**« Pet Rescue Saga » \_ Cahier de charges**

**Les animaux :**

* Se trouvent sur les blocs ;
* Se déplacent avec les blocs OU tout seuls (vérifier déplacement de côtés) ;
* S’ils arrivent jusqu’au “sol”  sauvés et disparaissent ;
* Sauvetage de chaque animal apporte X points au joueur ;
* Peuvent être ajoutés au cours de partie avec un/des blocs colorés ou sans ;
* Ne sont pas tués si un bloc tombe par-dessus, ou s‘ils tombent, ou si la bombe s’explose à côté ;

**Les blocs colorés mobiles :**

* Ont qq couleur (jaune, rouge, bleu, vert, violet) ;
* Peuvent se rejoindre dans un groupe selon la couleur (1 groupe = 2 blocs ou plus de la même couleur a coté c’est-à-dire qui ce touche par un coté.) ;
* Le groupe disparaît si on appuie sur n’importe quel bloc du groupe ;
* Disparition de chaque bloc apporte X points (10 ? 20 ?) au joueur ;
* Sont ajoutés au cours de la partie si la longueur de colonne de blocs devient plus petite que la hauteur du plateau de jeu ;
* Est-ce qu’ils sont ajoutés par les côtés ? Ou bien que par-dessus ?

**Les blocs immobiles (environnement) :**

* Ne peuvent être détruits par qq outils ;
* Ne bougent pas ;

**Les outils :**

1. Pics

* Sert à faire disparaître un seul bloc coloré ;
* Comment joueur les obtient pendant la partie ?
* Joueur peut l’acheter à tout moment en payant de l’or (1 pic = Y lingots d’or) ;
* Pour l’activer il faut avoir les coordonnées de bloc à détruire, puis « activer » ou « annuler » .

1. Ballons

* Ont qq couleur (jaune, rouge, bleu, vert, violet) ;
* Ils apparaissent pendant la partie à partir de qq niveau du jeu (apparition est-elle random ou programmée ?)
* Si le joueur fait activer un ballon de couleur C, tous les blocs de couleur C sur le plateau disparaissent ;

1. Bombe

* Explose si le jouer l’appuie ;
* En s’explosant elle fait disparaître +1 bloc tout autour de lui (vérifier si c’est le cas, en tous dimensions ?), sauf les animaux et les bloc d’environnement ;
* Ils apparaissent pendant la partie à partir de qq niveau du jeu (apparition est-elle random ou programmée ?)

1. Fusée

* S’explose si le jouer l’appuie ;
* En explosant elle fait disparaître une colonne de blocs, sauf les animaux et les bloc d’environnement (vérifier si c’est le cas) ;

Les outils peuvent s’obtenir dans les blocs spéciaux (des caisses) Les caisses ce casse lorsque un blocs voisin est cassé. (Soit les caisses donne un outils aléatoire, soit il est dessiné sur la caisse.

**Compte de joueur :**

* Points obtenus au cours de la partie (vérifier tous les moyens de l’obtention);
* Lingots d’or (comment joueur les obtient ? si le joueur sauve plus d’animaux que c’était précisé pour cette partie ?)

**Partie de jeu :**

* Différentes conditions selon le niveau de jeu ;
* Partie se termine si :
* Tous les animaux pour cette partie sont été sauvés  parie est gagnée ;
* Il n’y a plus de coup possible, i.e. il n’y a plus de groupe de blocs, ni de ballons, ni de pics, ni de bombe, ni de fusée MAIS il y a >= 1 animal non-sauvé  parie est perdu.
* Le temps de partie s’écoule – pour qqs niveaux de jeu (est-ce qu’on a besoin de ça ? cela veut dire qu’il faut ajouter class Timer)  parie est perdu;
* Le nombre de coups devient = 0 – pour qqs niveaux de jeu (est-ce qu’on a besoin de ça ?) ;
* Étoiles caractérisent à quel point de succès la partie a été jouée ? (Élaborer des critères ex : temps, nombre de coups restant)
* Report de partie : nombre d’étoiles pour partie, nombre de points gagnés, lingots d’or gagnés ;
* Mise à jour le compte de joueur (nombre de points gagnés, lingots d’or gagnés) ;
* Mise à jour le Map de jeu (+1 tour apparaissent comme joué, nombre d’étoiles pour ce tour, des tours disponibles à jouer (si accès est progressif)) ;

**Complexité de partie (nivaux du jeu)**

* Demander au joueur de sauvé certain animaux ou tous.
* Faire jouer le joueur avec plus de couleur différentes
* Partie sur une carte plus grande.

**Map de jeu :**

Simple = tableaux avec des chiffres ? (int [ ][ ])   
Ou liste chaînée avec des cases voisines (haut, bas, gauche, droite) : avantage ça permet de facilement traiter les cases voisines récursivement Inconvénient c’est un peu plus long a mettre en place.

Accessibles = couleurs vives (avec drawRect et filRect ca devrais pas être trop dur)

sans accès = couleurs pale/gris ? (Je crois que c’est juste dans le terminale cette partie la, on peu afficher un nombre pour chaque couleur, une lettre pour chaque animale et des lettres grec (ou des majuscules) pour les autres trucs.)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |