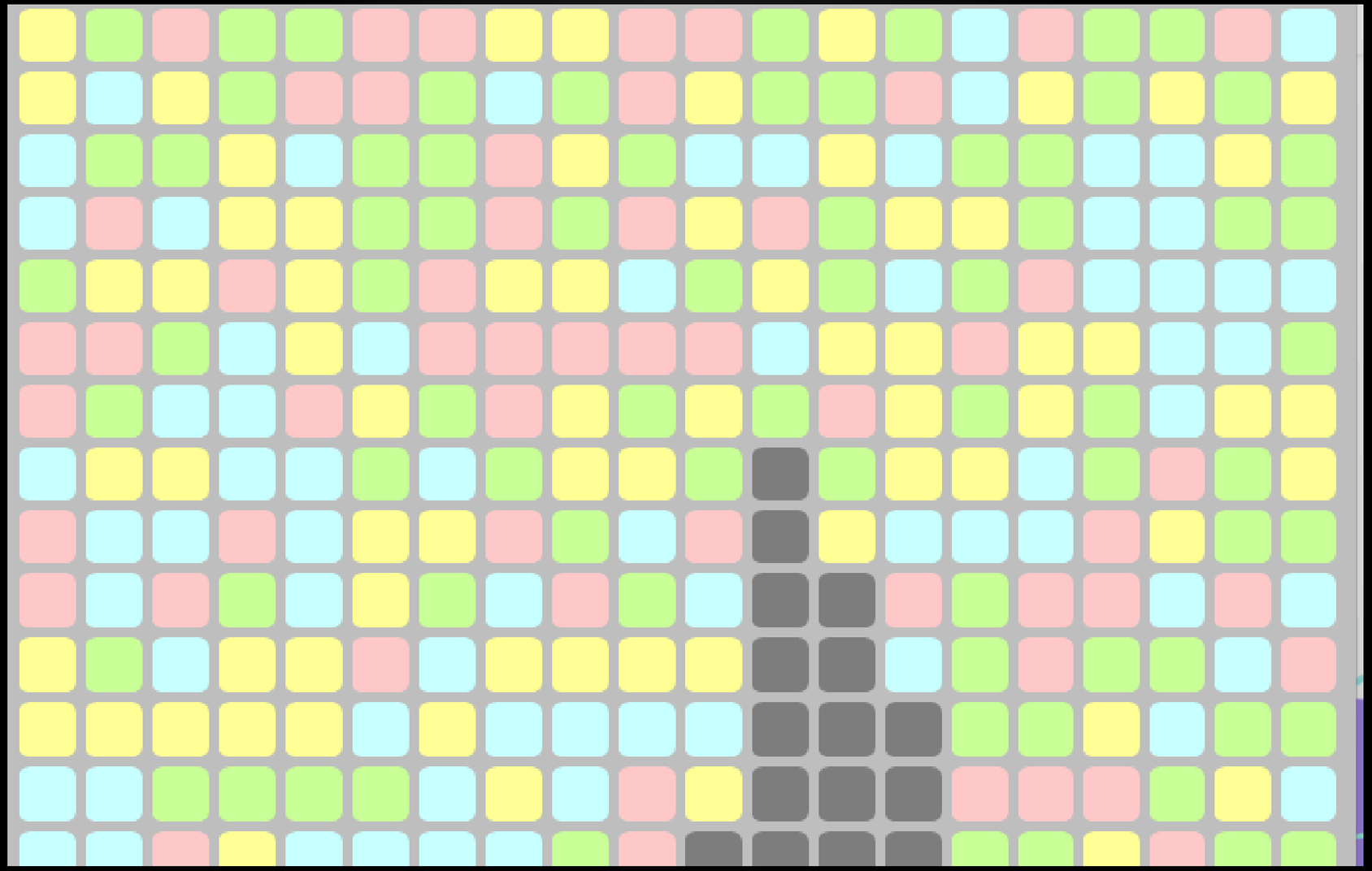
Qu'est-ce que SameGame ?

