

1. SYNOPSIS

Un *pokétudiant* est un petit monstre possédant des aptitudes magiques, définies à sa naissance. Il peut être capturé par un *enseignant-dresseur* dans la nature, où il aime se cacher dans les hautes herbes. Une fois capturé, le pokétudiant voue sa vie à son enseignant-dresseur, qui le contrôle alors totalement. Un enseignant-dresseur a pour but de collectionner un maximum de pokétudiants, et de les entraîner pour qu'ils deviennent suffisamment puissants pour devenir le meilleur dresseur de pokétudiants.

Le but du projet est de réaliser un petit jeu multijoueur en réseau dans lequel chaque joueur incarne un enseignant-dresseur. Lorsque votre projet sera terminé, les joueurs pourront créer des parties auxquelles d'autres joueurs pourront participer. Les joueurs auront la possibilité de se déplacer sur une carte, sur laquelle les autres joueurs seront visibles, de combattre et capturer des pokétudiants sauvages, ainsi que de défier les autres joueurs de la partie en combat.

2. POKÉTUDIANTS

1. Variétés de pokétudiants

Il existe plusieurs *variétés* de pokétudiants, chacune disposant de caractéristiques qui lui sont propres. Le personnage incarné par le joueur, l'enseignant-dresseur, sera considéré lui aussi comme un pokétudiant, et pourra de ce fait participer aux combats.

Une variété de pokétudiant dispose d'un *type*, parmi {Noisy}, {Lazy}, {Motivated} et {Teacher} (ce dernier étant réservé aux enseignants-dresseurs), qui influe sur les dommages subis par les attaques, en fonction de leurs propres types.

Certaines variétés de pokétudiants ont la faculté *d'évoluer* : en les entraînant suffisamment, les pokétudiants de ces variétés se transforment en d'autres variétés de pokétudiants plus puissantes. Ces variétés évoluées ne peuvent cependant pas être découvertes dans la nature, tout comme les enseignants-dresseurs. Les informations de base concernant les variétés de pokétudiants sont présentées dans la Table 1.

Variété	Type	Capturable	Évolution
Parlfor	Noisy	oui	∅
Ismar	Noisy	oui	Rigolamor
Rigolamor	Noisy	non	∅
Procrastino	Lazy	oui	∅
Couchtar	Lazy	oui	Nuidebou
Nuidebou	Lazy	non	∅
Alabourre	Lazy	oui	∅
Buchafon	Motivated	oui	∅
Belmention	Motivated	oui	Promomajor
Promomajor	Motivated	non	∅
Enseignant-dresseur	Teacher	non	∅

TABLE 1 – Variétés de pokétudiants. Le symbole ∅ dans la colonne « évolution » signifie que le pokétudiant correspondant ne dispose pas d'évolution.

Les pokétudiants disposent, en plus de leur type, de trois statistiques majeures : *l'attaque*, *la défense*, et les *points de vie maximum (PV max.)*. Plus l'attaque d'un pokétudiant est importante, plus ses attaques infligeront de dégâts (c'est-à-dire qu'elles consommeront plus de points de vie du pokétudiant qui subit l'attaque). À l'inverse, plus la statistique de défense d'un pokétudiant est importante, et moins les attaques qu'il subira infligeront de dégâts.

Variété	Attaque	Défense	PV max.
Parlfor	60	40	60
Ismar	50	30	40
Rigolamor	85	55	70
Procrastino	40	60	60
Couchtar	30	50	40
Nuidebou	55	85	70
Alabourre	75	95	65
Buchafon	50	50	60
Belmention	30	50	40
Promomajor	70	70	70
Enseignant-dresseur	100	100	100

TABLE 2 – Statistiques de base des différentes variétés de pokétudiants.

Attaque	Type	Puissance
Bavardage	Noisy	10
Groboucan	Noisy	15
Mégaphone-vocal	Noisy	20
Bâillement	Lazy	10
Pti'somme	Lazy	15
Superdodo	Lazy	20
Rateletrain	Lazy	15
Oboulo	Motivated	10
Exo-maison	Motivated	15
Max-reviz	Motivated	20
Tit'question	Teacher	10
Poser-colle	Teacher	15
Fatal-interro	Teacher	20

TABLE 3 – Attaques des pokétudiants.

