

Master 1 GIL - Gestion de projet
Document d'Architecture Logiciel du "Splendor"
Agora V3-1

Groupe 1 Agora V3-1 : Partie bibliothèque de jeux

23 janvier 2024

Version	3
Date	20 mai 2024
Rédigé par	BINGINOT Etienne CAUCHOIS Niels DUCROQ Yohann KHABOURI Izana MAZUY Axelle MONTAGNE Erwann THIBERVILLE Malvina VAN LIEDEKERKE Florian

Mises à jour du document

Version	Date	Modification réalisée
1	4 décembre 2023	Création du document
2	23 janvier 2024	Mise à jour après les retours de l'enseignant de gestion de projet
3	20 mai 2024	Mise à jour des services

Table des matières

1	Objet du document	4
2	Documents de références	5
3	Terminologie	5
4	Architecture statique	6
4.1	Les entités	6
4.1.1	Structure	6
4.1.2	Description des composants	6
4.2	Les services	11
4.2.1	BirthMYRService	11
4.2.2	DataManagementMYRService	11
4.2.3	EventMYRService	12
4.2.4	HarvestMYRService	13
4.2.5	MYRService	14
4.2.6	WinterMYRService	18
4.2.7	WorkerMYRService	19
4.2.8	WorkshopMYRService	22
4.3	Les contrôleurs	24
4.3.1	Structure	24
4.3.2	Description des composants	25
5	Architecture dynamique	26
6	Maquette de vue	37

1 Objet du document

Ce document a pour but de décrire l'architecture logicielle d'une version digitale du jeu Myrmes. Nous allons utiliser la bibliothèque que l'on va créer pour l'implémentation de jeux.

Ce jeu constitue une base à analyser pour alimenter la bibliothèque de manière générale et regrouper avec les autres analyses de jeux les points communs.

Ce jeu sera rendu fonctionnel aux joueurs inscrits et implémenté avec l'utilisation du framework Symfony.

2 Documents de références

Nos références utilisées sont les DAL des précédentes années concernant le projet AGORA, la feuille de présentation du projet, les STB générale et de Myrmes réalisées ainsi que les compte-rendus des réunions avec le client et le référent technique.

3 Terminologie

Marqueur : donne une information particulière sur le jeu.

Saison/Tour : dans le jeu "Myrmes" une saison correspond à un tour. Un tour est une phase d'un jeu constituée d'une action de chacun des joueurs. Un tour est fini lorsque tous les joueurs ont effectué leur action.

Partie : jeu en cours, une fois les paramètres et règles déterminés, conférant une action au tour par tour à chaque joueur.

Année/Manche : une année correspond à une manche dans le lexique de Myrmes. Une manche est une phase d'un jeu constituée d'un ensemble de tours, après lesquels les paramètres de la partie sont remis à zéro et les scores sont tenus.

La fourmilière : correspond au plateau personnel du joueur.

Ouvrière : fourmi que le joueur peut utiliser pour fabriquer des ressources dans la fourmilière, ou sortir sur le plateau pour réaliser toutes sortes d'actions.

Soldat : fourmi utiliser par le joueur lorsqu'une ouvrière rencontre une proie pour chasser, ou envahir une phéromone adverse et permettent de garder des ressources pour l'hiver.

Nourrice : fourmi spéciale (en forme d'octogone) et essentielle à l'avancée de la fourmilière, qui va permettre de donner naissance aux fourmis souhaitées dans la fourmilière, de réaliser des objectifs ou d'améliorer la fourmilière en se plaçant dans l'atelier, ou encore de produire des larves. Lorsque la nourrice réalise un objectif celle-ci est perdue.

Larve : ressource spéciale qui permet différents types d'actions dans la partie.

Proie : pièce sur le plateau principal pouvant être chassée pour obtenir des ressources et/ou des points de victoire.

Ressources : représentées par trois sortes de cubes. Le cube vert correspond à la "nourriture", le cube marron correspond à la terre, et le cube gris à la pierre.

Tuile de phéromones : correspond à une pièce spéciale pouvant être posée par une ouvrière sur le plateau principale qui permet de récupérer les ressources correspondantes sur lesquelles la tuile est placée, et permet de se déplacer plus rapidement.

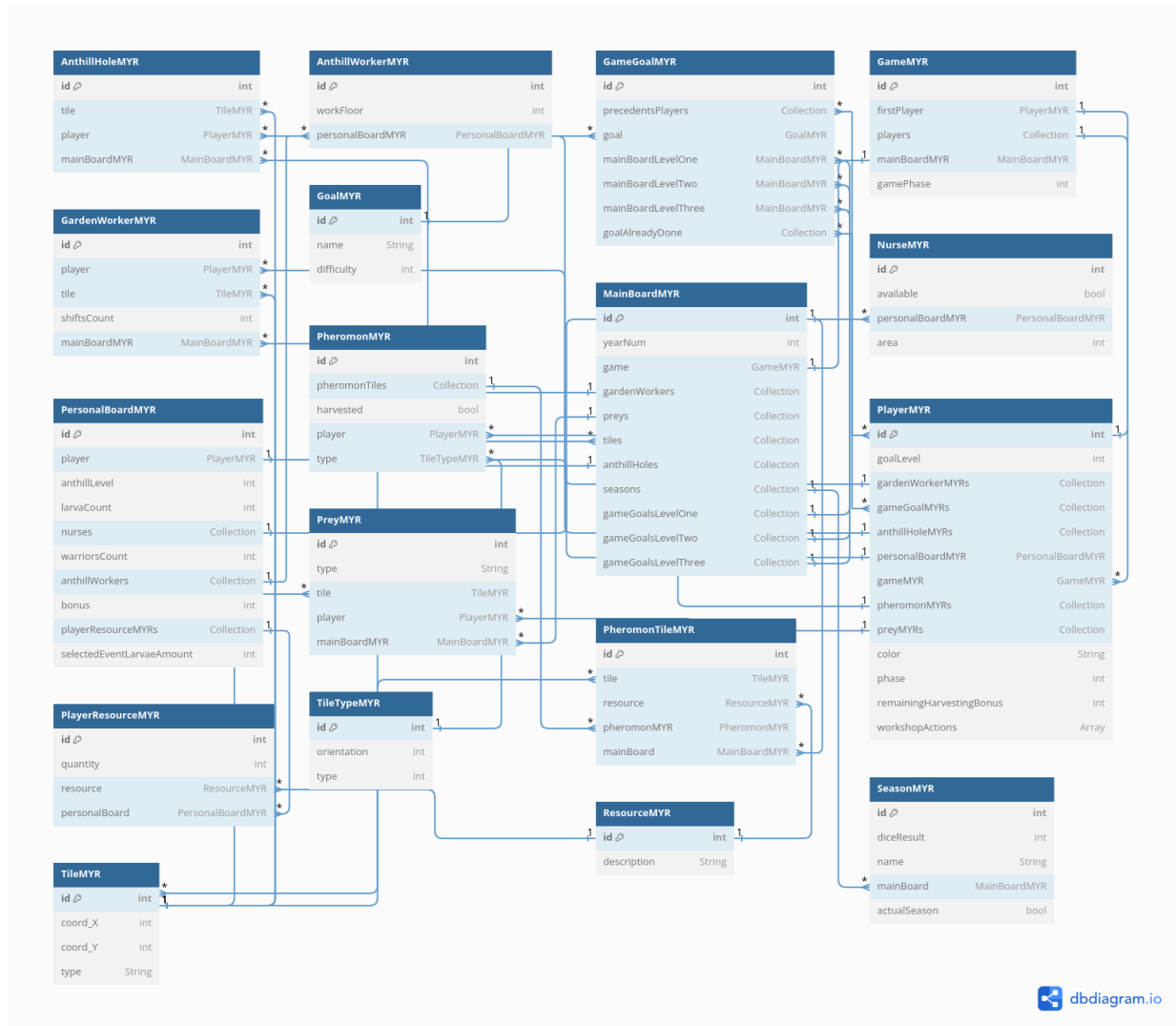
Points de victoire : nombre de points qu'un joueur possède qui va déterminer le classement final et qui est attribué en fonction des objectifs que le joueur va réaliser.

4 Architecture statique

4.1 Les entités

4.1.1 Structure

L'architecture statique se présente par les entités du jeu que l'on retrouve en base de données, permettant ainsi de stocker les informations d'une partie en cours.



4.1.2 Description des composants

Nom de la classe : GameMyr

Description : Cette classe permet d'identifier une partie du jeu "Myrmes" en cours.

Attributs	Méthodes
gameId : un entier identifiant une partie en cours	
playersId : un tableau d'entiers identifiant les joueurs d'une partie en cours	
firstPlayer : un entier référençant un joueur qui sera le premier à jouer	

Nom de la classe : PlayerMyr

Description : Cette classe permet d'identifier un joueur d'une partie en cours.

Attributs	Méthodes
gameId : un entier identifiant une partie en cours	
playerId : un entier identifiant le joueurs d'une partie en cours	
personalBoard : un entier référençant le plateau personnel du joueur	
score : un entier représentant le nombre de points du joueur	
goalLevel : un entier représentant le niveau des objectifs du joueurs	

Nom de la classe : MainBoardMyr

Description : Cette classe permet d'identifier le plateau de jeu principal d'une partie de Myrmes.

Attributs	Méthodes
gameId : un entier identifiant une partie en cours	
yearNum : année en cours sur le plateau	
seasonsId : un tableau d'entiers référençant les 3 saisons de l'année en cours	
actualSeason : un entier représentant la saison en cours.	

Nom de la classe : PersonalBoardMyr

Description : Cette classe permet d'identifier le plateau de jeu personnel d'un joueur.

Attributs	Méthodes
gameId : un entier identifiant une partie en cours	
personalBoardId : un entier identifiant le plateau d'un joueur d'une partie en cours	
anthillLevel : un entier représentant le niveau de la fourmilière	
larvaNumber : un entier représentant le nombre de larves dans la fourmilière.	
nursesId : un tableau d'entiers référençant les nourrices du plateau personnel.	
warriorsNumber : un entier représentant le nombre de combattantes sur le plateau personnel.	
anthillWorkersId : un tableau d'entiers référençant les fourmis travaillant dans la fourmilière appartenant au joueur.	
gardenWorkersId : un tableau d'entiers référençant les fourmis travaillant dans le jardin appartenant au joueur.	
bonus : un entier représentant le numéro de la case associée au bonus du joueur en début de manche.	
huntedPreyNumber : un entier représentant le nombre de proies chassées par le joueur.	

Nom de la classe : AnthillWorkerMyr

Description : Cette classe permet d'enregistrer les informations concernant les fourmis travaillant dans la fourmilière d'un joueur dans une partie.

Attributs	Méthodes
gameId : un entier identifiant une partie en cours	
playerId : un entier identifiant le joueur d'une partie en cours	
workerId : un entier identifiant la fourmi travailleuse	
workFloor : un entier représentant l'étage de travail de la fourmi.	

Nom de la classe : GardenWorkerMyr

Description : Cette classe permet d'enregistrer les informations concernant les fourmis travaillant dans le jardin d'un joueur dans une partie.

Attributs	Méthodes
gameId : un entier identifiant une partie en cours	
playerId : un entier identifiant le joueur d'une partie en cours	
workerId : un entier identifiant la fourmi travailleuse	
compartmentTileId : un entier représentant la tuile où se trouve la fourmi.	
shiftsNumber : un entier représentant le nombre de déplacement restant à la fourmi.	

