

NEED YOUR LOVELY HELP

Sous-titre : Quel modèle choisir ?

Glossaire

Planification

En sport, la « *planification* » c'est le fait de définir **SI** et **QUAND** on va faire **QUOI**. Plus précisément, c'est donc créer un (rétro-)planning pour la (ou les) saisons à venir pour - principalement- déterminer les échéances. Planifier c'est établir les objectifs (et voir s'ils peuvent être atteint).

Par exemple : il y a une compétition le 20 février 2024 (ça, c'est un fait), il faut donc que le gymnaste puisse réaliser ses deux séries (imposé + libre) le 20 février ; mais dans un monde parfait, il faudrait qu'il sache faire ses séries 3 à 4 semaines (disons le 20 janvier pour l'exemple) avant afin d'être prêt le jour J. Et, dans une de ses séries il doit faire un mouvement X qu'aujourd'hui il ne sait pas encore faire. Et comme il devrait savoir faire ses séries à partir du 20 janvier, il faut donc qu'il sache faire X pour le 20 janvier aussi au plus tard. Il faudrait même qu'il sache le faire avant puisque le 20 janvier il devrait pouvoir l'enchaîner. Il faudrait donc qu'il sache le faire le 20 octobre (toujours pour l'exemple), ...

Programmation

La « *programmation* » c'est définir **COMMENT** on va faire le **QUOI** défini dans la planification. Quand on sait pour quelles échéances telles ou telles choses (série, mouvement, ...) doivent être pouvoir réalisées, on va maintenant réfléchir à comment arriver à tenir ses échéances : comment (et quand) nous allons travailler quoi par rapport au gymnaste. En gros, on établit un plan de bataille en tenant compte d'où le gymnaste en est maintenant, où il devrait arriver quand, ...

Programmer c'est le chemin qu'on choisit pour atteindre l'objectif choisi. Poussé à son extrême, ça finit par définir les passages à faire pour l'élève X à la date Y.

Les deux termes (planification et programmation) sont difficilement totalement dissociables, il y a dans la pratique un recouvrement des deux notions.

Passage (class Passe)

Moment de travail lors d'un entraînement (i.e. entraînement = ensemble de passages).

Combinaison (class Routine)

Pour un *passage* dès qu'il y a plus qu'un *skill* (i.e. plusieurs *skills* ou plusieurs itérations d'un même *skill*) on dit que c'est une combinaison.

Enchaînement

Un enchaînement (parfois appelé « série » à tort !) est un ensemble de *Skill* qui se suivent (c-à-d. sans une forme de pause entre eux). Un enchaînement est donc en cas particulier de combinaison.

Série (Routine en anglais)

Une série est type *d'enchaînement* particulier : c'est un enchaînement *compétitif*. Pour être compétitif, un enchaînement se compose de :

- 10 sauts (ni plus ni moins) avec
- un départ (de la première figure) sur les pieds
- une arrivée (de la dernière figure) sur les pieds
- aucune arrivée sur les mains.

Nous avons donc :

Combinaison > Enchaînement > Série

Dans mes programmes actuels (fichier Excel), les enchaînements peuvent être notés de deux façons :

- sur la droite, les séries (sous-entendu « série de compétition » qui sont un cas particulier d' « enchaînement ») en général indiqué par un label (exemple : R1, R2, blah, série des assis, ...) ;
- et dans le programme, des enchaînements précis demandés. Ces derniers s'écrivent comme une suite de *Skill* liés par des tirets. Exemple : *skill_1 – skill_2 – skill_3* - ...

				Enchaînement (série)				Félix			
/ +		Vendredi		- / +		Samedi		- / +			
		Libre				Libre					
		/ 1, 2 & 3				/ 1, 2 & 3					
		/ & /4				/ & /4					
		(et)									
		BL2+3									
		.* beb 2									
		.41* beh 2									
		4.2/ - 4.2/ - 4.2/ 2				4.2/ - 4.2/ - 4.2/ 2					
		.43/ - .43/ - .43/ 2				.43/ - .43/ - .43/ 2					
		.8-1o - .8-1o - .8-1o 2				R1					
		.8-1< - .8-1< - .8-1< 2				R1					
		.8-3o - .8-1o - .8-3o 2				.8-1< - ... 4					
		.8-3< - .8-1< - .8-3< 2				o 2 2 2					
		8.-o - .41o - ... 3									
		8.-< - .41o - ... 3				8.11o - - ... 3					
		8.-/ 3				8.11< - - ... 3					
		8.11o - 4.-/ - ... 3				R2 [7-] 2					
		8.11< - 4.-/ - ... 3				R2 [2-7]					
		R1				.72-* 2					
		R1				.821< 2					
		R2 [7-] 2				.3-/ - .9-1o					
		R2 [2-7]				5.3-/ - passer tête 3					