Ces statistiques dépendent en grande partie de la variété du pokétudiant ; en effet, certaines variétés sont très offensives, quand d'autres sont capables d'encaisser beaucoup de dégâts. Les statistiques de base pour les différentes variétés de pokétudiants (au niveau 1, voir la section 2.3 pour les niveaux des pokétudiants) sont données dans la Table 2.

Pour une même variété, les aléas de la nature peuvent offrir à deux pokétudiants des statistiques différentes. Chacune des trois statistiques est en effet modulée par un coefficient réel compris entre 0.9 et 1.1.

Exemple. Au niveau 1, un Couchtar possède une attaque comprise entre 27 et 33, une défense comprise entre 45 et 55, et des PV max. compris entre 36 et 44.

2. Attaques des pokétudiants

Chaque pokétudiant dispose de deux *attaques*. La première est choisie aléatoirement parmi celles dont le type est le même que celui du pokétudiant. La deuxième est choisie de manière aléatoire parmi celles dont le type est différent du pokétudiant, à l'exception des attaques du type {Teacher}. En plus de leur type, les attaques disposent d'une statistique appelée *puissance* : plus la puissance d'une attaque est importante, plus elle inflige de dommages. Les types et puissances des attaques sont donnés par la Table 3.

Certains types d'attaque sont particulièrement *efficaces* contre des pokétudiants d'un certain autre type ; dans ce cas, la puissance de l'attaque est doublée. Cela se produit dans les cas suivants :

- attaque de type {Noisy} et pokétudiant défenseur de type {Lazy},
- attaque de type {Lazy} et pokétudiant défenseur de type {Motivated},
- attaque de type {Motivated} et pokétudiant défenseur de type {Noisy},

— attaque de type {Teacher} et pokétudiant défenseur de type {Noisy}, {Lazy} ou {Motivated}.

Exemple. Un Buchafon lance l'attaque Groboucan (de type Noisy) sur un Nuidebou (de type Lazy). L'attaque étant très efficace de par son type contre les pokétudiant de type Lazy, sa puissance est doublée pour passer de 15 à 30.

Une fois la puissance effective de l'attaque déterminée, le nombre de points de vie à retirer au pokétudiant qui subit cette attaque est calculé en fonction de l'attaque du pokétudiant attaquant et de la défense du pokétudiant défenseur, et est donné par la formule suivante :

$$dmg = k \cdot \frac{att_{\text{attacker}}}{def_{\text{defender}}} \cdot power,$$

où k est un nombre réel choisi aléatoirement entre 0.9 et 1.1. Une fois la valeur dmg calculée, son montant est retiré au nombre de points de vie restant au pokétudiant défenseur. Si les points de vie tombent en dessous de 1, le pokétudiant défenseur est alors KO et ne peut plus se battre.

3. Expérience et évolution des pokétudiants

Lorsque le joueur met KO un pokétudiant ennemi, l'ensemble des pokétudiants du joueur ayant participé au combat contre le pokétudiant KO reçoit de l'expérience (voir section 3 pour les combats). Le montant d'expérience reçu par chacun des pokétudiants est égal au montant d'expérience dû par le pokétudiant KO divisé équitablement (en utilisant la division entière) par le nombre de pokétudiants. Le montant dû est égal à 10% du montant total de l'expérience engrangée par le pokétudiant KO.

Lorsqu'un pokétudiant a engrangé suffisamment d'expérience, il monte d'un niveau : ses statistiques (attaque, défense, PV max.) augmentent et il peut évoluer. La quantité d'expérience nécessaire pour monter de niveau est donnée par la formule suivante :

$$xp_{\text{next}} = 500 \cdot \frac{1 + lvl}{2},$$

où xp_{next} est la quantité d'expérience nécessaire pour passer du niveau lvl au niveau $lvl + 1$. Le niveau minimal d'un pokétudiant est de 1, son niveau maximal est de 10 (une fois au niveau 10, l'expérience qu'il gagne n'est plus comptabilisée).

