

Table des matières

1. Cahier des charges	1
2. Etat stocké dans SQL	4
3. GUI du joueur actif.....	7
4. GUI du joueur passif	7

1. Board Game Arena

1.1. Software versions

Software Versions

Versions currently used by BGA framework:

- Dojo Toolkit 1.15
- PHP: 8.2
- SQL: MySQL 5.7
- JS/CSS/HTML: limited by what minimization tools support: JS minimization
- Font Awesome: 4.7 <https://fontawesome.com/v4.7/icons/> (available as `<i class="fa fa-clock" />`)
- Font Awesome: 6.4.0 <https://fontawesome.com/v6/search?o=r&m=free> (available as `<i class="fa6 fa6-clock" />`)

PHP Extensions Used

The following PHP extensions are - as of May 8th, 2022 - in use in BGA Studio and available:

date, libxml, openssl, pcre, zlib, filter, hash, Reflection, SPL, session, standard, sodium, apache2handler, mysqlnd, PDO, xml, apcu, bz2, calendar, ctype, curl, dom, mbstring, FFI, fileinfo, ftp, gd, gettext, gmp, iconv, igbinary, json, exif, msgpack, mysqli, pdo_mysql, apc, posix, readline, shmop, SimpleXML, sockets, sysvmsg, sysvsem, sysvshm, tokenizer, v8js, xmlreader, xmlwriter, xsl, zip, Phar, memcached, Zend OPcache

Javascript minimization (after July 2020)

For performance reasons, when deploying a game the javascript code is minimized using terser (<https://github.com/terser/terser>). This minifier works with modern javascript syntax. From your project "Manage game" page, you can now test a minified version of your javascript on the studio (and revert to the original).

NB: it has been reported that there is an issue with this minifier and percentage values for opacity.

<https://github.com/terser/terser>:

A JavaScript mangler/compressor toolkit for ES6+

1.2. BGA architecture/protocole

- Début du jeu :
 - o PHP : appel de « setupNewGame »
 - Create the right SQL statements to initialize the database according to the game rules
- Chargement du jeu :
 - o PHP : appel de « getAllDatas »
 - o JS :
 - Create some dynamic HTML page ; let the client do the game setup ;
 - The HTML generated embed automatically the data previously generated (in « json » format).
 - Load images, javascript and css stylesheet
 - Appel de « setup »
- Joueur actif :
 - o JS : « OnClicMyMethod » envoie la requête par URL « /mygame/mygame/playAcard.html?card=99 »
 - o PHP : « playAcard » est appelée
 - o SQL : mettre à jour l'état du jeu
- Tous les joueurs :

- « notifyAllPlayers » est renvoyé à tous les joueurs (actif ou pas).
- La notification par un nom dédiée telle que « takeMoneyToken »
- JS : la méthode « notif_takeMoneyToken » est appelée avec des arguments (qui vient de jouer et quoi).
- Ensuite :
 - D'autres notifications sont envoyées :
 - Pour d'autres mises à jour des artefacts du jeu
 - Et finalement, pour indiquer le nouveau « joueur courant ».

1. Cahier des charges

- Pas d'IA
- Montrer les coups possibles
- Pas de demande de validation d'un coup.
 - Mais possibilité d'annuler un coup qui n'est pas complètement joué (comme dans l'implémentation Python)
- Montrer le même état du plateau à l'adversaire.
 - Même les états intermédiaires ; donc les hésitations de l'adversaire.
- Montrer la trace du dernier coup joué par l'adversaire, comme dans l'implémentation Python.
- En partie différée, juste montrer la trace du dernier coup.
- Sur demande, montrer la liste des coups joués.
- Montrer graphiquement les cubes capturés.

2. Etat stocké dans SQL

Groupe/Table	Variable	Type/Domaine	Commentaire
game_state	game_is_terminated	Bool	
(1 row)	game_winner	Black/White/None	
	game_credit	Int	game_credit >= 0
	game_turn	Int	game_turn >= 0 ; game_turn >= 1 dès que la partie a commencé
	current_player	Black/White/None	
	last_player	Black/White/None	Le dernier joueur qui a joué
	last_turn	Int	Le numéro du tour déjà joué
	last_legend	String	La légende du dernier coup joué ; vide si game_turn = 0
	last_src_hexagon	Int/None	La trace du dernier coup joué : position source
	last_int_hexagon	Int/None	La trace du dernier coup joué : position intermédiaire
	last_dst_hexagon	Int/None	La trace du dernier coup joué : position destination
Groupe/Table	Variable	Type/Domaine	Commentaire
last_hexagon_states	hexagon_index	Int	Key (hexagon_index >= 0)
(n rows)	hexagon_position_u	Int	Pourrait être codé en dur dans PHP
	hexagon_position_v	Int	Pourrait être codé en dur dans PHP
	hexagon_label	String	Pourrait être codé en dur dans PHP
	hexagon_config	Void/Single/Stack	
	hexagon_color	Black/White/None	Ce champ et les deux suivants pourraient se coder en une seule
	hexagon_bottom	Rock/Paper/Scissors/Wise/None	lettre : {'R', 'P', 'S', 'W', 'r', 'p', 's', 'w', '_'} avec '_' qui encode « vide »
	hexagon_bottom	Rock/Paper/Scissors/Wise/None	
Groupe/Table	Variable	Type/Domaine	Commentaire
last_captured_cubes	cube_color	Black/White	Key (part 1)
(n rows)	cube_type	Rock/Paper/Scissors/Wise	Key (part 2) ; ou bien une clé compacte {'R', 'P', 'S', 'W', 'r', 'p', 's', 'w'}
	cube_count	Int	Value
Groupe/Table	Variable	Type/Domaine	Commentaire
turn_legends	turn_index	Int	Key ; turn_index >= 0 ; ne pas afficher dans la liste des coups joués

(n rows)	turn_legend	String	
----------	-------------	--------	--

Groupe/Table	Variable	Type/Domaine	Commentaire
current_hexagon_states	hexagon_index	Int	Key ; hexagon_index >= 0
(n rows)	hexagon_position_u	Int	Pourrait être codé en dur dans PHP
	hexagon_position_v	Int	Pourrait être codé en dur dans PHP
	hexagon_label	String	Pourrait être codé en dur dans PHP
	hexagon_config	Void/Single/Stack	
	hexagon_color	Black/White/None	Ce champ et les deux suivants pourraient se coder en une
	hexagon_bottom	Rock/Paper/Scissors/Wise/None	seule lettre : {'R', 'P', 'S', 'W', 'r', 'p', 's', 'w', '_'} avec '_' qui encode « vide »
	hexagon_bottom	Rock/Paper/Scissors/Wise/None	
Groupe/Table	Variable	Type/Domaine	Commentaire
current_captured_cubes	cube_color	Black/White	Key (part 1)
(n rows)	cube_type	Rock/Paper/Scissors/Wise	Key (part 2) ; ou bien une clé compacte {'R', 'P', 'S', 'W', 'r', 'p', 's', 'w'}
	cube_count	Int	Value

Groupe/Table	Variable	Type/Domaine	Commentaire
turn_state	action_index	1/2	1 ^{ère} ou 2 ^{nde} action
(1 row)	action_src_hexagon	Int/None	None ⇔ pas de position sélectionnée
	action_src_config	Single/Stack/Top	Top : le cube du haut d'une pile
	action_dst_hexagon	Int/None	None ⇔ pas de position sélectionnée
	action_choices	List[Int]	Liste calculée en PHP des cases sélectionnables

- Les hexagones de « action_choices » sont tous à cliquer et ils sont rendus :
 - en **surbrillance** claire **au survol** de souris si « action_src_hexagon == None » ou si « action_dst_hexagon == None »
 - en **surbrillance** foncée **fixe** si « action_src_hexagon != None »
- La valeur « action_index » pourrait être rendue dans le GUI, y compris celui de l'adversaire.

3. GUI du joueur actif

- Minimum d'intelligence
- Trois zones :
 - o Le plateau des hexagones
 - o Les cubes capturés
 - o La légende des coups déjà joués
- Deux boutons :
 - o Pour montrer ou cacher la zone contenant la liste des coups déjà joués
 - o Pour annuler le coup en préparation
- Seulement deux types d'événement sur les hexagones
 - o **Survol** de certains hexagones : changement immédiat de l'affichage
 - o **Click** de certains hexagones : renvoi de l'événement vers le serveur
- Piloté par l'état courant SQL :
 - o Quoi afficher et comment
 - o Quelles sont les hexagones dynamiques (associés aux deux types d'événements)

4. GUI du joueur passif

- Aucune d'intelligence
- Réplication de l'état courant du plateau : après un coup joué ou même l'état intermédiaire
 - o Uniquement les hexagones sélectionnés par l'adversaire
 - o Pas les réactions au survol de la souris par l'adversaire.