Jeu de puzzle implémenté en Pharo avec le framework graphique Bloc.

But du jeu

Cliquer sur des groupes de cases de même couleur pour les faire disparaître. (ex: Rouge, Bleu, Vert, Jaune)

INTERFACE DU JEU



ARCHITECTURE DE BASE

Structure MVC (Modèle–Vue)

SGBoard (Modèle) est découplé de SGBoxElement (Vue).
Sépare les règles du jeu de l'affichage pour faciliter la maintenance.

Gestion des États (State)

SGBox délègue sa couleur à des Singletons (SGBlueState, SGRedState…).
Évite de recréer les Box à chaque changement.

Réactivité (Announcements)

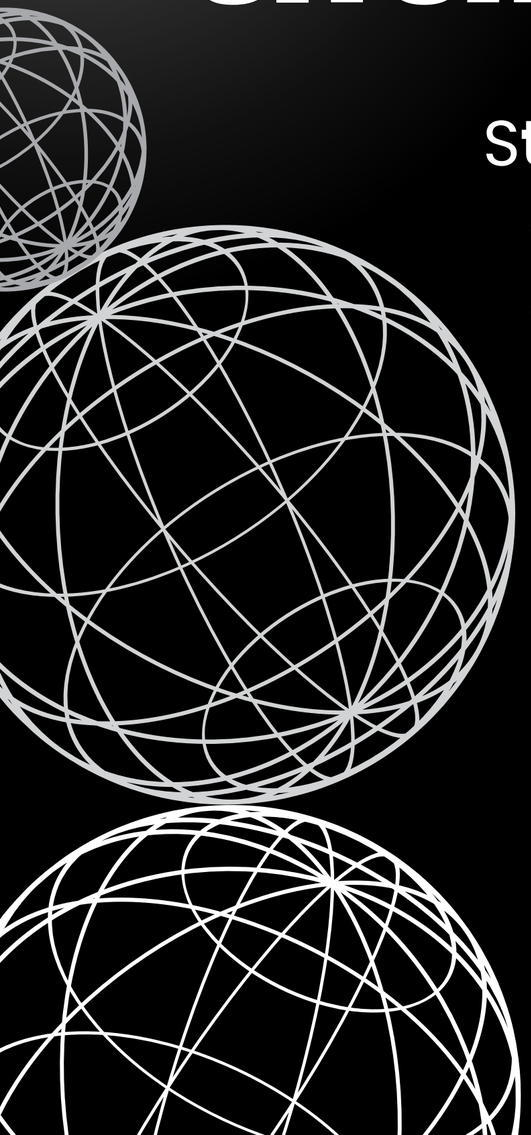
Observateur via SGStateChangedAnnouncement.
La Vue s'abonne au Modèle et se met à jour automatiquement.

FEATURE TAILLE VARIABLE

Nom de la fonctionnalité : Sélection de Taille de Grille Variable

Objectif : Permettre au joueur de choisir la dimension de la grille (Small, Medium, Large) avant le lancement d'une partie pour adapter la durée et la difficulté du jeu, au lieu d'être contraint à une taille unique.

SITUATION INITIALE



S**tructure de départ :**

- La méthode SameGame class >> open lançait directement le jeu.
- La taille de la grille (20x20) était codée en dur directement dans la méthode d'ouverture.

- **Problème :**

Aucune flexibilité, le joueur ne pouvait pas choisir une partie rapide ou longue.

MODIFICATIONS APPORTÉES

Classe SGSizeSelector : Une nouvelle interface utilisateur présentant 3 boutons de choix (Small/Medium/Large).

Méthode openWidth:height: : Une nouvelle méthode capable d'initialiser le jeu avec n'importe quelles dimensions valides.