Exemple. Un pokétudiant de niveau 2, ayant accumulé un total de 1200 points d'expérience (500 au niveau 1, 700 au niveau 2) met KO (seul) un pokétudiant ayant accumulé 2000 points d'expérience. Il se retrouve alors avec une expérience totale de 1400 points et est maintenant au niveau 3 (500 points d'expérience au niveau 1, 750 au niveau 2, et 150 au niveau 3).

En montant de niveau, il se produit les choses suivantes pour un pokétudiant : + ses statistiques d'attaque, de défense, et de PV max. sont augmentées de 10% des valeurs qu'elles avaient au niveau₁ ; + il est soigné (ses points de vies effectifs sont égaux à ses PV max.) ; + s'il fait partie d'une espèce capable d'évoluer, il est susceptible d'évoluer : + s'il vient d'atteindre le niveau 3, il a 20% de chances d'évoluer, + s'il vient d'atteindre le niveau 4, il a 37.5% de chances d'évoluer, + s'il vient d'atteindre le niveau 5, il évolue.

En évoluant, le pokétudiant change de variété (pour adopter celle en laquelle il évolue), et ses statistiques sont renforcées pour atteindre celles qu'il aurait eues s'il avait été de cette variété depuis le niveau 1. En d'autres termes, il faut partir des statistiques de base de sa variété, appliquer les mêmes coefficients que ceux qui avaient donné ses statistiques effectives lors de sa création, puis appliquer les augmentations de statistiques dues à chacune de ses montées de niveau.

Exemple. Un Belmention est généré au niveau 1, dont les coefficients d'attaque, défense, et PV max. sont respectivement égaux à 1.1, 0.9 et 1 ; ce Belmention a ainsi des statistiques effectives de 33, 45 et 40. En atteignant le niveau 2, puis le niveau 3, ses statistiques effectives atteignent respectivement $\lfloor 33 \times 1.2 \rfloor = 39$, $\lfloor 45 \times 1.2 \rfloor = 54$ et $\lfloor 40 \times 1.2 \rfloor = 48$.

Ce Belmention évolue au niveau 3 en Promomajor, dont les statistiques de base sont 70 pour chacune des statistiques. Les statistiques effectives de ce pokétudiant sont alors respectivement $\lfloor (70 \times 1.1) \times 1.2 \rfloor = 92$, $\lfloor (70 \times 0.9) \times 1.2 \rfloor = 75$ et $\lfloor (70 \times 1) \times 1.2 \rfloor = 84$.

3. COMBAT ENTRE POKÉTUDIANTS

Il existe deux types de combat auxquels peut participer un enseignant-dresseur : un combat contre un pokétudiant sauvage ou un combat contre un enseignant-dresseur rival contrôlé par un autre joueur de la partie.

1. Combat contre un pokétudiant sauvage

Au début du combat, le premier pokétudiant de l'équipe de l'enseignant-dresseur (voir section 4 pour l'équipe de l'enseignant-dresseur) fait face au pokétudiant sauvage. Le combat se décompose en un ensemble de *tours* se succédant tant que la bataille n'a pas pris fin.

Un tour se compose des étapes suivantes :

1. le joueur choisit une action à effectuer parmi :
 1. lancer la première attaque de son pokétudiant,
 2. lancer la deuxième attaque de son pokétudiant,
 3. changer de pokétudiant,
 4. tenter de capturer le pokétudiant sauvage,
 5. tenter de fuir le combat ;
2. l'action du joueur est effectuée (ce qui peut mettre fin au combat) ;
3. le pokétudiant sauvage lance une de ses deux attaques, la sélection étant faite au hasard (ce qui peut aussi mettre fin au combat).

En ce qui concerne les attaques, on suit les règles dictées à la section 2.2. Pour le changement de pokétudiant, on demande au joueur par quel autre membre de son équipe il souhaite remplacer le pokétudiant courant ; le remplacement est effectué, et le tour continue (le pokétudiant sauvage attaque).