Nom de la classe : NurseMyr

Description : Cette classe permet d'enregistrer les informations concernant les nourricières d'un joueur dans une partie.

Attributs	Méthodes
gameId : un entier identifiant une partie en cours	
playerId : un entier identifiant le joueur d'une partie en cours	
nurseId : un entier identifiant la nourricière	
position : un entier représentant la position actuelle de la nourricière.	

Nom de la classe : TileMyr

Description : Cette classe permet d'identifier une case du plateau de Myrmes.

Attributs	Méthodes
x_minCoord : un entier représentant la coordonnée minimale (la plus à gauche) de la case sur un axe horizontal	
x_maxCoord : un entier représentant la coordonnée maximale (la plus à gauche) de la case sur un axe horizontal	
y_coord : un entier représentant la coordonnée de la case sur un axe vertical	
tileId : un entier référençant la tuile case.	
type : une chaîne de caractères décrivant le type de case.	

Nom de la classe : AnthillHoleMyr

Description : Cette classe permet d'enregistrer les informations concernant les trous de fourmilière d'un joueur.

Attributs	Méthodes
gameId : un entier identifiant une partie en cours	
playerId : un entier identifiant le joueur d'une partie en cours	
holeId : un entier identifiant l'un des trous de la fourmilière du joueur dans la partie.	
tileId : un entier représentant la case où se trouve ce trou.	

Nom de la classe : PlayerStockMyr

Description : Cette classe permet d'enregistrer les informations concernant le stock d'un joueur.

Attributs	Méthodes
gameId : un entier identifiant une partie en cours	
resourceId : un entier identifiant la ressource stockée.	
quantity : un entier représentant la quantité de la ressource stockée.	
personalBoardId : un entier référençant le plateau personnel d'un joueur.	

Nom de la classe : ResourceMyr

Description : Cette classe permet d'enregistrer les informations concernant les ressources.

Attributs	Méthodes
resourceId : un entier identifiant une ressource.	
description : une chaîne de caractères décrivant la ressource.	

Nom de la classe : GardenTileMyr

Description : Cette classe permet d'enregistrer les informations concernant les tuiles de jardin.

Attributs	Méthodes
gameId : un entier identifiant une partie en cours	
gardenTileId : un entier identifiant la tuile	
playerId : un entier identifiant le joueur d'une partie en cours	
type : une chaîne de caractères décrivant le type de tuile. Une tuile se définit comme suit : 1) si phéromone ou tuile spéciale 2) quel phéromone ou quelle tuile spéciale 3) orientation de la tuile	
resourcesId : un tableau d'entiers référençant les ressources d'une tuile.	
compartmentTilesId : un tableau d'entiers référençant les cases sur lesquelles sont posées la tuile.	
hasAlreadyBeenHarvested : un booléen indiquant si une ressource a déjà été récoltée sur la tuile de type phéromone ce tour	

Nom de la classe : GameGoalMyr

Description : Cette classe permet d'enregistrer les informations concernant les objectifs de la partie en cours.

Attributs	Méthodes
gameId : un entier identifiant une partie en cours	
precedentPlayersId : un tableau d'entiers référençant les joueurs ayant réalisés cet objectif.	
goalId : un entier identifiant l'un des objectifs d'une partie en cours	

Nom de la classe : GoalMyr

Description : Cette classe permet d'enregistrer les informations concernant les objectifs.

Attributs	Méthodes
name : une chaîne de caractères décrivant l'objectif.	
difficulty : un entier représentant la difficulté de l'objectif.	
goalId : un entier identifiant l'un des objectifs d'une partie en cours	

Nom de la classe : PreyMyr

Description : Cette classe permet d'enregistrer les informations concernant les proies d'une partie.

Attributs	Méthodes
preyId : un entier identifiant une proie	
type : un entier représentant le type de proie.	
tileId : un entier référençant la tuile sur laquelle se trouve une proie.	
gameId : un entier référençant la partie en cours.	

Nom de la classe : SeasonMyr

Description : Cette classe permet d'enregistrer les informations concernant une saison.

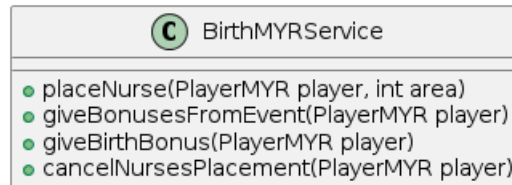
Attributs	Méthodes
gameId : un entier identifiant une partie en cours	
seasonId : un entier identifiant la saison.	
diceResult : un entier représentant le résultat obtenu par le dé de la saison	

4.2 Les services

L'architecture statique se présente par les services du jeu, permettant ainsi de manipuler les informations d'une partie en cours.

4.2.1 BirthMYRService

4.2.1.1 Structure



4.2.1.2 Description des composants

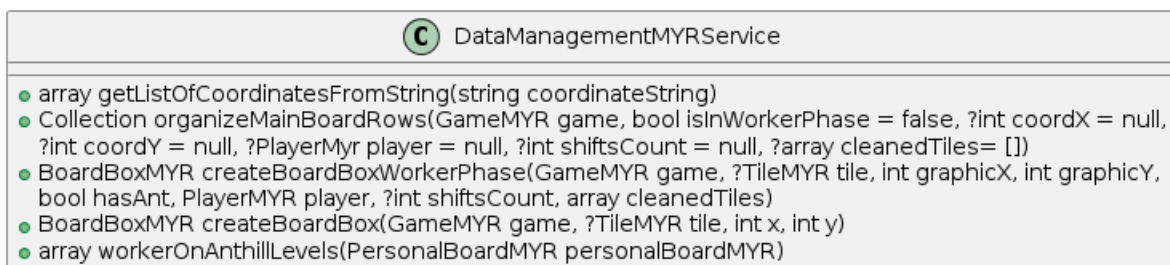
Nom de la classe : BirthMYRService

Description : Cette classe permet d'implanter les fonctionnalités du Myrmes concernant la phase naissance.

Méthode	Type de retour	Description	Référence
placeNurse(PlayerMYR player, int area)	void	Place une nourrice du joueur dans la zone demandée	
giveBonusesFromEvent (PlayerMYR player)	void	Accorde au joueur son éventuel bonus lié à sa phase événement	
giveBirthBonus (PlayerMYR player)	void	Accorde au joueur ses bonus de naissance en fonction de la position de ses nourrices	
cancelNursesPlacement (PlayerMYR player)	void	Retire toutes les nourrices du joueur de la piste de naissance	

4.2.2 DataManagementMYRService

4.2.2.1 Structure



4.2.2.2 Description des composants

Nom de la classe : DataManagementMYRService

Description : Cette classe permet de manipuler les données de Myrmes et de les convertir dans du contenu affichable.

Méthode	Type de retour	Description	Référence
getListOfCoordinatesFromString (string coordinateString)	array	Transforme une chaine de type "[X_Y ... X_Y]" en un tableau de coordonnées	
organizeMainBoardRows (GameMYR game, bool isInWorkerPhase = false, ?int antCoordX = null, ?int antCoordY = null, ?PlayerMYR player = null, ?int shiftsCount = null, ?array cleanedTiles = [])	Collection	Retourne une collection de lignes de tuiles représentant le plateau principal du haut vers le bas	
createBoardBoxWorkerPhase (GameMYR game, ?TileMYR tile int graphicX, int graphicY, bool hasAnt, PlayerMYR player, ?int shiftsCount, array cleanedTiles)	BoardBoxMYR	Crée une tuile de type BoardBox contenant la tuile, proie et pheromone données en paramètre, durant la phase ouvrière	
createBoardBox (GameMYR game, ?TileMYR tile int x, int y)	BoardBoxMYR	Crée une tuile de type BoardBox contenant la tuile, proie et pheromone données en paramètre, durant une autre phase que la phase ouvrière	
workerOnAnthillLevels (PersonalBoardMYR personalBoardMYR)	array	Renvoie un tableau où chaque indice indique si une ouvrière se trouve à l'étage correspondant	

4.2.3 EventMYRService

4.2.3.1 Structure



4.2.3.2 Description des composants


Nom de la classe : EventMYRService

Description : Cette classe permet d'implanter les fonctionnalités du Myrmes concernant la phase événement.

Méthode	Type de retour	Description	Référence
upBonus (PlayerMYR player)	void	le joueur augmente son bonus d'un cran	
lowerBonus (PlayerMYR player)	void	le joueur baisse son bonus d'un cran	
confirmBonus (PlayerMYR player)	void	le joueur confirme son bonus	

4.2.4 HarvestMYRService

4.2.4.1 Structure

 HarvestMYRService
<ul style="list-style-type: none">● bool areAllPheromonesHarvested(PlayerMYR player)● bool canStillHarvest(PlayerMYR player)● harvestSpecialTiles(PlayerMYR player)● harvestPheromone(PlayerMYR player, TileMYR tile)● harvestPlayerFarms(PlayerMYR player)● harvestPlayerQuarry(PlayerMYR player, PheromonMYR pheromone, string resource)● harvestPlayerSubAnthill(PlayerMYR player)

4.2.4.2 Description des composants

Nom de la classe : HarvestMYRService

Description : Cette classe permet d'implanter les fonctionnalités du Myrmes concernant la phase récolte.

Méthode	Type de retour	Description	Référence
areAllPheromonesHarvested(PlayerMYR player)	bool	indique si toutes les phéromones du joueur ont été récoltées	
canStillHarvest(PlayerMYR player)	bool	indique si le joueur doit encore être en phase récolte	
harvestSpecialTiles(PlayerMYR player)	void	récolte toutes les tuiles spéciales du joueur	
harvestPheromone(PlayerMYR player, TileMYR tile)	void	récolte une phéromone du joueur sur la tuile donnée	
harvestPlayerFarms(PlayerMYR player)	void	récolte tous les élevages du joueur	
harvestPlayerQuarry(PlayerMYR player, PheromonMYR pheromone, string resource)	void	le joueur récolte une fouille et obtient la ressource demandée	
harvestPlayerSubAnthill(PlayerMYR player)	void	récolte toutes les sous-fourmilières du joueur	

4.2.5 MYRService

4.2.5.1 Description des composants

Nom de la classe : MyrService

Description : Cette classe permet de manipuler les entités afin d'en changer le comportement.

