HS21 - Introduction to Computer Science

Project Specification

Lux AI – Automated agent

Project Name	Lux AI (Kaggle Competition)
Project Link	https://www.kaggle.com/c/lux-ai-2021
Team	Group 10 - Luca Keiser & Mirco Bazzani

Intro

“The night is dark and full of terrors. Two teams must fight off the darkness, collect resources, and advance through the ages. Daytime finds a desperate rush to gather the resources that can carry you through the impending night whilst growing your city. Plan and expand carefully -- any city that fails to produce enough light will be consumed by darkness.”

“The Lux AI Challenge is a competition where competitors design agents to tackle a multi-variable optimization, resource gathering, and allocation problem in a 1v1 scenario against other competitors. In addition to optimization, successful agents must be capable of analyzing their opponents and developing appropriate policies to get the upper hand.”

Lux AI is a Kaggle-competition based on a strategy game where teams need to code an automated agent that plays the game.

Goals

Initiation and SetUp

- Create an environment that is able to run and play a basic game of Lux AI.
- Set up a GitHub-Repo to make it reproducible
  => Is there a GitHub-integration for Python, similar to RStudio?
- Understand the basic code provided by the Competition to be able to optimize it.
- Define KPIs
  => I think there are 2 possibilities here:
  - 1) we define a desired game result (e.g. that we have survived X CityTiles, Y Workers and Z Units after finishing a game).

  - 2) we define a desired ranking in the Kaggle ranking (better than half of the participants?)

Strategy

- Formulate a strategy on how we want to improve the basic agent.
  - Part ML (Gradient descent?)
  - Part Hard coded
- Write first pseudo-code

Optimization

- Write the first optimized version of the agent
  - Measure how it performs
  - Where can it be further optimized?
  - Should the approach be scrapped?
- Try out multiple approaches

Users

Identify the Users

Our project is not stakeholder-oriented and is not going to be used by third parties.

Resources

Identify Related Work

- what open source projects and software are there, that you can use or that solve the same (sub-)problem(s)
Information is provided via:
- Official Kaggle-Webpage
- GitHub
- Official Discord-Server of the project

- what are their strengths and weaknesses?
The materials provided are a good starting point. However, there is still a lot of work left.

- What is your contribution?

As it is an open source competition we want to share our codebase / findings with the community

Specs and Scope

- Specify Functional Specifications

    - What functionality is your software going to offer? (in-scope)

    We develop an automated agent for a specific task (a game in our case). The automated agent should act independently and complete a full game.

    - What functionality is your software not going to offer? (out-of-scope)

    Even though it is an AI-Challange the biggest chunk of our code will be hard coded (with some hyper parameter tuning) and not rely on deep learning or more complex applications of neural networks.

- Specify Technical Specs

    - Where is your program going to run?

    Local, not server based.

    - How are you going to distribute your program?

    Via GitHub (it is open source)

    - What programming language(s) are you going to use?

    Python and some JS (in the background)

    - Are there any browser or operating system compatibility issues that you could face?

    We will find out soon...

    - Expected performance (loading times, general speed, GPU/CPU, ...)

    The basic agent is going to work smoothly, problems could arise when we use ML to finetune the agent.

    - Do you have any other constraints that could influence your design or solution?

    Lucas laptop may explode

Timeline

- Planning

    - Create a roadmap for your key results

    - 21.11.2021 - 28.11.2021: Initial setup (install challenge locally, including GitHub); lecture of some strategies competitors use.
    - 28.11.2021 - 12.12.2021: Optimize the agent by just using hand coded strategies – no ML included.
    - 12.12.2021 - 19.12.2021: Determine where ML would make sense. Determine what algorithms we want to use.
    - 19.12.2021 - 09.01.2021: Optimize the Agent, swap Models.
    - 02.01.2021 - 09.01.2021: Identify the best agent, optimize further.

    - Try to split the key results into concrete work packages which you can distribute within your team of 2

    For the first phase (21.11.2021 - 12.12.2021):
    We want to try a consensus-approach with one single master-branch that we fork and optimise individually. In recurring meetings we then decide what gets merged into the new master branch and is then used as the updated status quo.

    For the second phase (12.12.2021 - 09.01.2021)
    The ML-process is divided up. One party is training the model and then reporting to the second team member. We then swap positions to reduce waiting times while training the model.
    At the end we identify the best result and select it for our automated agent.