



Corewar

Frédéric CHIPOT / Florian DAYRE / Louis DUPLANTIER / Alex FOURNIER / Gael HALNAUT



1) Introduction au projet

Corewar :

- Jeu de contrôle d'espace mémoire
- “Guerriers” codés en RedCode s'affrontent dans une machine virtuelle MARS

1) Introduction au projet

Corewar :

- Jeu de contrôle d'espace mémoire
- “Guerriers” codés en RedCode s'affrontent dans une machine virtuelle MARS

DAT	terminate process
MOV	move from A to B
ADD	add A to B, result in B
SUB	subtract A from B, result in B
MUL	multiply A by B, result in B
DIV	divide B by A, result in B if A is not zero, else DAT
MOD	divide B by A, remainder in B if A is not zero, else DAT
JMP	execute at A
JMZ	execute at A if B is zero
JMN	execute at A if B is not zero
DJN	decrement B, if B is not zero, execute at A
SLT	skip if A is less than B
SEQ	skip if A is equal to B
CMP	
SNE	skip if A is not equal to B
NOP	no operation
SPL	new task at A

1) Introduction au projet

Corewar :

- Jeu de contrôle d'espace mémoire
- “Guerriers” codés en RedCode s'affrontent dans une machine virtuelle MARS

DAT	terminate process
MOV	move from A to B
ADD	add A to B, result in B
SUB	subtract A from B, result in B
MUL	multiply A by B, result in B
DIV	divide B by A, result in B if A is not zero, else DAT
MOD	divide B by A, remainder in B if A is not zero, else DAT
JMP	execute at A
JMZ	execute at A if B is zero
JMN	execute at A if B is not zero
DJN	decrement B, if B is not zero, execute at A
SLT	skip if A is less than B
SEQ	skip if A is equal to B
CMP	
SNE	skip if A is not equal to B
NOP	no operation
SPL	new task at A

.A	Instr. read and write A-fields
.B	Instr. read and write B-fields
.AB	Instr. read A-field of A-instr. and B-field of B-instr. and write B-field
.BA	Instr. read B-field of A-instr. and A-field of B-instr. and write A-field
.F	Instr. read both A&B fields of A&B instr. and write to both A&B fields (Ato A and B to B).
.X	Instr. read both A&B fields of A&B instr. and write to both A&B fields (Ato B and B to A).
.I	Instr. read and write Instr., Modifier, Modes, A & B fields

1) Introduction au projet

Corewar :

- Jeu de contrôle d'espace mémoire
- “Guerriers” codés en RedCode s'affrontent dans une machine virtuelle MARS

DAT	terminate process
MOV	move from A to B
ADD	add A to B, result in B
SUB	subtract A from B, result in B
MUL	multiply A by B, result in B
DIV	divide B by A, result in B if A is not zero, else DAT
MOD	divide B by A, remainder in B if A is not zero, else DAT
JMP	execute at A
JMZ	execute at A if B is zero
JMN	execute at A if B is not zero
DJN	decrement B, if B is not zero, execute at A
SLT	skip if A is less than B
SEQ	skip if A is equal to B
CMP	
SNE	skip if A is not equal to B
NOP	no operation
SPL	new task at A

.A	Instr. read and write A-fields
.B	Instr. read and write B-fields
.AB	Instr. read A-field of A-instr. and B-field of B-instr. and write B-field
.BA	Instr. read B-field of A-instr. and A-field of B-instr. and write A-field
.F	Instr. read both A&B fields of A&B instr. and write to both A&B fields (A to A and B to B).
.X	Instr. read both A&B fields of A&B instr. and write to both A&B fields (A to B and B to A).
.I	Instr. read and write Instr., Modifier, Modes, A & B fields

#	<i>immediate</i>
\$	direct
@	indirect using B-field
<	predecrement indirect using B-field
>	postincrement indirect using B-field
*	indirect using A-field
{	predecrement indirect using A-field
}	postincrement indirect using A-field



1) Introduction au projet

Corewar :

- Jeu de contrôle d'espace mémoire
- “Guerriers” codés en RedCode s'affrontent dans une machine virtuelle MARS
- Un vainqueur ou des ex-aequo



1) Introduction au projet

Corewar :

- Jeu de contrôle d'espace mémoire
- “Guerriers” codés en RedCode s'affrontent dans une machine virtuelle MARS
- Un vainqueur ou des ex-aequo

Objectif :

- Développer un algorithme d'évolution permettant de générer des “guerriers”



2) Présentation de l'existant

Article présentant un algorithme d'évolution

An Evolutionary Approach Generates Human Competitive Corewar Programs

Barkley Vowk¹, Alexander (Sasha) Wait², Christian Schmidt³

¹*University of Alberta, Department of Mathematical and Statistical Sciences, Edmonton, AB T6G2E1, Canada*

²*Harvard Medical School, Department of Genetics, 77 Avenue Louis Pasteur, Boston, MA 02115, USA*

³*Glasower Damm 4R, 15831 Mahlow, Germany*

fizmo@corewar.info



2) Présentation de l'existant

Article présentant un algorithme d'évolution

ExMars

- Simulateur de MARS
- Amélioration de Exhaust

2) Présentation de l'existant

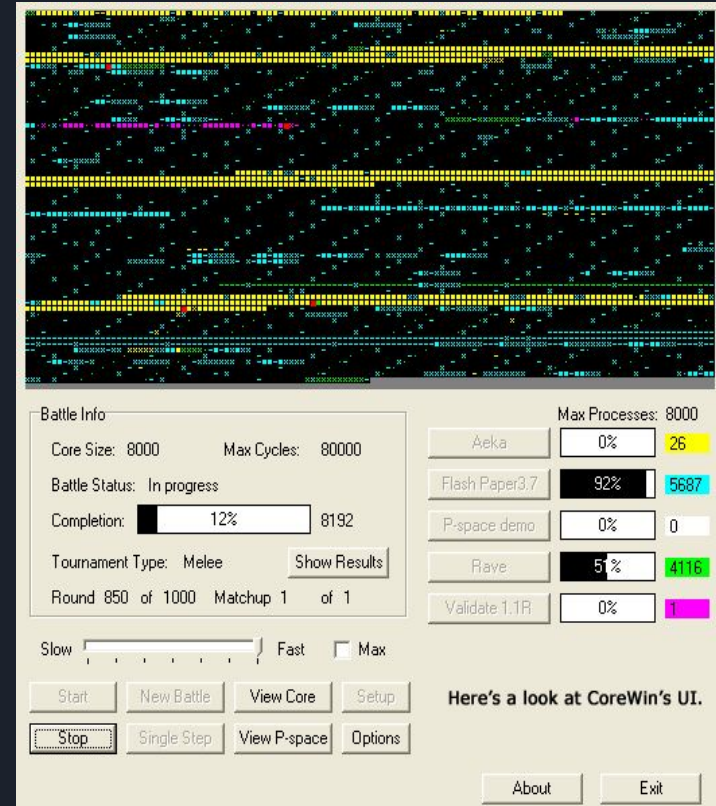
Article présentant un algorithme d'évolution

ExMars

- Simulateur de MARS
- Amélioration de Exhaust

Corewin

- Interface graphique
- Facilité à lancer des combats





2) Les besoins principaux

- Algorithme d'évolution
- Interface entre Mars et l'algorithme d'évolution
- Statistiques pour la génération 0
- Convertisseur de RedCode
- Implémentation des combats
- Paramètres d'algorithme d'évolution
- Suivi d'exécution d'algorithme

3) Les réalisations

```
{
  "version": 31,
  "generations": 1000,
  "battles_by_generation": [{
    "gen_number": 0,
    "value": 20
  }, {
    "gen_number": 75,
    "value": 25
  }],
  "num_warriors": [{
    "gen_number": 0,
    "value": 30
  }, {
    "gen_number": 75,
    "value": 25
  }],
  "A_value_modifier_min": 0,
  "A_value_modifier_max": 7999,
  "B_value_modifier_min": 0,
  "B_value_modifier_max": 7999,
```

```
  "survivors": [{
    "gen_number": 0,
    "value": 0.2
  }, {
    "gen_number": 100,
    "value": 0.5
  }],
```

```
  "breedings": [{
    "gen_number": 0,
    "value": 0.3
  }, {
    "gen_number": 100,
    "value": 0.2
  }],
```