Méthode	Type de retour	Description	Référence
getNumberOfFreeWorkerOfPlayer(PlayerMyr player)	int	retourne le nombre d'ouvrières disponible et appartenant à player	
getNursesInWorkshopFromPlayer(PlayerMyr player)	Collection	retourne les nourrices se trouvant dans l'atelier et appartenant à player	
getTileTypeFromTypeAndOrientation(int type, int orientation)	TileTypeMYR	retourne les tuile selon le type de tuile, ainsi que son orientation	
getPlayerResourceAmount(PlayerMyr player, string resourceName)	int	retourne la quantité de la ressource dont le nom est resourceName et appartenant au joueur	
getAvailableLarvae(PlayerMyr player)	int	retourne le nombre de larves disponible que possède le joueur	
canOnePlayerDoWorkshopPhase(GameMYR game)	bool	vérifie si au moins un joueur possède encore une nourrice dans la zone de l'atelier	
isInPhase(PlayerMyr player, int phase)	bool	vérifie si le joueur est dans la phase donnée	
getPlayerFromNameAndGame(GameMYR game, string name)	PlayerMYR	retourne un joueur dont son nom est name et participant dans la partie donnée	

Méthode	Type de retour	Description	Référence
initialize NewGame(GameMYR game)	void	initialise la partie associé à game	
getPheromones FromType(int type)	ArrayCollection	retourne une collection de toutes les orientations des phéromones, ou de tuiles spéciales dont son type est celui donnée en paramètre	
getDiceResults(GameMYR game)	ArrayCollection	retourne une collection de toutes les valeurs aléatoires pour chaque saison	
getActualSeason(GameMYR game)	SeasonMYR	retourne la saison actuelle	
getPlayerResourceOfType(PlayerMyr player, string type)	PlayerResourceMYR	retourne les ressource du joueur du type donnée	
isGameEnded(GameMYR game)	bool	vérifie si la partie est finie	
getNursesAtPosition(PlayerMYR player, int position)	ArrayCollection	retourne les nourrices disponibles se trouvant à la position donnée	
getActualPlayer(GameMYR game)	PlayerMYR	retourne le joueur qui doit jouer à son tour	
manageNursesAfter BonusGive(PlayerMYR player, int nurseCount, int positionOfNurse)	void	manipule les nourrices ayant été utilisé pour les différents bonus	
setPhase(PlayerMYR player, int phase)	void	change la phase de jeu pour le joueur et change la phase pour les autres joueurs en cas de synchronisation	
setNextPlayerTurn(PlayerMYR actualPlayer)	void	passse le tour au prochain joueur suivant le joueur actuel	
endPlayerRound(PlayerMYR player)	void	met fin au tour du joueur donné suivant le joueur actuel	
canManageEndOfPhase (GameMYR game, int phase)	bool	vérifie si tous les joueurs ont fini de joué la phase donnée	
manageEndOfRound (GameMYR game)	void	réalise toutes les actions liées à la fin d'une manche	
exchangeLarvaeForFood (PlayerMYR player)	void	le joueur réalise l'échange de trois larves pour une ressource	
initializeNewSeason (GameMYR game, string seasonName)	void	initialisation d'une nouvelle saison	
resetGameGoalsDone DuringThe Round(GameMYR game)	void	à la fin d'une manche, réinitialise les objectifs accomplis durant la manche	
discardLarvae(PlayerMYR player)	void	retire les larves sélectionnés par le joueur donné	
replaceWorkers(PlayerMYR player)	void	tous les fourmis ouvrières sont remis à la zone ouvrière	
replaceNurses(PlayerMYR player)	void	tous les nourrices du joueur sont remis à leur base	
initializePreys(GameMYR game)	void	initialisation aléatoire des proies sur le plateau principal	

Méthode	Type de retour	Description	Référence
initializePrey(GameMYR game, string type, array position)	void	initialisation d'une seule proie	
initializePlayerData(PlayerMYR player, string color)	void	initialisation des données du joueur lors du démarrage d'une partie	
initializePlayer WorkshopActions(PlayerMYR player)	void	initialisation des actions de l'atelier par rapport au joueur donné	
initializePlayerResources(PlayerMYR player)	void	initialisation des ressources du joueur, par défaut la quantité est à 0	
initializeNurse(PlayerMYR player)	void	initialisation des nourrices pour le joueur sélectionné	
initializeWorker(PlayerMYR player)	void	initialisation une nouvelle ouvrière pour le joueur sélectionné	
initializeAnthillHole ForPlayer(PlayerMYR player, array position)	void	initialisation d'un nouveau trou de fourmi pour le joueur sélectionné	
initializeColorForPlayer(PlayerMYR player, string color)	void	modifie la couleur du joueur donné	
initializeEventBonus(GameMYR game)	void	initialisation du bonus de saison pour chaque joueur participant à la partie	
initializeMainBoardTiles(GameMYR game)	void	initialisation du plateau principal en fonction du nombre de joueurs	
manageTurnOfPlayerNewPhase(Collection players, int phase)	void	modifier le tour des joueurs en fonction de la phase donnée	
getOrderOfPlayers(GameMYR game)	Collection	retourne les joueurs du jeu dans l'ordre du tour de jeu	
canDoGoal(PlayerMYR playerMYR, GameGoalMYR goalMYR)	bool	vérifie si le joueur donné a accompli l'objectif de niveau inférieur ou n'a pas réalisé l'objectif donné	
computePlayerRewardPoints WithGoal(PlayerMYR playerMYR, GameGoalMYR goalMYR)	void	consulte et donne les points associé à l'objectif donné	
endRoundOfFirstPlayer(GameMYR game)	void	le rôle du premier joueur est passé au joueur suivant	
endSeason(GameMYR game)	void	réalise une fin de saison actuel ou fin d'une année	
initializeNewYear(GameMYR game)	void	initialise une nouvelle année	
initializeGoals(GameMYR game)	void	initialise toutes les objectifs de la partie donnée	
createGameGoal(GoalMYR game, MainBoardMYR mainBoard)	void	crée un objectif de jeu	
clearSeasons(GameMYR game)	void	réinitialise les saisons lors d'une nouvelle année	

Méthode	Type de retour	Description	Référence
resetWorkshopActions(PlayerMYR player)	void	réinitialise les actions de l'atelier du joueur donné	
canPlayersStillDoWorkerPhase Phase(GameMYR game)	bool	vérifie si au moins un joueur peut réaliser la phase ouvrière	
setPheromonesHarvestedIf NoResourcesOnIt(Collection players)	void	récolte toutes les phéromones des joueurs si ces dernières n'ont pas de ressources	
makePheromonesHarvestable(PlayerMYR player)	void	faire des phéromones appartenant au joueur donné un composant récoltable	

Sémantique des codes de retour :

- -1 (error) : code d'erreur
- 1 (success) : code de succès
- Pour chooseNurse 2 (noCraft) : ne peut pas réaliser une fabrication
- Pour chooseNurse 3 (goal) : le joueur va réaliser un objectif
- Pour chooseNurse 4 (anthillHole) : le joueur va réaliser un trou de fourmilière
- Pour doEndOfTurnForPlayer 5 (endOfGame) : la partie se termine

4.2.6 WinterMYRService

4.2.6.1 Structure

 WinterMYRService
<ul style="list-style-type: none">• bool mustDropResourcesForWinter(PlayerMYR player)• bool canManageEndOfWinter(GameMYR game)• bool canSetPhaseToWinter(GameMYR game)• removeCubeOfWarehouse(PlayerMYR player, PlayerResourceMYR playerResource)• retrievePoints(PlayerMYR player)• manageEndOfWinter(GameMYR game)• beginWinter(GameMYR game)

4.2.6.2 Description des composants


Nom de la classe : WinterMYRService

Description : Cette classe permet d'implanter les fonctionnalités du Myrmes concernant la phase hiver.

Méthode	Type de retour	Description	Référence
mustDropResourcesForWinter(PlayerMYR player)	bool	indique si le joueur doit jeter des ressources pour l'hiver	
canManageEndOfWinter(GameMYR game)	bool	indique si tous les joueurs ont jeté leurs ressources en trop	
canSetPhaseToWinter(GameMYR game)	bool	indique si la partie doit passer en phase hiver	
removeCubeOfWarehouse(PlayerMYR player, PlayerResourceMYR playerResource)	void	retire un cube de ressource de l'entrepôt du joueur	
retrievePoints(PlayerMYR player)	void	retire la nourriture et les points, si nécessaire, du joueur en fonction de l'année	
manageEndOfWinter(GameMYR game)	void	pour chaque joueur de la partie retire leurs nourritures et points et gère la fin de manche	
beginWinter(GameMYR game)	void	commence la phase hiver. Si aucun joueur ne doit jeter de ressources, la fin de l'hiver est gérée automatiquement	

4.2.7 WorkerMYRService

4.2.7.1 Structure

 WorkerMYRService
<ul style="list-style-type: none"> ● int getNumberOfGardenWorkerOfPlayer(PlayerMYR player) ● ArrayCollection getAvailablePheromones(PlayerMYR player) ● ArrayCollection getAllAvailablePositions(PlayerMYR player, TileMYR tile, TileTypeMYR tileType, int antCoordX, int antCoordY, array availableTiles) ● bool canWorkerMove(PlayerMYR player, GardenWorkerMYR gardenWorker, int direction) ● int getNeededSoldiers(int coordX, int coordY, GameMYR game, PlayerMYR player, array cleanedTiles) ● bool isPreyOnTile(int coordX, int coordY, GameMYR game, array cleanedTiles) ● int getNeededMovementPoints(int coordX1, int coordY1, int coordX2, int coordY2, GameMYR game) ● bool isValidPositionForAnt(?TileMYR tile) ● bool canCleanPheromone(PheromonMYR pheromone, int playerDirtQuantity) ● ?PheromonMYR getPheromoneFromTile(Game game, TileMYR tile) ● string getStringCoordsOfPheromoneTiles(PheromonMYR pheromone) ● ArrayCollection getAllCoordinatesFromTileType(PlayerMYR player, TileMYR tile, TileTypeMYR tileType, array \$availablePositions = []) ● int getPlayerMovementPoints(PlayerMYR player) ● cleanPheromone(PheromonMYR pheromone, PlayerMYR player) ● placeAntInAnthill(PersonalBoardMYR personalBoard, int anthillFloor, ?String lvlTwoResource) ● takeOutAnt(PersonalBoardMYR personalBoard, AnthillHoleMYR exitHole) ● placeAnthillHole(PlayerMYR player, TileMYR tile) ● bool canPlacePheromone(PlayerMYR player, TileMYR tile, int antCoordX, int antCoordY, TileTypeMYR tileType, array \$availablePositions = []) ● placePheromone(PlayerMYR player, TileMYR tile, TileTypeMYR tileType, int antCoordX, int antCoordY) ● killPlayerGardenWorker(PlayerMYR player) ● workerMove(PlayerMYR player, GardenWorkerMYR gardenWorker, int direction) ● ?TileMYR getTileFromCoordinates(int coordX, int coordY) ● ArrayCollection getPheromonesFromListOfids(array pheromonelds) ● ArrayCollection getPlayerPheromones(PlayerMYR playerMYR)

4.2.7.2 Description des composants

Nom de la classe : WorkerMYRService

Description : Cette classe permet d'implanter les fonctionnalités du Myrmes concernant la phase ouvrière.