Flying Acrobatics Trampoline Club													
Coach : Gregory													
D1	Semaine 35												
	Date												
	Lundi												
1	Libre												
2	/ 1, 2 & 3												
3	/ & /4												
4	BL2+3 2												
5	4.* beh 2												
6	41* beh 2												
7	4.2/ - 4.2/ - 4.2/ 2												
8													
9													
10													
11													
12													
13													
14													
15													
16													
17													
18													
19													
20													
21													
22													
23													
24													
25													
26													
27													
28													
29													
30													
31													
32	Points à corriger :												

Lundi		Mercredi		Vendredi		Samedi		Séries	
Libre		Libre		Libre		Libre		L1	
1	(&)	20	(&)	20	(&)	20	(&)	20	4.-/
2	...	30	[1, 2 & 3	30	[1, 2 & 3	30	[1, 2 & 3	30	A1/
3	...	6	[& /4	6	[& /4	6	[& /4	6	A1/
4	...	6	[& /4	6	[& /4	6	[& /4	6	A1/
5	...	9	S0B 3	9	S0B 3	9	S0B 3	9	A1/
6	...	9	BL1 3	9	BL1 3	9	BL1 3	9	A1/
7	...	6	BL2 2	6	BL2 2	6	BL2 2	6	A1/
8	...	6	A1* - 3.<	6	A1* - 1./	6	A1* - 1.<	6	A1/
9	...	6	4.* - 3.<	6	4.* - 1./	6	4.* - 1.<	6	A1/
10	...	6	L1 [8-] 2	6	L1 [6-] 2	10	L1 [6-] 2	10	A1/
11	...	10	L1 [6-] 2	10	L1 [3-6] 2	8	L1 [3-6] 2	8	A1/
12	...	8	L1 [3-6] 2	8	L1 [3-] 2	8	L1 [3-] 2	8	A1/
13	...	8	L1 [3-] 2	8	L1 [3-] 2	10	L1 [3-] 2	10	A1/
14	...	10	L1 [3-] 2	10	L1 [3-] 2	10	L1 [3-] 2	10	A1/
15	...	2	3.-/ - dos - POo	2	3.-/ - dos - POo	2	3.-/ - dos - POo	2	A1/
16	...	2	1/ - 5.<	2	1/ - 5.<	2	1/ - 5.<	2	A1/
17	...	4	PP - BO <	4	PP - Piqué	4	PP - Piqué	4	A1/
18	...	4	3.-/ - 6.<	4	3.-/ - 6.<	4	3.-/ - 6.<	4	A1/

Un **entraînement**, c'est une activité sportive qui se passe à un moment (**date/date+heure**) et pendant lequel un gymnaste va s'entraîner. Un **entraînement est constitué de plusieurs passages**.

Il peut y avoir récurrence : un gymnaste peut avoir un entraînement tous le X (par exemple tous les lundis ou tous les mercredis ou tous les lundi ET mercredi, ...) mais -pour moi- c'est plus une vision de l'esprit, une exagération. Ce sont des entraînements différents, avec des coaches différents, des passages différents, ... Le moment d'entraînement est récurrent mais l'entraînement en lui-même ne l'est pas. Il faut voir si nous parlons de la généralité « entraînement » ou d'un entraînement précis (je parle ici du précis, du ponctuel, pas du général).

Un **passage**, c'est un moment de travail : l'élève fait une chose (mouvement (*skill*), un enchaînement série (*routine*), un enchaînement spécifique (*routine*), une combinaison de mouvement, ...) qu'il doit répéter un certain nombre de fois avant de laisser la place au gymnaste suivant (ou de se reposer).

Une combinaison est un ensemble de mouvement **non enchaînés** (donc *pas liés par des tirets*).

