

# Interfaces Tactiles et Interfaces Réparties sur Plusieurs Supports

## Spécifications de l'application FAST : Fast-Action Space Tactics

<b>Description du sujet</b>	<b>4</b>
Problème d'usage	4
1. Orientation	4
2. Utilisation des smartphones	4
3. Compréhension des mini-jeux	4
4. Complexité du jeu	4
5. Equilibrage	5
Système interactif actuel	5
Ecran partagé	5
Appareils personnel dédié	5
Multijoueur synchrone	5
Personas, cycle de vie et Scénarii	6
Personas	6
Paul, le global élite	6
Lucas, le noob	6
Cycle de vie de l'application	6
Scenarii	7
Analyse de la situation de crise:	7
Déplacement de l'équipage:	7
Résolution d'un problème situé dans une salle:	8
<b>Système interactif visé</b>	<b>9</b>
Table tactile	9
Réalisation	9
Smartphone (mini-jeux)	9
Réalisation	9
Ordinateur avec webcam reliée à opencv (mini-jeu de tir)	9
Réalisation	10
Atomes	10
Communication dans l'application	10
Rouge : broadcast d'événement Smartphone	11
Verte : interaction avec le vaisseau	11
Violette : communication serveur-smartphone(s)	12
Jaune : communication serveur-table	12
Rose : communication entre le serveur et le module de tir	12
Matériel nécessaire	12

Agencement du système de jeu	12
Experience utilisateur	13
Difficulté adaptative	13
Negative Feedback loops	14
Identité visuelle	14
Coopération	14
Au travers des mini-jeux	14
Broadcast d'événement	15
Coopération directe	15
Coopération globale	15
Coopération indirecte	15
<b>Règle générale des jeux</b>	<b>16</b>
Template général	16
Splash Screen	16
Interface en jeu	17
Protocole commun	18
Données d'initialisation globale	18
Données d'échange	18
Liste exhaustive des mini jeux	18
<b>Extinction d'incendie</b>	<b>18</b>
Template visuel	19
Splash Screen	19
Mini jeu	19
Variante Solo et Multijoueurs	20
Variante Solo	20
Variante Multijoueurs	20
<b>Protection météore</b>	<b>20</b>
Template visuel	21
Variante Solo et Multijoueurs	22
Variante Solo	22
Variante Multijoueurs	22
<b>Appuyer sur l'interrupteur</b>	<b>22</b>
Template visuel	23
Variante Solo et Multijoueurs	23
Variante Solo	23
Variante Multijoueurs	23
<b>Balistique</b>	<b>23</b>
Template visuel	24
Smartphone	24
PC	24

Variante Solo et Multijoueurs	26
Variante Solo	26
<b>Repartition du travail</b>	<b>26</b>
Alison BERLIOZ:	26
Nicolas BIGHETTI:	26
Flavian JACQUOT:	26
Steven LEQUIENT:	26
<b>Annexe</b>	<b>26</b>
Protocol réseau	27
<b>Violet : Communication double sens Serveur-Smartphone</b>	<b>28</b>
Serveur vers Smartphone	28
Connection à la partie	28
Début d'un mini-jeu	28
Début d'un mini jeu multi	28
Fin d'un mini jeu	29
Protocole par mini jeu	29
Données d'initialisation globale	29
Données d'échange	29
FAST_GAME_FIRE	29
Initialisation	29
Broadcast update	30
Format des données	30
Type de feu	30
Données d'initialisation	30
Données échangés durant la partie	30
FAST_GAME_METEOR	30
Format des données	31
Données d'initialisations	31
Contact avec le vaisseau	31
FAST_GAME_SWITCH	31
Format des données	31
FAST_GAME_BALLISTIC	31
Format des données	31
Données de dégâts	31
Smartphone vers Serveur	31
Inscription à un mini jeu	31
Fin du mini-jeu	32
<b>Rouge : Distribution des events Smartphone</b>	<b>32</b>
Smartphone vers serveur	32
Serveur vers smartphone	32

Liste exhaustive des events	32
<b>Orange : Communication Table-Serveur</b>	<b>33</b>
Liste exhaustive des codes d'avarie	33
Déplacement personnage	33
Activation bonus	33
Apparition d'une avarie	34
Résolution d'une avarie	34

# Description du sujet

**FAST : Fast-Action Space Tactics** est un jeu d'action/stratégie à 4 joueurs, utilisant une table tactile ainsi qu'un smartphone par joueur.

Le jeu a pour but d'avoir un rythme très nerveux et chaotique, orienté sur la gestion de crise ainsi que la communication.

Le jeu s'oriente autour de combat spatial, où la carte des pièces du vaisseau est affichée sur la table tactile, certaines pièces ayant des fonctions précises, dans lesquelles les joueurs devront se rendre rapidement afin d'effectuer des réparations ou d'activer des mécanismes, au moyen de mini-jeux sur smartphone.

Les personnages sont représentés par des atomes que les joueurs peuvent déplacer sur la table tactile de pièce en pièce.

## Problème d'usage

Le jeu pose plusieurs problèmes d'usage:

### 1. Orientation

Le jeu se passe principalement sur la table tactile, et a pour but d'être rapide et chaotique. Les joueurs doivent être capables de prendre des décisions rapides à n'importe quel moment, depuis n'importe quel côté de la table. Les icônes, commandes et informations doivent toutes être visibles et compréhensibles rapidement depuis n'importe quel côté de la table.

### 2. Utilisation des smartphones

Chaque joueur doit avoir son smartphone pour effectuer les mini-jeux à tout moment, et l'on peut donc présumer qu'ils doivent alors effectuer toutes leurs interactions avec la table tactile d'une seule main.

De même, toutes les transitions entre la table tactile et le smartphone doit être effectué avec un court délai, ou une activation du mini-jeu, le temps que le joueur s'équipe de celui-ci.

### 3. Compréhension des mini-jeux

Les mini-jeux se doivent d'être très faciles à comprendre, les joueurs ne devant pas mémoriser chaque type, mais plutôt être capable de réagir rapidement au problème posé, tout en restant concentré sur le reste de la situation de crise, et ce malgré la diversité des mini-jeux proposé.

#### 4. Complexité du jeu

La complexité d'un jeu est la quantité d'apprentissage nécessaire afin de pouvoir y jouer correctement. Plus l'on rajoute de mécaniques et de problèmes à régler, plus le jeu sera chaotique, mais aussi aussi complexe, avec de nombreuses choses à retenir.

Il ne faut pas confondre la complexité avec la profondeur, qui est la quantité de choix et de possibilités donnant de la durée et rejouabilité au jeu.

Une grande complexité a donc un fort impact négatif dans l'utilisabilité du jeu, à savoir essentiellement dans son efficience, mais aussi dans la satisfaction, selon les utilisateurs.

#### 5. Équilibrage

Sans un bon équilibrage du jeu, à savoir une gestion de la difficulté, mais aussi de l'importance de certaines actions par rapports à d'autres, le jeu peut rapidement devenir insatisfaisant, mais aussi complexe à prendre en main.

## Système interactif actuel

Actuellement, les expérience de jeux coopératives se présentent de plusieurs manière différentes:

### Écran partagé

Les joueurs jouent tous sur le même écran, soit cet écran est partagé en n-zones pour n joueurs.

L'avantage est qu'il n'y a besoin que d'un seul écran pour joueur. Le désavantage est que les joueurs sont souvent contraint à utiliser des contrôleurs qu'ils ne connaissent pas. Ce qui peut entraîner des confusions sur les contrôles à utiliser pour interagir avec le jeu.

### Appareils personnel dédié

Les joueurs disposent chacun d'un appareil dédié à l'activité de jeu en groupe.

L'avantage est que chacun utilise un appareil qu'il connaît pour jouer, le désavantage est que bien trop souvent la coopération est rompu par le repli des participants sur leur appareils personnel.

### Multijoueur synchrone

Les joueurs, à l'aide de leur smartphone, fournissent à tour de rôle des données (question, réponse, couleur, etc) à une application web. Celle ci utilise ensuite les données fournies par les joueurs pour générer la question d'un quizz, les réponses à une question proposée, ou encore un test quelconque aux joueurs.

L'avantage ici est que l'expérience est commune, tout en laissant les joueurs utiliser des appareils avec lesquels ils sont familier. Toutefois, on est bien loin des jeux plus "traditionnels" et l'expérience reste bien trop encadré, voir dirigiste. Tout est remplaçable par des petits bouts de papier et un saladier.

# Personas, cycle de vie et Scénarii

## Personas

Paul, le global élite



**Nom:** Paul Ratorze

**Age :** 22 ans

**Profession :** Streamer professionnel

**Niveau de formation:** Très grande expertise en jeux vidéos

**Connaissance et expérience des jeux vidéos:** Joueur régulier, expérimenté. A une bonne connaissance des jeux et n'a pas besoin d'un tutoriel en profondeur, s'acclimate rapidement à de nouveaux concept mais a besoin d'une expérience diversifiée.

**Usage des technologies :** Professionnel et personnel

**Motivation pour utiliser le système :** S'amuser, pousser la coopération au maximum en dépit d'un challenge croissant et demandeur.

