

Modèle de données By Year

Objets Unity

- `SimulationByYear` : simulation et options propres à la famille;
- `ActivityByYear` : année courante, états, calendrier et données de l'activité;
- `TeamByYear` : équipe et données runtime;
- `TeamDecisionsByYear` : décisions par année;
- `ResultsByYear` : résultats annuels;
- `TeamEvaluationByYear` / `StudentEvaluationByYear` : évaluations;
- `StaticDataByYear` : base de static data des jeux annuels.

Controllers

Les classes `ActivityByYearController`, `SimulationByYearController`, `TeamByYearController`, `TeamEvaluationByYearController` et `StudentEvaluationByYearController` servent de couche runtime entre ces modèles et l'UI.

Documents serveur

```
flowchart TD
    SIM[(simulation)] --> ACT[(activity)]
    ACT --> TEAM[(teams)]
    ACT --> GAME[(actDataGameSpec)]
    TEAM --> DEC[(team decisions / activities)]
    ACT --> RES[(results)]
    TEAM --> ET[(evaluationTeam)]
    SIM --> ES[(evaluation)]
    TEAM --> LOG[(teamsLogbooks)]
```

currentYear / currentYearState

L'activité garde l'année courante et son état. Les transitions serveur utilisent notamment les états “not started”, “year started”, “generating” et “year finished”. Les contrôles Unity affichent l'écran approprié en fonction de cet état.

years[]

Chaque entrée annuelle peut porter les dates de début/fin et le flag de pratique. Le calendrier et le mode de gestion déterminent si les transitions se font manuellement ou automatiquement.

Ajouter un champ de décision ou de résultat By Year

Décision étudiante

1. Ajouter le champ dans le modèle Unity `TeamDecisions<Game>` ou `YearData` selon son scope.
2. Ajouter la désérialisation avec une valeur compatible avec les anciennes équipes.
3. Ajouter le contrôle UI qui édite cette décision.
4. Ajouter la custom command serveur ou étendre celle qui possède déjà la décision.
5. Sauvegarder dans le bon `yearData` de l'équipe.
6. Envoyer un RTT game-specific si les coéquipiers doivent voir le changement.
7. Consommer la décision dans `GenerateResults<Game>`.
8. Mettre à jour `anyDecisionsTaken()` si cette décision compte pour déterminer si l'équipe a commencé.
9. Mettre à jour les bots si une équipe bot doit prendre cette décision.

Résultat annuel

1. Ajouter le champ au modèle serveur de résultat produit par `generateCurrentYearResults()`.
2. Le persister dans `results`.
3. Ajouter le modèle Unity `Results` / `ResultsByYear` et désérialiser.
4. Ajouter report/tooltip/UI.
5. Si le champ participe à ActPerf, mettre à jour `Eval<Game>.computeActPerfScore()`.
6. Si le champ participe au ranking, mettre à jour `updateRankings<Game>()`.

7. Vérifier `deleteResults()` /reopen year.

Où placer une donnée annuelle

Donnée	Emplacement
Décision qui change chaque année	<code>TeamDecisions<Game>.yearData[year]</code> ou structure équivalente.
État d'équipe durable entre années	<code>Team<Game></code> / team data persistante.
Résultat calculé d'une année	<code>results</code> .
Configuration immuable	Static data.
Donnée commune à toute l'activité	<code>actDataGameSpec</code> .

Compatibilité anciens documents

Une nouvelle propriété de team/activity doit être tolérante à l'absence. Si sa valeur doit absolument exister sur les vieux documents, utiliser la migration correspondante dans `_utils/version/`.

Revision #2

Created 2026-09-18 19:19:41 UTC by Mathis

Updated 2026-09-18 20:17:48 UTC by Mathis