

Projet BrakBox

Groupe 10

Matthieu BARRAUD

Noe CHOC

Clement CORRENOZ

Nicolas GERMANI

Maxence LUCAS

Elia LAWRIV

Valentin WILBAILLE

SOMMAIRE

1. Cadrage
 - a. Définition des objectifs
 - b. Définition du contexte
 - c. Analyse de l'existant
 - d. Contraintes et risques
2. Expression des besoins
 - a. Identification des besoins fonctionnel
 - b. Priorisation des besoins fonctionnels
 - c. Mise en avant de critères qualité logicielle
 - d. Mise en avant des critères ergonomiques
3. Solutions
 - a. Présentation et analyse des solutions techniques possibles
 - b. Gestion du travail (organisation du travail et outils)
 - c. Description d'un modèle de développement
4. Annexes (maquettes non fonctionnelles)

1. Cadrage

Définition des objectifs

Nous pensons faire une application web qui sera une agrégation de la mallette. Elle devra être minimaliste pour être ergonomique sur mobile ou tablette.

On pourra consulter dessus les documents utiles aux résolutions des énigmes.

Cette application sera sous 2 versions : une pour éditer (qui sera accessible uniquement pour l'administrateur) et une pour jouer les parties.

L'application devra être :

- Cross-plateforme
- Accessible via un QR code ou par un lien « classique » depuis n'importe quel navigateur
- Multilingue (français, anglais pour commencer)
- Ergonomique

Pour l'application jouable par le public :

Une fois sur la page d'accueil, il y aura :

- Un choix pour les scénarios
- Un système de comptes pour s'enregistrer ou se connecter pour que seulement ceux en possession de la mallette puissent accéder au site (ça sera fait seulement s'il nous reste du temps)

Une fois le scénario choisi, nous arriverons sur la page principale du jeu où il y aura plusieurs « modules » non modifiables :

- Le nom du scénario
- Le scénario en question
- Les annexes
- Un système d'indices (optionnel)

Tous ces éléments se trouveront dans une base de données.

Pour l'éditeur uniquement destiné à l'administrateur :

L'objectif est de créer un éditeur d'aventure pour que l'administrateur :

- Pourra supprimer, éditer et créer de nouveaux scénarii depuis la page d'accueil du site
- Pourra supprimer, éditer et créer de nouveaux résumés, indices, informations...

La base de données de l'application et de l'éditeur est la même.

Définition du contexte

L'escape game est un genre de jeu de plus en plus en tendance depuis les années 2010. Avant il se jouait surtout dans des salles mais avec la forte croissance de l'informatique dans nos vies, l'escape game s'est modernisé. En effet aujourd'hui, les parties des épreuves des escape game peuvent se jouer en virtuelles (via casque VR/jeux video) et c'est là qu'apparaît le projet Brak Box.

Analyse de l'existant :

Nous allons étudier les cas d'application étant complémentaire avec des jeux de société/ escapes games.

UNLOCK :

Unlock, créé par Space Cowboy a souvent été cité les professeurs et l'intervenant comme étant similaire à ce que l'on devrait faire pour la Bräk Box, nous allons donc l'étudier.

Unlock fonctionne grâce à une application et un deck de carte.

L'application contient un menu qui permet de changer d'aventure.

Un écran d'aventure contient :

- Le timer
- Des onglets (indices, machines, code)
- Le bouton pénalité

Les cartes contiennent le scénario et, le plus souvent, des codes, pour l'application ou retourner d'autres cartes.

Les cartes sont faces cachées et doivent être retournées selon les codes d'autres cartes.

Lorsque qu'un code indice de carte est rentré dans l'onglet indice, du texte donnant un indice apparaît.

Lorsque qu'un code machine de carte est rentré dans l'onglet machine, une énigme (qu'il faut résoudre grâce aux indices) apparaît. Une fois l'énigme terminée, l'on obtient de quoi progresser (le code d'une carte à retourner).

Le code final, lorsque rentré dans l'onglet « code », met fin à l'aventure par une victoire, affichant le score.

L'aventure est un échec si le timer arrive à zéro. Si une carte le demande, l'on doit appuyer sur le bouton pénalité, ce qui fait perdre du temps.