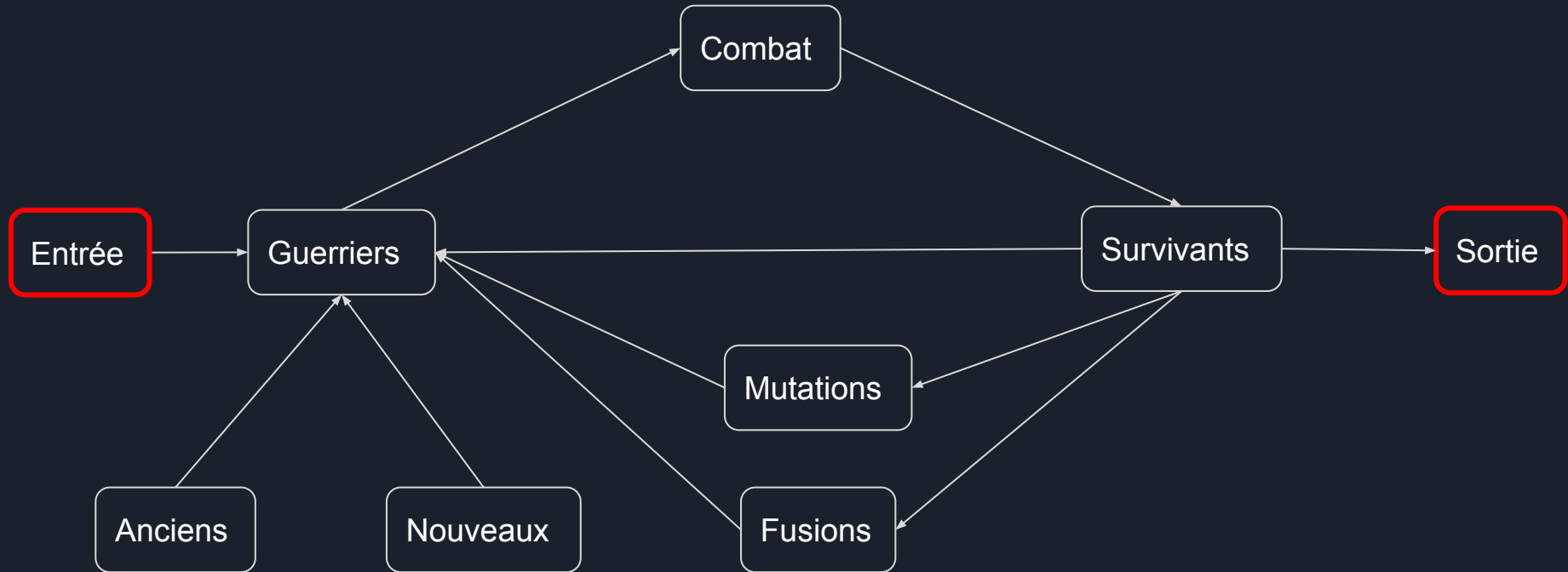
```
  "mutations": [{
    "gen_number": 0,
    "value": 0.3
  }, {
    "gen_number": 100,
    "value": 0.2
  }],
```

```
  "fresh": [{
    "gen_number": 0,
    "value": 0.15
  }, {
    "gen_number": 100,
    "value": 0.05
  }],
```

```
  "old": 0.05,
  "p_mutation_line": 0.1,
  "p_mutation_field": 0.2,
  "breed_length_min": 4,
  "breed_length_max": 10,
  "generator_path":
    "../data/Warriors/Infinite",
  "generator_depth": 2,
  "time_tested": 4,
  "score": 123.68325042724609
}
```

3) Les réalisations

Déroulement de l'algorithme d'évolution



3) Les réalisations

```
C:\Users\lshigh\Projects\core-war\src\x64\Release\Evolution algorithm.exe
best warrior of this battle : 23_40_121_451_477_560_670_703_745_773_795_818_877
**** Generation 43 ****
#####
best warrior of this battle : 23_40_121_451_477_560_670_703_745_773_795_818_877_898_917
**** Generation 44 ****
#####
best warrior of this battle : 23_40_121_451_477_560_670_703_745_773_795_818_877_898_917_956
**** Generation 45 ****
#####
best warrior of this battle : 23_40_121_451_477_560_670_703_745_773_786_824_870_893_915_946_962
**** Generation 46 ****
#####
best warrior of this battle : 23_40_121_451_477_560_670_703_745_773_795_818_877_898_917_956_979
**** Generation 47 ****
#####
best warrior of this battle : 23_40_121_451_477_560_670_703_745_773_786_824_853_900_918_941_959_989
**** Generation 48 ****
#####
best warrior of this battle : 23_40_121_451_477_560_670_703_745_773_786_824_870_893_915_946_962_991
**** Generation 49 ****
#####
best warrior of this battle : 23_40_121_451_477_560_670_703_745_773_786_824_853_900_918_941_959_989_1006_1048
**** Generation 50 ****
#####
best warrior of this battle : 23_40_121_451_477_560_670_703_745_773_786_824_853_900_918_941_959_989_1006_1077
**** Generation 51 ****
#####
best warrior of this battle : 23_40_121_451_477_560_670_703_745_773_786_824_853_900_918_941_959_989_1006_1038_1060_1090
**** Generation 52 ****
#####
_
```

