

Sujet : 1

Titre : Jeu de Go

Description :

Développer un jeu de Go (https://fr.wikipedia.org/wiki/Jeu_de_go).

Liberté au niveau des langages : C(++), Java, Python, Haskell, Prolog, ...

Objectifs :

- développer une version texte du jeu fonctionnelle pour deux joueurs humains (en règles chinoises)
- optionnel : intégrer une intelligence artificielle qui fait des coups au hasard
- optionnel : intégrer une intelligence artificielle de type Monte Carl

Ressources :

Thèmes :

algo., intelligence artificielle

Commentaires de l'équipe d'étudiant :

Sujet : 2

Titre : Mastermind

Description :

Le Mastermind

(confère <http://fr.wikipedia.org/wiki/Mastermind>) ou Master Mind est un jeu de société pour deux joueurs dont le but est de trouver une combinaison de pions de couleur, autrement dit le but est de « casser un code ». Il en existe de nombreuses variantes.

Présentation : le jeu se présente sous la forme d'un plateau perforé ayant généralement 10 rangées de quatre trous pouvant accueillir des pions de couleurs. Il y a généralement 8 couleurs : rouge ; jaune ; vert ; bleu ; orange ; blanc ; violet ; fuchsia. De part et d'autre de cette rangée principale il y a de petits trous pour accueillir des petits pions blancs et rouges (ou noirs) utilisés pour donner des indications à chaque étape du jeu.

Principe : chaque joueur joue un tour. Le premier joueur commence par placer son choix de 4 pions (couleur et position) sans qu'ils soient vus de l'autre joueur. Ce dernier doit ainsi trouver la combinaison choisie en un nombre donné d'étapes. Le but est de trouver le plus rapidement possible en se basant sur les indications que fournies le premier joueur à chaque étape.

Étape d'un tour : à chaque étape, le second joueur propose une combinaison en plaçant des pions. Le premier joueur lui donne ensuite des indications : le nombre de pions de la bonne couleur bien placés en utilisant le même nombre de petits pions rouges (parfois noirs selon la version du jeu) et le nombre de pions de la bonne couleur mal placés en utilisant de petits pions blancs.

Tactique : essayer d'obtenir le maximum d'indications, notamment en se basant sur les tours précédents, d'autant que le nombre de propositions possibles est limité.

Fin du jeu : le second joueur gagne s'il trouve la bonne combinaison le plus tôt possible et sans dépasser le nombre de combinaisons possibles.

Objectifs :

- Cahier des charges
- Conception (diagrammes de cas d'utilisation, scénarios, diagramme de classes, diagrammes de séquences et éventuellement diagrammes d'états-transitions ou diagrammes d'activité).
- Faire une application paramétrable.
- Développement

Ressources :

Thèmes :

conception - algo - programmation - tests.

Commentaires de l'équipe d'étudiant :

Sujet : 3

Titre : Organisation d'un tournoi interne de badminton

Description :

Un club nantais de badminton organise tous les ans un tournoi interne entre ses nouveaux et ses anciens adhérents. Ce tournoi a lieu sous la forme de plusieurs matchs de doubles dans lesquels chaque paire est composée d'un nouvel adhérent et d'un ancien adhérent. A l'issue du tournoi, le meilleur nouveau et le pire ancien sont désignés.

Objectifs :

réaliser un prototype fonctionnel

Ressources :

Thèmes :

algorithmique, base de données, web

Commentaires de l'équipe d'étudiant :

Sujet : 4

Titre : Organisation d'un tournoi interne de badminton

Description :

Un club nantais de badminton organise tous les ans un tournoi interne entre ses nouveaux et ses anciens adhérents. Ce tournoi a lieu sous la forme de plusieurs matchs de doubles dans lesquels chaque paire est composée d'un nouvel adhérent et d'un ancien adhérent. A l'issue du tournoi, le meilleur nouveau et le pire ancien sont désignés.

Objectifs :

prototype fonctionnel

Ressources :

Thèmes :

algo, base de données, web

Commentaires de l'équipe d'étudiant :

Sujet : 5

Titre : Motus

Description :

Ce projet consiste à développer le Motus avec une interface graphique.
Le choix du langage de programmation est libre.

Objectifs :

Ressources :

Thèmes :

Algorithmique, programmation,

Commentaires de l'équipe d'étudiant :

Sujet : 6

Titre : Jeu Bataille navale

Description :

Ce projet consiste à développer le jeu bataille navale avec une interface graphique.

Le choix du langage de programmation est libre.