**Bête noire:** S'ennuyer. Les tutoriaux beaucoup trop long. La simplification à outrance.

**Citation:** "Quand je gagne 20 fois contre toi d'affilé, tu dois commencer à te poser des questions, gamin !"

Lucas, le noob



**Nom:** Lucas Sual

**Age:** 15 ans

**Profession :** Lycéen

**Niveau de formation :** Novice

**Connaissance et expérience des jeux vidéos:** Joue à des jeux smartphone linéaires et peu complexes.

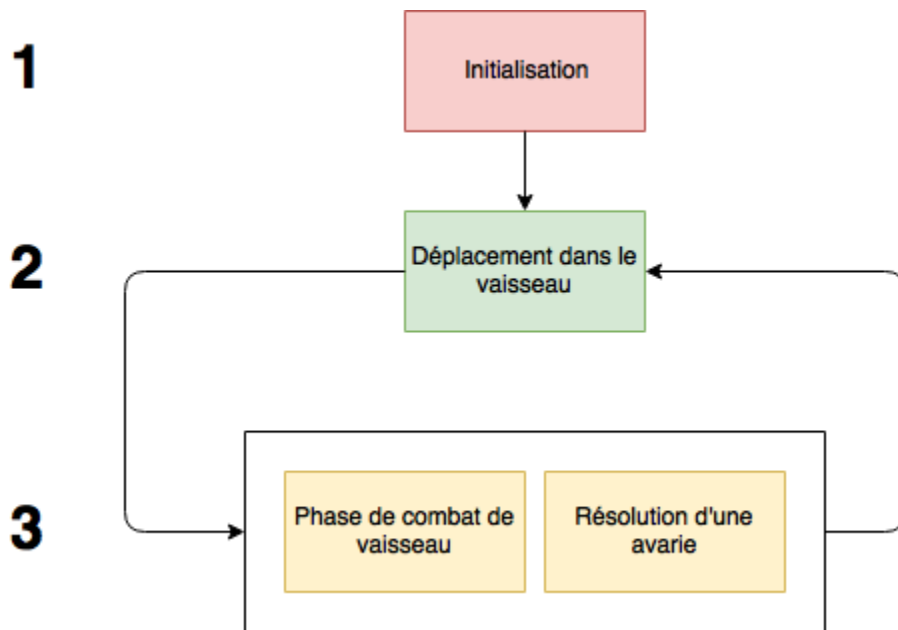
**Usage des technologies :** Personnel

**Motivation pour utiliser le système :** Veut jouer avec ses amis pour passer un moment agréable avec eux. Ne veut pas faire d'effort pour comprendre un système.

**Bête noire:** Lire le moindre texte. Analyser rapidement.

**Citation :** "J'ai pas compris, j'ai pas lu, c'est nul, je me casse"

## Cycle de vie de l'application



## Scenarii

Analyse de la situation de crise:

**Utilisateur:** Paul Ratorze, le pro

### Scénario:

Paul Ratorze: "Un combat contre un autre vaisseau a commencé. Je regarde alors la table tactile afin de voir les différents problèmes. En effet, il y a un incendie dans une salle, des meteores qui arrivent vers une autre, ainsi qu'un problème de disjoncteur dans une dernière. Je décide alors en vitesse d'un plan d'action en décidant de me diriger vers le problème le plus proche, et en disant aux autres les tâches qu'ils devraient accomplir."

### Dispositif visé:

Table tactile

Quel utilisateur: **Tous**

Tâche à plusieurs **collaborative**

Où effectuer la tâche? **Table tactile et téléphone.**

A quel étape de l'application se situe la tâche? Étape 3: **Phase de combat du vaisseau**

Quel type de donnée? **Public**

Quelle interaction? Pourquoi? **Tactile et tangible (il faut déplacer les atomes)**

Quel widget / représentation, pourquoi? **Plan avec Icônes rondes/multiples pour analyser la situation**

Déplacement de l'équipage:

**Utilisateur:** Lucas Sual, le noob

### Scénario:



Lucas Sual: "Notre vaisseau se fait tirer dessus, et j'ai vraiment besoin d'aller vers l'incendie à trois pièces de là. Du coup je saisis la figurine qui représente mon personnage, et je la pose sur la salle où j'ai besoin d'être. Je vois alors mon personnage se déplacer depuis la où il était jusqu'à la salle où j'ai posé la figurine. Franchement, c'est pratique, je sais forcément où se trouve mon personnage quand il ne bouge pas, et je peux suivre sur la table tactile ses déplacements quand il bouge."

**Dispositif visé:**

Table tactile et atomes

Quel utilisateur: **Tous**

Est-ce une tâche à plusieurs? **Seul**

Où effectuer la tâche? **Table et atome**

A quel étape de l'application se situe la tâche? **Étape 3: Phase de combat du vaisseau**

Quel type de donnée? **Public**

Quelle interaction? Pourquoi? **Interaction tangible pour l'immersion**

Quel widget / représentation, pourquoi? **Figurine, pour l'ambiance de jeu**

Résolution d'un problème situé dans une salle:

**Utilisateur:** Lucas Sual, le noob

**Scénario:**

Lucas Sual: "J'arrive dans la salle en feu, et je concentre alors toute mon attention sur mon téléphone qui affiche l'incendie. Je me dépêche, car je sais que l'incendie abîme le vaisseau. Je touche les flammes de ma couleur, et je souffle sur mon téléphone pour les éteindre."

**Dispositif visé:**

Smartphone

Quel utilisateur: **Tous**

Est-ce une tâche à plusieurs? **Possiblement (selon la gravité de l'incendie, il peut nécessiter plusieurs personnes)**

Où effectuer la tâche? **Smartphone**

A quel étape de l'application se situe la tâche? **Étape 3: résolution d'une avarie**

Quel type de donnée? **Public**

Quelle interaction? Pourquoi? **Interaction vocale, car c'est original et satisfaisant**

Quel widget / représentation, pourquoi? **Représentation de la salle en feu avec des flammes qui diminuent puis s'éteignent**

# Système interactif visé

Le système interactif utilisé est composé d'une table tactile, en combinaison avec 4 atomes ainsi que 4 smartphones, chaque joueur en possédant un.

La table tactile affiche la carte du vaisseau, ainsi que des informations liées à son état actuel.

Elle affiche aussi les informations sur l'état courant de la partie.

Chaque joueur se voit attribuer un atome en début de partie, à l'effigie de son personnage, et dont la couleur correspond aux éléments avec lesquels il peut interagir. Celui-ci le représente dans le jeu. Il l'utilise pour se déplacer dans le vaisseau et pour interagir avec l'environnement de jeu. Cette couleur est mise en avant afin qu'à tout moment l'utilisateur puisse identifier son rôle et/ou sa position dans le jeu en cours, qu'il soit concentré sur la table ou sur son smartphone.

Les smartphones permettent de réaliser des actions au niveau de la salle occupée par son personnage.

## Table tactile

**But** : donner corps à la coopération, centraliser les informations générale

### Réalisation

Interface lisible de n'importe quel côté de la table

Affichage des informations critique le plus lisiblement possible

Permet un déplacement facile grâce aux atomes, permettant de savoir à tout moment la position de son personnage.

## Smartphone (mini-jeux)

**But** : donner à chacun un accès individuel à la coopération.

### Réalisation

Vibration et lumière en cas d'échec

Multiplicité des interactions pour tenir le joueur sur ses gardes et varier l'expérience

Mini jeu multi joueur pour faire face à l'évolution des situations de crises mal contrôlées.

## Ordinateur avec webcam reliée à opencv (mini-jeu de tir)

**But** : avoir un mini-jeu original forçant les joueurs à se concentrer sur autre chose que la table tactile et le smartphone, ajoutant au chaos

## Réalisation

Lancer d'une balle colorée en scratch sur une surface en ouate

Caméra tournée vers la surface en ouate sur laquelle sont placés des repères colorés

## Atomes

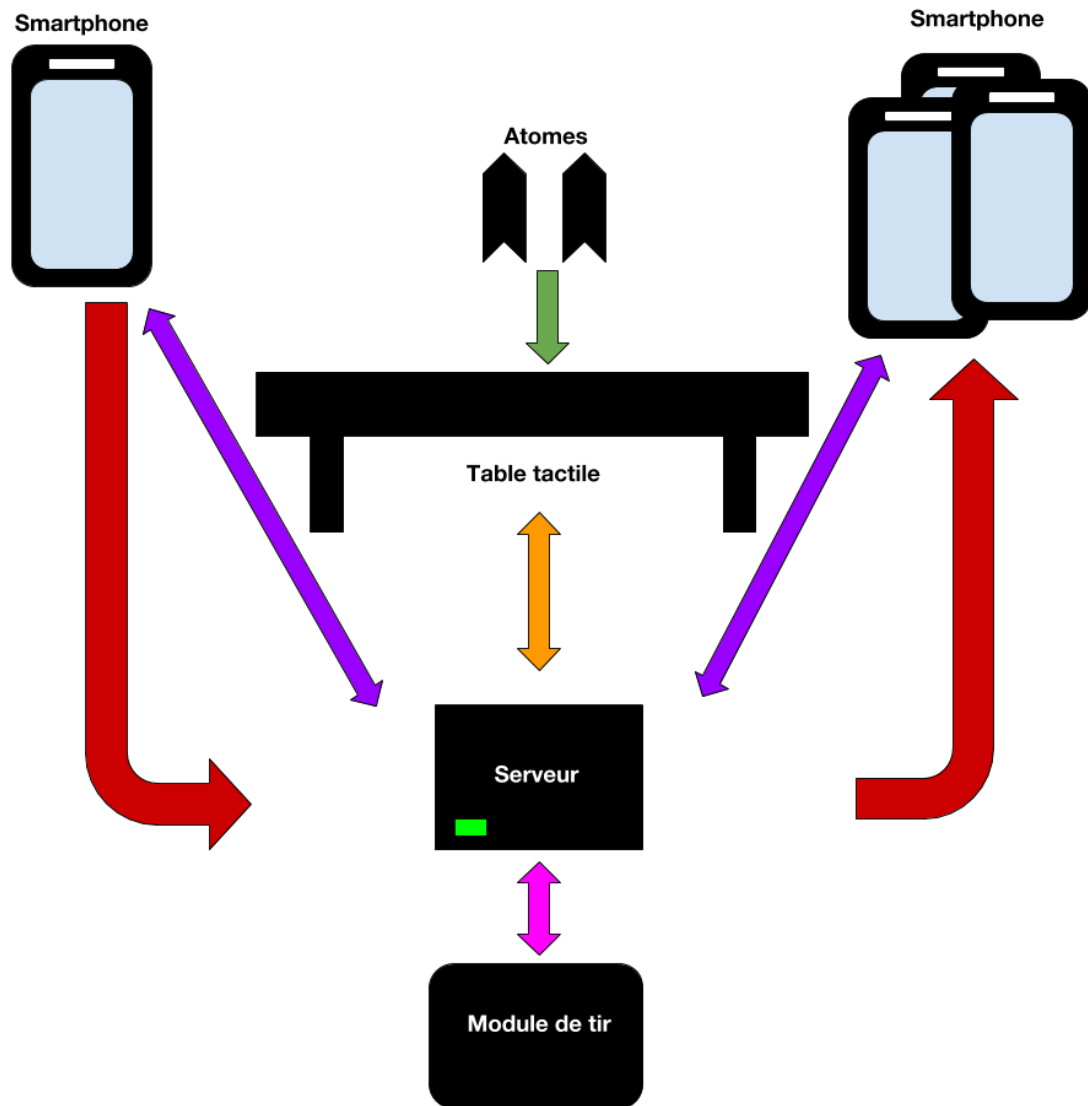
But: Concrétiser l'information pour permettre une lecture du jeu plus clair aux joueurs dans l'action.

Réalisation:

Déplacement

## Communication dans l'application

Les différents éléments communiquent comme indiqué sur le schéma, nous allons cependant détailler le sens derrière les flèches:



#### Rouge : broadcast d'événement Smartphone

Durant les mini-jeux, divers événements peuvent avoir une répercussion sur le téléphone: affichage coupé, vibration. Puisque les joueurs sont tous à bord du même vaisseau, refléter ses événements sur l'ensemble des appareils permet de tenir au courant les autres joueurs de l'évolution de la situation au travers du smartphone.

Lorsque l'un de ses événements se produit, le téléphone notifie le serveur qui va broadcaster à tous les joueurs présents dans la partie l'événement. Leur smartphone pourront alors reproduire l'événement (vibration, affichage coupé, son).

#### Verte : interaction avec le vaisseau

Les interactions avec le vaisseau se font au travers d'atomes disposés sur la table tactile. Les déplacements et les activations de bonus se font au travers des atomes.

Violette : communication serveur-smartphone(s)

Le serveur orchestre l'état des smartphones, le lancement des mini-jeux, le liens lors de la coopération dans la gestion des avaries, etc...

Jaune : communication serveur-table

Le serveur sert de relais entre la table et les parties prenantes.

Rose : communication entre le serveur et le module de tir

Le module de tir étant représenté par une cible de fléchette physique, et d'un appareil équipé d'une caméra, la communication entre ce module et le reste de l'application passe par le serveur, plus précisément par un canal dédié.

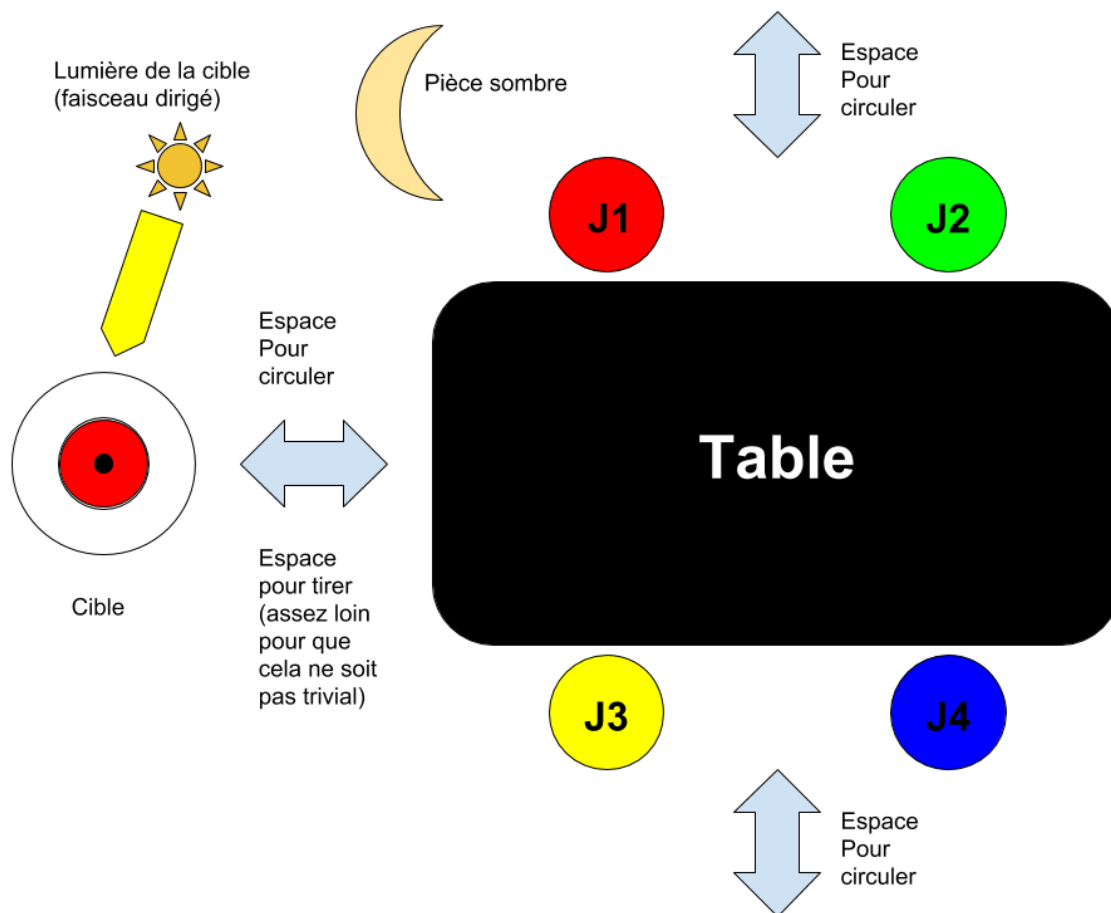
## Matériel nécessaire

*Tout le matériel ci-dessous devra être connecté au même réseau WI-FI local (dans la mesure du possible)*

- 1 table tactile surface compatible tag
- 4 smartphone (Android v5 min) avec les capteurs/Actionneurs suivants
  - Gyroscope
  - Micro
  - Camera
- 4 pions / tag
- 1 système openCV ready
- 1 webcam
- 1 cible + 1 projectile
  - le projectile en scratch doit avoir une couleur dont les bornes en hsv sont les plus différentes possibles et les plus différentes possibles des autres couleurs visibles par la caméra
- 1 lampe avec ampoule à lumière blanche avec faisceau directionnel

## Agencement du système de jeu

L'éco-système de jeu devra être disposé, dans un environnement adapté, de la façon suivante



## Experience utilisateur

Notre projet est un jeu, ce qui présente la contrainte supplémentaire de devoir donner envie d'y jouer. Ainsi, en plus de tout le travail effectué au niveau du développement, un travail de conception concernant le design du jeu doit être effectué afin de cerner avec précision ce qui le rend intéressant à jouer, et de trouver des astuces pour améliorer l'expérience utilisateur.

L'intérêt du jeu est ici de répondre le plus rapidement à une situation urgente, au moyen d'interactions originales. L'échec et la réussite sont ici des facteurs d'égales importances pour le plaisir de jeu.

## Difficulté adaptative

Notre jeu s'oriente autour d'une action chaotique et nerveuse devant inciter à la précipitation et à une prise de décision rapide en situation de crise. Cela insinue donc que la situation soit en effet une crise, ce qui peut ne pas être le cas dans le cas d'utilisateurs expérimentés. A ces fins, plutôt que d'implémenter plusieurs niveaux de difficulté, il est plus intéressant de s'inspirer ici de Resident Evil IV et de Left 4 Dead 2, et d'inclure de la difficulté adaptative ([cf vidéo additionnelle explicative, durée : 5mn59](#)).

Le principe ici est de conserver une situation de stress pour les joueurs en augmentant la difficulté, via le nombre et l'importance des incidents à résoudre, quand leur barre de vie est

pleine, et de la diminuer au fur et à mesure que celle-ci se vide, faisant que chaque action à un impact critique sur la survie des joueurs.

## Negative Feedback loops

Un autre problème est l'adaptation est la difficulté des mini-jeux, qui doivent être fini dans un temps limité, qui peut être trop dur à aborder pour des joueurs novices, ou trop confortable pour des joueurs expérimentés. On utilise alors ici le principe des negative feedback loops, chargés de faciliter la vie des joueurs en difficulté, et d'intensifier les problèmes des joueurs en situation de confort. On utilise pour cela le temps moyen de résolution des derniers incidents, afin de jauger la compétence du groupe, et l'on ajuste alors la difficulté ([cf vidéo additionnelle explicative, durée 13mn10](#)).

