TODO :

9/4/2021 et 10/4/2021

Gerer le cas ou l’utilisateur entre une autre valeur que les coordonnee dans case apres le prompt.

Limiter le nombre de guerriers qui peuvent etre misent dans une case cependant le type de la tuile. [done]

Le regex utilizer pour valider le choix de tuile devrait etre definit par les limites de la Map.

[done]

Le regex devrait utiliser le string {nbRepetition} si la taille du map depasse le 9 suivant le height ou le width

[done]

Gerer le ca sou l’utilisateur entre une autre valeur qu’une taille valid de l’armee :

* Si la taille de l’armee depasse le nombre de querriers disponibles [done]
* Si la taille de l’armee depasse la limit definit par la tuile [done]
* Si le string entree par l’utilisateur n’est pas une chiffre valid (inferieur a 1) [done]
* Si le string entree par l’utilisateur n’est pas du tout valid (un String quelconque) [done]

Lorsque la taille de la map devienne tres grande l’impression deviendra inorganisé a cause des espaces dans le header :

* les indices ne sont pas au milieu de la colonne des tuiles ; [done]
* Et a gauche si on a deux chiffres comme indice en passant aux nombres > 9 il y a une difference entre les espaces ; [done]

Des fois l’algorithme de genration de tuile nous done des resultat indesirables : des tuiles touts seul ☹

[done]

Sans utilisation de instanceof dans le projet [done]

Touts les actions(les attacks/supports sur les tuiles adjacentes) [done]

11/4/2021

Une manière pour récolter et gérer les ressources des tuiles-> la production doit dépendre du nombre de guerriers déploie ou non ? on va choisir oui par ce que sinon la production ne sera pas suffisante pour soutenir les armees deploees. Le nombre de ressource chaque tour depend de la taille de l’unitee et de la tuile. On doit différencier entre les diffèrent types de ressources pour les convertir après avec la commande ; 2 : Exchange (ressources) ;

Système pour gerer le gold des unites/players

Système pour gerer le food des unitees/players

ESSAYON D’ECRIRE UNE DESCRIPTION DU JEU :

1. Choix du jeu :

Au debut l’utilisateur va choisir le type du jeu, soit jeu de development agricole soit jeu de guerre ;

L’utilisateur ensuite va choisir le nombre de joueurs (entre 2 et 4)

Si on est dans debug mode les noms des joueurs sont choisis automatiquement sinon il seront entrer par l’utilisateur

Finalement le joueur va choisir la taille de la plaque (map size)

1. Debut du jeu :
2. Choosing a game type and players set up:

the users are prompted to chose which type of game to play: WAR or FARM.

the users are prompted to enter the number of Players to play the game.

the Players are prompted to enter their names.

the Players are asked to choose map size.

1. Game start:

A map object is created with the size given by the users (not necessarily square, could be rectangle).

Each player is assigned a name inside the game object

The first player to play is chosen in a random way.

Check if current game is war\_game

If true

check if the current player has soldiers left; if not just skip to *nextTurn()*

print (“make choice…”)

if 1:

prompt user to chose tile

check if tile *isEmpty()*

if not empty reprint prompt to check for user choice

get the tileType to set the limit of soldier that could be put in it

chose how many users to make army

make army then put the soldiers (army) in the tile

1. Update the number of soldiers left for the user

2. Update the owner and the number of units inside the tile

3. set the production type for the army

4. set how much gold is consumed nextRound() (IN THE TILE OR IN THE ARMY??)

5. set how much resources are produced for nextRound()

1. Next turn: