

# 로보코드 코리아컵 2004

---

16강 진출 로봇 전략 모음

한국IBM developerWorks

## 16강 진출자 및 로봇

- 김가영 KillRobot\_1.0 (미제출)
- 김현연 Poseidon\_1.0
- 김형찬 ZLibra\_1.11
- 김후림 LinuxJoa\_3.53
- 박성수 sungsuZZANG\_1.0
- 박성열 Pazik\_1.0
- 신인철 Mirae\_1.1
- 손주호 SigmaType\_A\_0.8
- 양인상 Dotdog\_1.5
- 정일영 ExplorEr3\_1.0
- 정제현 OperatorZeze\_1.05
- 조병민 Panicroom\_1.1V
- 최문수 Trickster\_1.02
- 최종관 RedBot\_0.1
- 허찬 XII
- 황준식 Chun\_1.0

# Poseidon\_1.0

## 김현연

### ■ 움직임 :

#### 1. Melee시

- 1) 최적 위치를 검색하여 목표 좌표를 선정한다.
- 2) 목표 좌표로 이동 시 반중력 알고리즘을 사용하는데 목표좌표에 중력을 발생하여 이동한다.
- 3) 최적 위치의 반경 100픽셀 안에 들었을 경우 가장 가까이 있는 적에 대해 90도 각을 유지하며 램덤한 속도로 움직인다.
- 4) 최적 위치 검색 알고리즘
  - 전장을 20X20의 작은 블록으로 나눈다
  - 각 블록에서 적과의 거리가 가까울수록 패널티를 가산 한다.
  - 현 위치에서 각 블록으로 이동시 적과 적사이를 통과하는 경우 패널티를 가산한다.
  - 가장 낮은 점수를 얻은 블록을 최적 위치로 선정한다.

#### 2. OneOnOne

- 1) 적과 항상 90도의 각을 유지하게 움직인다.
- 2) 이동할 좌표가 전장의 밖이면 이동 각을 조정하여 전장의 안으로 좌표를 수정하여 이동한다.

# Poseidon\_1.0

## (Continue)

- Targeting :
  1. 적 스캔시 가속도를 구하여 저장하고 가상 탄환을 발사한다.
  2. 가상탄환의 예상 적중 시간에 실제 적의 위치와 탄환의 위치 사이의 차이를 구하여 저장 한다.
  3. 현재까지 저장된 적의 가속도에 대한 평균을 구하여 적용하는데 적과의 거리가 가까울수록 적용하는 가속도의 수를 적게 하여 최근의 Trend를 반영한다.
  4. 마지막으로 구해진 가속도를 가지고 선형 타겟팅에 적용하여 적의 예상 좌표를 구하여 가상탄환의 오차값을 평균을 내어 적용 발사한다.

# ZLibra\_1.11

## 김형찬

- 공격전략 :  
wave를 이용한 guessFactorTargeting을 사용했다.
- 움직임:  
wave를 이용한 WaveSurfing과 GuessFactor를 사용했고 벽에 가까이 가면 부드럽게 슬라이딩 하도록 벽면 임계값 및 WallSmoothing을 사용했다.
- 레이더:  
sweep을 이용한 최적탐색을 했다.
- 데이터:  
한 라운드가 끝날때마다 저장하고 시작할때 읽어들이었다.

# LinuxJoa\_3.53

## 김후림

- LinuxJoa의 주요 핵심은 많은 테스트를 통하여 얻은 자료를 바탕으로 만들어 졌다는 것이다.

구할 수 있는 모든 로봇과의 대결을 통하여, 로봇들의 행동패턴을 크게 몇 가지로 분류하여 그에 맞는 독자적인 사격방법과 로봇의 움직임을 만드는데 중점을 두었다. 또한, 이전 경기의 결과를 파일로 저장, 다음 경기에 참고할 수 있게 하여 가장 적절한 사격 방법과 로봇의 움직임을 선택하도록 하였다.

많은 변화를 주기보다는, 간단하지만 꼭 필요한 것을 그 때 그 때 사용하도록 만든 로봇이 바로 LinuxJoa 이다.