Capture d'écran de la génération des guerriers par l'algorithme d'évolution

3) Les réalisations

```
ORG          START
MOV.B  @      -98, >    242
MOV.BA $    3618, >    643
SPL.BA # -3973, {   -293
JMZ.F  } -2270, {   3415
SPL.BA # -3973, {   -293
SPL.BA #  2899, {   -293
MOV.I  @    3470, }    242
JMN.BA -1452, * -1193
SPL.BA # -3973, {   -293
SPL.BA # -3973, {   -293
MOV.F  }    3470, }  -834
JMN.BA -1452, * -1193
SPL.BA # -3973, {   -293
SPL.BA # -3973, {   -293
MOV.F  }    3470, }  -834
START SPL.BA # -3973, {   -293
      SPL.BA # -3973, {   -293
      MOV.I  #   3114, }    242
      SPL.BA >  3618, *   -293
      MOV.B  * -1777, *  2299
      SPL.F  # -3973, > -1064
      SPL.BA #  -547, #   2086
      SPL.BA <  3155, #    682
      SPL.I  <   971, {   1330
      SPL.BA # -3973, {   -293
      MOV.I  @    3470, }    242
      SPL.A  } -1036, {   -293
      SPL.A  } -1036, {   -293
      END          START
```



4) L'architecture

Architecture modulaire



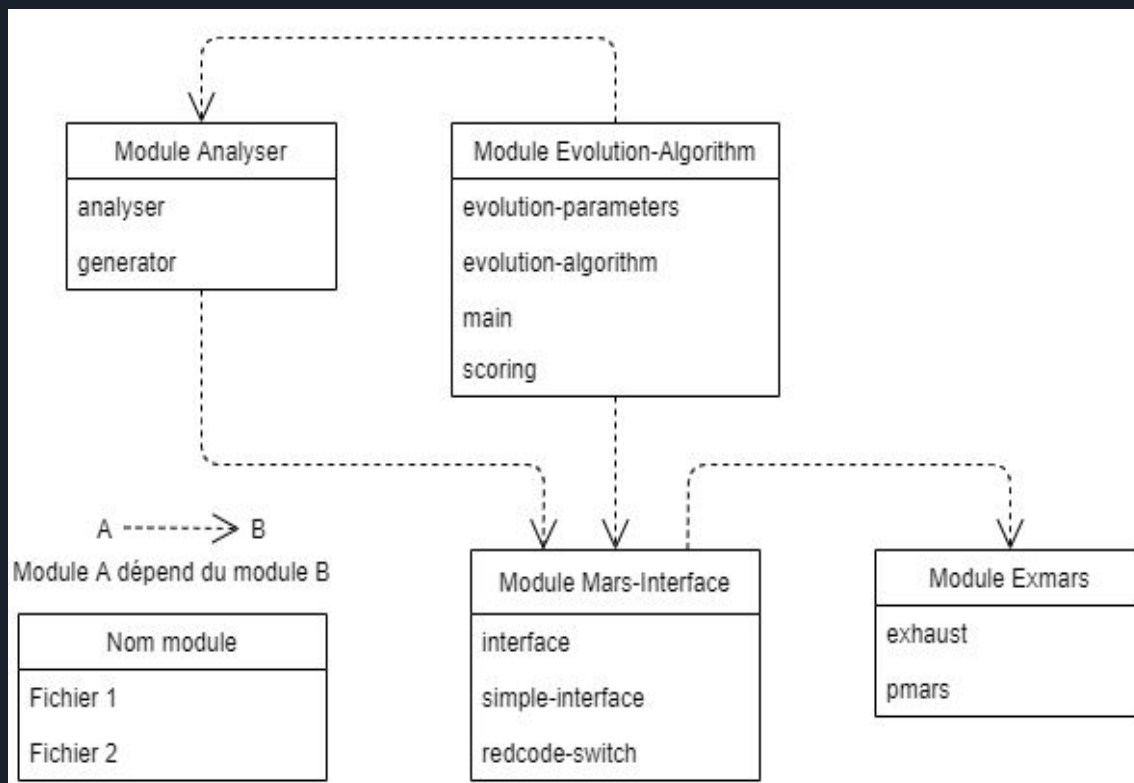
4) L'architecture

Architecture modulaire

4 modules :