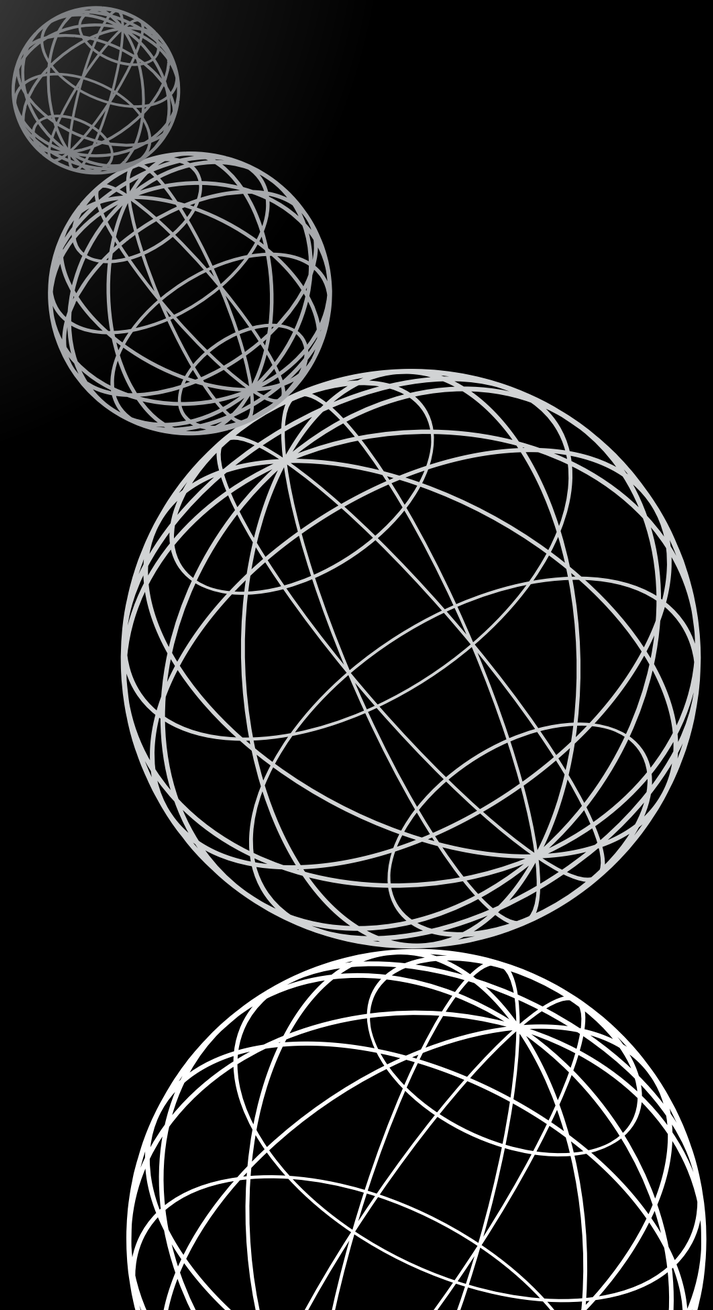
Refactorisation de open : Modification pour qu'elle n'ouvre plus le jeu directement, mais lance d'abord le sélecteur SGSizeSelector.