Mais comment le designer ? (En fait je m'oriente de plus en plus vers la version normalisée qui est beaucoup plus propre et respectueuse)

/!\ ATTENTION :

- ne pas s'attarder sur les noms des modèles, le problème n'est pas (encore) là.
- Le nom des concepts et modèles déjà existant ne sont pas toujours corrects (ex. « *routine* » que j'ai voulu renommer en « *combinaison* » devrait en réalité peut-être s'appeler « *linking* » ou « *sequence* » ou ...)

Sur base des explications, je modéliserais un programme avec deux ¹classes : une classe **entraînement** et une classe **passage** :

- un modèle « *Training*² » constitué de
 - o ID (int)
 - o gymnast_id (int)
 - o date (ou datetime begin) (date/datetime) (hérité de *seasonisable* ?)
 - o (datetime end si datetime begin) (date/datetime)
 - o informations
- un modèle « *TrainingProgram*³ » constitué de
 - o un ID (int)
 - o une FK (int) vers *Training* (un passage est lié à (au moins un) entraînement)
 - o un nombre (int) de répétition demandée (combien de fois le mouvement ou la combinaison de mouvement doit être effectuée)
 - o une difficulté théorique (quel est la difficulté si l'élève fait ce qui est demandé) (float)

Le modèle *Training* devrait être réfléchi à l'aune des modèles *NumberOfRoutineDone*, *Intensity*, *Chrono*, *LearnedSkill* et *MindState*

- <https://sources.grimbox.be/Sulley/Jarvis/src/branch/master/jarvis/followup/models.py> -
puisque ces modèles représentent une information qui ne peut être produite que pendant un entraînement : tous ces modèles dérivent d'ailleurs de modèle virtuel « *Seasonisable* ».

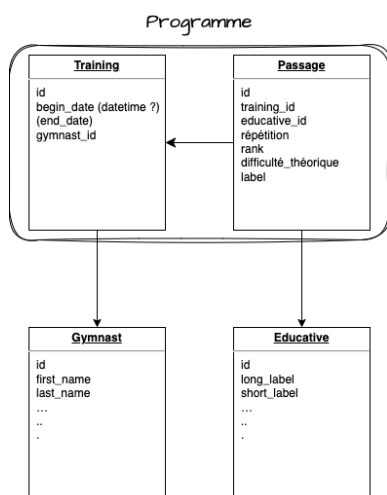
On pourrait donc imaginer sur *Training* hérite de *Seasonisable* (ainsi que les autres) et que les autres aient en plus une FK nullable vers *Training*.

¹ Sur ce point là, je pense qu'on est d'accord

² Ne pas faire attention au nom

³ Ne pas faire attention au nom

Solution simpl(ist)e



Ici le modèle passage (*Passé*) contient un **ManyToMany** vers le modèle *Educative* (pour rappel : *Educative* est un modèle « virtuel » (pas vraiment en fait) qui permet de pointer vers *Skill* (élément unique) ou *Combinaison* (éléments enchaînés).

Le lien M2M permet de demander la réalisation de plusieurs choses au cours d'un même passage, par exemple : | 1, 2 & 3 ; | & | 4 ; (et) ; WC x & y ; ...

a supprimé: aucours

Avantages

- Tout est basé sur « *educative* »

Inconvénients

- Tord le modèle « *educative* » (selon moi) : certains passages (de ma manière de travailler actuelle : | 1, 2 & 3 ; | & | 4 ; (et) ; WC x & y ; ... cf. fichier Excel) ne sont ni des *skill* ni des *enchaînements*. Je peux les mettre dans un de ces deux modèles MAIS ce n'est pas parce que ça rentre que c'est vraiment ça (ce n'est pas parce qu'un rond rentre dans un carré que ça en fait un carré pour autant)
- Pas de normalisation
- Beaucoup de répétition d'informations

Commenté [GT1]: Pour être précis, certains sont des enchaînements

Ces passages peuvent être mis sous forme de séries et comme le lien avec *Educative* est un M2M, il y a bien moyen de faire le lien. Exemple :

- | 1, 2 & 3 = | 1 (série) + | 2 (série) + | 3 (série)
- | & | 4 = | (série) + | 4 (série)
- (&) = ((série) +) (série)

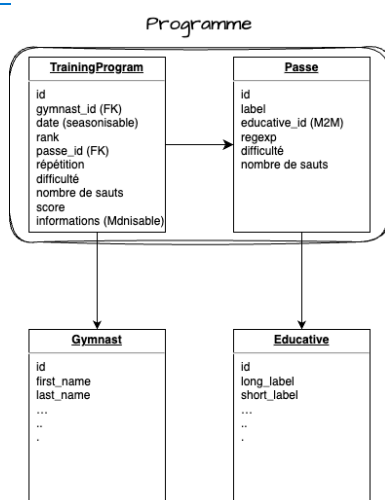
Par contre, tout ne peut pas être un éducatif, par exemple WC x & y.