Pour la tentative de fuite du combat, les probabilités de réussite sont les suivantes :

- 0% (resp. 100%) de chances de réussite si le pokétudiant du joueur possède au moins trois niveaux de moins (resp. de plus) que le sauvage,
- 25% (resp. 90%) de chances de réussite si le pokétudiant du joueur possède deux niveaux de moins (resp. de plus) que le sauvage,
- 40% (resp. 75%) de chances de réussite si le pokétudiant du joueur possède un niveau de moins (resp. de plus) que le sauvage,
- 50% si les deux pokétudiants ont le même niveau.

En cas de fuite réussie, le combat se termine. Pour la tentative de capture, la probabilité de réussite est donnée par la formule

$$2 \cdot \max \left(\frac{1}{2} - \frac{pv_{\text{eff}}}{pv_{\text{max}}}, 0 \right)$$

où pv_{eff} est le nombre de points de vie qu'il reste au pokétudiant sauvage, et pv_{max} est son nombre de points de vie maximum. Un pokétudiant capturé n'est pas soigné.

Si à la fin d'un tour le pokétudiant courant du joueur est KO, il doit en choisir un de son équipe qui n'est pas KO pour continuer le combat. Si tous ses pokétudiants sont KO, alors le joueur a perdu le combat ; ses pokétudiants sont soignés mais il perdent tous 20% de leur expérience totale acquise, ce qui peut leur faire perdre des niveaux mais pas « désévoluer ». En revanche, si une attaque du pokétudiant du joueur met KO le pokétudiant sauvage, alors l'expérience obtenue est divisée entre tous les pokétudiants du joueur ayant participé au combat, comme décrit à la section 2.3.

2. Combat contre un enseignant-dresseur rival

Il s'agit d'un combat joueur contre joueur. En début de tour, chacun des joueurs choisit son action, qui peut être de lancer une des deux attaques du pokétudiant courant, ou bien changer de pokétudiant courant. On distingue trois cas différents : + si les deux joueurs souhaitent changer de pokétudiants, les changements sont réalisés dans un ordre quelconque ; + si un des joueurs souhaite attaquer alors que l'autre souhaite changer de pokétudiant courant, le changement est effectué puis l'attaque est lancée ; + si les deux joueurs souhaitent attaquer, le premier à lancer son attaque est choisi aléatoirement, puis le deuxième attaque s'il n'a pas été mis KO.

Contrairement aux combats contre des pokétudiants sauvages, vaincre un pokétudiant adverse n'apporte pas d'expérience. Lorsqu'un pokétudiant est mis KO, son joueur doit en sélectionner un de son équipe pour continuer le combat. S'il n'en a plus, ses pokétudiants perdent de l'expérience de la même manière que lors d'un combat contre un pokétudiant sauvage. Le joueur gagnant ne gagne rien d'autre que le droit de narguer son adversaire.

4. ÉQUIPE DU JOUEUR

Un joueur dispose d'une *équipe*, composée de son enseignant-dresseur et d'un maximum de deux pokétudiants. Au début de la partie, l'équipe est composée d'un enseignant-dresseur initialisé aléatoirement, au niveau 1, ayant 0 points d'expérience.

Lors de la capture d'un pokétudiant sauvage, celui-ci est placé dans l'équipe si une place est disponible, sinon il est libéré : il disparaît du jeu.

Quand il n'est pas en combat, un joueur a la possibilité de changer l'ordre dans lequel sont rangés les pokétudiants de son équipe, ou de libérer un de ses pokétudiants (à l'exception de son enseignant-dresseur).

5. TRAVAIL À RÉALISER

Le projet est à réaliser en binôme, et est composé de deux programmes : un serveur écrit dans le langage de votre choix, et un client écrit également dans le langage de votre choix (les langages pour le serveur et le client doivent être différents). Ils devront être déposés sur Moodle au plus tard le 13 décembre. Aucune librairie externe ne doit être utilisée (sauf demande auprès de l'enseignant), et les communications réseaux ne peuvent utiliser que les fonctions et objets vus en TP. Les communications entre clients et serveurs doivent respecter les spécifications de la section 5.2. Les questions devront être posées avec le forum de Moodle.

1. Description du jeu

1. Rejoindre une partie

À son lancement, le client envoie un message sur tout le réseau (*broadcast* sur le port UDP 9000) à la recherche de serveurs. Tout serveur recevant ce message lui répond de manière à se manifester auprès du client. Le client dispose d'un bouton permettant de mettre à jour la liste des serveurs en répétant ce processus.

Le client sélectionne alors un serveur parmi ceux disponibles, et s'y connecte (port TCP 9001) pour découvrir la liste des parties ouvertes sur ce serveur. L'utilisateur dispose alors de deux alternatives : soit rejoindre une partie (qui peut accueillir jusqu'à quatre joueurs), soit en créer une nouvelle (un serveur peut accueillir jusqu'à quatre parties). S'il choisit d'en créer une nouvelle, il doit lui donner un nom composé de caractères alphanumériques et d'espaces. Le joueur entre dans cette partie, qui restera ouverte tant qu'au moins un joueur y sera connecté, et sera supprimée dès que plus personne n'y participe. Le client dispose d'un bouton permettant de rafraîchir la liste des parties disponibles.

Une fois le client dans une partie, le joueur dispose d'une interface présentant la carte du jeu sur laquelle sont placés les joueurs, d'un *chat* pour dialoguer avec les autres joueurs, d'une interface montrant son équipe, ainsi que d'un bouton permettant de quitter la partie. Les échanges d'informations avec le serveur se déroulent en utilisant la même socket TCP que précédemment. L'ensemble des informations, y compris la position du joueur et son équipe, sont enregistrés sur le serveur : le client ne s'occupe que de l'affichage du jeu et de la transmission des actions que le joueur souhaite exécuter.

2. Sur la carte

Le joueur peut se déplacer sur la carte, composée d'une grille de cases. Cette carte est enregistrée dans un fichier (au format de votre choix), et chargée par le serveur à la création d'une partie. Certaines de ses cases représentent des zones qui soignent les pokétudiants de l'équipe (les points de vie courants sont augmentés jusqu'aux points de vie maximum), d'autres sont susceptibles de déclencher un combat ; les autres cases sont neutres.

Le joueur peut se déplacer sur la carte, d'une case vers la gauche, le haut, la droite ou le bas. Le client ne s'occupe pas lui-même de déplacer le joueur sur la carte : il n'a qu'un rôle d'affichage. Ainsi, pour se déplacer, le client envoie la direction dans laquelle il doit aller, et attend que le serveur lui renvoie la carte pour la mettre à jour. Si ce déplacement crée une rencontre avec un pokétudiant sauvage, le serveur prévient le client qui se retrouve en combat (le pokétudiant est généré aléatoirement, avec une expérience faisant qu'il est entre les niveaux 1 et 3). Si un combat contre un pokétudiant sauvage n'est pas déclenché et si ce déplacement se fait sur une case où un enseignant-dresseur rival est, et qu'il n'est pas en train de combattre, cela provoque un combat entre les deux joueurs : le serveur prévient chacun d'eux.

3. Dialoguer avec ses rivaux

Le chat permet d'envoyer et de recevoir des messages à tout moment. Le nom de l'expéditeur, tel que donné par le serveur, doit apparaître en tête du message. Le client envoie le message de l'utilisateur, mais ne met pas à jour directement le contenu du *chat* : encore une fois, on attendra que le serveur nous transmette le message.

4. Équipe

L'utilisateur dispose en permanence d'un panneau permettant de voir l'état de son équipe. Celui-ci permet aussi de demander au serveur de changer l'ordre des pokétudiants ou d'en libérer un. L'état de l'équipe est envoyé par le serveur lorsque celui-ci change.

5. En combat

Lorsqu'un combat a lieu, l'affichage de la carte est remplacé par un affichage dédié permettant d'afficher le nombre de pokétudiants adverses (la taille de l'équipe, ou 1 dans le cas d'un pokétudiant sauvage), chacun des deux pokétudiants en combat, et des boutons pour choisir quelle action effectuer.

Les communications en rapport avec le combat s'effectuent de manière synchrone : le serveur demande au client quelle action est à effectuer. Lorsque l'action de chacun des protagonistes est connue, le serveur les effectue et envoie au client chacune des conséquences de ces actions. Le client et le serveur communiquent aussi de manière synchrone pour un changement de pokétudiant (que celui-ci ait été demandé par le joueur ou qu'il soit dû à un KO).

Lorsque le combat se termine, le serveur le signale au client, et on repasse à l'interface de la carte.

2. Description des protocoles

Les différentes communications s'effectuent par envois de messages composés de lignes de texte terminées par le caractère `\n`.

1. Recherche des serveurs de jeu

Il s'agit d'un simple échange de messages.

— requête du client (*broadcast*} UDP 9000) :

```
looking for poketudiant servers
```

— réponse des serveurs :

```
i'm a poketudiant server}
```

2. Demande de la liste des parties

Le serveur envoie dans un premier temps le nombre de parties *N*, puis dans un deuxième temps pour chaque partie *X* le nombre de joueurs *players-in-game-X* qu'elle contient ainsi que son nom *game-name-X*.

— requête du client (TCP) :

```
require game list
```

— réponse du serveur :

```
number of games N
players-in-game-1 game-name-1
...
players-in-game-N game-name-N
```

3. Créer/rejoindre une partie

Le joueur demande la création d'une partie en indiquant son nom `game-name` composé de caractères alphanumériques et d'espaces (le nom est écrit tel quel, sans guillemets). Si le serveur réussit à créer une partie, un message le signalant est envoyé ; sinon c'est un message d'erreur qui est transmis.

— requête du client (TCP) :

```
create game game-name
```

— réponse du serveur (ok) :

```
game created
```

— réponse du serveur (erreur) :

```
cannot create game}
```

Le procédé est similaire pour rejoindre une partie.

— requête du client (TCP) :

```
join game game-name
```

— réponse du serveur (ok) :

```
game joined
```

— réponse du serveur (erreur) :

```
cannot join game
```

4. Carte

Pour la transmission de la carte, le serveur envoie dans un premier temps ses dimensions (son nombre de lignes, puis de colonnes ; la carte est une grille) au client. Dans un deuxième temps, il envoie la carte ligne par ligne, tel que le caractère espace est une zone neutre, le caractère + est une zone de soins, le caractère * est une zone où un pokétudiant sauvage est susceptible d'apparaître, le chiffre 0 représente la position du joueur et un chiffre entre 1 et 4 donne la position d'un joueur rival. Un rival garde le même numéro de son arrivée dans la partie jusqu'à ce qu'il en sorte.

— envoi du serveur (TCP) :

```
map nb-rows nb-cols
*** 0 +++ 1 ** ...
...
***** ...
```

Pour signaler une action de déplacement, le client envoie la direction souhaitée.

— déplacement vers la gauche :

```
map move left
```

— déplacement vers le haut :

```
map move up
```

— déplacement vers la droite :

```
map move right
```

— déplacement vers le bas :

```
map move down
```

5. Chat

Pour demander la transmission d'un message, le client l'envoie sans guillemets ; c'est le caractère `\n` qui donne sa fin. De même quand le serveur envoie au joueur un message d'un rival. Dans ce dernier cas, le nom du joueur (son adresse et le port de sa socket, composé uniquement de caractères alpha-numériques) est aussi envoyé.

- envoi du message `msg` par le client :
`send message msg`
- envoi du message `msg` du rival `rival-name` par le serveur :
`rival message rival-name msg`

6. Gestion de l'équipe

Dans les commandes suivantes, `X` représente l'indice du pokétudiant concerné (compris entre 0 et 2) dans la liste composant l'équipe.

- remonter le pokétudiant d'une position dans l'équipe :
`poketudiant X move up`
- descendre le pokétudiant d'une position dans l'équipe :
`poketudiant X move down`
- libérer le pokétudiant :
`poketudiant X free`

Lorsque le serveur transmet l'équipe, il envoie dans un premier temps le nombre de pokétudiants `N` qu'elle contient. Il envoie ensuite un pokétudiant par ligne, en précisant dans l'ordre : variété, type de la variété, niveau, expérience dans le niveau courant, expérience à gagner pour le niveau suivant, points de vie actuels, points de vie maximum, attaque, défense, noms et types des attaques.

- envoi de l'équipe :
`team contains N`
`variety type lvl xp-current-lvl xp-next-lvl current-pv max-pv att def attack1-name`
`attack1-type attack2-name attack2-type`
`...`
`variety type lvl xp-current-lvl xp-next-lvl current-pv max-pv att def attack1-name`
`attack1-type attack2-name attack2-type`

3. Combats

Lorsqu'un combat est lancé, le serveur envoie un message au client lui signalant s'il s'agit d'un combat contre un pokétudiant sauvage ou un rival. Il indique aussi le nombre de pokétudiants adverses `N`.

- combat contre un pokétudiant sauvage :
`encounter new wild N`
- combat contre un rival :
`encounter new rival N`

La fin d'un combat ou la mise KO d'un pokétudiant est annoncée par un message indiquant s'il s'agit d'une victoire ou d'une défaite. Si un joueur se déconnecte pendant un combat contre un rival, on donne la victoire à l'autre joueur. Si de l'expérience est gagnée par un pokétudiant, s'il gagne un niveau, ou s'il évolue, le serveur le signale également en fournissant son indice (sa place dans l'équipe, entre 0 et 2).

- mise KO du pokétudiant adverse :
`encounter KO opponent`
- mise KO du pokétudiant du joueur :
`encounter KO player`
- victoire du combat :
`encounter win`
- défaite du combat :
`encounter lose`
- pokétudiant sauvage capturé :
`encounter catch ok`
- échec de la capture du pokétudiant sauvage :
`encounter catch fail`
- fuite face à un pokétudiant sauvage :
`encounter escape ok`

- échec de la fuite face à un pokétudiant sauvage :
encounter escape fail
- le pokétudiant d'indice X du joueur gagne N point d'expérience :
encounter poketudiant xp X N
- le pokétudiant d'indice X du joueur gagne N niveaux :
encounter poketudiant level X N
- le pokétudiant d'indice X du joueur évolue en la variété Y :
encounter poketudiant evolution X Y

Pendant un combat, le serveur peut demander une action, ou un indice de pokétudiant pour un changement (changement demandé par le joueur ou dû à un KO). Le serveur peut aussi signaler qu'une mauvaise commande a été entrée (commande interdite en combat contre un rival ou indice de pokétudiant invalide) ; après une erreur, il soumet à nouveau la requête pour laquelle l'erreur s'est produite.

- demande d'action :
encounter enter action
- demande d'indice de pokétudiant :
encounter enter poketudiant index
- action interdite :
encounter forbidden action
- indice de pokétudiant invalide :
encounter invalid poketudiant index

Il peut enfin envoyer les informations sur les pokétudiants courants du joueur et de l'adversaire, lorsqu'ils entrent en combat ou qu'ils sont blessés : nom de la variété, niveau, arrondi supérieur du pourcentage de points de vie restant, et lorsqu'il s'agit du joueur les noms et types des attaques. Enfin, le serveur transmet aussi l'équipe du joueur lorsqu'un de ses pokétudiants est blessé ou mis KO, avec le même message que décrit précédemment (section 5.2.6).

- informations sur le pokétudiant du joueur :
encounter poketudiant player variety-name lvl pv-percent attack1-name attack1-type
attack2-name attack2-type
- informations sur le pokétudiant adverse :
encounter poketudiant opponent variety-name lvl pv-percent

Les réponses du client aux requêtes du serveur (action, indice de pokétudiant) sont données ci-dessous.

- lancer l'attaque 1 :
encounter action attack1
- lancer l'attaque 2 :
encounter action attack2
- demander un changement du pokétudiant courant :
encounter action switch
- tenter de capturer un pokétudiant sauvage :
encounter action catch
- tenter de fuir le combat contre un pokétudiant sauvage :
encounter action leave
- sélectionner le pokétudiant d'indice X :
encounter poketudiant index X