# sungsuZZANG\_1.0

## 박성수

- 공격 :  
targeting 은 크게 두 가지 전략으로 나누었다. 하나는 적 탱크가 있는 위치 그대로 총알을 발사하는 것이며 다른 하나는 적 탱크의 이동을 예측하여 예측된 곳으로 총알을 발사하는 것이다. 적 탱크의 이동은 현재 적 탱크의 속도와 위치 등을 고려하여 계산하였다. 처음 전투시 두 가지 전략을 혼용하여 사용하다가 총알 hitting ratio 를 계산하여 어느 정도가 넘어가면 한 가지 전략을 사용하도록 하였다. Virtual bullet, pattern matching 등의 고급 기술은 사용하지 않았다.
- 방어(움직임) :  
1) melee  
전장의 모든 탱크를 vector로 저장하여 scan시마다 정보를 update했으며 현재 가장 가까운 적을 target으로 삼았다. Target에게 멀어지는 방향으로 움직이도록 했으며 가능한 한가한 지역으로 움직이도록 설계 하였다. 반중력은 사용하지 않았으며 전장의 중심을 피하지 않는다.

# sungsuZZANG\_1.0

## (Continue)

- 방어(움직임) :  
2) one and one  
제일 중요하게 생각한 point는 최대한 랜덤하게 움직이도록 설계하고자 하였다. 그를 위하여 chaos의 logistic map 방정식을 이용하여 움직임에 가중치로 주었다. 기본적인 움직임은 적의 에너지 변화를 감지하여 적이 총알 발사했을 때 반대방향으로 움직이도록 하였으며, 적과 탱크의 몸체가 수직을 이루게 하였다.
- 간략한 전략 소개 :  
적의 속도와 위치를 통해 이동 방향을 예측하여 총알을 발사하고 chaos logistic map을 적용하여 최대한 랜덤하게 움직이도록 하여 적의 총알을 회피하며 적 탱크를 중심으로 앞뒤로 원 운동한다.

# Pazik\_1.0

## 박성열

- 기본컨셉 :  
적과의 거리를 위주로 모든상황을 판단하여 전투한다.
- 공격 :  
거리가 가까울 수록 강한 공격을 한다.
- 방어 :  
벽 회피, 적을 중심으로 원운동, 모든 움직임은 적과의 거리에 따라 속도를 달리한다.

# Mirae\_1.1

## 신인철

- 이동 : 반중력이론 + 시계방향 이동
- 안테나 : 최소 이동각 계산으로 최소 이동각으로 스캔(다수적인 경우)
- 총알 : 적의 이동 방향과 속도를 연산하여 예상 지점으로 발사

# SigmaType\_A\_0.8

## 손주호

- 처음에 로보코드를 보면서 생각했던 것이 단체전일 때와 개인전일 때의 상황이었다. 대회에서도 예선 때와 본선 때의 게임 진행 방식이 틀리다는 것을 명시하고 있었기에 각각의 상황에 대한 로봇의 대전방식이 필요하다고 생각했다.

우선 단체전일 때는 레이더를 팔방으로 회전시켜서 모든 적들의 정보를 얻는 것이다. 그 정보들을 이용해서 가장 가까이 있는 적부터 우선 공격하게 한다. 그리고 총알이 남발하는 상황에서 잘 맞지 않아야 한다. 이것은 강화된 반중력 움직임을 이용해서 최대한 적게 맞게 했다. 몇 번의 테스트를 거쳐 보았지만 확실히 어떤 확률에 의해서 점수를 많이 획득하는 경우가 많았다. 단체전에서 이기기 위해선 이 확률을 높이는 것이 관건이라고 생각한다.

개인전에서는 적에 대한 정보를 끊임없이 읽어오고, 그 정보에 따라 로봇의 움직임을 제어하는 것이 이길 확률이 높아졌다. 적이 총알을 발사했을 경우 좌우로 회피하는 코드를 추가시켜서 조금 덜 맞게 되었다.

그리고 포신과 총알을 강화해야 했다. 적을 조금 더 잘 맞춰서 에너지를 뺏어오는 것이야말로 후반으로 갈수록 이길 확률이 상승한다. 공식에 의한 로봇의 에너지가 상승하기 때문이다.