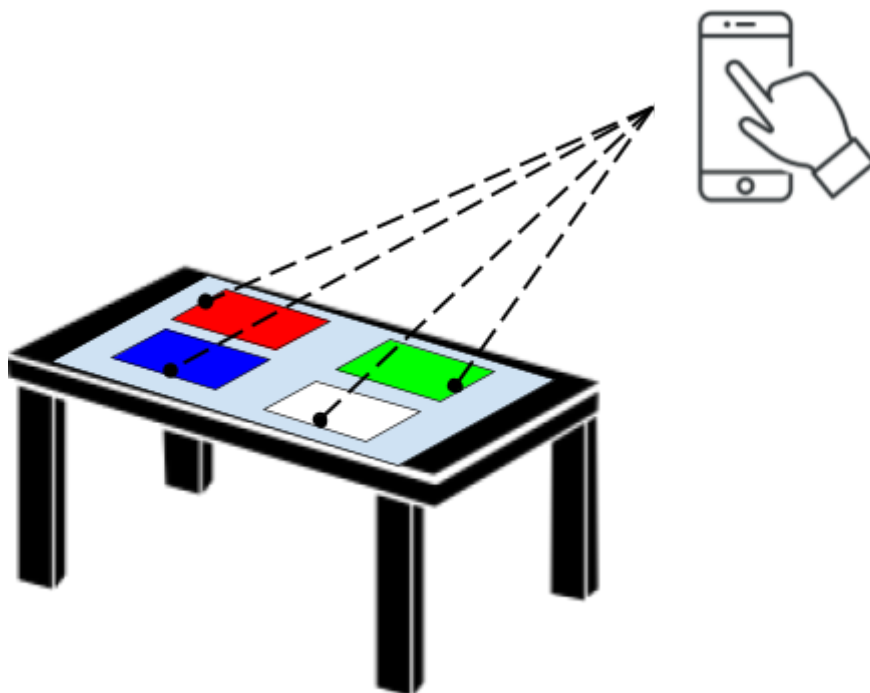
## Color IO : Liaison table et téléphone

Au lieu d'utiliser un qr code pour se connecter à la table, nous avons pensé à une autre interaction. L'importance est dans le geste et la praticité.

Il suffit de poser son téléphone sur la zone de couleur affichée par la table et d'effectuer un touché sur celui-ci. L'interaction devrait être tellement simple qu'un enfant pourrait le faire.

Dans la première version la zone de couleur était petite et sans instruction.

Dans la seconde version il est prévu d'agrandir la zone et de rajouter soit un texte explicatif, soit une représentation illustrée de l'action à effectuer comme ceci:



(non représenté, les figurines correspondants aux personnage posé sur ces mêmes zones. Cette interaction nous permet d'avoir une connexion ludique et originale dès le début du jeu car elle se produit juste après le démarrage d'une partie.

## Identité visuelle

Afin de conserver une cohérence visuelle dans l'application, nous avons décidé d'avoir l'intégralité des sprites du jeu avoir une apparence similaire, proche du pixel art.

Les couleurs des flammes de l'incendie dédiés à chaque joueur ont inspiré le choix d'associer une couleur précise à chaque joueur correspond aux 4 éléments, et de changer l'incendie en tempête élémentaire.

Ainsi, nous avons la terre/nature en vert, l'eau en bleu, le feu en rouge et l'air en violet. Le violet reste une couleur communément associée à l'air, même s'il est moins courant que le blanc ou le jaune.

Afin d'avoir des pierres vertes, la terre est représenté par des émeraudes, ce qui contribue à l'impression de puissance du personnage correspondant, mais aussi de l'élément dans les tempêtes élémentales.

Afin d'identifier les personnages plus facilement, Alison a dessiné chacun des 4 personnages, chacun ayant son élément associé afin de pouvoir s'occuper de celui-ci dans les incidents correspondants.

Le design de chacun de ces personnages est passé par la recherche d'inspirations dans de nombreux jeux et films d'animation.

## Coopération

### Au travers des mini-jeux

Le principal moyen de faire coopérer les joueurs est au travers des mini jeux.

Dans un premier temps, les joueurs doivent coopérer pour résoudre les avaries au travers des mini jeux tout en survivant le plus longtemps possible.

La difficulté adaptative va forcément faire basculer le jeu à un point où il y aura plus d'avaries à résoudre que de joueurs disponibles. Il est alors essentiel que les joueurs communiquent afin de démêler la situation en minimisant les pertes.

De plus certains mini jeux nécessitent une coopération direct, au travers de mini jeux multi-joueurs. Les avaries concernées par les mini-jeux multijoueurs ne peuvent évidemment pas être résolues si le nombre de joueurs rassemblés n'est pas suffisant. Il faudra donc que les joueurs parviennent à se répartir entre les différents incidents tout en trouvant le temps de se rassembler pour résoudre les problèmes le nécessitant.

### Broadcast d'événement

Au cours d'une partie, des événements sont diffusés vers tous les smartphones des joueurs. Ces événements, actuellement limités à des vibrations, ont pour but de donner de la



cohérence au jeu indépendamment de l'état actuel de la table de jeu ou de l'état des smartphones connectés.

Par exemple, si un joueur échoue à protéger le vaisseau d'un projectile, alors tous les téléphones vibreront.

Cette vibration a deux objectifs :

- D'abord elle permet de prévenir tous les joueurs présent que le vaisseau a subi des dégâts. Et donc qu'ils s'éloignent de leur objectif principal : survivre en tant que groupe.
- Deuxièmement, elle va aussi interférer avec d'autres mini-jeux qui utilisent la vibration pour communiquer une information. Ainsi la réussite du groupe passe par la réussite individuel à un mini jeu, et ce même s'il n'est pas à première vu multijoueurs.

On augmente donc la coopération du groupe indépendamment du nombre de joueur impliqué dans la résolution d'une avarie, même lors d'une activité en solo il n'y a pas d'isolement.

C'est aussi un moyen pour les joueurs d'être tenu au courant de l'évolution de la situation : peu de vibration signifie que le vaisseau subit potentiellement moins de dégâts que s'il y a plus de vibrations.

On dispose donc d'un vecteur d'émission supplémentaire pour les informations publiques du système.

### Coopération directe

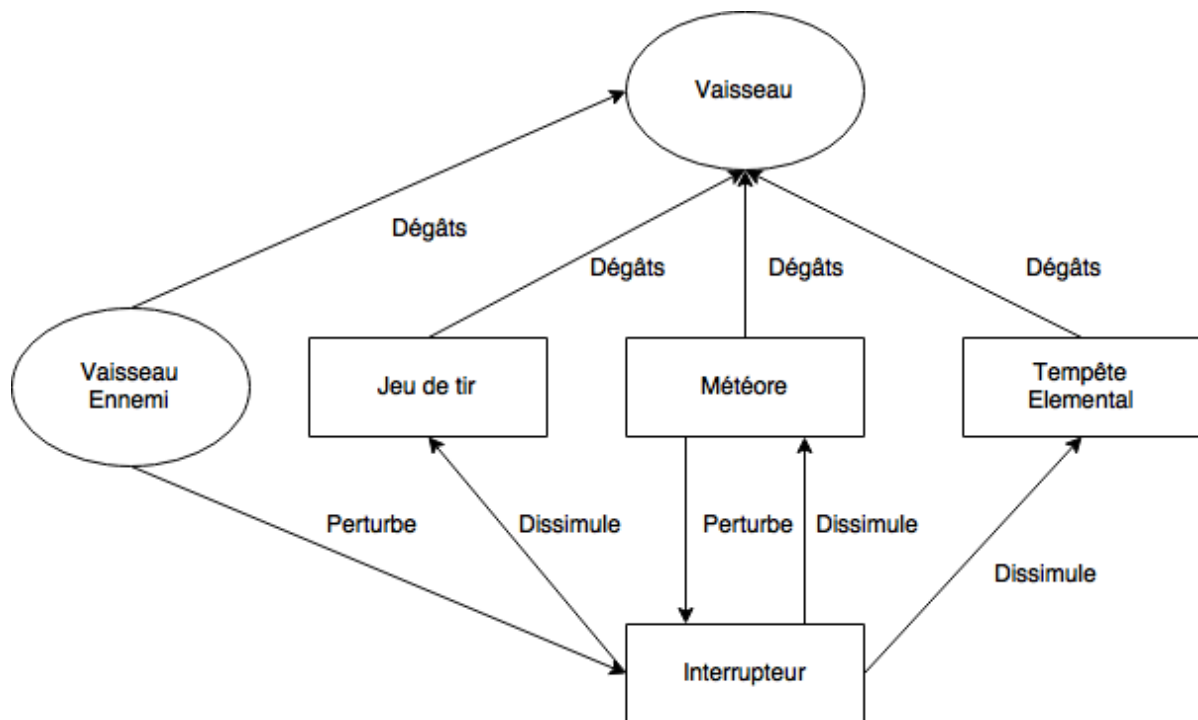
La coopération direct est la coopération qui a lieu lors d'une partie d'un mini jeu multi joueur. Les joueurs sont rassemblé dans la même partie et doivent alors coopérer pour résoudre la situation chacun depuis leur smartphone

### Coopération globale

La coopération globale est la coopération qui a lieu lorsque les joueurs se répartissent les tâches à réaliser pour sauver le vaisseau. La communication est la clé pour réussir à survivre aux différentes avaries qui se produiront en parallèles. Cet aspect régule aussi la difficulté adaptative qui permet à tous les types d'équipages de profiter du jeu à leur rythme.

### Coopération indirecte

La résolution des avaries est d'une certaine façon régie de manière hiérarchique. En effet, comme nous pouvons le constater avec le schéma ci-dessous :



On voit que l'ensemble des avaries/mini-jeux interagissent. Il est crucial d'en résoudre certains pour faciliter, voir tout simplement permettre, la résolution des autres, tout en survivant.

## Règle générale des jeux

Le vaisseau subit des avaries aléatoires tant que le vaisseau ennemi n'est pas détruit. Il faut alors résoudre les mini-jeux afin d'éviter de prendre des dégâts, enlever un malus, ou attaquer (selon son type).

Si la vie du vaisseau atteint 0, la partie est terminée.

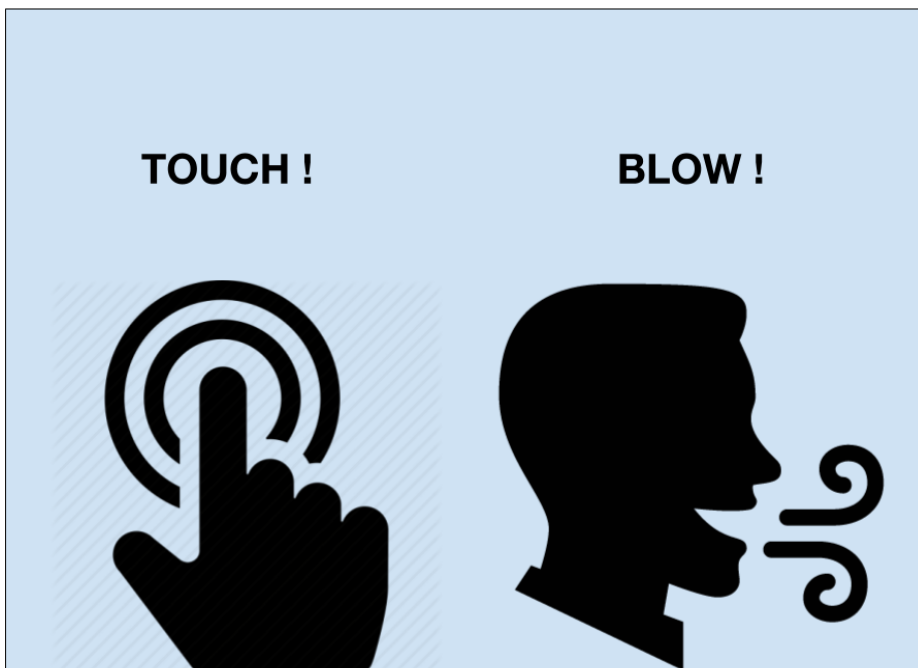
A l'exception du jeu de tir, qui nécessite que le joueur se lève afin de pouvoir viser et tirer sur la cible, tous les mini jeux se déroulent autour de la table, ou les joueurs, assis, résolvent les avaries depuis leur smartphones.

Encore une fois à l'exception du jeu de tir, tout les mini jeux peuvent avoir lieu dans n'importe laquelle des salles du vaisseau, il n'est pas possible de prédire de manière certaine le lieu d'incident d'une avarie dont l'emplacement n'est pas fixe.

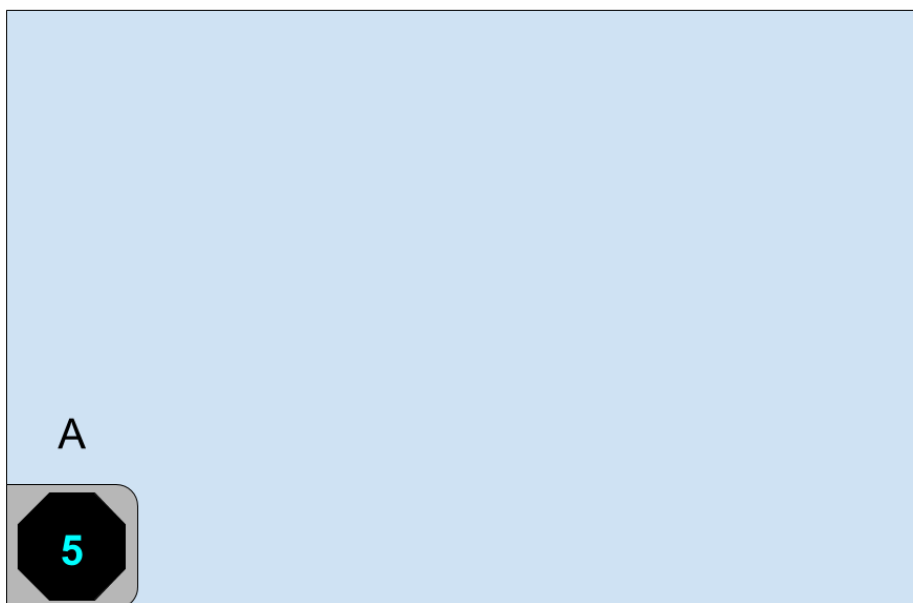
## Template général

### Splash Screen

Affiché en début de partie, donne l'indication aux joueurs des interactions dont il devra faire usage pour le mini jeu à venir. De plus pour chaque interaction utilisé dans le mini jeu à venir, le splash screen réagi à cette interaction ce qui permet à l'utilisateur de se former durant le chargement ou en attendant les autres joueurs pour un mini jeu multi-joueur.



Interface en jeu



A : Indique le temps restant en secondes pour résoudre l'incident avant une pénalité appliqué au vaisseau

## Protocole commun

### Données d'initialisation globale

```
{
    time_limit : int,
    game_type : [cf identifiant mini jeux],
    game_data : [voir les données d'initialisation au cas par cas]
}
```

### Données d'échange

```
{
    timestamp: long,
    id: string,
    data: any
}
```

## Liste exhaustive des mini jeux

Mini jeu	Identifiant
<a href="#">Incendie</a>	FAST_GAME_FIRE
<a href="#">Météore</a>	FAST_GAME_METEOR
<a href="#">Switch</a>	FAST_GAME_SWITCH
<a href="#">Ballistic</a>	FAST_GAME_BALLISTIC

## Extinction d'incendie

**But:** arreter une tempête pour éviter au vaisseau de perdre continuellement de la vie. L'avarie peut se produire à tout moment, étant donné qu'elle est causée par le vaisseau ennemi qui attaque l'équipe. L'ampleur de l'avarie est déterminée par un facteur aléatoire dépendant de la difficulté.

### Interaction:

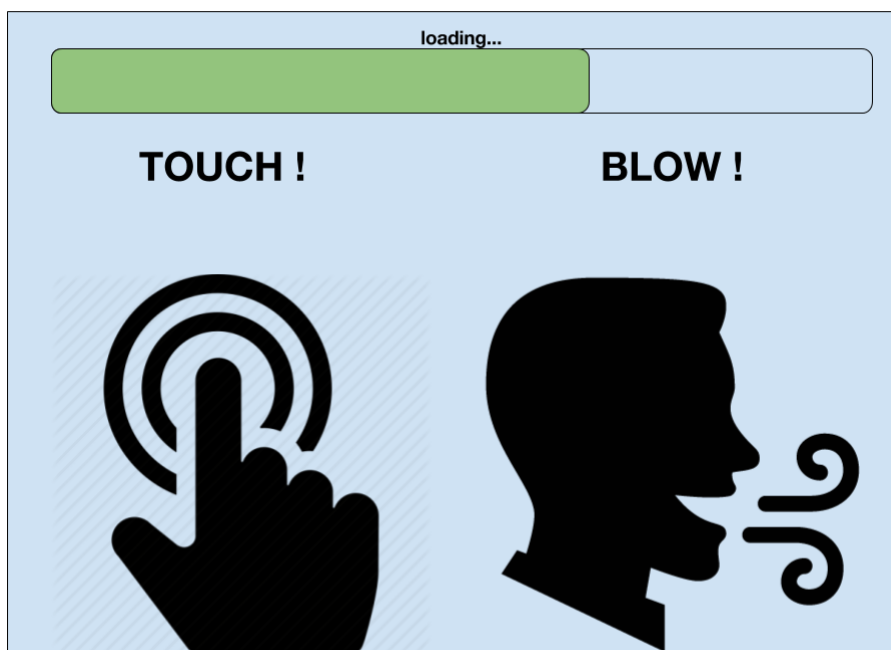
On touche les éléments correspondant à celui de son personnage, et on souffle pour les détruire (Sur smartphone).

Avarie associée	Incendie
-----------------	----------

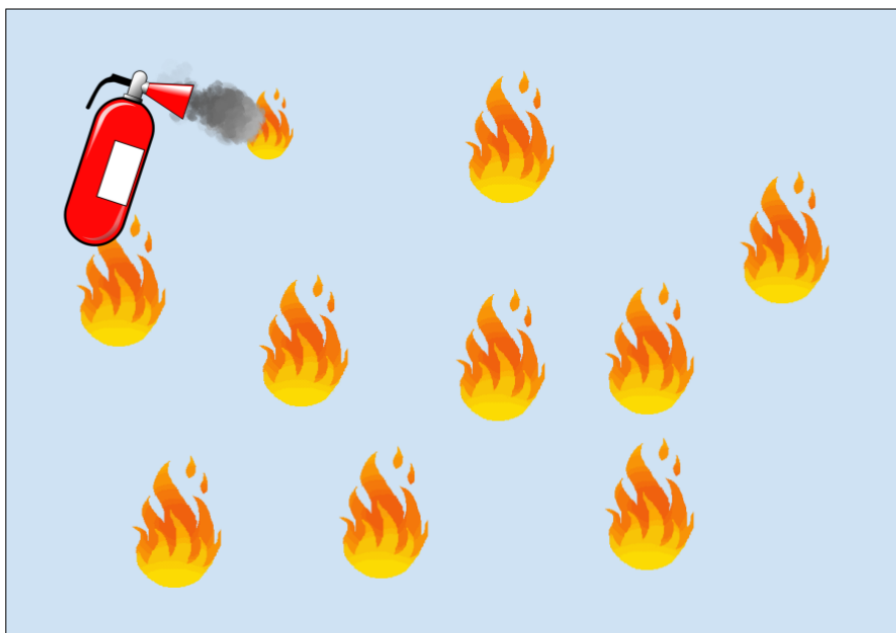
<b>Solo</b>	Oui
<b>Multijoueur</b>	Oui
<b>Nombre de joueurs</b>	1 à 4
<b>Contrôle</b>	Tactile / Micro (souffle)
<b>Condition de réussite</b>	Éteindre l'incendie dans le temps imparti

## Template visuel

### Splash Screen



## Mini jeu



## Variante Solo et Multijoueurs

### Variante Solo

Toucher l'écran tactile positionne l'extincteur, souffler dans le micro expulse de la mousse étouffante.