Principales similitudes Bräk Box / Unlock :

- Énigmes à résoudre
- Applications + support physique
- Codes à récupérer
- Parties chronométrées

Unlock, bien que similaire à la Bräk Box, a dans son fonctionnement de majeures différences :

- Unlock n'est pas une appli web, contrairement à la l'appli de la Bräk Box
- Le timer est dans l'application, a l'inverse de la Bräk Box qui a un timer dans la mallette
- Obtient les codes depuis les cartes pour débloquent des énigmes, alors que les énigmes de la Bräk Box sont toutes disponibles depuis le départ
- Dans la Bräk Box Il faut rentrer les codes obtenus grâce l'application dans la mallette (l'inverse de Unlock)

A part cela et d'autres différences de fonctionnement mineures, les fondations sont relativement les mêmes.

Il est à noter qu'il n'existe pas d'autres jeu comme la Bräk Box ou Unlock assez populaires pour être trouvé, Unlock est donc le seul cas que l'on peut étudier pour développer la Bräk Box.

Néanmoins la Bräk Box est vaguement similaire à un jeu vidéo : « keep talking and nobody will explode », où 2 joueurs, un possédant le manuel (format PDF, ou papier), et un sur le jeu, en possession d'une bombe à désamorcer, doivent s'entraider pour ne pas exploser.

La comparaison vient du fait que les informations contenues dans le manuel servent à désamorcer la bombe. L'apparence de la Bräk Box ressemble fortement à la bombe de ce jeu. Cependant l'aspect coopération ne se fait pas de la même manière pour la Bräk Box (tous les joueurs ont accès au « manuel » et à la bombe).

Contraintes et risques

CONTRAINTES

Manque d'argent :

Cette contrainte naturelle nous impose d'utiliser des moyens totalement gratuits.

- Travailler en équipe sans sous-traitance :

L'intégralité des tâches devront être réalisées par des membres de l'équipe qui est limitée en capacité(7 personnes). Il nous est impossible de déléguer des tâches.

- Juridique :

Les documents utilisés devront être libres de droits ou de notre création.

- langages de programmation imposés :

Le commanditaire nous impose d'utiliser un certain langage (le typescript) et une certaine base de donnée (firebase).

- Le support doit être une application web :

Il faut que ce support soit facilement accessible. Le plus simple est donc de développer une application web dans laquelle les créateurs d'énigme pourrons les créer et les joueurs les voir.

- Nécessité d'être accessible à un large public :

L'appli doit être le plus ergonomique possible pour que tous les joueurs puissent comprendre facilement comment accéder aux énigmes et aux compliments.

RISQUES

- R1 : Discorde ou mésentente au sein de l'équipe :

Il est possible que certains des membres de l'équipe soient en désaccord. Cela pourrait nous ralentir voir même nous bloquer. Cependant le risque que ça arrive est moyen. risque moyen. Pour éviter cela il faut communiquer et savoir être à l'écoute.

- R2 : Manque de compétence :

Sachant qu'on nous impose certains langages, il faut apprendre à les utiliser. Il est dangereux de se lancer dans le projet sachant qu'aucun de nous ne les maîtrise. risque moyen Il va falloir s'améliorer dans les compétences concernées et se perfectionner pour atteindre les objectifs fixés.

- R3 : Absence d'un membre :

Il est possible qu'un membre soit absent temporairement ou définitivement

Risque faible → Être capable de comprendre et faire ce que les autres font et doivent faire.

- R4 : Perte de contact avec le commanditaire :

La perte de contact est une éventualité à ne pas négliger car il est très probable que l'on ai pas un emploi du temps incompatible avec le commanditaire .

Risque faible → Il faut se coordonner avec le commanditaire.

- R5 : Changement des conditions de travail :

En ces temps de virus, les conditions de travail sont plus complexes. Ces conditions peuvent donc être un frein à l'avancée du projet.

Risque Constant

- R6 : Non-respect des délais :

Le mieux serait de rendre un projet terminé, cependant on sait qu'il est très fréquent qu'un projet ne soit pas prêt à l'échéance prévue.

Risque élevé → Se focaliser sur les choses principales

MATRICE DE CRITICITÉ

		Niveau de Gravité			
		insignifiant	marginal	critique	catastrophique
Probabilité	très probable				
	Probable		R3, R2		
	Possible		R1		
	Peu probable			R4	
	Tres improbable		R5	R6	

2. Expression des besoins

Identification des besoins fonctionnels et non fonctionnels

Lors de la phase de recherche, nous avons identifié différents besoins fonctionnels. Nous devons avant tout pouvoir naviguer entre les énigmes. On devra aussi pouvoir afficher l'introduction et les annexes. Un des besoins pour l'utilisateur sera de pouvoir consulter des indices au bout d'un certain temps (s'il reste longtemps bloqué à une étape). Pour la partie éditeur, le plus gros besoin sera de pouvoir créer, éditer et supprimer les énigmes.

Priorisation des besoins fonctionnels

Tous les besoins cités précédemment sont tout aussi important excepté un : la consultation des indices. En effet c'est une idée qui a été soumise lors de la première réunion et le commanditaire nous a dit de nous focaliser en premier temps sur les premières cités, ce-dernier sera un plus s'il nous reste du temps.

Mise en avant de critères qualité logicielle

Nous avons identifié plusieurs critères qualité logicielle. Un des principaux est la facilité de modification. En effet, l'administrateur devra souvent faire des modifications dans ses scenarios. C'est là que la stabilité entre en jeu car il faut que l'application doive restée opérationnelle lorsque l'administrateur modifie ses scenarios. Il faudra faire attention à que seul l'administrateur puisse modifier la base de données, il s'agit donc de sécurité.

Mise en avant de critères ergonomiques

Nous avons pour but de faire en sorte que pour tout type de personne qui utilisera l'application, celle-ci se doit être la plus simple d'utilisation possible. Il y aura donc un guidage où l'incitation, le feedback immédiat et la lisibilité seront de mise. Nous voudrions qu'il n'y ait pas trop de charge de travail, les textes d'indications devront être brefs et les actions que nous pourrions faire minimales.

Le contrôle sera explicite grâce aux actions explicites.

L'application devra être conçue pour faire en sorte que si quelqu'un n'a jamais joué au jeu, il pourra rapidement prendre l'application en main. C'est donc le critère de l'adaptabilité qui apparaît ici.

Dans toutes les pages de l'application, si des boutons ou textes réapparaissent, ils devront être positionnés aux mêmes endroits pour garder la cohérence des pages.

Nous voudrions que tous ces critères soient respectés peu importe sur laquelle on joue au jeu.

3. Solutions

Présentation et analyse des solutions techniques possibles

Solutions techniques envisagées

-Une application web qui permet d'afficher les indications en plus du jeu.

Avantages : une meilleure maniabilité de l'histoire qui peut rester sous format numérique. Le fait qu'il n'y est plus de papier.

Complexité : Adaptation du format papier sur téléphone, liaison des téléphones et des joueurs

-Un éditeur de partie du jeu pour pouvoir créer une aventure depuis l'application.

Avantages : Une facilité d'ajouter ou de modifier des parties.

Complexité : Uniquement le commanditaire peut avoir accès à cette application.

-Base de données accessible en temps réel à des fichier textes, images.

Avantages : Tous les documents accessibles depuis le même endroit.

Complexité : Mise en œuvre d'une base de données.

-Un gestionnaire de tâche à effectuer, informant sur l'activité de chacun des membres du groupe (Trello)

-Un logiciel de création de maquette (Axure 10)

Besoin technique :

Angular :

Angular est un framework qui va nous servir à développer l'application web ainsi que l'éditeur de partie. Angular utilise le langage TypeScript.

Firebase :

Nous allons utiliser Firebase qui nous propose une base de données simple.

-Le firestore database qui permet de stocker efficacement des documents dans une base de données.

Gestion du travail (organisation du travail et outils)

Premièrement, pour gérer l'organisation des tâches, nous utilisons Trello, un outil permettant de se répartir les tâches et d'avoir une vue d'ensemble sur l'avancée de celles-ci.

Ensuite, nous avons créé un serveur discord pour communiquer plus facilement et s'échanger des fichiers lorsque l'on travaille depuis chez nous.

Pour la répartition des tâches, durant la première itération nous avons réparti la création du dossier comme ceci :

- Nicolas Germani s'occupe de la partie des objectifs.
- Elia Lawriw s'occupe de l'analyse de l'existant.
- Clément Correnoz et Matthieu Barraud s'occupent des contraintes et risques.
- Noé Choc s'occupe de la partie sur l'organisation de l'équipe et du projet.
- Valentin Wibaillie et Maxence Lucas s'occupent de la partie environnement technique.
- Enfin pour la partie Besoins fonctionnels et non-fonctionnels, c'est Nicolas et Valentin qui s'en chargent.

Pour la deuxième itération, Elia, Noé et Maxence feront les diagrammes de séquences tandis que Nicolas, Valentin, Clément et Matthieu feront les graphiques UML.

Ensuite au niveau du code en lui-même, Matthieu et Valentin s'occuperont de créer et gérer la base de données puis une fois celle-ci finie, ils aideront sur la partie la moins avancée entre l'éditeur d'énigme et l'application Web. Nicolas, Clément et Maxence développeront l'appli Web et enfin Noé et Elia programmeront l'éditeur d'énigme.

Description d'un modèle de développement :

Nous appliquerons un modèle de développement itératif pour ce projet, c'est-à-dire que nous aurons un premier prototype que nous améliorerons au fur et à mesure.

Au niveau de l'organisation du projet, nous avons réparti les premières choses à faire pour le début et nous nous sommes mis d'accord sur des maquette. Pour la suite du projet, nous comptons commencer à coder un prototype dès la deuxième itération du projet.

En effet nous voulons d'abord mettre en place la base du projet dans cette première itération puis nous commencerons à coder cette base dans la deuxième itération après l'avoir modéliser dans des graphiques UML et des diagrammes de séquences.

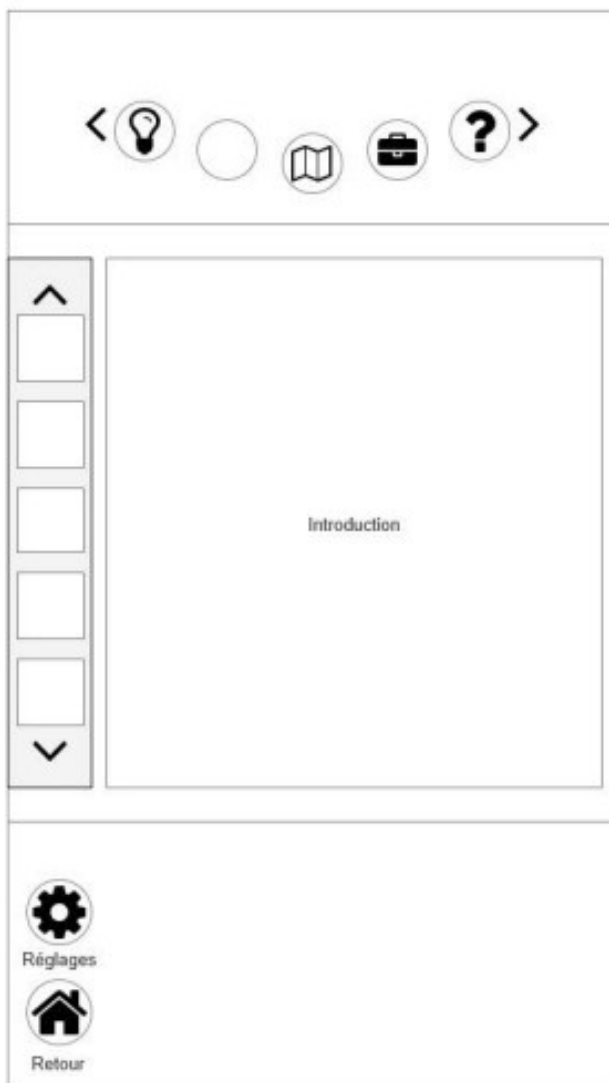
Enfin durant la troisième itération, nous verrons en fonction de l'avancée du projet pour rajouter des fonctionnalités tout en faisant en sorte d'avoir un prototype valable le jour de l'évaluation.

