Lecture d'un programme en RedCode Reader et Parser Traitement

L'objet Instruction Les erreurs

CoreWar

Classes principales Début du ieu La houde du

Génération programme L'algorithme génétique Combat Sélection Evolution

Croisement

Travail Personnel Approfondi: CoreWar

Chaid Akacem Jasmine Baudin Hugo

Université de Caen

21 avril 2019

Introduction

Lecture d'un programme en RedCode Reader et Parser Traitement L'objet

Les erreurs CoreWar

Classes principales Début du jeu La boucle du jeu

Génération de programme L'algorithme génétique Combat Sélection Evolution

Croisement

CoreWar

- Développer une machine virtuelle
- Construire des programmes performants

Introduction

Lecture d'un programme en RedCode

Reader et Parser Traitement

L'objet Instruction Les erreurs

Les erreu

CoreW

Classes principales Début du jeu La boucle du jeu

Génération de programme L'algorithme

L'algorithm génétique Combat Sélection Evolution Croisement

Canalusian

on

1 Lecture d'un programme en RedCode

Reader et Parser

Traitement

L'objet Instruction

Les erreurs

2 CoreWar

Classes principales

Début du jeu

La boucle du jeu

3 Génération de programme

L'algorithme génétique

Combat

Sélection

Evolution

Croisement

4 Conclusion

Introduction

Lecture d'un programme en RedCode

Reader et Parser

Traitement L'objet Instruction Les erreurs

C===\\/=

Classes principales Début du jeu La boucle du jeu

Génération de programme L'algorithme génétique Combat Sélection Evolution

Deux classes pour effectuer la lecture

Classe Reader

- Lecture basique d'un fichier texte
- Pré-traitement

Classe Parser

- Création d'un programme exécutable
- Découpage d'une ligne
- Vérification
- Conversion

Introduction

Lecture d'un programme en RedCode Reader et

Traitement L'objet Instruction

Instruction Les erreurs

CoreWa

Classes principales Début du jeu La boucle du jeu

Génération de programme

L'algorithme génétique Combat Sélection Evolution Croisement

• Découpage de la ligne

• Traitement différencié selon opérateur



Figure – Schéma général d'une ligne RedCode

Exemple traitement différencié

La ligne ADD #0 1 est orienté vers le traitement par défaut. La ligne JMP 5 est orienté vers le traitement pour les opérateurs à un champ.

Introduction

Lecture d'un programme en RedCode Reader et

Reader et Parser Traitement

L'objet Instruction Les erreurs

CoreWa

Classes principales Début du jeu La boucle du jeu

Génération de programme L'algorithme génétique Combat Sélection

Evolution Croisement

Les instructions

- Instruction : Classe abstraite
- Trois méthodes principales

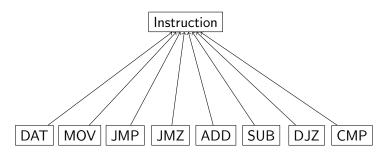


Figure – Héritage de l'instruction

Introduction

Lecture d'un programme en RedCode Reader et Parser Traitement L'objet

Les erreurs

CoreW/a

Classes principales Début du jeu La boucle du jeu

Génération de

programme
L'algorithme
génétique
Combat
Sélection
Evolution
Croisement

nclusion

Une gestion des erreurs pendant la lecture

Erreurs de syntaxe

- Ligne non conforme
- Opérateur/Mode d'adressage non reconnu
- Adresse qui n'est pas un nombre

Instructions non exécutables

- Ligne syntaxiquement correcte
- Nécessite qu'un objet Instruction soit instancié
- Bloquée avant son exécution

Exemple Instruction non exécutable

La ligne ADD 0 1 n'est pas exécutable La ligne DAT 0 n'est pas exécutable (Raison de *cohérence*)

Introductio

Lecture d'un programme en RedCode Reader et Parser

Traitement L'objet Instruction

Les erreurs

CoreW

Classes principales Début du jeu La boucle du

Génération de programme L'algorithme génétique Combat Sélection Evolution

Croisement

La partie CoreWar : deux objets principaux

La classe VirtualMachine

- Machine virtuelle du logiciel
- Exécution
- Stockage
- Insertion des programmes

La classe Warrior

• Un programme à exécuter

Introduction

Lecture d'un programme en RedCode Reader et Parser Traitement L'objet

Les erreurs CoreWar

Classes principales

Début du jeu La boucle du jeu

Génération de programme L'algorithme génétique Combat Sélection Evolution

Croisement

Deux initialisations de la mémoire

- Avec des données (Ligne DAT #0)
- Avec les deux programmes qui vont se combattre
 - 1 Placement du pointeur du premier programme
 - 2 Placement du premier programme
 - 3 Placement du pointeur du second programme selon les contraintes
 - 4 Placement du second programme

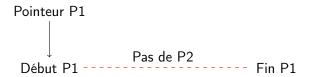


Figure - Contraintes de placement

Introduction

Lecture d'un programme en RedCode Reader et Parser Traitement L'objet Instruction Les erreurs

CoreWar

Classes principales Début du jeu La boucle du jeu

Génération de programme L'algorithme génétique Combat Sélection Evolution

Conclusion

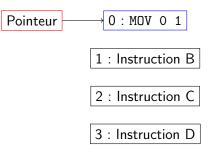


Figure – Boucle de jeu

Introduction

Lecture d'un programme en RedCode Reader et Parser Traitement L'objet Instruction

Les erreurs

Classes principales Début du jeu La boucle du jeu

Génération de programme L'algorithme génétique Combat Sélection Evolution

Conclusion

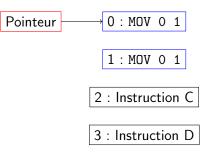


Figure – Boucle de jeu

Introduction

Lecture d'un programme en RedCode Reader et Parser Traitement L'objet Instruction Les erreurs

CoreWar

Classes principales Début du jeu La boucle du jeu

Génération de programme L'algorithme génétique Combat Sélection Evolution

Conclusion

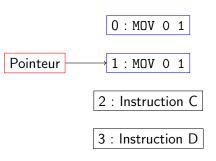


Figure - Boucle de jeu

Introduction

Lecture d'un programme en RedCode Reader et Parser Traitement L'objet Instruction Les erreurs

CoreWar

CoreWar

Classes

principales

Début du jeu

La boucle du
jeu

Génération de programme L'algorithme génétique Combat Sélection Evolution

Conclusion

0 : JMP 1
Pointeur 1 : DAT #0
2 : Instruction C
3 : Instruction D

Figure - Boucle de jeu

Introductio

Lecture d'un programme en RedCode Reader et Parser

Traitement L'objet Instruction Les erreurs

CoreWa

Classes principales Début du jeu La boucle du jeu

Génération de programme

L'algorithme génétique Combat Sélection

Evolution Croisement

nclusion

- Créer un programme performant
- Analogue à la biologie
 - Sélection
 - Mutations
 - Croisement des individus sélectionnés
- Début aléatoire

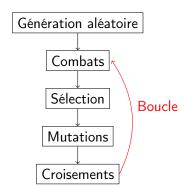


Figure – Déroulement de l'algorithme

Lecture d'un programme en RedCode Reader et

Parser Traitement L'objet Instruction Les erreurs

Classes principales Début du ieu La houcle du

Génération programme L'algorithme génétique

Combat Sélection

Evolution Croisement

Partie Combat

- Méthode fight
- Adaptation au problème
- Statistiques sur les Warriors

Notre implémentation : Tournoi

- Complexité raisonnable
- Les meilleurs confrontés aux meilleurs

Lecture d'un programme en RedCode Reader et

Parser Traitement L'objet Instruction Les erreurs

Classes principales Début du ieu La houcle du

Génération programme L'algorithme génétique Combat

Sélection Evolution Croisement

Partie Sélection

- Méthode doSelection
- Garder la meilleure partie de la population
- A partir des statistiques générées

Notre implémentation : Les 20 meilleurs

- Adapté à notre implémentation du combat
- Complexité d'un algorithme de tri

Partie Evolution

CoreWar

Introductio

Lecture d'un programme en RedCode

Parser Traitement L'objet Instruction Les erreurs

CoreW

Classes principales Début du jeu La boucle du jeu

Génération de programme L'algorithme génétique Combat Sélection

Evolution Croisement

nclusion

- Méthode mutAllWarriors
- Outils de base
- Réutilisation avec un seul outil, muni des probabilités
- Intervient sur les programmes

Notre implémentation

Outils de base

- Remplacement d'une ligne
- Délétion d'une ligne
- Rajout d'une ligne

Le tout encapsulé dans une classe qui va utiliser les trois outils successivement.

Introductio

Lecture d'un programme en RedCode

Reader et Parser Traitement L'objet

Les erreurs

CoreWa

Classes principales Début du jeu La boucle du jeu

Génération de programme L'algorithme génétique Combat Sélection

Croisement

Partie Croisement

- Méthode cross et crossAll
- Nombre de croisements définis
- Prend les instructions des meilleurs programmes
- Créé de nouveaux programmes

Notre implémentation

- Sélection aléatoire de deux programmes
- Sélection aléatoire de ligne

Introduction

Lecture d'un programme en RedCode Reader et Parser Traitement L'objet

Instruction Les erreurs

CoreWa

Classes principales Début du jeu La boucle du jeu

Génération de programme L'algorithme génétique Combat Sélection Evolution Croisement

Conclusion

Conclusion

- Un CoreWar complétement orienté objet
- Une gestion des erreurs avant exécution
- Une génération de programme modulable selon les besoins