A 1 joueur il n'y a qu'un seul type de flamme qu'il faut éteindre le plus rapidement possible.

### Variante Multijoueurs

Toucher l'écran tactile positionne l'extincteur, souffler dans le micro expulse de la mousse étouffante.

A 2, 3 ou 4 joueurs, il y a autant de couleurs de flammes que de joueurs qu'il faut éteindre le plus rapidement possible. Le nombre de joueur varie en fonction de la gravité de l'avarie.

## Protection météore

**But:** Déplacer un bouclier pour empêcher des météores de frapper le vaisseau

Des météores s'abattent sur le vaisseau, et le joueur doit déplacer un bouclier afin de les bloquer, pour éviter de perdre de la vie, et de faire vibrer tous les smartphones.

Pour cela, le joueur doit tourner son téléphone afin d'utiliser son gyroscope.

**Interaction:**

Incliner le téléphone à gauche ou à droite afin de déplacer le bouclier, afin de le mettre sur la trajectoire des météores. Le bouclier ne se déplace que sur un axe horizontal en bas de l'écran en format paysage, tandis que les météores arrivent par le haut en se déplaçant vers le bas.

Les vibrations causées par l'échec du mini-jeu sont envoyés sur les smartphones de tous les joueurs, qu'ils soient en mini-jeu ou non.

<b>Avarie associée</b>	Pluie de météore
<b>Solo</b>	Oui
<b>Multijoueur</b>	Non
<b>Nombre de joueurs</b>	1
<b>Contrôle</b>	Gyroscope
<b>Condition de réussite</b>	Bloquer tout les météores

**Template visuel**

## Variante Solo et Multijoueurs

### Variante Solo

Incliner le téléphone contrôle l'intercepteur.

Un météore qui entre en contact avec l'intercepteur est détruit. Si le météore dépasse l'intercepteur, il touche le vaisseau. Cela entraîne alors des dégâts sur le vaisseau, et tout les téléphones des joueurs vibrent alors.

### Variante Multijoueurs

Pas de variante multijoueurs

## Appuyer sur l'interrupteur

**But:** Trouver un interrupteur dans le noir pour rétablir le courant

La lumière de la salle s'éteint, car il n'y a plus de courant. La salle devient noire, et il est nécessaire pour le joueur qui s'y trouve d'utiliser les vibrations pour trouver l'interrupteur et résoudre l'avarie.

L'avarie empêche de voir le contenu de la pièce, et empêche de résoudre les autres avaries. Il est crucial de résoudre l'incident au plus vite pour avoir suffisamment d'informations et résoudre les avaries supplémentaires si elles existent.

De plus, allumer l'interrupteur est nécessaire à la résolution des autres avaries de la pièce. Il est ainsi par exemple impossible de se défendre d'une pluie de météores depuis la pièce si l'on met trop longtemps à résoudre l'avarie.

### Interaction:

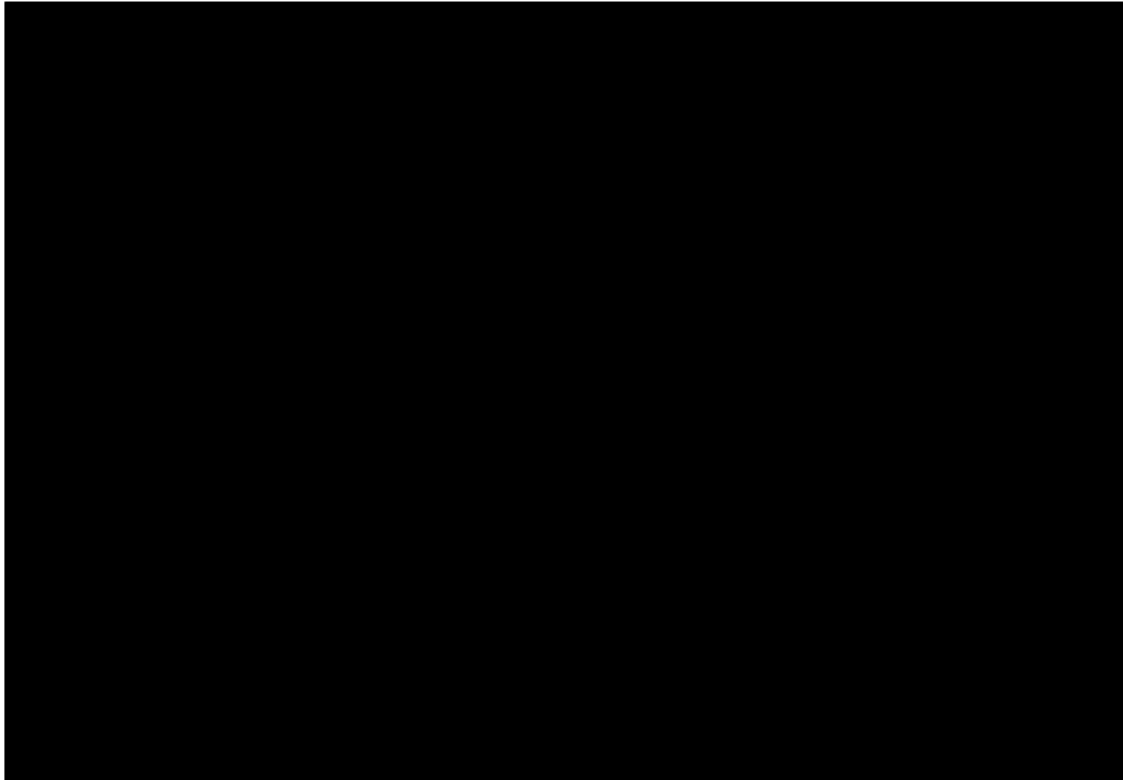
Toucher l'écran permet d'avoir plus ou moins de vibrations en fonction de la distance entre le point ciblée et l'interrupteur. Ainsi, le but est de chercher à déplacer son doigt afin d'intensifier les vibrations. (Smartphone)

Les vibrations causées par l'échec du mini-jeu de blocage des météores interfèrent avec ce mini-jeu.

<b>Avarie associée</b>	Coupure de lumière
<b>Solo</b>	Oui
<b>Multijoueur</b>	Non
<b>Nombre de joueurs</b>	1
<b>Contrôle</b>	Tactile
<b>Condition de réussite</b>	Trouver et toucher l'interrupteur



## Template visuel



## Variante Solo et Multijoueurs

### Variante Solo

Toucher l'écran du téléphone le fait vibrer. La durée de la vibration est inversement proportionnel à la distance qui sépare l'interrupteur caché du doigt.

Lorsque la vibration est constante, c'est que l'interrupteur est sous le doigt de l'utilisateur. Il n'a plus qu'à toucher l'écran tactile pour allumer la lumière et finir le jeu.

### Variante Multijoueurs

Pas de variante multijoueurs

## Balistique

**But:** Tirer avec suffisamment de précision pour infliger des dégâts au vaisseau ennemi et se rapprocher de la victoire.

**Interaction:**

Envoyer une balle de velcro colorée avec le plus de précision possible vers le centre d'une cible en ouate accrochée au mur filmée par une caméra.

<b>Avarie associée</b>	Aucune (Mini jeu de tir)
<b>Solo</b>	Oui
<b>Multijoueur</b>	Non
<b>Nombre de joueurs</b>	1
<b>Contrôle</b>	Envoi d'une balle
<b>Condition de réussite</b>	Avoir plus qu'un seuil minimum

Ce mini jeu diffère fortement des autres car il s'effectue non plus sur smartphone mais sur une vraie cible :

**Qui:**

Joueur sans nécessiter de smartphone.

**Quoi:**

Afin d'attaquer le vaisseau ennemi, il faut se déplacer dans la salle le permettant, et ensuite tirer sur la cible avec la balle de velcro. Il faut alors ensuite aller la récupérer une fois le tir enregistré, dans le temps restant imparti indiqué sur le téléphone.

Les tirs proviennent de la tourelle présente vers l'avant du vaisseau, qui peut être tournée en plaçant puis pivotant un atome à sa base.

**Où:**

L'occasion de tirer apparaît dans une salle spécifique afin que les joueurs soient attentifs à son changement d'état.

La tourelle à faire pivoter se situe à droite des salles les plus à droite du vaisseau.