Méthode	Type de retour	Description	Référence
getNumberOfGardenWorkerOfPlayer(PlayerMYR player)	int	retourne le nombre d'ouvrières du jardin du joueur	
getAvailablePheromones(PlayerMYR player)	ArrayCollection	retourne un triplet (type, quantité, nb orientations) des phéromones disponibles pour le joueur	
getAllAvailablePositions(PlayerMYR player, TileMYR tile TileTypeMYR tileType int antCoordX int antCoordY array availableTiles	ArrayCollection	retourne toutes les positions disponibles pour une phéromone d'un certain type et orientation	
canWorkerMove(PlayerMYR player, GardenWorkerMYR gardenWorker, int direction)	bool	indique si l'ouvrière peut se déplacer sur la tuile dans la direction demandée	

Méthode	Type de retour	Description	Référence
getNeededSoldiers(int coordX, int coordY, GameMYR game, PlayerMYR player, array cleanedTiles)	int	retourne le nombre de soldates nécessaires pour se déplacer sur la nouvelle tuile	
isPreyOnTile(int coordX, int coordY, GameMYR game, array cleanedTiles)	bool	indique si une proie se trouve sur la tuile	
getNeededMovementPoints(int coordX1, int coordY1, int coordX2, int coordY2, GameMYR game)	int	retourne le nombre de points de déplacement nécessaires pour se déplacer sur la nouvelle tuile	
isValidPositionForAnt(?TileMYR tile)	bool	indique si la tuile peut contenir une fourmi	
canCleanPheromone(PheromonMYR pheromone, int playerDirtQuantity)	bool	indique si le joueur peut nettoyer la phéromone avec cette quantité de terre	
getPheromoneFromTile(GameMYR game, TileMYR tile)	?PheromonMYR	renvoie l'éventuelle phéromone posée sur la tuile	
getStringCoordsOfPheromoneTiles(PheromonMYR pheromone)	string	renvoie une chaîne représentant les coordonnées de chaque tuile recouverte par la phéromone	
getAllCoordinatesFromTileType(PlayerMYR player, TileMYR tile, TileTypeMYR tileType, array availablePositions = [])	ArrayCollection	renvoie toutes les tuiles qu'une phéromone recouvrirait	
getPlayerMovementPoints(PlayerMYR player)	int	renvoie le nombre de points de déplacement du joueur	
cleanPheromone(PheromonMYR pheromone, PlayerMYR player)	void	le joueur nettoie la phéromone	
placeAntInAnthill(PersonalBoardMYR personalBoard, int anthillFloor, ?String lvlTwoResource)	void	le joueur place une fourmi dans sa fourmilière au niveau demandé	
takeOutAnt(PersonalBoardMYR personalBoard, AnthillHoleMYR anthillHole)	void	le joueur sort sa fourmi dans le jardin et la transforme en ouvrière du jardin	
placeAnthillHole(PlayerMYR player, TileMYR tile)	void	place un nouveau trou de fourmi pour le joueur	
canPlacePheromone(PlayerMYR player, TileMYR tile int antCoordX, int antCoordY TileTypeMYR tileType, array availablePositions = [])	bool	indique si le joueur peut placer une phéromone dans la position demandée	

Méthode	Type de retour	Description	Référence
placePheromone(PlayerMYR player, TileMYR tile, TileTypeMYR tileType, int antCoordX, int antCoordY)	void	le joueur place une phéromone dans la position demandée	
killPlayerGardenWorker(PlayerMYR player)	void	tue l'ouvrière du jardin du joueur	
workerMove(PlayerMYR player, GardenWorkerMYR gardenWorker int direction)	void	déplace l'ouvrière sur le plateau principal	
getTileFromCoordinates(int coordX, int coordY)	?TileMYR	renvoie l'éventuelle tuile aux coordonnées données	
getPheromonesFromListOfIds(array pheromoneIds)	ArrayCollection	renvoie toutes les phéromones dont l'id se trouve dans le tableau	
getPlayerPheromones(PlayerMYR player)	ArrayCollection	renvoie toutes les phéromones du joueur	

4.2.8 WorkshopMYRService

4.2.8.1 Structure

 WorkshopMYRService
<ul style="list-style-type: none">● ArrayCollection getAvailableAnthillHolesPositions(PlayerMYR player)● ?AnthillHoleMYR getAnthillHoleFromTile(TileMYR tile, GameMYR game)● bool canPlayerDoGoal(PlayerMYR player, GameGoalMYR gameGoal)● doGoal(PlayerMYR player, GameGoalMYR gameGoalMYR, NurseMYR nurse)● doStoneOrDirtGoal(PlayerMYR player, GameGoalMYR gameGoal, NurseMYR nurse, int stoneQuantity, int dirtQuantity)● doSpecialTileGoal(PlayerMYR player, GameGoalMYR gameGoal, NurseMYR nurse, Collection specialTiles)● doPheromoneGoal(PlayerMYR player, GameGoalMYR gameGoal, NurseMYR nurse, Collection pheromones)● manageWorkshop(PlayerMYR player, int selectedCraft, TileMYR tile = null)● ArrayCollection playerAvailableGoals(PlayerMYR player, GameMYR game)● ?PlayerResourceMYR getPlayerResourcesFromSelectedType(PlayerMYR player, string selectedType)● Collection getSpecialTilesOfPlayer(PlayerMYR player)

4.2.8.2 Description des composants

Nom de la classe : WorkshopMYRService

Description : Cette classe permet d'implanter les fonctionnalités du Myrmes concernant la phase atelier.

Méthode	Type de retour	Description	Référence
getAvailableAnthillHolesPositions (PlayerMYR player)	ArrayCollection	Renvoie toutes les positions possibles pour un nouveau trou de fourmi pour le joueur	
getAnthillHoleFromTile (TileMYR tile, GameMYR game)	?AnthillHoleMYR	Renvoie l'éventuel trou de fourmi posé sur la tuile	
canPlayerDoGoal (PlayerMYR player, GameGoalMYR gameGoal)	bool	Indique si le joueur peut accomplir l'objectif demandé	
doGoal (PlayerMYR player, GameGoalMYR gameGoalMYR, NurseMYR nurse)	void	Le joueur réalise l'objectif	
doStoneOrDirtGoal (PlayerMYR player, GameGoalMYR gameGoal, NurseMYR nurse, int stoneQuantity, int dirtQuantity)	void	Le joueur réalise l'objectif de terre ou pierre	
doSpecialTileGoal (PlayerMYR player, GameGoalMYR gameGoal, NurseMYR nurse, Collection specialTiles)	void	Le joueur réalise l'objectif de tuile spéciale	
doPheromoneGoal (PlayerMYR player, GameGoalMYR gameGoal, NurseMYR nurse, Collection pheromones)	void	Le joueur réalise l'objectif de phéromone	
manageWorkshop (PlayerMYR player, int selectedCraft, TileMYR tile = null)	void	Réalise les différentes actions en fonction des nourrices posées dans l'atelier	
playerAvailableGoals (PlayerMYR player, GameMYR game)	ArrayCollection	Renvoie la liste de tous les objectifs atteignables pour le joueur	
getPlayerResourcesFrom SelectedType(PlayerMYR player, string selectedType)	?PlayerResourceMYR	Renvoie l'entité représentant la ressource du joueur demandée	
getSpecialTilesOfPlayer (PlayerMYR player)	Collection	Renvoie une liste de toutes les tuiles spéciales du joueur	

4.3 Les contrôleurs

4.3.1 Structure

Les contrôleurs permettront aux utilisateurs d'accéder aux fonctionnalités de l'application

 MyrController
<ul style="list-style-type: none">● <code>HTMLResponse changeSeasonBonus(GameMyr game, int newValue)</code>● <code>HTMLResponse placeNurse(GameMyr game, int oldPosition, int newPosition)</code>● <code>HTMLResponse takeOutAnt(GameMyr game)</code>● <code>HTMLResponse sendAntToAnthill(GameMyr game, int anthillLevel)</code>● <code>HTMLResponse moveGardenAnt(GameMyr game, TileMyr oldTile, TileMyr newTile)</code>● <code>HTMLResponse placePheromon(GameMyr game, TileMyr tile, int pheromonType)</code>● <code>HTMLResponse placeSpecialTile(GameMyr game, TileMyr tile, int specialTileType)</code>● <code>HTMLResponse clearPheromon(GameMyr game, TileMyr tile)</code>● <code>HTMLResponse endRoundWorkerAnt(GameMyr game, AnthillWorkerMyr ant)</code>● <code>HTMLResponse recoltPheromon(GameMyr game, TileMyr tile)</code>● <code>HTMLResponse chooseNurse(GameMyr game, NurseMyr nurse)</code>● <code>HTMLResponse placeAnthillHole(GameMyr game, TileMyr tile)</code>● <code>HTMLResponse doGameGoal(GameMyr game, GameGoalMyr goal)</code>● <code>HTMLResponse removeResource(GameMyr game, ResourceMyr resource)</code>

4.3.2 Description des composants

Nom de la classe : MyrController

Description : Cette classe permet d'implanter les fonctionnalités du Myrmes.

Méthode	Type de retour	Description	Référence
changeSeasonBonus(GameMyr game, int newValue)	HTMLResponse	Change le bonus de saison du joueur accédant à la route dans la partie game à la valeur newValue si possible	STB-GES-MYR-10
placeNurse(GameMyr game, int oldPosition, int newPosition)	HTMLResponse	Déplace la nourrice du joueur accédant à la route dans la partie game de oldPosition vers newPosition	STB-GES-MYR-11, 29
takeOutAnt(GameMyr game)	HTMLResponse	Déplace la fourmi du joueur accédant à la route dans la partie game sur le plateau central	STB-GES-MYR-12, 14
sendAntToAntHill(GameMyr game, int anthillLevel)	HTMLResponse	Déplace la fourmi du joueur accédant à la route dans la partie game dans la fourmilière pour travailler au niveau anthillLevel	STB-GES-MYR-13, 27
moveGardenAnt(GameMyr game, TileMyr oldTile, TileMyr newTile)	HTMLResponse	Déplace la fourmi du joueur sur le plateau central accédant à la route dans la partie game de oldTile vers newTile	STB-GES-MYR-14, 15, 16, 17
placePheromon(GameMyr game, TileMyr tile, int pheromonType)	HTMLResponse	Place une phéromone pour le joueur accédant à la route dans la partie game sur tile avec l'orientation pheromonType	STB-GES-MYR-18, 19, 20
placeSpecialTile(GameMyr game, TileMyr tile, int specialTileType)	HTMLResponse	Place une tuile spéciale pour le joueur accédant à la route dans la partie game sur tile avec l'orientation specialTileType	STB-GES-MYR-18, 18, 21
clearPheromon(GameMyr game, TileMyr tile)	HTMLResponse	Efface la phéromone pour le joueur accédant à la route dans la partie game présente sur la tuile tile	STB-GES-MYR-22, 23, 24
endRoundWorkerAnt(GameMyr game, AnthillWorkerMyr ant)	HTMLResponse	Met fin au déplacement de la fourmi ant du joueur accédant à la route dans la partie game	STB-GES-MYR-14
recoltPheromon(GameMyr game, TileMyr tile)	HTMLResponse	Récolte les ressources présentes sur la phéromone à l'emplacement tile pour le joueur accédant à la route	STB-GES-MYR-25
chooseNurse(GameMyr game, NurseMyr nurse)	HTMLResponse	Permet de choisir la nourrice nurse pour réaliser une fabrication pour le joueur accédant à la route	STB-GES-MYR-26
placeAnthillHole(GameMyr game, TileMyr tile)	HTMLResponse	Permet de placer un trou de fourmilière à la position tile pour le joueur accédant à la route	STB-GES-MYR-04
doGameGoal(GameMyr game, GameGoalMyr goal)	HTMLResponse	Permet de réaliser l'objectif goal si le joueur possède les ressources pour le joueur accédant à la route	STB-GES-MYR-34, 35
removeResource(GameMyr game, ResourceMyr resource)	HTMLResponse	Permet au joueur de choisir une ressource à enlever de son inventaire pour le joueur accédant à la route	STB-GES-MYR-27

5 Architecture dynamique

Diagramme de séquence avec l'action "Le joueur change son marqueur de bonus"