점수를 많이 얻기 위한 최고의 전략은 ‘최대한 적게 맞고, 최대한 적을 많이 맞추는 것이다.’ 여기에 초점을 두고 로봇을 제작하였다.

# SigmaType\_A\_0.8

## (Continue)

- 전략 줄임 :  
로보코드를 분석하면 다음 세 가지를 제어하는 게 곧 전략이라고 생각한다.
- 1) 이동 :  
적의 사격을 혼란시키기 위해서 반중력 이론을 기본토대로 제작하였다.  
그리고 문제점들을 고쳐나가면서 좀 더 나은 움직임이 나오도록 했다. 회피율을 높이는 것이야말로 로봇 이동의 최대 관건이다.
  - 2) 포신(총알) :  
사격은 승리를 보장한다. 적을 최대한 많이 맞추는 것이 이기는 지름길이다. 또한, 거리에 따른 총알의 세기를 결정하는 것도 중요하다. 이것은 로봇의 에너지를 효율적으로 이용하기 위한 것이다.
  - 3) 레이더 :  
적의 정보를 얻어내기 위해서 가장 중요한 부분이다.  
레이더로 적들의 정보를 알 수 있기 때문에 전략이 적용될 수 있다.  
개인전일 땐 레이더가 적을 놓치는 인터벌이 적어야 한다.

# Dotdog\_1.5

## 양인상

- 제1회 대회에 출전했던 DotDog v1.3을 업그레이드 하여 1.5버전과 1.6버전을 제작하였고 1.5버전은 공격위주, 1.6은 수비위주로 제작하였다. 1.5버전은 적과의 거리에 비례한 탄환의 알파각을 조정(거리가 멀수록 알파각을 넓힌다) 했으며, 많이 쏘면 언젠간 맞는다는 신념으로 탄환의 속도를 조절하여 탄환이 동시에 많이 발사 되도록 했다.

역시 이전 버전과 마찬가지로 특별한 복잡한 알고리즘을 멀리하고 단순하고 큐티한 방법으로 확률을 높였다.

# ExplorEr3\_1.0

## 정일영

- 스캔:  
적이 있는 영역만 스캔한다.
- Targeting:  
선형, circular, pattern-matching, GuessFactor를 이용한 방법 중에 확률이 높은 것을 선택해서 조준한다.
- 움직임:  
반중력을 사용하여, 벽과 적위치에 반중력을 걸고, 적이 없는 영역에 중력을 걸어서 한다.

# OperatorZeze\_1.05

## 정제현

- 1:1 전투시 적 로봇을 맞추기 위한 조준 방법은 **pattern-matching** 기법을 응용하여 구현하였으며, 현재 행동 패턴이 과거 행동과 유사할수록 매칭이 될 가능성이 높다고 보아 포탄의 세기를 강하게 발사하도록 하였다. 단체전에서는 패턴 매칭과 함께, 가장 가까운 적을 주 표적으로 하여 포탄을 발사한다.
- 움직임은 반중력 운동을 기본으로 하고 있으며, 상대의 포탄을 피하기 위한 움직임도 반중력을 이용하여, 상대의 포탄이 도달하리라 예측되는 부분을 **Random** 으로 선택하여 그 부분에 중력값을 할당, 유연한 움직임으로 포탄을 피할 수 있도록 설계하였다.

# Panicroom\_1.1V

## 조병민

- 움직임:  
Panicroom의 기본적인 이동 전략은 적이 탄환을 쏘을 때 피하는 것이다. 몸체를 움직이는 타이밍은 적의 로봇이 탄환을 발사함과 동시에 에너지가 감소하므로 그것을 통해 찾게 된다. 또한 적의 에너지 변화를 통해 탄환의 세기를 예측하고 움직이는 속도와 거리도 고려해 이동할 수 있도록 하였다. 또한 벽에 부딪치는 것을 막기 위해 벽 근처에 가게 되면 우회하도록 하였다.
- 공격:  
곡선 움직임에는 약한 방법이지만 직선 움직임을 예측하는 방법을 사용하였다. 이 방법은 적로봇의 위치와 속도, 방향을 계산한 후에 탄을 발사하게 된다. 전략적으로 곡선 움직임을 갖는 로봇에는 진짜로 약하다.
- 레이더:  
가장 많이 사용하는 방법인 적을 중심으로 좌우로 움직이는 전략을 사용했다. 때문에 적의 상태를 빠르게 관측할 수 있다.