Objectifs :

Ressources :

Thèmes :

Algorithmique, programmation

Commentaires de l'équipe d'étudiant :

Sujet : 7

Titre : Colons de Catane

Description :

1 Présentation du jeu

Le projet qui vous est demandé est un jeu de plateau *librement inspiré* des Colons de Catane.

https://fr.wikipedia.org/wiki/Les_Colons_de_Catane

Le plateau est constitué de cases carrées contenant des ressources.

Les joueurs peuvent construire, sous certaines conditions, des routes, des colonies, ou transformer une colonie en ville.

Chaque construction se fait en dépensant une certaine quantité de ressources.

Les ressources sont récoltées autour des colonies (ou des villes).

Un joueur gagne lorsqu'il atteint 10 points de victoire. Les points de victoire se comptent en nombre de villes et colonies, plus certains bonus.

2 Début de partie

Au début de la partie, chaque joueur se voit attribuer une couleur et un nouveau plateau est créé.

La taille du plateau est variable et doit prendre en compte le nombre de joueur, il semble qu'une centaine de cases soit nécessaire pour jouer à deux, mais c'est à vous de décider ce que vous proposerez au joueur.

Le type de ressource de chaque case est fixé aléatoirement en début de partie.

Un numéro entre 2 et 12 est attribué aléatoirement pour chaque case. Bien que l'attribution des numéros aux cases soit aléatoire, tous les numéros doivent être attribués à peu près le même nombre de fois.

Une fois le plateau créé, les joueurs placent chacun leur tour une colonie où ils le souhaitent sur le plateau, ainsi qu'une route collée à cette colonie. Puis un deuxième tour de placement à lieu avec à nouveau une colonie et une route par joueur.

La partie peut maintenant commencer.

3 Déroulement de la partie

Les joueurs jouent chacun leur tour. Le tour d'un joueur se déroule de la façon suivante :

- Un lancé de dés 1 permet la récolte des ressources (voir section 3.1).
- Le joueur peut ensuite lancer une phase de commerce, lui permettant d'échanger des ressources. (voir section 3.2).
- Le joueur peut ensuite construire des routes, des colonies ou des villes.

Lorsqu'il le décide, il passe la main au joueur suivant.

3.1 Récoltes

Le lancer de dés donne un nombre entre 2 et 12. Tous les joueurs dont une colonie ou une ville est voisine d'une case numérotée par le nombre tiré récoltent les ressources correspondantes à cette case : une unité de ressource si c'est une colonie, deux si c'est une ville. Une case est voisine d'une autre si elles partagent un côté (les diagonales sont donc exclues).

Les différents types de ressources sont : bois, argile, laine, blés, pierre.

3.2 Commerce

Les joueurs peuvent convertir 4 unités d'une même ressource en 1 unité de la ressource de leur choix.

3.3 OPTIONS

3.3.1 Échange entre joueurs

Vous pourrez ajouter la possibilité d'échanger des

ressources entre joueur (les joueurs se mettent d'accord entre eux sur un taux d'échange).

3.3.2 Construction de ports

Certaines positions de la carte peuvent, à l'initialisation du jeu, être définies comme étant des ports de commerce. Si un joueur construit une colonie à une telle position il bénéficiera d'un meilleur taux de conversion de ressources.

Les ports se situent sur les bords de la carte, la carte étant entourée par la mer.

Il existe différents types de port, les ports généralistes, permettant d'échanger n'importe quelle ressource à un taux de 3 :1 (au lieu de 4 :1), et les ports spécialisés, ne permettant que d'échanger un type de ressource (contre n'importe quelle autre type) à 2 :1.

Par exemple, si je possède une colonie sur un port spécialisé dans le bois, je pourrais échanger 2 unités de bois contre 1 unité d'argile, de laine, de blé ou de pierre.

4 - Conditions de victoire

Pour gagner la partie il faut avoir 10 points de victoire à la fin de son tour de jeu. les points de victoire d'un joueur se calculent de la façon suivante :

- 1 point par colonie
 - 2 points par ville
 - 1 point supplémentaire pour le joueur possédant la route la plus longue
- Si deux joueurs ont la route la plus longue, le joueur ayant le premier construit sa route de taille maximale obtient le point de victoire.

5 -Règles de construction

Colonies:

Un joueur ne peut construire une colonie que sur une case voisine d'une route lui appartenant. Il ne peut construire une colonie à moins de deux cases d'une colonie (ou ville) existante.

La construction d'une colonie coûte une unité de bois, une d'argile, une de blé et une de laine.

Villes :

Un joueur peut transformer en ville une colonie lui appartenant.