- ExMars (existant)
- Mars-Interface
- Analyser
- Evolution-Algorithm

4) L'architecture





5) Tests

Test 1 : Comparatif des guerriers de génération 0 contre les guerriers de dernière génération

ID	Warrior	Author	Length	Score	Given	W%	L%	T%2	T%3	T%4	T%5	T%6
1	new4.red		0,99904	1166,6	1172,9	0,3	0,5	10,8	88,4	0,0	0,0	0,0
3	new10.red		0,99904	1139,4	1201,4	0,1	1,9	9,6	88,4	0,0	0,0	0,0
2	new7.red		0,99904	1005,9	1342,6	0,0	9,6	1,9	88,4	0,0	0,0	0,0
6	No Name		0,99904	0,0	3500,0	0,0	100,0	0,0	0,0	0,0	0,0	0,0
5	old8.red		0,99904	0,0	3500,0	0,0	100,0	0,0	0,0	0,0	0,0	0,0
4	old5.red		0,99904	0,0	3500,0	0,0	100,0	0,0	0,0	0,0	0,0	0,0



5) Tests

Test 2 : Modification de la profondeur pour la génération des statistiques

- Profondeur 2 : 101,3
- Profondeur 3 : 103,6
- Profondeur 4 : 101,6
- Profondeur 5 : 111,2
- Profondeur 6 : 96,3
- Profondeur 7 : 105,1

Résultat : Valeur optimale à profondeur 5



6) Conclusion

Bilan du projet :

- Algorithme d'évolution fonctionnel
- Génération de statistiques sur un pool de guerriers
- Sauvegarde des résultats des différents paramétrage
- Aucun paramètre intéressant trouvé avec un grand nombre de génération
- Les guerriers ne sont presque pas capable de gagner et se construisent autour de la survie



6) Conclusion

Extensions possibles :

- Algorithmes d'évolution supplémentaire
- Différents pool de guerriers pour les statistiques de la génération 0
- Mise en place d'un principe "d'espèce" de guerrier
- Outil de visualisation des paramètres et résultats amélioré



Bibliographie

Documentation sur Corewar et le Redcode :

- [1] John K. Lewis, Corewars for Dummies, <https://Corewar.co.uk/lewis/index.htm>, (Janvier 2021)
- [2] Ilmari Karonen, Beginner's Guide to Redcode, <https://vyznev.net/Corewar/guide.htm>, (Janvier 2021)

Simulateurs de MARS

- [3] Joonas Pihlaja, exhaust 1.9.2, 27 Juillet. 2004, <https://Corewar.co.uk/pihlaja/exhaust/index.htm>, (Février 2021)
- [4] Martin Ankerl, Corewars Simulators Overview, 29 Juilllet 2006, <https://martin.ankerl.com/2006/07/29/corwars-simulators-overview>, (Accédé en Février/Mars 2021).

Corewar et l'évolution

- [5] John Perry, Core Wars Genetics: The Evolution of Predation, Janvier 1991, <https://Corewar.co.uk/perry/evolution.htm>, (Février 2021)
- [6] Terry Newton, Using Core Wars to Simulate Evolution, 7 avril 2009, <http://newton.freehostia.com/cwevol.html>, (Février/Mars 2021)
- [7] Linus Thorsell, Evolving warriors, 1 novembre 1999, <https://Corewar.co.uk/thorsell/paper.htm>, (Février/Mars 2021)

Présentation de l'algorithme d'évolution pour Corewar

- [8] Barkley Vowk, Alexander (Sasha) Wait, Christian Schmidt, An Evolutionary Approach Generates Human Competitive Corewar Programs, 2004, <https://Corewar.co.uk/vowk/alife9ac.pdf>, (Janvier à Avril 2021)