Documentation Logicielle Splashmem

Contents

1 Jeu

1.1 Technologies utilisées

Le jeu est codé intégralement en C, en utilisant la librairie de rendu graphique SDL2 et la librairie de transfert de données libcurl.

1.2 Organisation du jeu

A la racine du projet on retrouve 3 dossiers et un fichier :

- assets/
- include/
- src/
- makefile

1.2.1 assets/

Le dossier assets/ contient tous les fichiers qui ne sont pas du code. On y trouve notamment les sprites pour le rendu graphique du jeu, la musique, la police de caractère, etc.

1.2.2 include/

Le dossier include/ contient tous les headers associés aux fichiers c du projet. On trouve un header pour chaque fichier c, et un header nommé params.h qui contient tous les paramètres du jeux.

1.2.3 src/

Le dossier src contient tous les fichiers sources du jeu.

1.3 Fonctionnement du jeu

Le code du jeu se découpe en trois phases : une phase d'initialisation, puis une boucle qui déroule le jeu et enfin une phase qui vient après la fin du jeu. C est un langage procédural mais le jeu a été pensé en termes d'interactions entre des objets, on retrouve donc des structures et fichiers C permettant de mimer le concept de classes :

- inits.c contient les méthodes d'initialisation
- action.c contient le code des actions effectuées par les joueurs
- handler.c contient le code permettant de gérer les événements clavier (appui sur q pour fermer le jeu par exemple)
- entities.c contient le code des entités du jeu (joueurs, bombes, map)
- finish.c contient les fonctions qui vont s'exécuter lors de la fin du jeu, lorsque tous les joueurs ont atteint 0 crédits.

1.3.1 phase d'initialisation des objets

La phase d'initialisation est assurée par plusieurs fonctions d'initialisation, qui vont venir mettre en place la fenêtre, le renderer, la map, les joueurs, etc. Ces fonctions d'initialisation se situent principalement dans les fichiers inits.c et entities.c.

1.3.2 phase de jeu

La phase de jeu se décompose en 3 parties :

Réalisation des actions Chaque joueur va effectuer une action de sa librairie, qui lui permet donc de se déplacer à chaque tour.

Rendu des images Tous les joueurs sont associés à des images qui se trouvent dans assets/images. Cela permet d'avoir des animations pour les coups spéciaux, et une impression de mouvement pour les déplacements.

Gestion des effets sonores Splashmem tourne avec une musique de fond (initialisée dans init_wav). Les sons servent aussi a supporter les animations des coups spéciaux et de certains mouvements comme le dash ou le teleport.

1.3.3 phase de fin

Le jeu se termine lorsque :

- La touche q est enfoncée
- Les crédits de tous les joueurs tombent à 0.

Lorsque le jeu est fini, un fichier results.csv est généré, contenant le pseudo du joueur ayant initié la partie, ainsi que les noms et librairies de chaque joueurs, associés au score effectué. Ce fichier est ensuite envoyé au serveur en PUT par la fonction send_results.

2 Site web

2.1 Technologies utilisées

Pour le frontend :

- HTML5
- CSS
- JavaScript

Pour le backend :

- PHP

La base de données est composée de fichiers CSV dont le contenu est séparer par des ";".

2.2 Organisation du serveur web

2.2.1 Frontend

Le frontend est réparti en 4 dossiers :

- assets : contient les images et autres fichiers

- css : contient les fichiers css
- html : contient les fichiers html
- js : contient les fichiers javascript

assets/ Le dossier assets contient un dossier img qui contient les images utilisées dans le site web, et un dossier fonts qui contient les polices utilisées dans le site web.

css/ Le CSS est réparti en deux fichiers : style.css et color.css. Le fichier style.css contient les styles généraux du site web, et le fichier color.css contient les couleurs utilisées dans le site web.

html/ Le dossier html contient les fichiers html du site web :

- landing.html : page d'accueil
- account.html : page de gestion du compte
- install.html : page contenant les instructions d'installation du jeu
- ranking.html : page contenant le classement
- rules.html : page contenant les règles et mouvements du jeu

js/ Le dossier js contient les fichiers javascript du site web :

- index_script.js : contient les requêtes AJAX pour la page d'accueil
- ranking_script.js : contient la requête AJAX pour la page ranking
- hide_gif.js : permet de cacher les gifs de la page rules.html

2.2.2 Backend

Le backend est réparti en 3 dossiers :

- à la racine du site on retrouve tous les fichiers php

- database : contient les fichiers csv constituants la base de données

- files : contient le package splashmem.deb ainsi que l'archive game.zip (de manière temporaire pour cette dernière)

- player_libs : contient les librairies uploadées par les joueurs

Le fichier JavaScript session_destroy_on_close.js est le seul fichier JS dédié au backend, puisqu'il permet de détruire la session lors de la fermeture de la page. Néanmoins, il n'existe pas vraiment de méthode pour gérer ça, donc l'efficacité de ce fichier est relative et il est possible que la session ne soit pas détruite, même à la fermeture de la page.
Ce fichier est nécessaire au bon fonctionnement du concept de connexion rapide. En effet comme la connexion rapide ne nécessite pas de mot de passe, l'idée est de détruire les données des utilisateurs ayant utilisé la connexion rapide, et ce même s'ils ne se déconectent pas eux même.

2.3 Fonctionnement du site web

2.3.1 Page d'accueil

Le code de la page d'accueil se trouve dans le fichier landing.html.
La page d'accueil est structurée de la manière suivante :

- Un header contenant les liens vers les autres pages

- le top 10 du classement

- un texte d'accueil

- Un formulaire de connexion contenant un champ pour le pseudo, un champ pour le mot de passe et un bouton de connexion

- Un formulaire d'inscription contenant un champ pour le pseudo, un champ pour le mot de passe, un champ pour la confirmation du mot de passe et un bouton d'inscription

- un formulaire de connexion rapide contenant un champ pour le pseudo et un bouton de connexion

- un formulaire de sélection des librairies composé de checkboxes pour chaque librairie ainsi qu'un bouton de validation

Formulaires de connexion Les trois formulaires de connexion de la page d'accueil envoient les données en POST à la page connection.php.
La méthode est la même pour les trois formulaires :

- si une session est ouverte, on la détruit en utilisant la fonction logout() du fichier functions.php, puis on crée une nouvelle session et on traite les données.

- sinon, on ouvre une session et on traite les données.

Le formulaire de connexion rapide permet de se connecter sans mot de passe. Il suffit de rentrer un pseudo et de cliquer sur le bouton de connexion. Celui-ci permet de créer des parties et d'uploader des librairies, mais toutes les entrées dans la base de données associées au pseudo choisi seront supprimées lors de la déconnexion. Il faut créer un compte si l'on veut conserver ses données.
Les pseudos sont limités en taille, pour faciliter leur récupération par le jeu. ainsi, un pesudo doit faire au maximum 25 caractère, puisque le nom d'une librairie est composée du pseudo ainsi que de 5 caractères de plus, cela permet de réserver 30 octets à la librairie.
Ainsi, un pseudo doit se conformer à l'expression régulière (RegEx) suivante :

```
1     pattern="^[a-zA-Z0-9-]{1,25}$"
2
```

Formulaire de sélection des librairies Les librairies sont rafraichies toutes les 20 secondes via une requête AJAX. Le code php permettant de récupérer les données de la base de données est contenu dans le fichier refresh_libs.php. La requête est dans le fichier index_script.js.
Le formulaire de sélection des librairies envoie les données en POST à la page game_setup.php. (voir section)
Le formulaire de sélection des librairies permet de sélectionner les librairies que l'on veut utiliser pour jouer. Il suffit de cocher les librairies que l'on veut utiliser et de cliquer sur le bouton de création. un lien de téléchargement apparaitera alors et en cliquant sur celui-ci, les librairies sélectionnées seront alors téléchargées dans un zip nommé game.zip.

Classement Le classement est rafraichi en temps réel via une requête AJAX. Le code php permettant de récupérer les données de la base de données est contenu dans le fichier refresh_ranking.php. La requête est dans le fichier index_script.js.

2.3.2 Page de gestion du compte

Le code de la page de gestion du compte se trouve dans le fichier account.html.
La page de gestion du compte est structurée de la manière suivante :

- Un header contenant les liens vers les autres pages

- Un formulaire de changement de mot de passe contenant un champ pour l'ancien mot de passe, un champ pour le nouveau mot de passe, un champ pour la confirmation du nouveau mot de passe et un bouton de validation

- Un formulaire de suppression de compte contenant un champ pour le mot de passe et un bouton de suppression

- Un formulaire de suppression des librairies permettant de sélectionner les librairies uploadées et de les supprimer.

Les trois formulaires ci-dessous envoient leurs données en POST, respecivement aux pages change_infos.php, delete_account.php et account.php.
Le header de la page de gestion du compte contient également un lien pour se déconnecter (voir section sur la déconnexion).

2.3.3 Page de classement

Le code de la page de classement se trouve dans le fichier ranking.html.
Contrairement au classement de la page d'accueil, celui-ci n'est pas limité à seulement le top 10. La navigation se fait via une propriété d'overflow scroll, évitant de surcharger visuellement la page.
Le classement est rafraichi en temps réel via une requête AJAX. Le code php permettant de récupérer les données de la base de données est contenu dans le fichier display_ranking.php. La requête est dans le fichier ranking_script.js.

2.3.4 Page des règles

Le code de la page des règles se trouve dans le fichier rules.html.
La page se compose de 2 catégories :

- Les règles du jeu

- Les mouvements du jeu

Mouvements Chaque mouvement a un nom et une illustration sous forme d'un gif. Le gif peut être caché en cliquant sur le nom du mouvement. Le code permettant d'afficher ou de cacher le gif se trouve dans le fichier hide_gif.js.
Il y a aussi une brève description du mouvement et son coût en crédits.