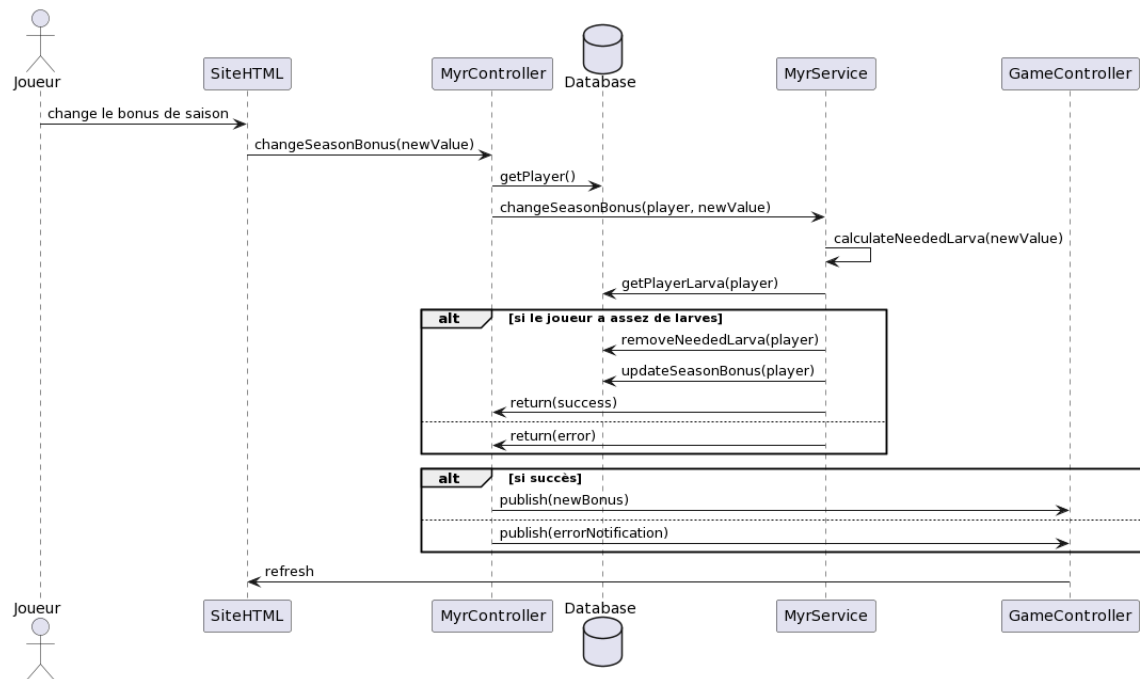


Diagramme de séquence avec l'action "Le joueur place une nourrice"

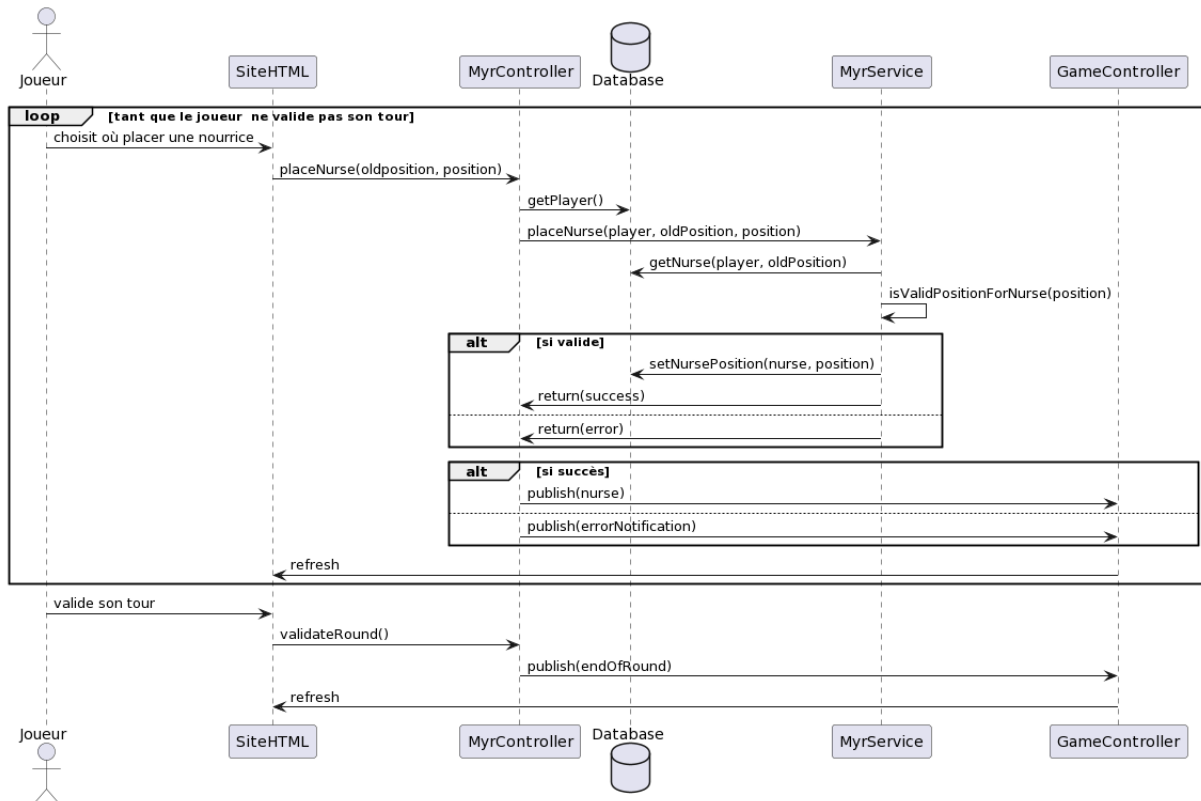


Diagramme de séquence avec l'action "Le joueur sort sa fourmi"

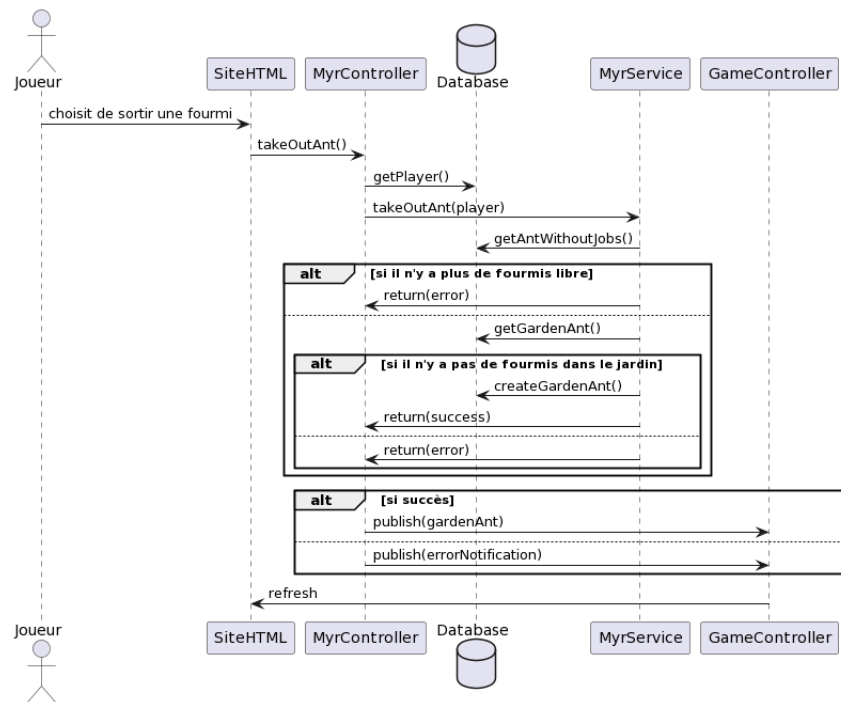


Diagramme de séquence avec l'action "Le joueur envoie une fourmi dans sa fourmière"

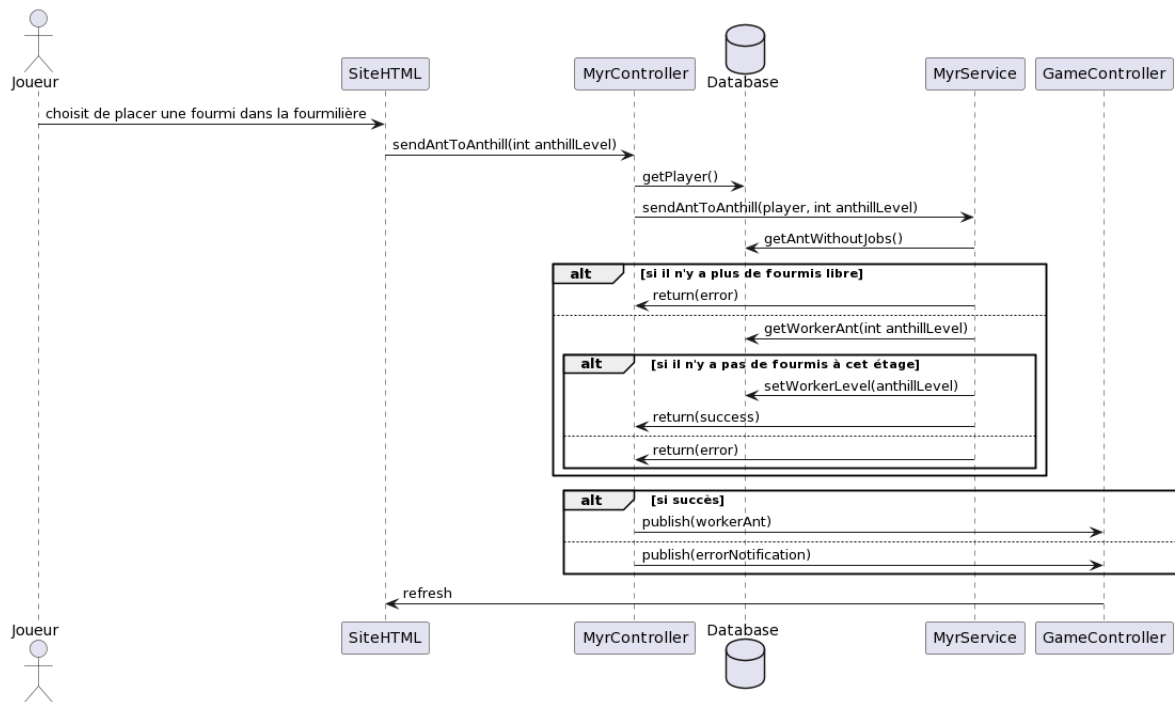


Diagramme de séquence avec l'action "Le joueur choisit de déplacer la fourmi dans le jardin"

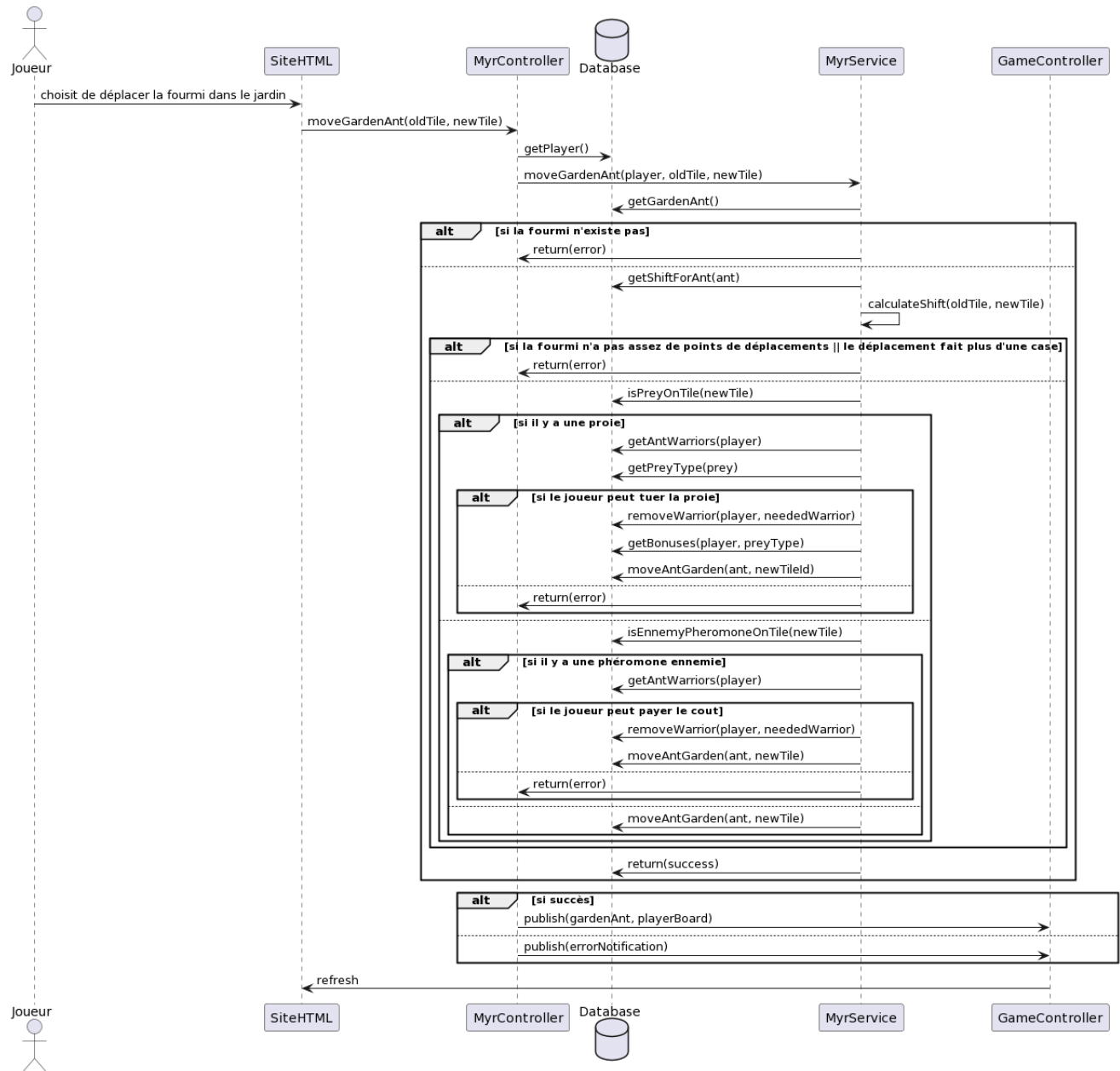


Diagramme de séquence avec l'action "Le joueur choisit de placer une phéromone"

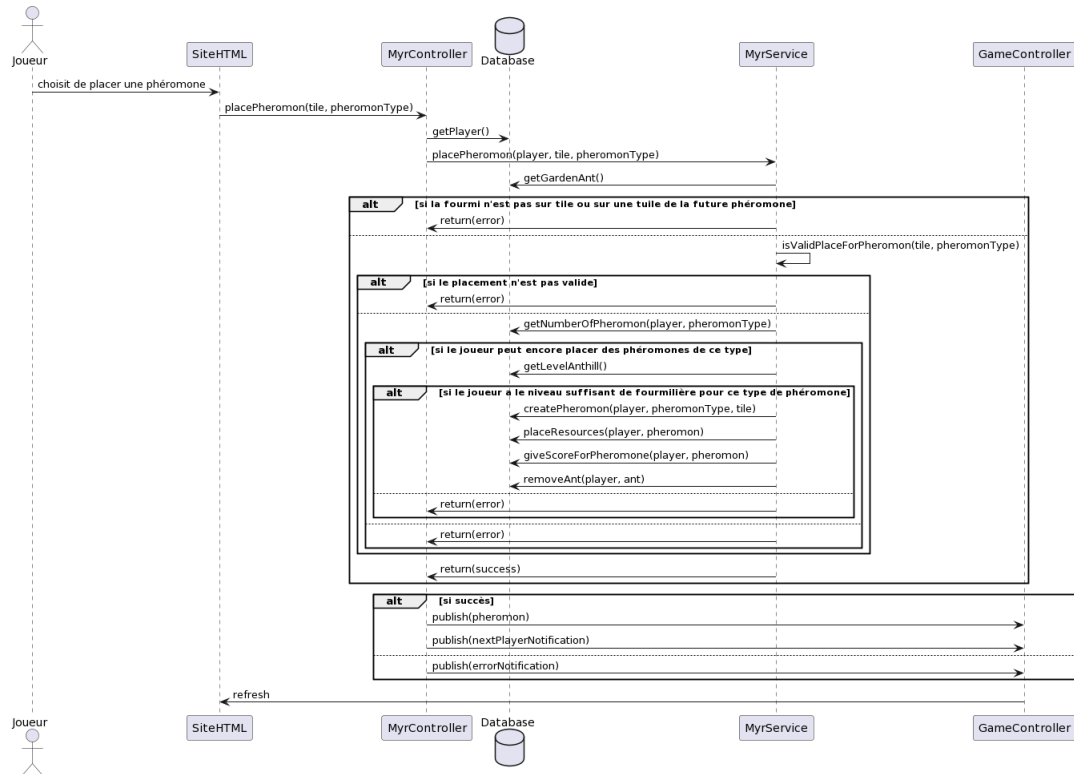


Diagramme de séquence avec l'action "Le joueur choisit de placer une tuile spéciale"

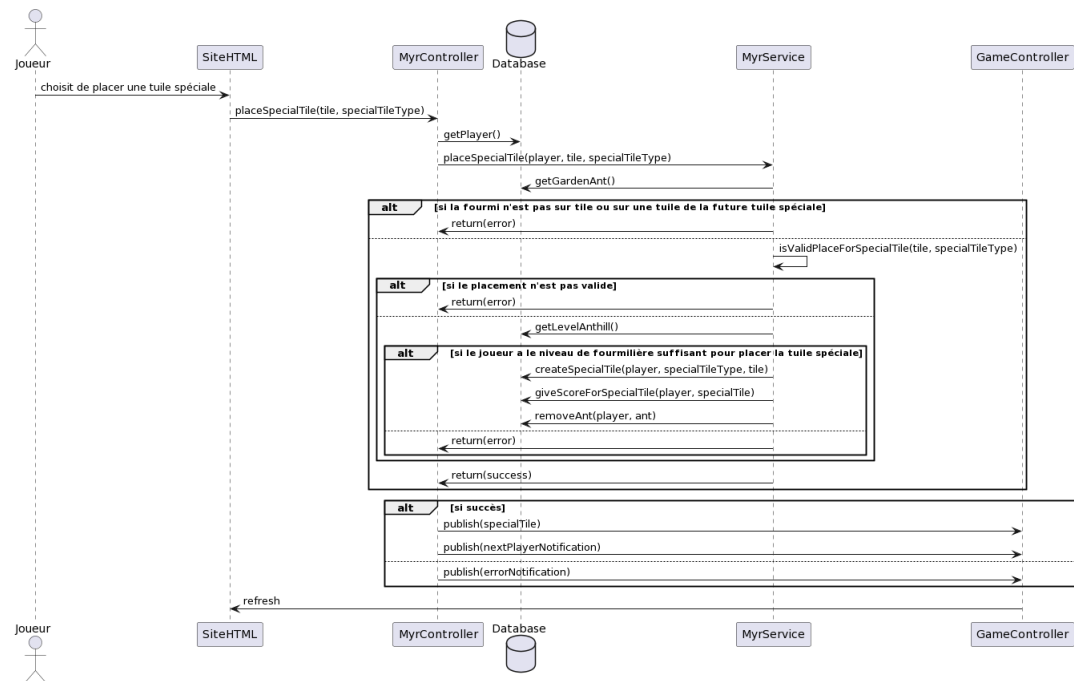


Diagramme de séquence avec l'action "Le joueur choisit de nettoyer une phéromone"

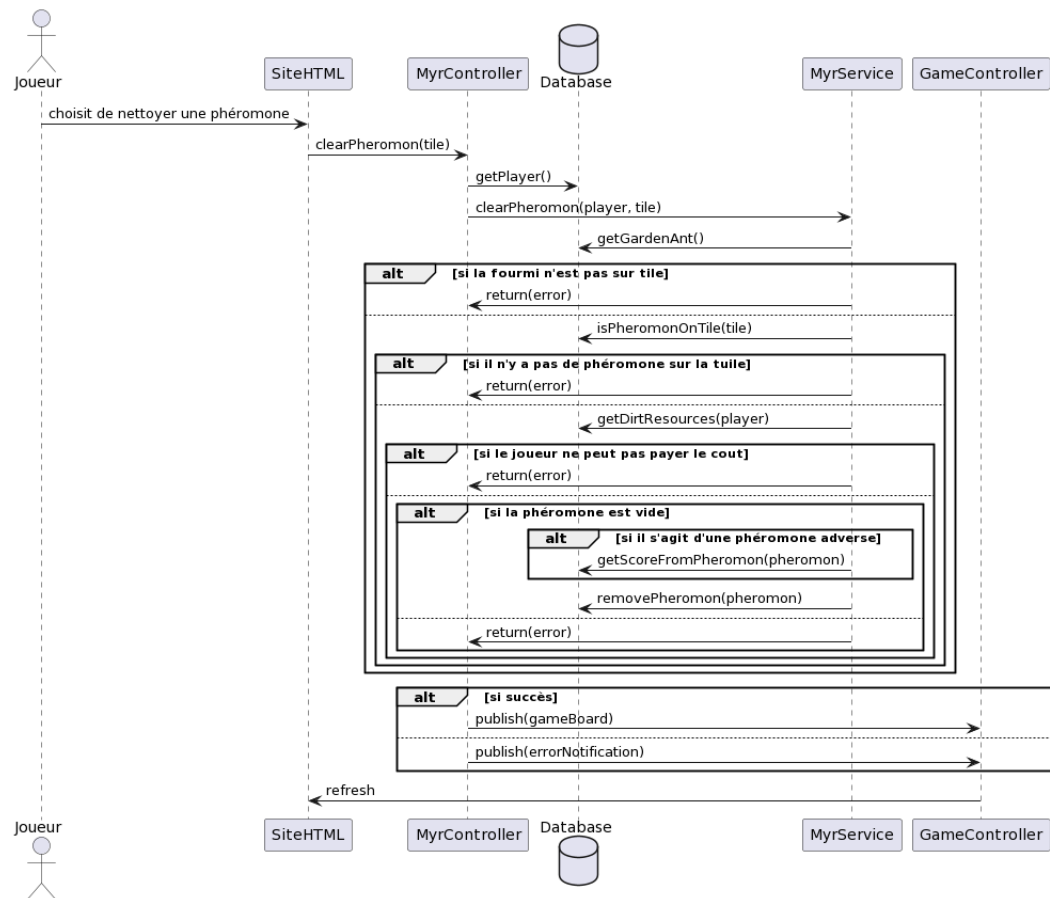


Diagramme de séquence avec l'action "Le joueur choisit de finir le déplacement de son ouvrière"