NOUVELLE INTERFACE

FEATURE SKIN

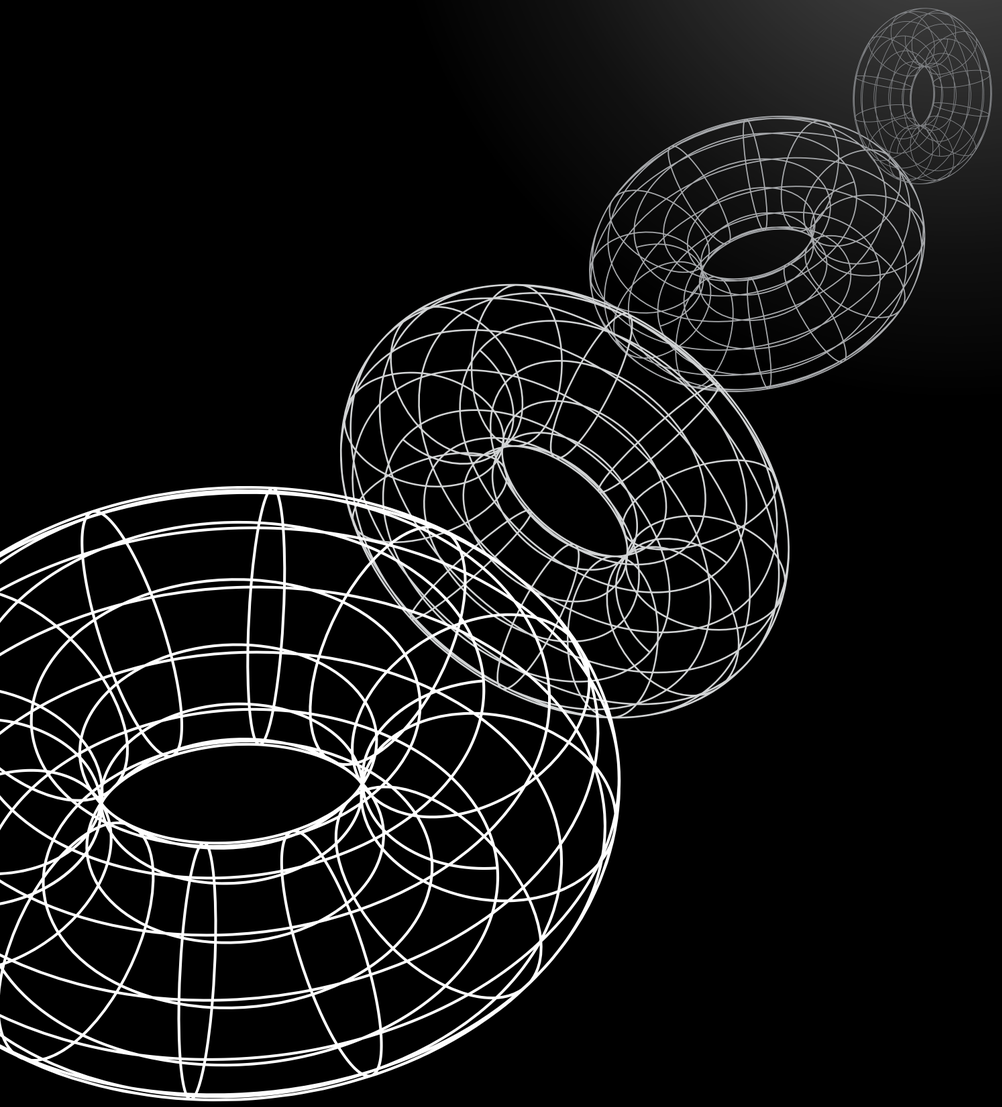
Nom de la fonctionnalité : Système de Thèmes Dynamique (Skins)

Objectif : Permettre à l'utilisateur de changer l'apparence visuelle du jeu (les couleurs des blocs) instantanément pendant une partie, sans avoir à redémarrer le jeu ni à perdre sa progression.



CONCEPTION DE DEPART

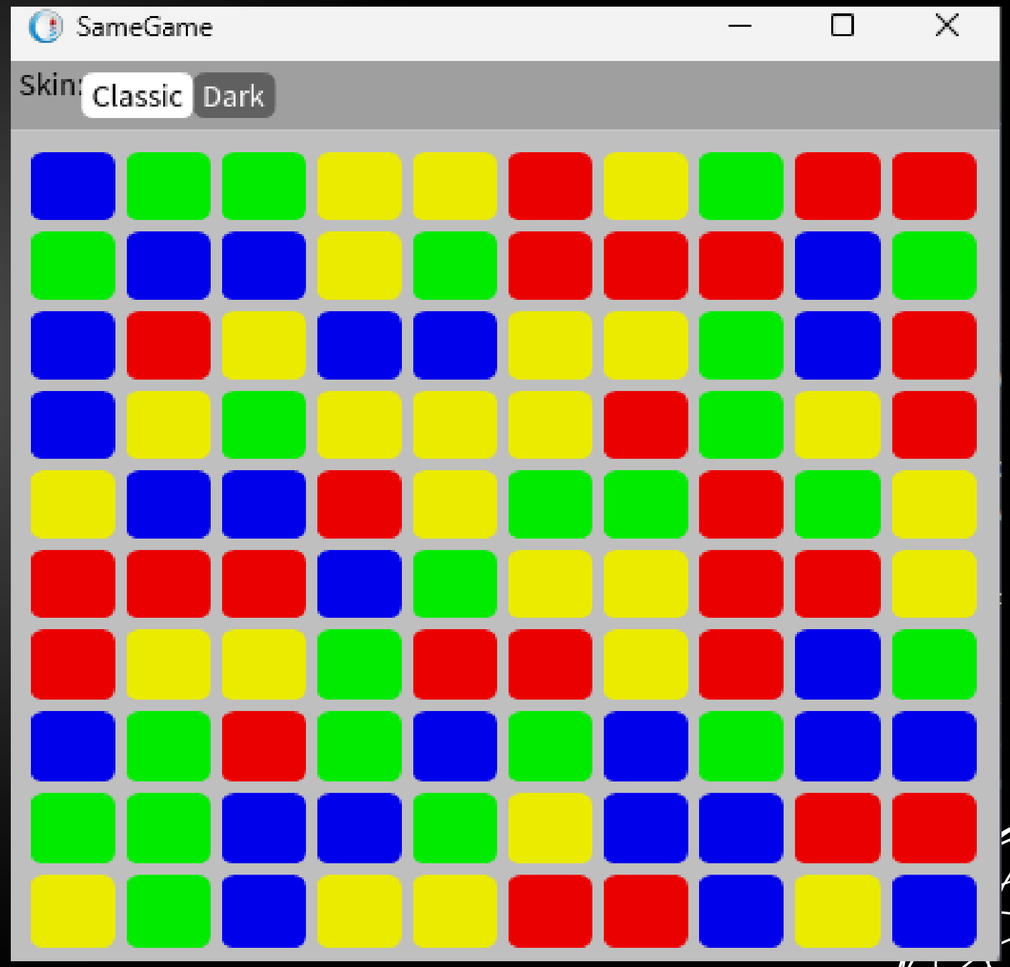
- Structure codé en dur
- Simple polymorphisme
- Pas de pattern utilisé pour nos couleurs



CONCEPTION D’ARRIVÉE

- Pattern Strategy (darkSkin-ClassicSkin)
- pattern Visitor (logique des blocs couleurs)
- pattern Observer (gestionnaire d’apparence skin)

NOUVELLE INTERFACE



NOUVEAUX TESTS AJOUTÉS

1. Tests des Thèmes (SGAppearanceTest)
Vérifie que les skins (Classic/Dark) renvoient les bonnes couleurs.
Validations : couleurs correctes, notification envoyée lors du changement de skin.

2. Tests Réactivité UI (SGBoxElementAnnouncementTest)
Garantit que l'interface se met à jour sans redémarrage.
Validations : une case écoute ses propres changements d'état et les changements de thème.

3. Tests Configuration Taille (SGSizeConfigurationTest)
Valide les presets de taille de grille.
Validations : Small = 10×10, Medium = 20×20, Large = 30×30.

OBJECTIF DE NOTRE PROJET

L'objectif principal était de rendre l'apparence du jeu “dynamique et extensible”.

Séparation des préoccupations : Découpler complètement la structure de la logique.

MERCI POUR VOTRE ATTENTION