Tâches	24/09/21	01/10/21	08/10/21	15/10/21	20/10/21	22/10/21	29/10/21	10/11/21	19/11/21	26/11/21	03/12/21	10/12/21	17/12/21	07/01/22	14/01/22	17/01/22
Prise en main du sujet																
Création du dossier																
Modélisation (Graphe UML et séquence)																
Programmation de l'appli Web																
Programmation de l'éditeur d'énigme																
Gestion de la base de données																

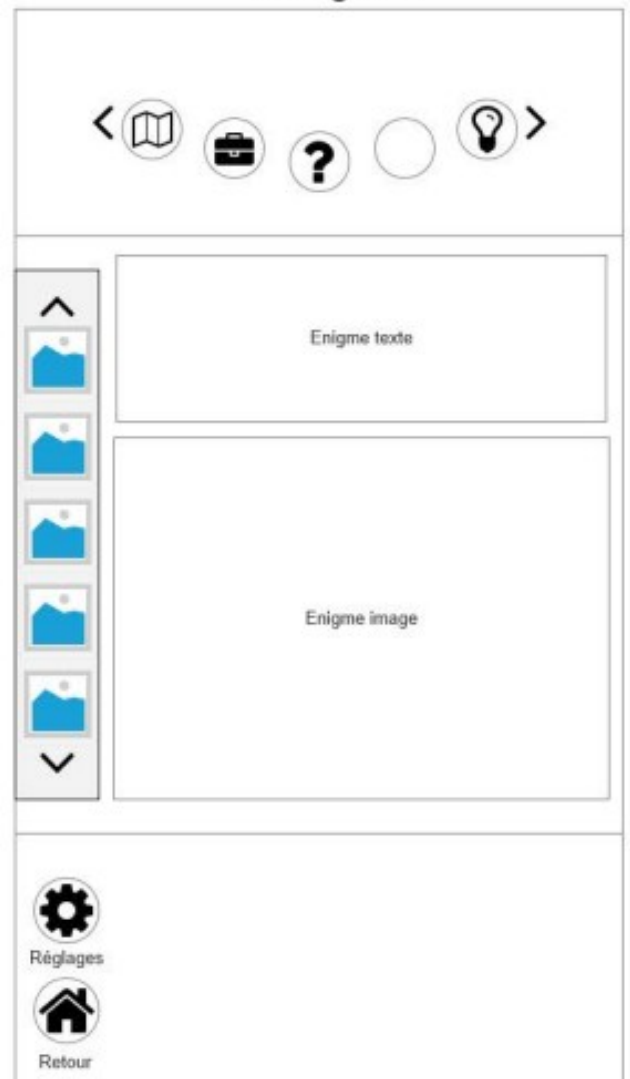
4. Annexes

Maquette de l'application

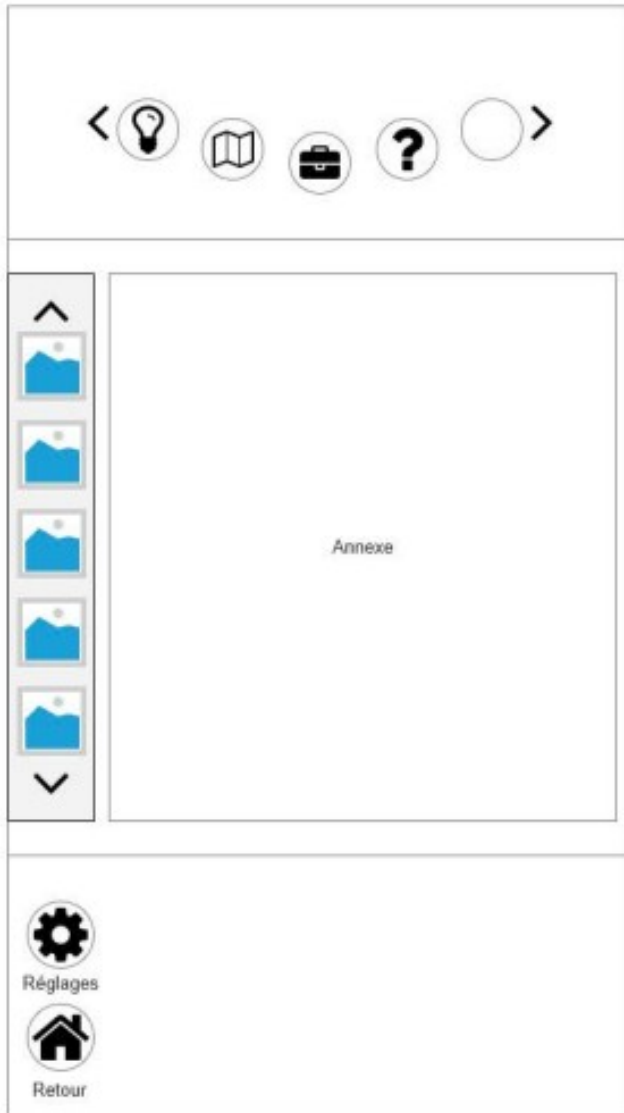
intro



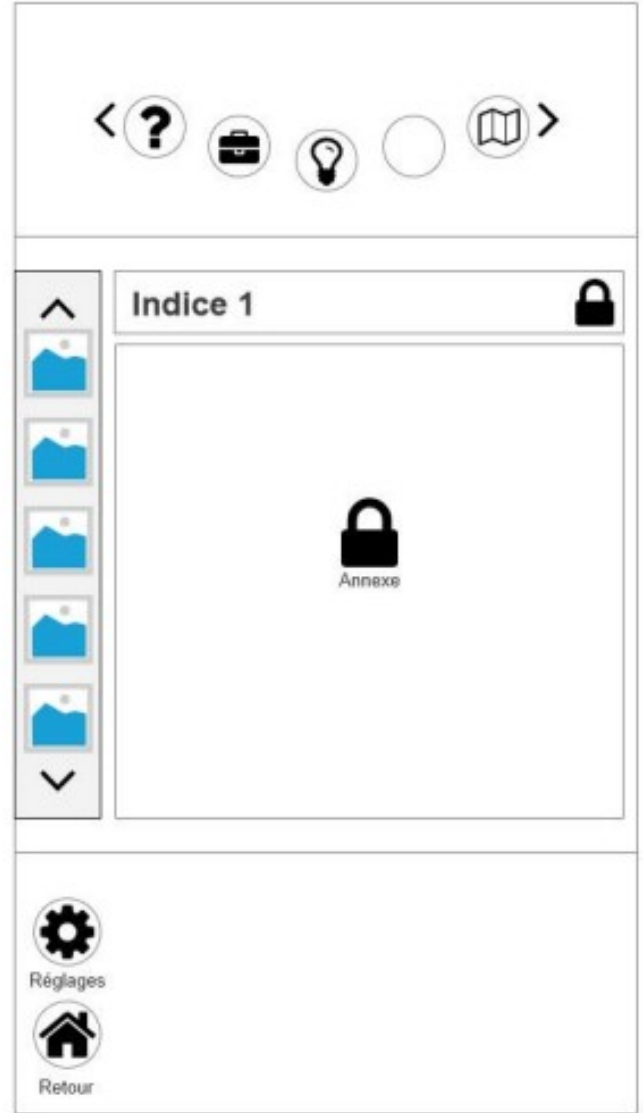
Enigme



Annexes



Indice



Maquette de l'éditeur

Annuler

Paramètres

Enigmes

Annexes

Finalisation

Sauvegarde

Nom:

Aventure

Nombre d'énigmes:

6

Temps:

60

image:

Continuer

Annuler

Paramètres