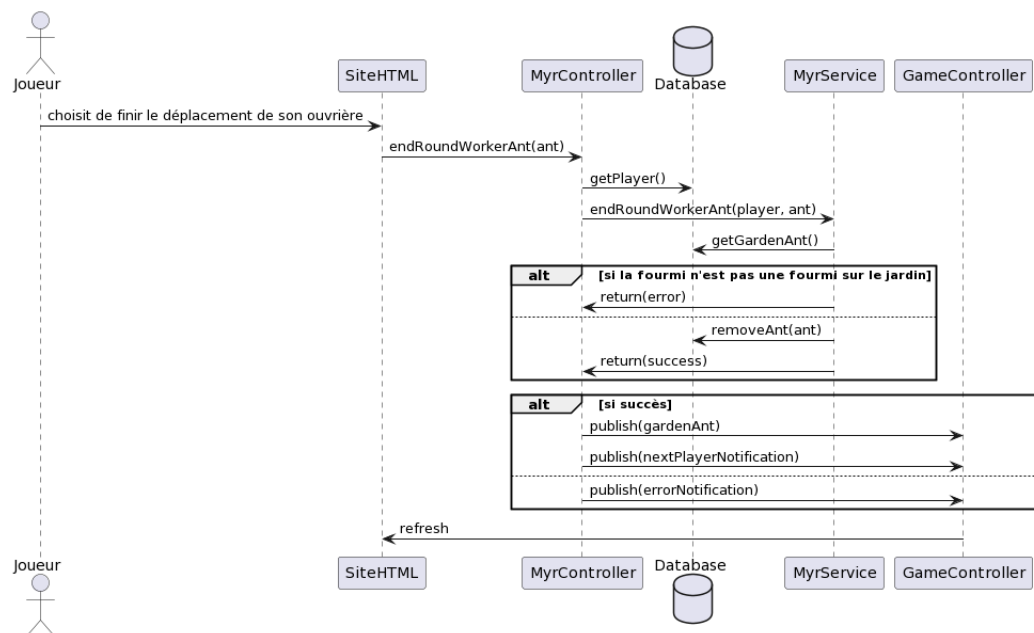


Diagramme de séquence avec l'action "Le joueur récolte une phéromone"

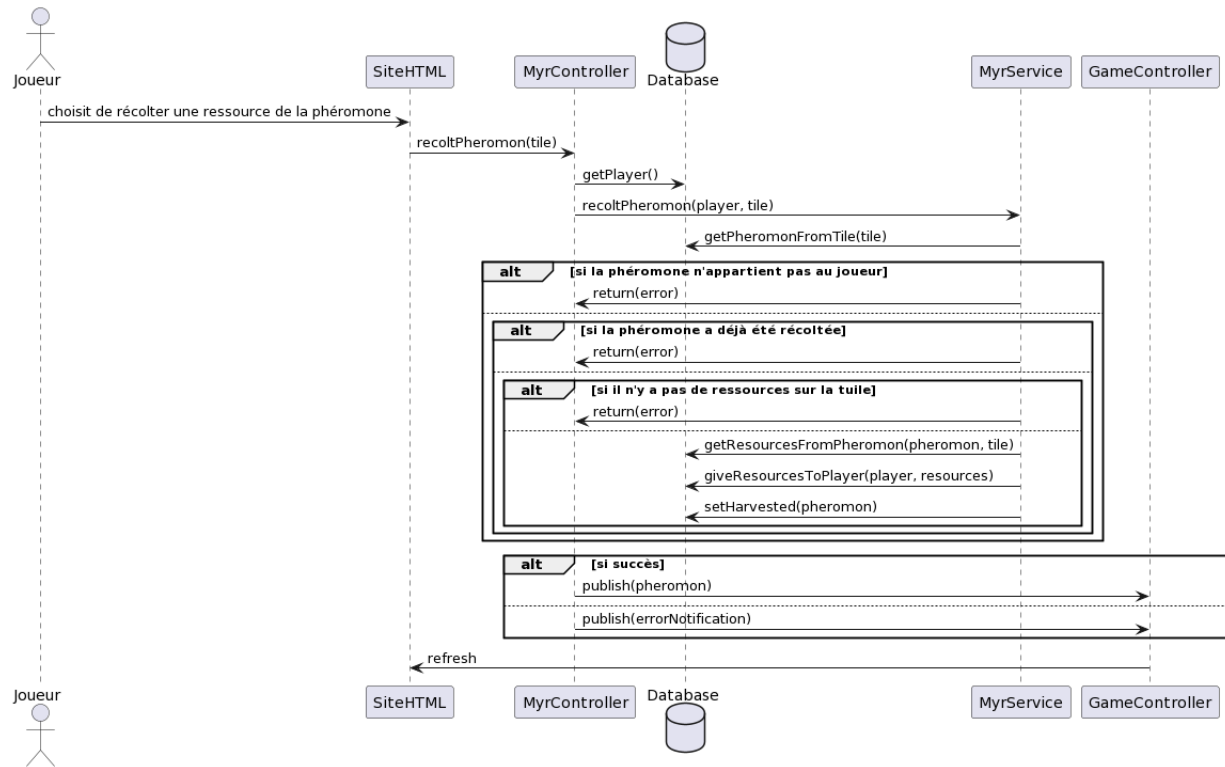


Diagramme de séquence avec l'action "Le joueur choisit la première nourrice à utiliser pour fabriquer"

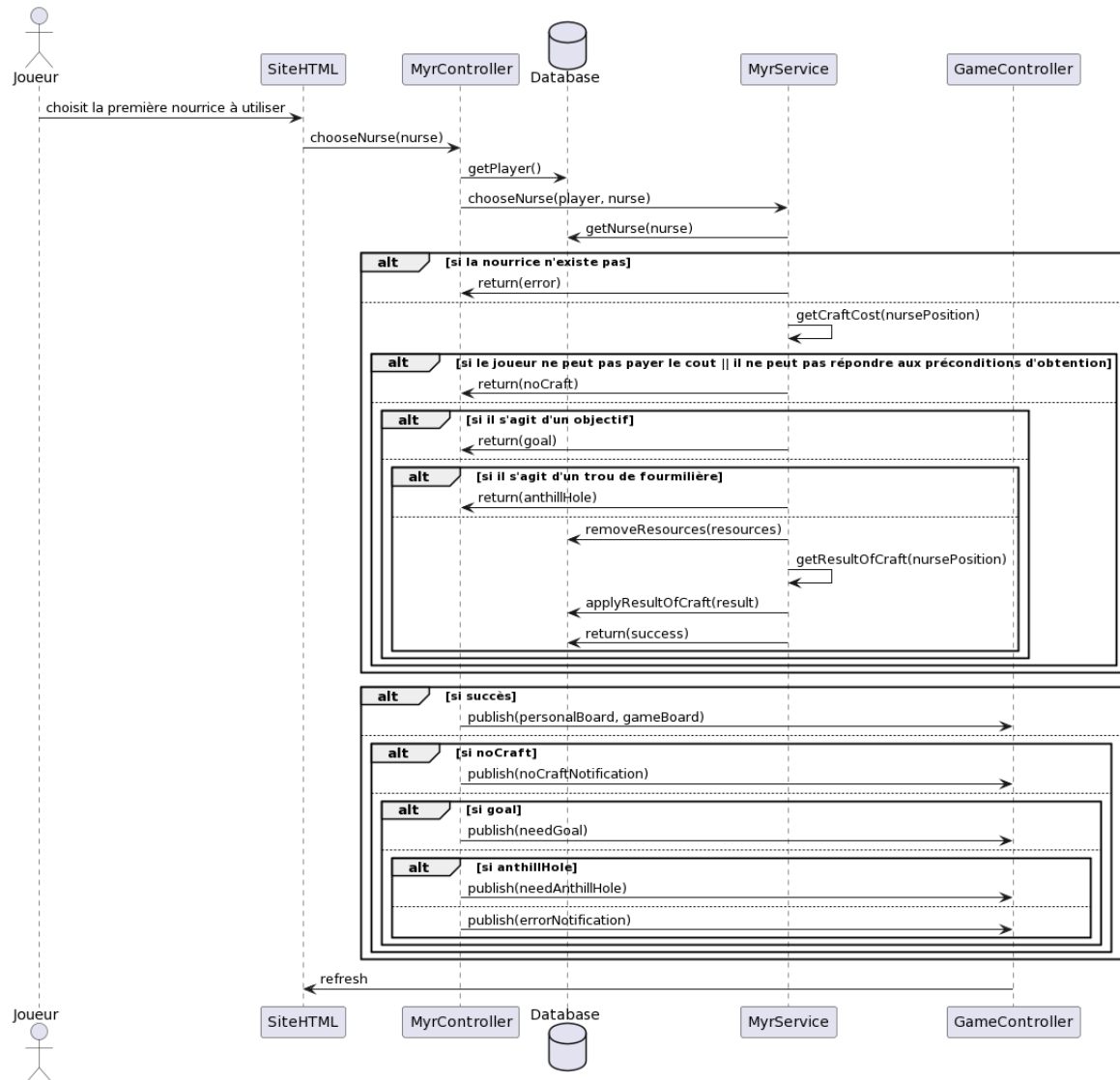


Diagramme de séquence avec l'action "Le joueur crée un trou de fourmilière"

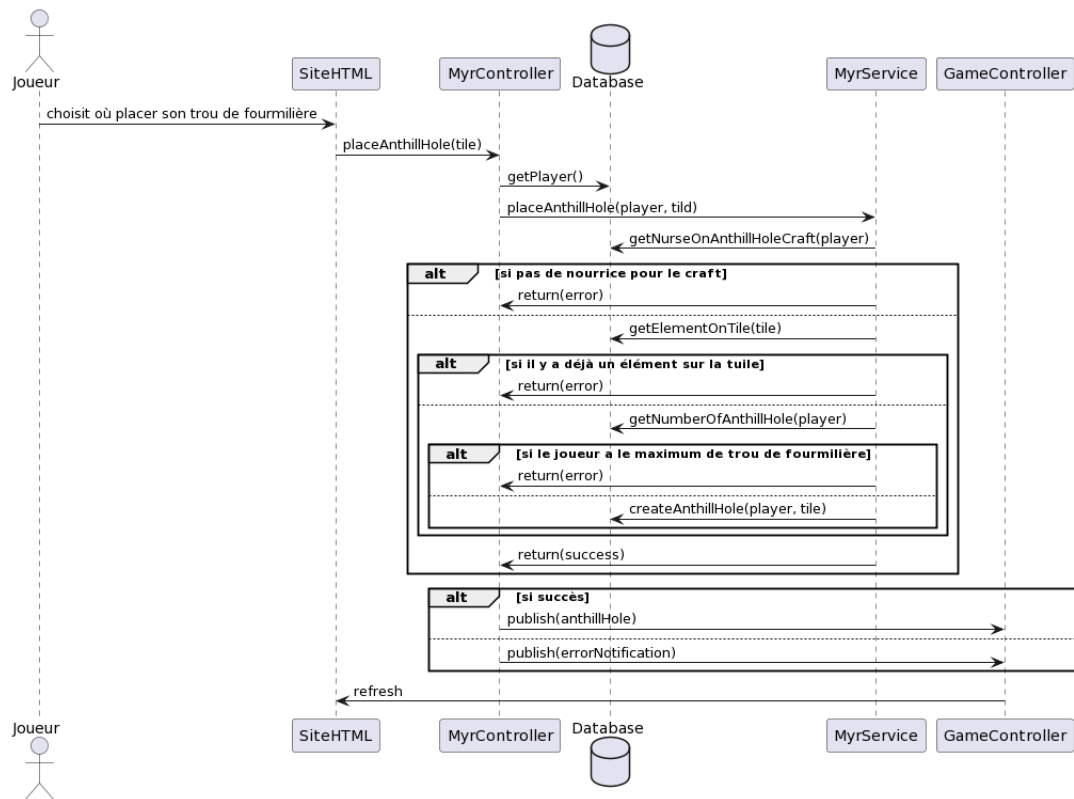


Diagramme de séquence avec l'action "Le joueur choisit un objectif"