# Trickster\_1.02

## 최문수

- 레이더 :  
라운드 시작시 최악 5틱 안에 모든 적을 스캔한다.  
이후 적들의 절대각을 갱신 정렬하여 모두를 포함하는 최단 각으로 왕복한다.  
일대일에서는 적의 중심으로 약 30도 크기의 아치를 그리며 추적한다.
- 몸체 :  
반중력을 사용하며 적의 접근과 빠른 반응에 밀려다니는 단점을 보완하고자 적의 반발력은 충돌하지 않을 만큼의 반발력만 주었고 실질적으로 움직임은 주도하는 것은 반중력 처리되는 적의 총알이다.  
매틱 위치가 갱신되며 회피 유무는 웨이브의 개념을 사용하고 상황에 따라 예측된 총알 헤딩의 변화를 주어 다양한 에임에 대비한다.
- 총 :  
패턴매칭을 사용하며 적의 속도, 헤딩변화율, 베어링차, 이 세 가지의 패턴을 누적하여 두 가지 타게팅을 만들어낸다. 각 타게팅별로 나타난 결과 값에 따른 가상 총알은 히트 유무를 판별하여 실제 총알에 반영한다. 에너지 관리차 파워 조절을 하고 있으며 조절된 파워는 정확도를 위해 한번 더 조절된다.

# RedBot\_0.1

## 최종관

- 로봇 전략 :
  - a.RedBot\_0.1은 RoboLeague를 이용하여, 다양한 Factor와 수치를 변경하면서, 많은 전투를 수행하고, 승률을 검사하여, 최고의 승률을 보일수 있도록 노력하였다. 현재 abc.Shadow\_2.49.1을 상대로 최적화되어있다.
- OneOnOne 전략 :
  - 공격은 GuessFactor를 이용하여 공격을 수행하며, 상대의 총알회피전략을 감안하여 총알이 발사된 Wave인 경우 가중치를 두었다.
  - 이동은 상대가 GuessFactor로 정보를 저장한다고 가정하고, 동일한 정보를 기록하여, 이를 바탕으로 보다 안전한 회전방향을 결정하고, 현재 가장 접근한 Wave의 중심 즉, 총알이 발사된 곳을 중심으로 회전하여 총알을 회피한다.
- Melee 전략 :
  - 공격은 상대의 위치에 계산없이 공격을 하며, 단지, 최대 Power를 결정하는 거리만을 조정하였다.
  - 이동은 다수의 상대에 대한 위험수치를 방향별로 계산하여, 가장 안전한 방향을 결정하고, 이동한다.
  - 이 위험수치를 결정하는 요인으로는 에너지의 차이, 이동위치의 각도 차이, 거리, 상대의 표적 여부, 공격받은 시간등을 이용하여 계산한다.

## XII 허찬

- 이 로봇은 모든 면에서 간단하다.

레이더는 단순히 한 로봇을 주시하는 형식이고 조준 방식도 적의 위치를 따라가서 좁은 범위 안에 무작정 포탄을 발사하는 형태이다.

움직임도 매우 단순하여 적과의 일직선상에서 적당한 각도를 유지하며 가만히 있다가 적의 에너지가 포탄을 발사한 정도로 감소하면 그 때 전후 중에 하나를 선택하여 랜덤한 거리만큼 피하게 했다.

# Chun\_1.0

## 황준식

- Chun1.0의 움직임은 순수하게 랜덤 무브먼트로 이루어져있다. 방향과 거리를 랜덤하게 설정하여 움직인다.
- Chun1.0의 큰 특징은 사격 시스템에 있는데, 패턴인식과 통계 예측사격의 하이브리드된 사격 시스템을 가지고있다.
- 레이더는 모든 로봇의 정보를 최신으로 유지하는 방향으로 움직이도록 되어 있다.