Alexandru-Ilie Știrbu

(+40) 751-973-748 | stirbu.alexandru.net@outlook.com | Akrielz | alexandru-știrbu-748068177

Education

University "Alexandru Ioan Cuza"

Iasi, Romania

Bachelor Degree in Computer Science

Expected graduation date:Summer 2021

- **Faculty of Computer Science**, Iasi, Romania — *Started in Fall 2018*

University "Alexandru Ioan Cuza"

Iasi, Romania

Master Degree in research on artificial Intelligence

Expected graduation date:Summer 2023

- **Faculty of Computer Science**, Iasi, Romania — *Expected start in Fall 2021*

Work

Bentley Systems

Iasi, Romania

Software Engineer I

Started in: March 2021

- My **main responsibility** is **developing efficient algorithms** for diverse **3D geometry tasks** related to the new product OpenTunnel Modeler.
- Implemented in **C++** and **C#** core **algorithms** for: creating solid tunnels, dividing solid tunnels and generating diverse 3D solids required for the tunnels.
- Designed a **C++ API** that allows any user to easily **create** / **get** / **edit** / **delete** any specific **item** from any **internal database**. This was latter used in the product to work with diverse items that resemble relevant parts of the designed tunnel.

Projects

StarCraft II - Zerg AI (Winter 2020)

- A **StarCraft II Zerg AI** in **Python** (made with **Burny's SC2 API**) that is able to constantly **defeat the official Elite AIs** on every class, and occasionally to **defeat even the official Cheater Elite AIs** as well. The **AI chooses optimally** what structures, upgrades and units to **build based on the enemy's strategy** and based on the given environment. Every **unit controlled by the AI is played optimally**, using all the unit's abilities and making decisions of moving and attacking such that every units lives as much as possible and deals as much damage as possible. **Every fight that is about to be taken is pre-evaluated**, and the AI decides to retreat if it is not possible to win the fight, or engage if it considers that it has a significant advantage.

Water World - Deep Q-Network (Winter 2020)

- Water World is a **game** where all entities are molecules. The player must eat green molecules and avoid the red ones. This project resembles a **Deep Q-Network AI** made in **Python** that managed to learn how to play the **game**, knowing only the current score and the state of his own vision sensors. The **game** was re-created in **Python** using **PyGame**, and the **Neural Network** was created to receive as input the state of the sensors from the current frame, and it gives back as output a direction towards to move for that frame. The **learning process** is based on a internal **buffer** that the AI holds, from where it extracts the valuable information for a specific frame, and the outcome of the next frame, trying to improve previously made choice.

VR Card Game (Winter 2020)

- **A online VR Mobile Card Game** made in **C#** using **Unity**, that combines main mechanics from games like Hearthstone, Chess and Heroes of Might & Magic. The game uses an **AI** that takes frames from the phone's **video camera** uses them in order to **determine the position and gesture of the hands**, which is used for the user's input to the game.The **game** is a **1v1 turn-based card game**, where the action takes place on a 6x6 board, and the players have to manage their resources in order to use cards that either summon creatures or casts spells. In the game, there are 4 elements that each player can choose, each element featuring 4 types of minions and 5 types of spells. The player that manages to destroy the enemy's base is victorious. The **online connection** was made with **Photon**.

!FII Bolnav (Summer 2020)

- The main application is a website that is meant to offer guidance regarding medical attention. I worked on a **submodule** that splits a given image into patches of 50x50, feeding it to a **Neural Network** with 2 functionalities: detecting whether the breast cancer is benign or malignant and detecting brain tumors, written in **Python**. This submodule hosts a **server**, offering access for an **API** for the neural networks. The submodule is inside a **docker image** to solve the compatibility problems for debugging / hosting.

Universal Storage Tool (Summer 2020)

- A **website** that allows any user to sign up/sign in, and link their existent **accounts** of **DropBox**, **Google Drive** and **Microsoft OneDrive** in this platform, allowing to **upload/download files** through the **web application**, according to what settings the user has. The application is modeled on the **Model-View-Controller** paradigm, having the **back end** written in **NodeJS**, and the **front end** in **HTML**, **CSS** and **JavaScript**, using **MongoDB** for **databases**, and the **Google Drive / Microsoft OneDrive / DropBox APIs** for any required operations. The application also offers an **API** for normal users in order to use its services, and an **API** that requires credentials of the **admin accounts**, to easily have control over the application's data and states.

Mailing System (Winter 2019)

- A networking project written from scratch that implements a **Mailing System**. The **server** side is written in **C/C++**, while The client side has a **visual interface** made in **QT** with all the functionalities written in **C/C++**, that allows the user to sign up/sign in, create new mails, send them to multiple subjects, CC and BCC, forward them, visualize them, and check for new mails. The project comes with a **hand-made architecture for a database** for mails, each mail having an id, a subject, destinations, an owner, plain-text, HTML-text, emojis, attached files, and a category.

Romanian Pseudo-code Interpreter (Winter 2019)

- An **interpreter** written in **YACC**, **LEX** and **C/C++**, that analyzes **syntactically** and **semantically** the given code. The **interpreter** will signal any syntactical error, specifying the line and what was expected instead. The syntax allows: basic input-output (read, print), basic control structures (if, for, while, repeat until), strong type variable declarations(int, float, bool, string) with scope declaration, basic operations, arrays, functions with parameters, classes and objects. The semantics allows evaluation of expressions, receiving input from keyboard, printing expressions and mapping of all the variables with values.

Skills

Languages C/C++, Python, Java, C#, SQLPlus, AutoIt, JavaScript, NodeJS, Assembly

Knowledge Artificial Intelligence, Machine Learning, Artificial Neural Networks, Math, Genetic Algorithms, Data Structures, Graph Theory, Object Oriented Programming, Algorithms, Game Tactics, Compilers, Computer Networking, Computer Graphics, Databases, Computer Architecture, Probabilities

Academic Awards

MO-Ariel - International Mathematical Olympiad Championship (Individual competition)

December 2020

8TH PLACE GLOBALLY, GOLD AWARD

MO-Ariel - International Mathematical Olympiad Blitz (Individual competition)

December 2020

5TH PLACE GLOBALLY, GOLD AWARD

Cloudflight Coding Contest

October 2020

2ND PLACE IN IAȘI, 42ND GLOBALLY (TOP 2.95%)

HashCode - International Google's Coding Competitions

February 2020

1ST PLACE IN IAȘI, 13TH IN ROMANIA, 265TH GLOBALLY (TOP 2.47%)

MO-Ariel - International Mathematical Olympiad Championship (Team competition)

December 2019

1ST PLACE GLOBALLY, GOLD AWARD

MO-Ariel - International Mathematical Olympiad Blitz (Team competition)

November 2019

1ST PLACE GLOBALLY, GOLD AWARD

Personal Achievements

Chess.com

2132 RAPID RATING (TOP 0.1%), 2044 BLITZ RATING(TOP 1.1%)

StarCraft II

MASTER AT 2V2 (TOP 4%), DIAMOND II AT 1V1 (TOP 13%)

League of Legends

GRANDMASTER SOLO (TOP 0.057%), GRANDMASTER FLEX (TOP 0.057%), CHALLENGER 3V3 (TOP 0.017%)