**Quand:**

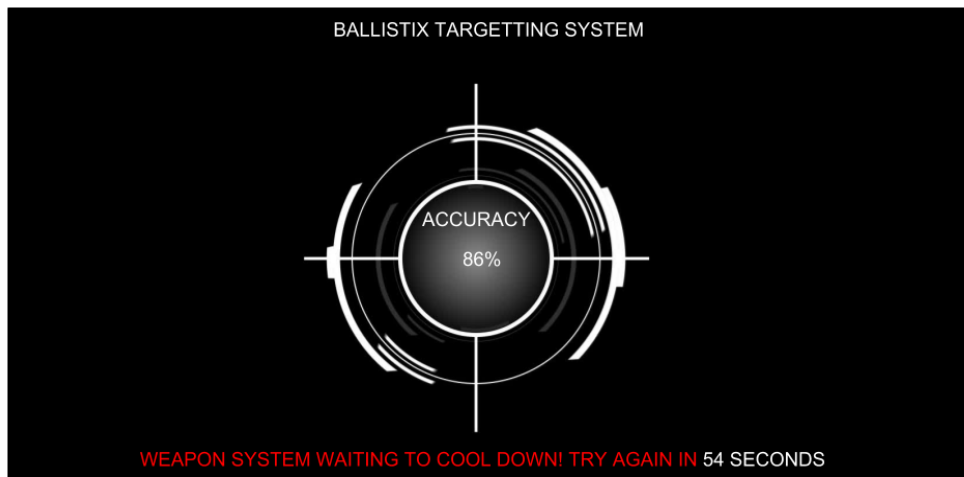
Pendant la phase 3, de combat avec le vaisseau, la salle des armes est disponible pour utilisation. Une fois que le tour à tiré, il faut quelques secondes pour que le tir soit validé, puis la salle des armes devient indisponible pendant un certain temps, permettant au joueur d'aller chercher la balle, puis de s'occuper des autres avaries.

**Pourquoi:**

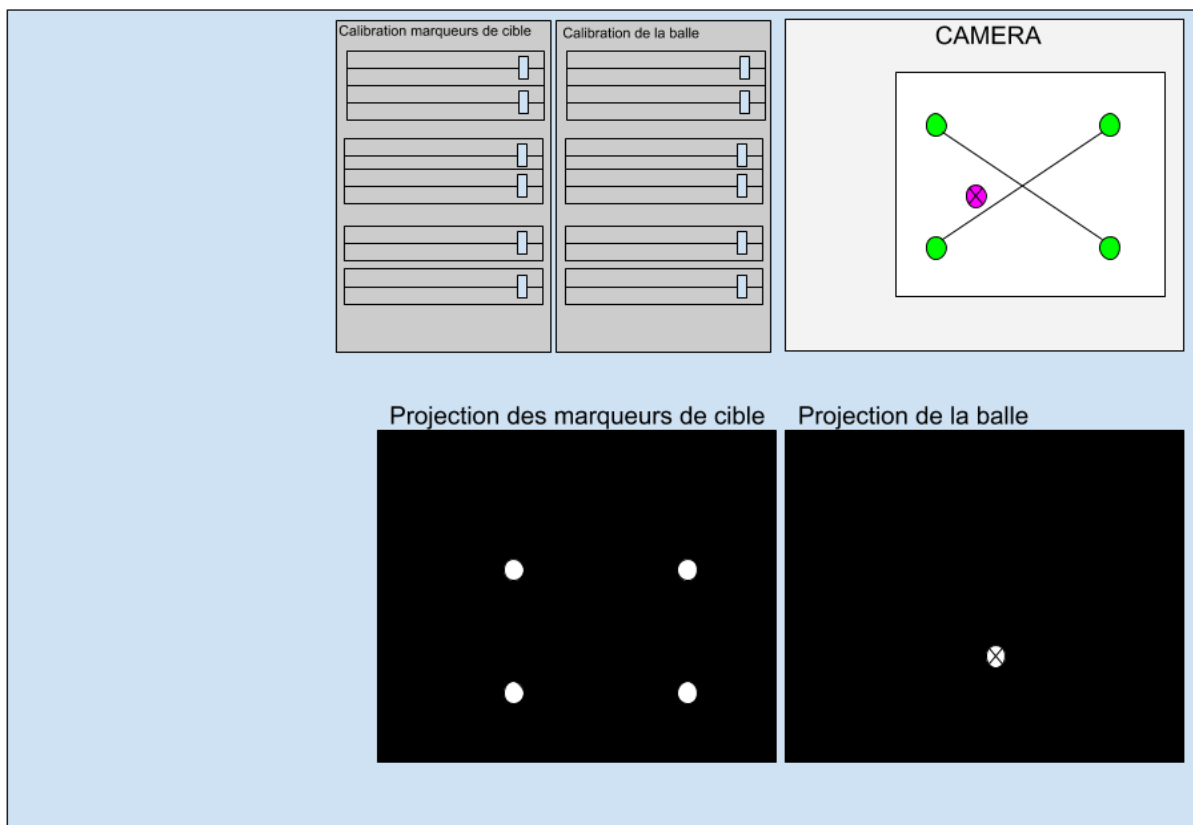
Ce mini-jeu est le seul moyen d'infliger des dégâts au vaisseau ennemi, et donc de gagner. Pour cela, il est nécessaire d'avoir la plus haute précision possible, afin de maximiser les dégâts infligés.

## Template visuel

### Smartphone



PC



## Variante Solo et Multijoueurs

### Variante Solo

Le joueur saisit la balle et la jette sur la cible. Selon la précision de son tir un score s'affiche sur le téléphone, et la tourelle du vaisseau tire en plus ou moins grande quantité, avec plus ou moins de précision.

La précision du lancer, relative à la position de la balle par rapport à la taille de la cible, indique la précision des tirs de la tourelle, pouvant aller de 0° à 180° maximum de déviation aléatoire entre la direction de la tourelle et chaque tir.

Le chiffre des dizaines indique quand à lui le nombre de tirs, 100% de précision donnant lieu à 10 tirs avec une précision absolue, tandis que 30% ne donne lieu qu'à 3 tirs très imprécis, chacun divergeant vers une direction différente.

### Variante Multijoueurs

Pas de variante multijoueurs

## Projet F.A.S.T.

Le projet, le code, les pré-requis matériels, les pré-requis technologique et la procédure de mise en place sont disponibles sur les Github suivant :

### F.A.S.T

<https://github.com/NicolasBighetti/F.A.S.T>

### F.A.Server.T

<https://github.com/NicolasBighetti/F.A.Server.T>

### F.App.S.T

<https://github.com/NicolasBighetti/F.App.S.T>

### F.A.S.Table

<https://github.com/NicolasBighetti/F.A.S.Table>

### Fire.A.S.T

<https://github.com/Epsiom/Ballistix>

# Repartition du travail

Alison BERLIOZ:

Design graphique

Nicolas BIGHETTI:

Mini jeux smartphone. Interaction avec l'aide des capteurs du smartphone dans les mini jeux.

Flavian JACQUOT:

Color connect

Interaction tangible et atomes sur la table tactile.

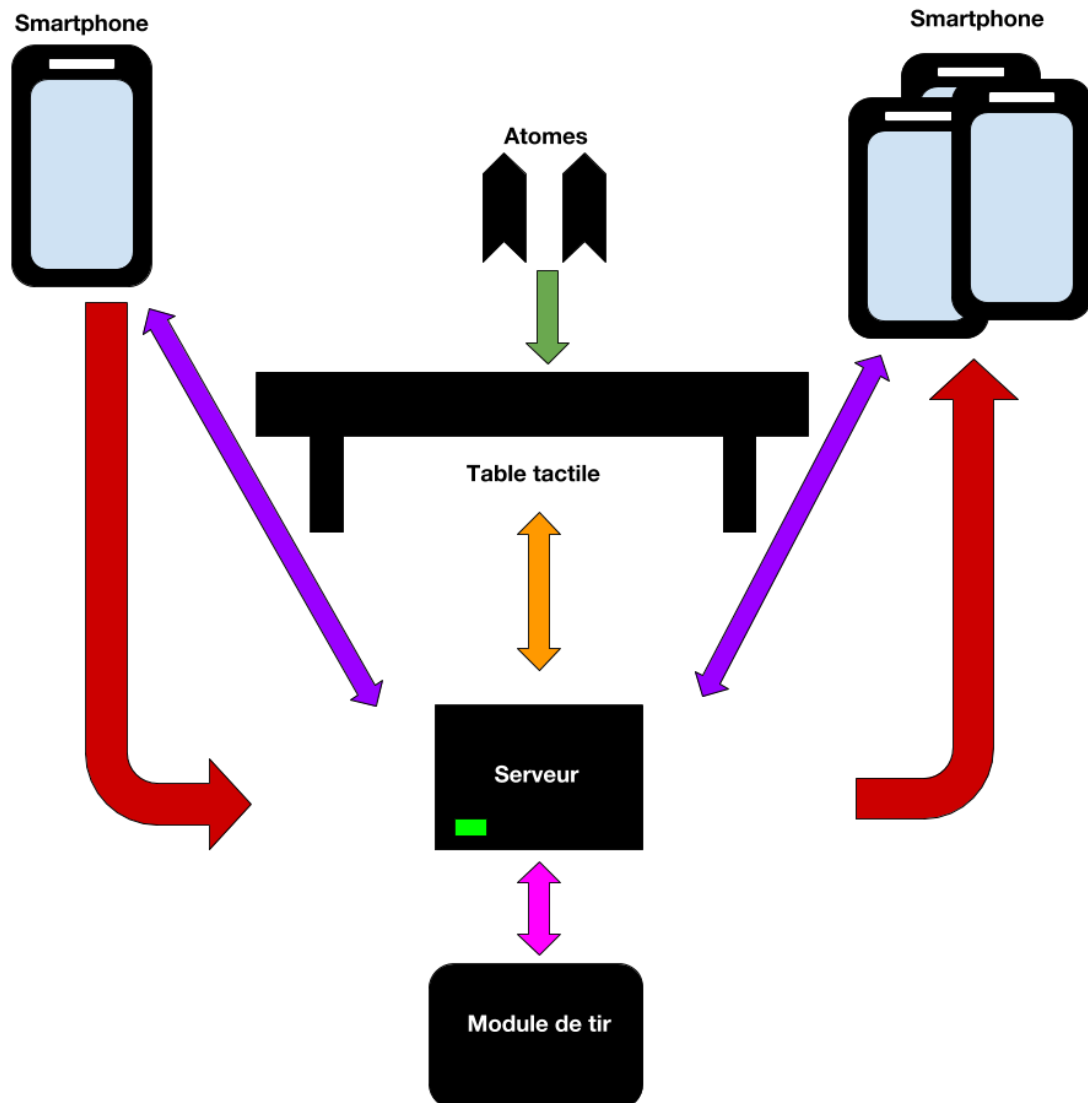
Déplacement des personnages et lancement des mini-jeux.