La construction lui coûte deux unités de blé et trois de pierre.

Routes : Un joueur ne peut construire une route que sur une case voisine d'une route, d'une colonie ou d'une ville lui appartenant.

La construction d'une route coûte deux unités de bois et une d'argile.

Objectifs :

Ressources :

Thèmes :

algo, prog objet

Commentaires de l'équipe d'étudiant :

Sujet : 8

Titre : Jeu Snake

Description :

Ce projet consiste à concevoir et à développer le jeu snake en utilisant le langage C, principalement la bibliothèque graphique SDL de ce langage.

Objectifs :

Ressources :

Thèmes :

programmation C, interface graphique

Commentaires de l'équipe d'étudiant :

Sujet : 9

Titre : Contrôle de la maison
- Pgm à la demande (S2)

Description :

Execution de programme de controle à la demande

La maison intelligente est une maison dotée d'équipements divers avec des dispositifs contrôlés (via logiciels).

Nous nous intéressons au contrôle donc au logiciel embarqué sur les équipements ou les dispositifs de contrôle.

Le logiciel a ici la particularité d'être en étroite interaction avec les équipements physiques contrôlés.

Dans le cadre de ce projet de S2 nous voulons expérimenter le controle à la demande pour différents équipements ou différentes parties dans la maison. Il peut y avoir des incohérences, des concurrences d'accès, etc

Les usagers de la maison peuvent programmer des controles différents pour les divers équipements partagés.

On ne peut donc pas exécuter directement les programmes ; un pretraitement pour identifier les anomalies est indispensable.

Les ordres de contrôle peuvent être donnés directemet sur un boitier principal ou à distance via un sms ou un tweet.

Une partie de l'application est déjà programmée. Nous voulons aller plus loin dans les possibilités :

- adaptation des contrôles logiciels à la demande ;

Les étudiants vont étudier, concevoir et expérimenter une plateforme de supervision des équipements connectés qui sont sous controle.

- Visualisation des ordres, des interactions et des eventuelles incohérences,
- Aide au diagnostic, à la planification et à la gestion d'alerte

Objectifs :

Prototype, poster, demonstration, documentation

Ressources :

documentation existante,
version précédente

Thèmes :

Algo, programmation ; interaction avec des processeurs arduino, bases de données,

Commentaires de l'équipe d'étudiant :

Sujet : 10

Titre : Extension Chrome de mise en évidence des mots-clés

Description :

Les mots-clés sont les mots (ou les expressions) qui permettent de décrire le contenu principal d'un document.

En s'appuyant sur les travaux récents en extraction automatique de mots clés, les étudiants devront réaliser une extension Chrome qui exploite les mots-clés de chaque page visitée par l'utilisateur.

Travail :

- 2 à 4 méthodes différentes devront être implémentées pour extraire les mots-clés (TextRank, TopicRank, etc.)
- Différentes exploitations des mots-clés extraits devront être proposées :

* Nuage de mots-clés

* Mise en évidence dans le texte

* Proposition de documents similaires, selon Google

* etc.

Les étudiants doivent soigner la modularité de l'extension.

De nouvelles méthodes d'extraction de mots-clés ainsi que de nouvelles exploitations de ceux-ci doivent pouvoir être aisément intégrées.

Objectifs :

Ressources :

L'encadrant fournit aux étudiants une bibliographie au sujet des méthodes d'extraction de mots-clés.

Thèmes :

Développement web ; conception.

Commentaires de l'équipe d'étudiant :

Sujet : 11

Titre : Gestion de RDV entre parents et professeurs

Description :

Chaque année, différents lycées ou collèges mettent en place une journée de rencontre entre les enseignants et les parents d'élèves. Un enseignant rencontre ces derniers dans une salle à une heure précise pour chaque parent.

L'objectif de ce projet est de fournir une interface en Java, dans un premier temps, pour gérer tous ses RDV.

au niveau enseignant :

- Afficher une liste de parents avec les différents RDV.
- Valider une rencontre et passer au parent suivant,
- Annuler un RDV, etc...

Au niveau parents :

Voir en temps réel les différents RDV et les différentes salles.

Langages : Java, Java Swing ou Awt.

SGBD : Oracle

Ce projet continuera en S3 et S4 avec une application web et une application sous Android.

Objectifs :

Ressources :

Thèmes :

Mise en place d'une base de données oracle,

Accès à la base de données oracle via JDBC (Java).

IHM en Java awt ou swing.