Enigmes

Annexes

Finalisation

Sauvegarde

énigme 1

énigme 2

énigme 3

énigme 4

énigme 5

énigme 6

Contenu

Indice

liste indice ▼

Contenu :

texte

Continuer

Annuler


Paramètres

Enigmes

Annexes

Finalisation

Sauvegarde

Ajouter des annexes : 

lien



Continuer

Annuler

Paramètres

Enigmes

Annexes

Finalisation

Sauvegarde

énigme 1

énigme 2

énigme 3

énigme 4

énigme 5

énigme 6

Contenu

Indice

Enoncé :

Texte

Document visuel



Continuer

Annuler

Paramètres

Enigmes

Annexes

Finalisation

Sauvegarde

Paramètres

Nom:

Nombre d'enigmes:

Temps:

Image: 

Modifier

Enigmes

Enigme 1

Enigme 2

Contenu

Index

énigme 2

Fichier texte

Texte

Document visuel



Modifier

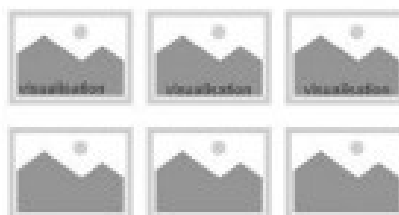
Enigme 3

Enigme 4

Enigme 5

Enigme 6

Annexes



Modifier

Continuer