Steven LEQUIENT:

Système de cible avec détection openCV

# Annexe

## Protocol réseau



Ce document est une des itérations de la description du protocol applicatif de notre application, F.A.S.T.  
Ce protocol sera amené à évoluer au cours du développement, ce document reflétera les changements effectués.

## Violet : Communication double sens

### Serveur-Smartphone

La communication entre le serveur et le smartphone s'effectue à l'aide de socket.io.

#### **Préfixe :**

FAST\_MINI\_GAME relatif aux mini-jeu

FAST\_PHONE communication vers le téléphone

### Serveur vers Smartphone

#### Connection à la partie

**Clé unique :** FAST\_PHONE\_CONNECT

**Destinataire :** Le téléphone qui se connecte

**Objet associé :** Les données de connection

**Format des données :**

```
{  
  PLAYER_ID: 1,  
  COLOR: "#0000FF",  
  ATOM_PLAYER_ID: 14,  
  ATOM_PHONE_ID: 20  
}
```

**Réponse :** -

#### Début d'un mini-jeu

**Clé unique :** FAST\_MINI\_GAME\_SOLO\_START

**Destinataire :** Participant au mini jeu

**Objet associé :** -

**Format des données :** -

**Réponse :** -

#### Début d'un mini jeu multi

**Clé unique :** FAST\_MINI\_GAME\_MULTI\_START

**Destinataire :** Participants d'un mini jeu multijoueur

**Objet associé :** -

**Format des données :** -

**Réponse :** -

Déclenche le début de la session privée entre les smartphones.

Fin d'un mini jeu

**Clé unique :** FAST\_MINI\_GAME\_MULTI\_STOP

**Destinataire :** Participants d'un mini jeu multi joueur

**Objet associé :** Si oui ou non la réparation a réussi

**Format des données :** boolean

**Réponse :** -

## Protocole par mini jeu

Le protocole par mini jeu est détaillé dans [le document relatif aux mini jeux](#). Toutefois le modèle général mis en place est le suivant:

Données d'initialisation globale

```
{
    time : int,
    id : identifiant_mini_jeu,
    data : [voir les données d'initialisation au cas par cas]
}
```

Données d'échange

```
{
    timestamp: ms,
    id: string,
    [voir les données d'échange au cas par cas]
}
```

FAST\_GAME\_FIRE

**Préfixe :** FAST\_GAME\_FIRE\_

Initialisation

**Clé unique :** FAST\_GAME\_FIRE\_INITIALISATION

**Destinataire :** Les participants du mini jeu

**Objet associé :** Les données d'initialisation

**Format des données :** [cf Forme des données](#)

**Réponse :** FAST\_GAME\_FIRE\_ACK



Broadcast update

**Clé unique** : FAST\_GAME\_FIRE

**Destinataire** : Smartphone

**Objet associé** :

## Format des données

Type de feu

Code	Type
FAST_GAME_FIRE_RED	Feu rouge
FAST_GAME_FIRE_BLUE	Feu bleu
FAST_GAME_FIRE_GREEN	Feu vert
FAST_GAME_FIRE_PURPLE	Feu violet

## Données d'initialisation

? : optionnel

On fournit la quantité initiale de flamme pour chaque couleur

```
{
    FAST_GAME_FIRE_RED: int,
    FAST_GAME_FIRE_BLUE?: int,
    FAST_GAME_FIRE_GREEN?: int,
    FAST_GAME_FIRE_PURPLE?: int
}
```

## Données échangés durant la partie

? : optionnel

On fournit la quantité courante de flamme pour chaque couleur.

```
{
    FAST_GAME_FIRE_RED?: int,
    FAST_GAME_FIRE_BLUE?: int,
    FAST_GAME_FIRE_GREEN?: int,
    FAST_GAME_FIRE_PURPLE?: int
}
```

## FAST\_GAME\_METEOR

**Clé unique** : FAST\_GAME\_METEOR

**Destinataire** : Serveur

**Objet associé :**

## Format des données

Données d'initialisations

```
{  
    FAST_GAME_METEOR_DATA: number  
}
```

Contact avec le vaisseau

```
{  
    FAST_GAME_METEOR_DAMAGE: number  
}
```

## FAST\_GAME\_SWITCH

**Clé unique :** FAST\_GAME\_SWITCH

**Destinataire :** Serveur

**Objet associé :**

## Format des données

Rien de plus que les données générales de début et de fin de partie.

## FAST\_GAME\_BALLISTIC

**Clé unique :** FAST\_GAME\_BALLISTIC

**Destinataire :** Serveur / Téléphone

**Objet associé :**

## Format des données

Données de dégâts

```
{  
    FAST_GAME_BALLISTIC_DAMAGE: number  
}
```

## Smartphone vers Serveur

Inscription à un mini jeu

**Clé unique :** FAST\_MINI\_GAME\_REGISTER

**Destinataire :** Serveur

**Objet associé :** L'identifiant du mini jeu choisi

**Format des données :** identifiant : String

**Réponse** : isRegistered : boolean, privateRoom : string

Fin du mini-jeu

**Clé unique** : FAST\_MINI\_GAME\_STOP

**Destinataire** : Serveur

**Objet associé** : -

**Format des données** : boolean

**Réponse** : -

## Rouge : Distribution des events Smartphone

Les events sont redistribués à l'aide d'une connexion socket.io établie sur un port différent de la connexion principale afin que les events puissent atteindre le serveur malgré la session privée.

**Prefixe** : FAST\_EVENT\_

### Smartphone vers serveur

**Clé unique** : FAST\_EVENT\_BROADCAST

**Destinataire** : Serveur

**Objet associé** : JSON contenant la liste des events à reproduire

**Format des données** : Array<String, option>

**Réponse** : Pas de réponse

### Serveur vers smartphone

**Clé unique** : FAST\_EVENT\_BROADCAST

**Destinataire** : Tout les smartphones sauf l'émetteur

**Objet associé** : JSON contenant la liste des events à reproduire

**Format des données** : Array<keyEvent : String, option : any>

**Réponse** : Pas de réponse

## Liste exhaustive des events

Event String	option	Comportement associé
FAST_EVENT_VIBRATION_WEAK		Faible vibration durant en ms
FAST_EVENT_VIBRATION_MEDIUM		Moyenne vibration durant en ms
FAST_EVENT_VIBRATION_STRONG		Forte vibration durant en ms

## Orange : Communication Table-Serveur

La communication entre la table et le serveur se fait à l'aide d'une connection socket.io.

On ne synchronise le serveur et la table que pour les données utiles pour la mise en relation avec les smartphones client.

**Préfixe** : FAST\_SHIP\_

### Liste exhaustive des codes d'avarie

**Préfixe** : DAMAGE\_

Code	Avarie associée
FAST_SHIP_DAMAGE_FIRE	Incendie
FAST_SHIP_DAMAGE_METEOR	Pluie de météore
FAST_SHIP_DAMAGE_POWER_OFF	Coupure de courant
FAST_SHIP_DAMAGE_SPACE_FIGHT	Attaque de vaisseaux ennemi

La ou les salles concernées sont envoyées en paramètre.

### Déplacement personnage

**Clé unique** : FAST\_SHIP\_PLAYER\_MOVE

**Destinataire** : Serveur

**Objet associé** : L'identifiant du joueur qui se déplace, la pièce vers laquelle il s'est déplacé.

**Format des données** : {id:string, position: int}

**Réponse** : -

### Activation bonus

**Clé unique** : FAST\_SHIP\_BONUS\_ACTIVATED

**Destinataire** : Serveur

**Objet associé** : Le bonus qui vient d'être activé

**Format des données** : String

**Réponse** : Boolean : activé ou non

## Apparition d'une avarie

**Clé unique** : FAST\_SHIP\_DAMAGED\_UP

**Destinataire** : Table

**Objet associé** : [L'identifiant de l'incident apparu](#), les salles il est apparu

**Format des données** : {avarie: String, room: [int]}

**Réponse** : -

## Résolution d'une avarie

**Clé unique** : FAST\_SHIP\_DAMAGED\_DOWN

**Destinataire** : Serveur

**Objet associé** : [L'identifiant de l'incident résolu](#), la salle ou il est résolu

**Format des données** : [avarie: String, room: int]

**Réponse** : -