2.3.5 Installation du jeu

Le code de la page d'installation se trouve dans le fichier install.html.
La page d'installation contient les instructions pour installer les librairies utilisées par le jeu, ainsi qu'un lien permettant de télécharger l'archive du jeu et les instructions permettant de l'installer.

2.3.6 Déconnexion

Pour les comptes en connexion rapide, la déconnexion se fait en cliquant sur le lien "Déconnexion" dans le header. Pour les comptes ayant un mot de passe, il faut passer par la page de gestion du compte. La page de gestion du compte est accessible dans les headers de toutes les pages.

Fonctionnement de la déconnexion Le script de déconnexion est relativement simple : on démarre une session, on appelle la fonction logout() codée dans le fichier functions.php et on redirige vers la page d'accueil.

2.3.7 Fonctions PHP

Un fichier functions.php contient les fonctions PHP utilisées dans les pages du site web.
Les fonctions sont les suivantes :

rename_lib_as_player_name($curr_name, $player_name, $nb_libs) Cette fonction permet de renommer les librairies uploadées par les joueurs.
Son fonctionnement est simple : on récupère en amont le pseudo du joueur et le numéro de version de la dernière librairie qu'il a uploadé, on incrémente ce numéro et on on appelle la fonction qui se charge de renommer la librairie de la manière suivante : [pesudo]_[numéro de version].so.

logout() Cette fonction permet à un utilisateur de se déconnecter. Elle est utilisée dans quatre cas :

- lors de la connexion d'un utilisateur
- lors de la déconnexion volontaire d'un utilisateur
- lors du fonctionnement du script JS session_destroy_on_close.js, pour déconnecter l'utilisateur
- lors de la suppression d'un compte

Le fonctionnement est le suivant :
Lors de la connexion, une variable de session $_SESSION[üser] est créée. Elle va prendre la valeur 0 si c'est une connexion rapide, ou la valeur 1 si c'est une connexion avec mot de passe. Lors de l'appel de la fonction logout(), on regarde la valeur de cette variable. Si c'est 0, on supprime les données de l'utilisateur dans la base de données. Si c'est 1, on ne fait rien.
Dans le cas d'une suppression de compte, on passe la valeur de $_SESSION[üser] à 0 avant d'appeler la fonction logout().

2.3.8 game_setup.php

Le fichier game_setup.php est appelé lors de la création d'une partie. Il permet de créer un zip contenant les librairies sélectionnées par l'utilisateur.
Le fonctionnement est le suivant :

- On récupère les librairies sélectionnées par l'utilisateur
- On crée un dossier libs
- On copie les librairies sélectionnées dans le dossier libs
- On crée un fichier start.sh permettant de lancer le jeu avec les librairies contenues dans le dossier libs
- On crée un fichier game_config.bin contenant la configuration du jeu
- On crée un zip contenant le dossier libs, le fichier start.sh et le fichier game_config.bin
- On recharge la page avec un lien permettant de télécharger le zip.

le zip créé contient :

- un dossier libs contenant les librairies séléctionnées
- un fichier start.sh permettant de lancer le jeu avec les librairies contenues dans le dossier libs
- un fichier game_config.bin contenant la configuration du jeu

Un fichier zip est créé à chaque fois que l'on clique sur le bouton de création. Une fois qu'on a cliqué sur le bouton "créer", un lien permettant de télécharger le zip apparaît. Une fois le zip téléchargé, il est supprimé du serveur.

start.sh Le fichier start.sh contient une ligne permettant de récupérer le chemin absolu du script, puis de lancer le jeu.
Par exemple, avec les librairies lib_1.so et lib_2.so, le fichier start.sh créé ressemblera à ça :

```
#!/bin/bash
directory=$(dirname -- $(readlink -fn -- "$0"))
./opt/spashmem/splash $directory/game_config.bin $directory/libs/lib_1.so $directory/libs/lib_2.so

```

game_config.bin Le fichier game_config.bin contient les informations permettant au jeu de renvoyer les scores au serveur. Il est structuré de la manière suivante :

- 25 octets pour le pseudo du joueur ayant créé la partie
- puis, pour chaque librairie contenue dans le dossier libs, on a 25 octets pour le pseudo du joueur qui l'a importée, et 30 octets pour le nom de la librairie

De la même manière, le fichier game_config.bin d'un jeu créé par pseudo0 avec les librairies pseudo0_1.so et pseudo1_2.so, importées par les joueurs pseudo0 et pseudo1, ressemblera à ça :

```
00000000: 7073 6575 646f 3000 0000 0000 0000 0000  pseudo0.........
00000010: 0000 0000 0000 0000 7073 6575 646f 3000  ........pseudo0.
00000020: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000030: 0000 0000 0000 7073 6575 646f 305f 312e  ......pseudo0_1.
00000040: 736f 0000 0000 0000 0000 0000 0000 0000  so..............
00000050: 0000 0000 7073 6575 646f 3100 0000 0000  ....pseudo1.....
00000060: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000070: 0000 7073 6575 646f 315f 322e 736f 0000  ..pseudo1_2.so..
00000080: 0000 0000 0000 0000 0000 0000 0000 0000  ................

```