Commentaires de l'équipe d'étudiant :

Sujet : 12

Titre : mini grep

Description :

Écrire un clone de l'outil Unix standard grep, qui permet de rechercher dans un fichier texte les lignes qui correspondent à un motif donné sous la forme d'une expression rationnelle.

Objectifs :

Objectif minimal : supporter les opérations de base sur les expressions rationnelles : produit, somme et étoile.

Ressources :

Algorithmes et documentations diverses

Thèmes :

algorithmique, mathématiques (théorie des automates), système

Commentaires de l'équipe d'étudiant :

Sujet : 13

Titre : Poker

Description :

Implémenter un jeu de Poker en C++ et en Haskell (ou scala)

Objectifs :

Ressources :

Thèmes :

algo/prog objet/ prog fonctionnelle/

Commentaires de l'équipe d'étudiant :

Sujet : 14

Titre : La chasse au Wumpu

Description :

Hunt the Wumpus (ou Chasse au Wumpus, en français) est un des premiers jeux informatiques. Il consiste en une partie de cache-cache avec un monstre, le Wumpus, tapi dans un réseau de cavernes souterraines.

Le Wumpus occupe l'un des sommets de ce dodécaèdre

Dans sa version d'origine, le joueur donne ses instructions par ligne de commande. Il peut se déplacer de salle en salle ou tirer une flèche selon un chemin donné. Le terrain de jeu consiste de 20 salles, chacune étant reliée à trois autres. Géométriquement, cette carte est équivalente à un dodécaèdre dont chaque sommet serait une salle. Quelques pièges sont également présents sous la forme de fosses de vase et de chauves-souris.

Par déduction, le joueur est capable de localiser la salle dans laquelle se trouve le Wumpus, sans y avoir jamais pénétré. À ce point, il lance une flèche dans la direction supposée du monstre. Si le Wumpus s'y trouve, il remporte la partie ; sinon, la partie est perdue.

dixit Wikipedia...

la description originale : <http://www.atariarchives.org/bcc1/showpage.php?page=247>

Objectifs :

Ressources :

Thèmes :

algo., prog. objet, intelligence artificielle

Commentaires de l'équipe d'étudiant :

Sujet : 15

Titre : Abalone

Description :

Réaliser un programme permettant de jouer à l'abalone. Le travail consiste à

- se familiariser avec les règles du jeu ;
- réfléchir aux structures de données permettant de coder le jeu ;
- implanter le jeu lui-même ;
- enfin, dans la mesure du possible, réfléchir sur la possibilité de doter le jeu d'une IA.

Objectifs :

Ressources :

Thèmes :

algo., prog. objet, intelligence artificielle

Commentaires de l'équipe d'étudiant :

Sujet : 16

Titre : Awele

Description :

Réaliser un programme permettant de jouer à l'awele. Le travail consiste à

- se familiariser avec les règles du jeu ;
- réfléchir aux structures de données permettant de coder le jeu ;
- implanter le jeu lui-même ;
- enfin, dans la mesure du possible, réfléchir sur la possibilité de doter le jeu d'une IA.

Objectifs :

Ressources :

Thèmes :

algo., prog. objet, intelligence artificielle

Commentaires de l'équipe d'étudiant :

Sujet : 17

Titre : Gestion des vœux des étudiants

Description :

Les étudiants d'info 2 développent une application destinée à faciliter la saisie des vœux de poursuite d'étude. C'est un projet conséquent, qui s'étale sur plusieurs années.

Objectifs :

Il s'agit d'améliorer la sécurité, l'ergonomie et les fonctionnalités de cette application.

Ressources :

Thèmes :

BDD, Sécurité

Commentaires de l'équipe d'étudiant :

Sujet :

Titre :

Description :

Objectifs :

Ressources :

Thèmes :

Commentaires de l'équipe d'étudiant :

Sujet :

Titre :

Description :

Objectifs :

Ressources :

Thèmes :

Commentaires de l'équipe d'étudiant :

Sujet :

Titre :

Description :

Objectifs :

Ressources :

Thèmes :

Commentaires de l'équipe d'étudiant :

Sujet :

Titre :

Description :

Objectifs :

Ressources :

Thèmes :

Commentaires de l'équipe d'étudiant :

Sujet :

Titre :

Description :

Objectifs :

Ressources :

Thèmes :

Commentaires de l'équipe d'étudiant :

Sujet :

Titre :

Description :

Objectifs :

Ressources :

Thèmes :

Commentaires de l'équipe d'étudiant :

Sujet :

Titre :

Description :

Objectifs :

Ressources :

Thèmes :

Commentaires de l'équipe d'étudiant :