MAIS WC x & y est une combinaison spéciale (WC) de deux saut (M2M vers Skill). Ça devrait donc pouvoir rentrer également.

Solutions testées

Sur base de cette solution simpliste, j'ai testé deux solutions. Je n'ai plus tous les détails de la première mais elle n'était clairement pas suffisante et manquait trop de normalisation (le champ *repetition* était, par exemple, dans la table *Passe*).

Deuxième solution testée :



Elle répond presque à tous mes besoins. **MAIS** il reste plusieurs problèmes :

- 1) Je n'ai pas une classe « entraînement » ce qui fait que je duplique beaucoup d'information : date, gymnast id, informations, ...
- 2) Je n'ai pas de classe « entraînement » ce qui m'empêche de pouvoir modéliser ce concept et de lui attribuer des infos (typiquement : informations et score)
- 3) J'ai un problème de concept : utilisation impossible de la notation Lx (cf. ci-dessous)

Problème de concept

Comme expliqué dans le commentaire de la page précédente, j'arrive à tout modéliser sous forme d' *Educative* via des *Routines* et le lien M2M de la table *Passe*.

Dans notre façon de travailler actuelle nous utilisons beaucoup les concepts de L1 (ou R1 ou Q1R1), L2 (ou R2 ou Q1R2), ... cela apparaît souvent dans les programmes :

	Lundi		Mercredi		Vendredi		Samedi		Séries
	Libre		Libre		Libre		Libre		L1 L2
1	(&)	20	(&)	20	(&)	20	(&)	20	4-/ 0.6
2	[1,2 & 3	30	[1,2 & 3	30	[1,2 & 3	30	[1,2 & 3	30	.41/ 0.6
3	& 4	6	& 4	6	& 4	6	& 4	6	.41< 0.6
4	SdB 3	9	SdB 3	9	SdB 3	9	SdB 3	9	< <
5	BL1 3	9	BL1 3	9	BL1 3	9	BL1 3	9	4-< 0.6
6	BL2 2	6	BL2 2	6	BL2 2	6	BL2 2	6	4-o 0.5
7	.41* -1-/	6	.41* -3-<	6	.41* -1.1/	6	.41* -1-<	6	.41o 0.6
8	4-* -1-/	6	4-* -3-<	6	4-* -1.1/	6	4-* -1-/	6	// //
9	L1 [6-] 2	10	L1 [B-] 2	10	L1 [6-] 2	10	L1 [6-] 2	10	3-/ 0.3
10	L1 [6-] 2	10	L1 [6-] 2	10	L1 [B-6] 2	8	L1 [B-6] 2	8	5lo 0.7
11	L1 [B-6] 2	8	L1 [B-6] 2	8	L1 [B-] 2	8	L1 [B-] 2	8	
12	L1 [B-] 2	8	L1 [B-] 2	8	L1 [B-] 2	8	L1 [B-] 2	8	
13	L1 [3-] 8	10	L1 [3-] 8	10	L1 [3-] 8	10	L1 [3-] 8	10	
14	L1 10	10	L1 10	10	L1 10	10	L1 10	10	
15	3-/ - dos - POo	2	3-/ - dos - POo	2	3-/ - dos - POo	2	3-/ - dos - POo	2	
16	1 - .5-o	2	1 - .5-<	2	1 - .5-<	2	1 - .5-o	2	
17	Piqué - BO <	4	PP - Piqué	4	PP - Piqué	4	PP - Piqué	4	
18	.3-/ - .6-o	4	.3-/ - .6-<	4	.3-/ - .6-<	4	.7-o	2	
19	.3-/ beh	2	.3-/ beh	2	.3-/ beh	2	.3-/ beh	2	
20									

- La *regexp* se trouve dans le record *Passe*
- Le *rank* se trouve dans le record *TrainingProgram*
- La *repetition* se trouve dans le record *TrainingProgram*

Le problème est la notion de « L1 » : Lx (ou Rx ou QxRy, peu importe comment on le note) représente une association temporellement limitée entre une série (de compétition) et un gymnaste.

Ce lien est stocké dans la table *GymnastHasRoutine*. Chaque record lie :

- un Gymnast à
- une Routine (lien vers la classe fille d'*Educative*) avec
- une date de début,
- une date de fin et
- un type (Q1R1, Q1R2, Q2R1, SF, F, ...).

Or dans ma solution actuelle je n'ai pas de lien (dans le modèle *Passe*) vers le modèle *GymnastHasRoutine* (et pas sur d'en avoir besoin dans l'absolu).

La fausse bonne idée : je pourrais, dans ma table *Passe*, créer un record avec un label « L1 » et un lien vers la série voulue MAIS :

- un gymnaste à plusieurs L1 au cours de sa carrière
- chaque gymnaste à un L1

Donc le label « L1 » ne suffit pas pour savoir de quelle série précisément je parle. Or j'ai besoin de pouvoir l'utiliser.

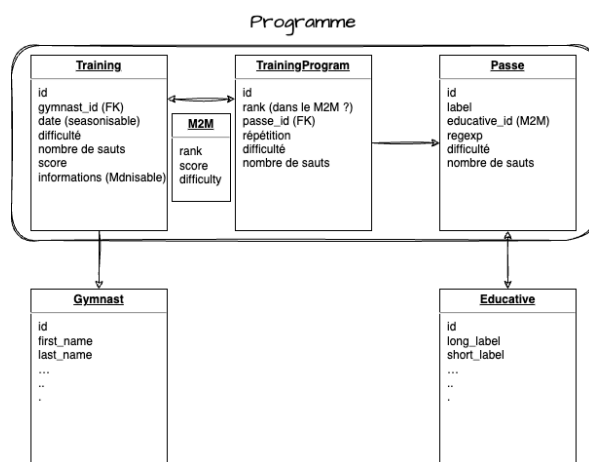
Amélioration proposée :

Ajouter un modèle *Training* avec les champs suivants :

- date (héritée de *Seasonisable*)
- gymnast (FK *Gymnast*)
- difficulté
- nombre de saut
- score
- informations (hérité de *Markdownisable*)
- relation M2M avec TrainingProgram

Cela résoudrait certains problèmes évoqués :

- beaucoup moins de duplication d'informations (date, gymnaste) et
- possibilité d'avoir des informations sur un entraînement (informations, score, ...).
- Augmentation de la normalisation de la DB.



Passe est lié à Educative par une relation M2M

Training est lié à TrainingProgram par une relation M2M

Question 1 :

La relation M2M entre *Training* et *TrainingProgram* devrait avoir un moins un champ supplémentaire : *score* (avis du gymnaste sur le passage en question). Mais peut/doit-elle contenir d'autre champ (par exemple : *rank*, *difficulté*, *repetition*, ...) ? Et, si oui, jusqu'à quel point : *TrainingProgram* ne pourrait-elle pas elle-même servir de M2M entre *Training* et *Passe* ?

Concernant le 3^{ème} problème (utilisation des notions de Lx/Rx/QxRy), voici une idée : rendre le champ *educative_id* nullable et écrire « L1 [...] », ... en *label* ET en *regexp* pour créer un lien « soft ».

Le lien serait fait à la volée lors de la liaison entre *Training* et *TrainingProgram* et l'information en découlant (*difficulty*) stocké dans la table de liaison avec le *score*. Imaginons un gymnaste affiche un entraînement dans lequel il doit faire :

- L1 [8-]
- L1 [6-9]
- L1 [3-7]
- L1 [-4]
- L1
- L1

Dans *Passe*, il est possible de calculer le *nombre de saut* mais pas la *difficulté* (pour rappel, le champ *educative* est null pour ces passages).

Dans *TrainingProgram* j'ai un *rank*, une *repetition*, le *nombre de saut* mais toujours pas de *difficulté* parce que le lien ne sera fait que sur base de la *date du record Training*.

Au moment où je lie le record *Training* et *TrainingProgram*, le lien est remonté : dans son programme il y a un *trainingpasse* « L1 [6-9] 2 », ce *training* a une *date*, donc le passage doit être fait à cette date donc L1 peut-être récupéré par rapport aux informations présente dans *GymnastHasRoutine* et à la date du *Training*.