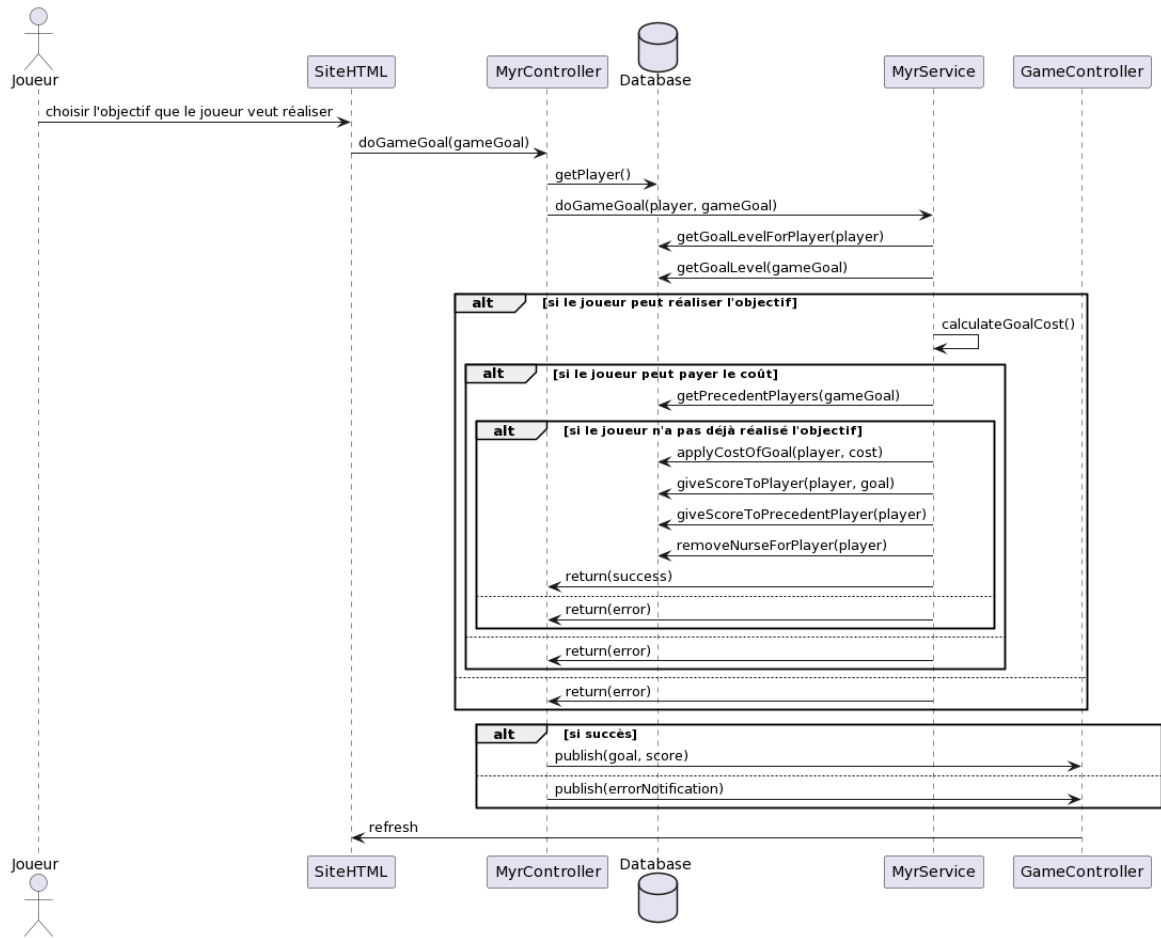
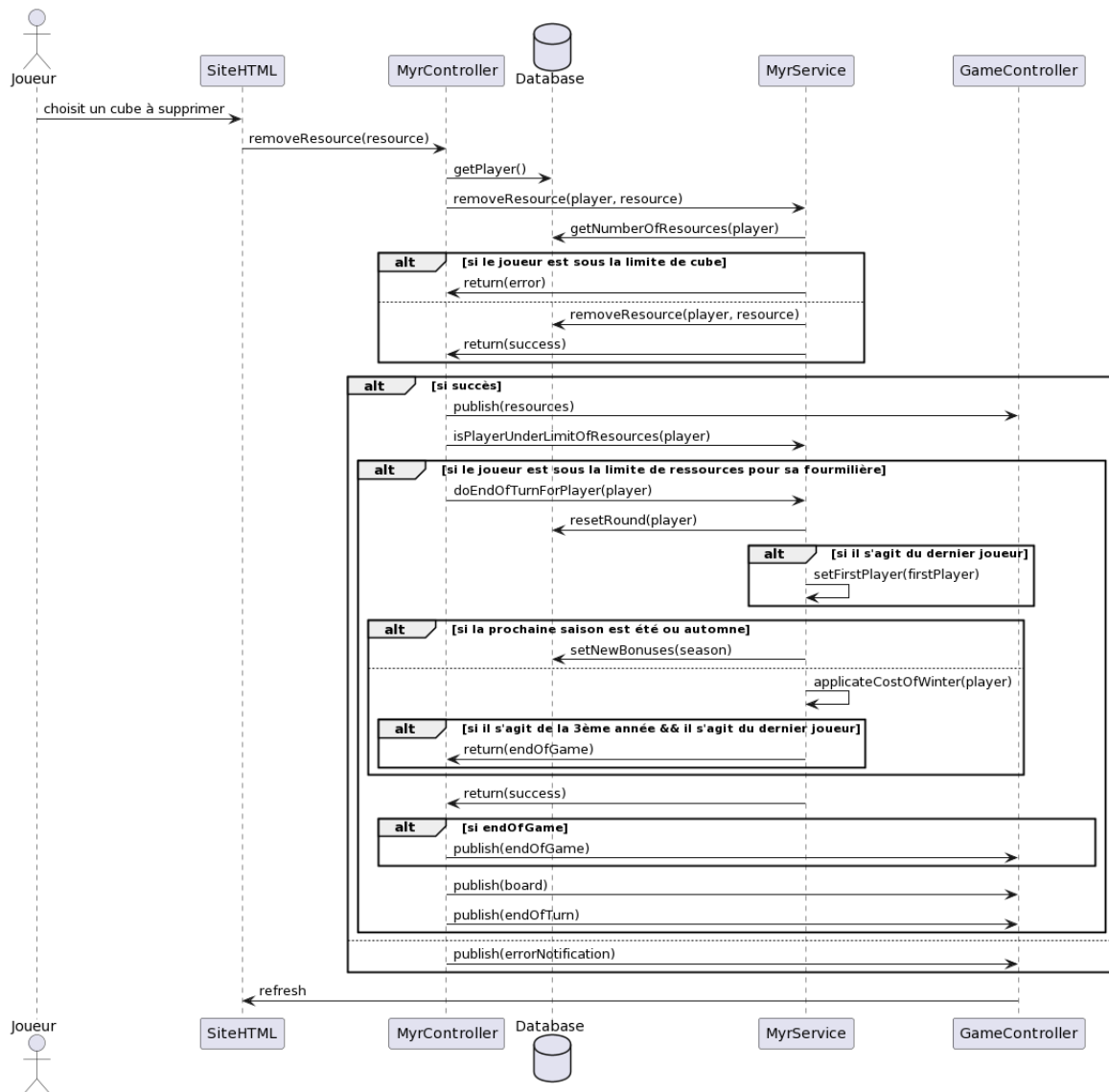


Diagramme de séquence avec l'action "Le joueur choisit un cube à supprimer"



6 Maquette de vue

Nous avons également réalisé une première version d'une maquette de vue, notamment pour la représentation générale du plateau. Celle-ci sert notamment à avoir une idée de la disposition des différents éléments, et est amenée à être affinée et améliorée au cours de l'avancée du projet. Il s'agit donc ici d'une version minimaliste :

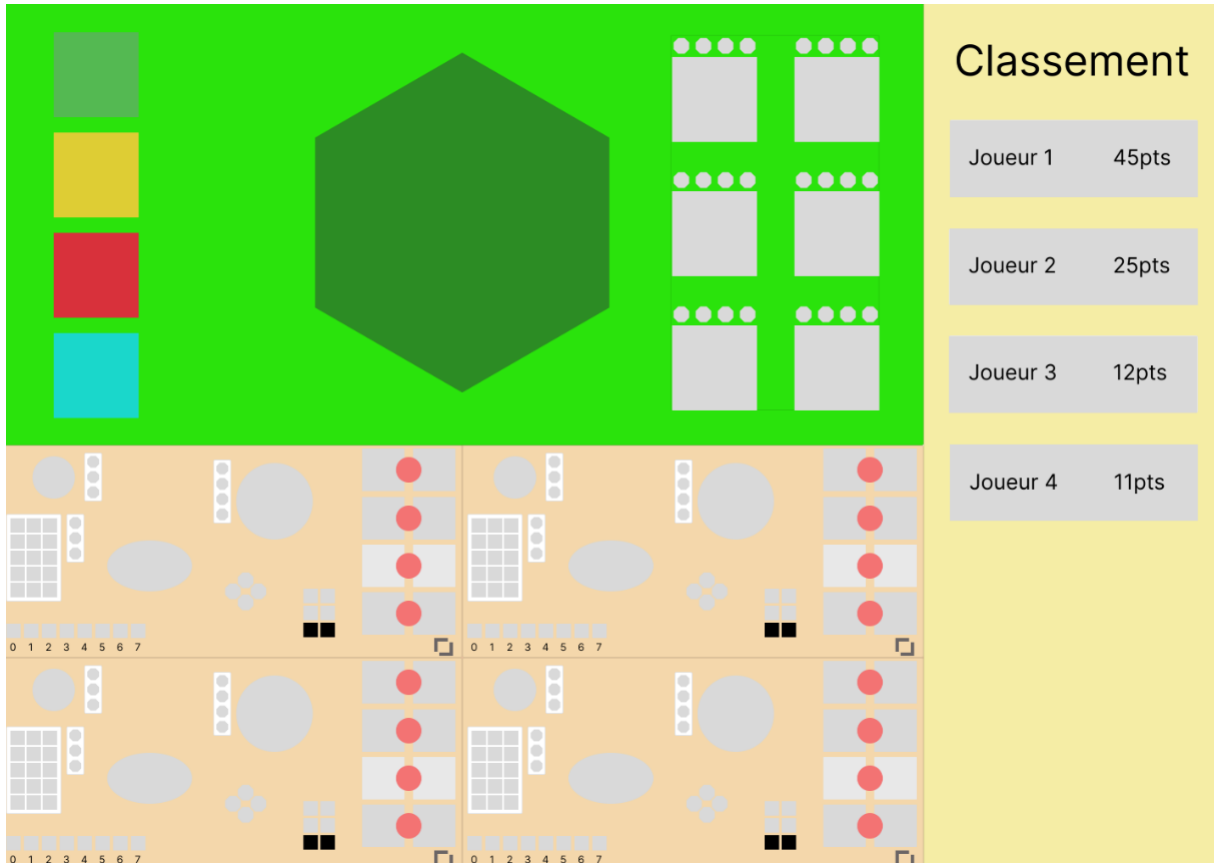


FIGURE 1 – Première version de la maquette de la vue du plateau principal du jeu 6 qui prend

On peut retrouver sur ce plateau différentes zone, comme le jardin (le plateau principal) qui est la zone verte en haut à gauche.

En bas se trouvent les plateaux des joueurs avec leurs nourricières, bonus, soldats, atelier, niveau de fourmilière etc. Il est possible d'agrandir chacun des plateaux, toutefois à certains moments du jeu, le visuel des plateaux sera masqué.

On peut également voir un classement des différents joueurs en fonction de leur nombre de points de victoire.