

# CE810 - Game Design 2

## Metrics

---

Joseph Walton-Rivers & Piers Williams

Tuesday, 22 May 2018

University of Essex

- We talked about this last week...

- We talked about this last week...
- Have a look at the evaluation slides

- We talked about this last week...
- Have a look at the evaluation slides
- We also did this in the asteroids code.

# Evaluation

---

- Look at AppMetrics
- Add your parameters using `addParameter`
- Write your fitness function
- get the best and print it out

# What we need

- AI agents (We've done this yesterday)

# What we need

- AI agents (We've done this yesterday)
- Some Parameters (EntityProp, EntityCost, etc...)



# What we need

- AI agents (We've done this yesterday)
- Some Parameters (EntityProp, EntityCost, etc...)
- An evaluation function

# What we need

- AI agents (We've done this yesterday)
- Some Parameters (EntityProp, EntityCost, etc...)
- An evaluation function
- Some maps? (One will do...)

- We did these yesterday.

- We did these yesterday.
- We should load the AIFactory

- We did these yesterday.
- We should load the AIFactory
- Then use the factory to build agents in the evaluate

- We did these yesterday.
- We should load the AIFactory
- Then use the factory to build agents in the evaluate
- `ai.buildAI("ProRuleRushRangedBlue", settings),`

## Some Parameters

- `evo.addParameter(new EntityProp("red_knight", "defRanged", 0, 10, 1));`

## Some Parameters

- `evo.addParameter(new EntityProp("red_knight", "defRanged", 0, 10, 1));`
- A min, max and step size



## Some Parameters

- `evo.addParameter(new EntityProp("red_knight", "defRanged", 0, 10, 1));`
- A min, max and step size
- The entity and property

# Evaluation Function

- Run the games

# Evaluation Function

- Run the games
- Collect some **metrics**

# Evaluation Function

- Run the games
- Collect some **metrics**
- Report the game fitness

# Evaluation Function

- Run the games
- Collect some **metrics**
- Report the game fitness
- (offline) analyse the metrics...

```
public Double evaluate(GameSettings settings) {  
}
```

## Less Basic

```
public Double evaluate(GameSettings settings) {
    GameState start = map.buildState(settings);
    int[] winCounts = new int[2];
    for (int i=0; i<10; i++) {
        Controller[] controllers = new Controller[] {
            ai.buildAI("ProRuleRushRangedBlue", settings),
            ai.buildAI("ProRuleRushRed", settings)
        };
        GameState state = new GameState(start);
        GameMetrics metrics = runGame(state, settings,
↪ controllers);

        Integer winner = metrics.getWinner();
        if (winner != null) {
            winCounts[winner]++;
        }
    }
}
```

Stats

---



- Stats about the games
- Stats about the turns
- You can write files per game played - see the example
- lots (and lots) of files...

# Graphs

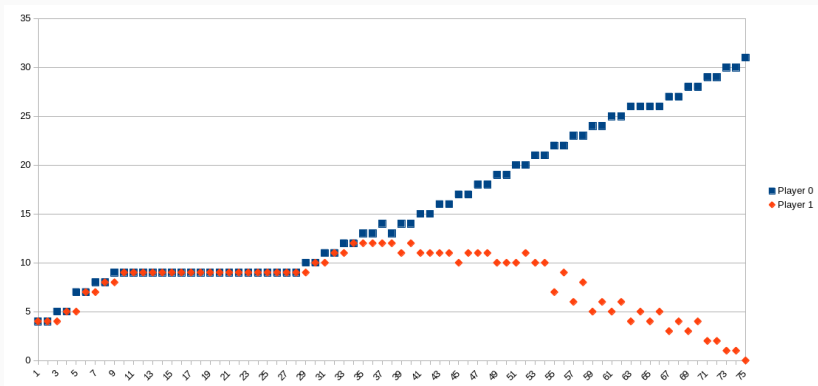


Figure 1: What can we learn from this?

# Results

defRanged	atkRanged	fitness
1	5	30
10	6	20
4	2	20
10	2	20

# Results

defRanged	atkRanged	fitness
1	5	30
10	6	20
4	2	20
10	2	20

So not a very good fitness function then...

# User Testing

---

- Put two versions of the game in front of someone

- Put two versions of the game in front of someone
- Ask them what one they enjoyed most

- Put two versions of the game in front of someone
- Ask them what one they enjoyed most
- Ideally should use a large sample size (or stats)



- Put two versions of the game in front of someone
- Ask them what one they enjoyed most
- Ideally should use a large sample size (or stats)
- But this isn't meant to be rigorous

- In inform out conclusions we could ask other things

# Qualitative Data

- In inform out conclusions we could ask other things
  - Basic demographic data,

- In inform out conclusions we could ask other things
  - Basic demographic data,
  - Experience with games,

- In inform out conclusions we could ask other things
  - Basic demographic data,
  - Experience with games,
  - Open ended questions,

- In inform out conclusions we could ask other things
  - Basic demographic data,
  - Experience with games,
  - Open ended questions,
  - Game Metrics...

# Advice

---

- Fast iterations - don't spend too long waiting for stuff



- Fast iterations - don't spend too long waiting for stuff
- Drop turn limits, repeats, generation counts...

- Fast iterations - don't spend too long waiting for stuff
- Drop turn limits, repeats, generation counts...
- See what kind of data you're getting and adapt your fitness.