L'information de *difficulté* peut alors être calculée à ce moment-là et être stockées dans la table M2M entre *Training* et *TrainingProgram*.

Inconvénients :

- La *regexp*

Il faut gérer la *regexp* à plusieurs endroits :

- Dans le *clean()* / *save()* du form
- A l'affichage (détails) d'une *Passe*
- A l'affichage d'un *Training/TrainingProgram*
- ... ?

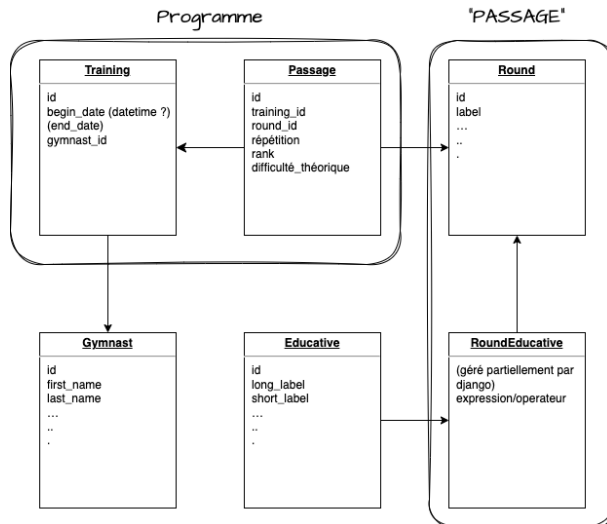
Et il y a plusieurs gestions :

- Validation individuelle (*clean()* / *save()*)
- Validation inter-champ (*clean()* / *save()*)
- Interprétation (détails *Passe/Training/TrainingProgram*)

ET UNE REGEXP C'EST PAS FACILE A BIEN GERER !!!!!!!!! :(

- Toujours pas de normalisation complète de la DB (même si c'est déjà bien mieux que mes solutions d'avant).

Solution normalisée ([abandonnée parce que fausse bonne idée](#))



Ici le modèle passage n'est PAS lié à *Educative* de manière directe, mais de manière indirecte. Dans cette solution *Passage* est lié à *Round*.

Round et *RoundEducative* vont me permettre de représenter les *combinaisons*⁴. Ces deux modèles vont avoir un rôle *similaire* que *Routine* et *RoutineSkill* **sauf** que le lien n'est pas le même ([ce n'est pas](#) des tirets !). Ce sont deux liens différents, créant deux objets différents et impliquant donc deux modèles différents. Ces deux modèles ont donc pour but de me permettre une liaison autre que la simple *Routine* (*Combinaison*).

Avantages

- Propre et ne tord aucun modèle
- Garde la DB normalisée

Inconvénients

- Plus compliqué à mettre en place et à gérer

[RoundEducative n'est pas nécessaire, il y a moyen de lier les Educatives comme je veux en faisant autrement \(cf. solutions précédentes\) et, me semble-t-il, plus simplement que cette solution-ci.](#)

⁴ Ne pas faire attention au nom

- (&)
 -) -) -) -) -) -) -) -) -)
 - (- (- (- (- (- (- (- (- (
- *Crevettes*
 - Dos – dos – dos – dos – dos ...
- | 1, 2 & 3
 - | 1 - | 1 - | 1 - | 1 - | 1 - | 1 - | 1 - | 1 - | 1
 - | 2 - | 2 - | 2 - | 2 - | 2 - | 2 - | 2 - | 2 - | 2
 - | 3 - | 3 - | 3 - | 3 - | 3 - | 3 - | 3 - | 3 - | 3
- | & | 4 (1 sur 2)
 - | - | - | - | - | - | - | - | - |
 - | 4 - | 4 - | 4 - | 4 - | 4 - | 4 - | 4 - | 4 - | 4
- SdB
 - Saut groupé
 - Saut carpé
 - Saut écart
- BL1
 - Assis – debout
 - Ventre – debout
 - Dos – debout
- BL1 + ½
 - Assis – ½ debout
 - Ventre – ½ debout
 - Dos – ½ debout
- BL2
 - ¾ avant o – debout
 - ¾ avant < – debout
 - ¾ avant / – debout
 - ¾ arrière o – debout
 - ¾ arrière < – debout
 - ¾ arrière / – debout