

Object Detection and Behavior Optimization of Agents in a Computer Game

Bc. Roland Marek Rak
Supervisor: doc. RNDr. Ľubomír Antoni, PhD.
Consultant: Ing. Matúš Semančík



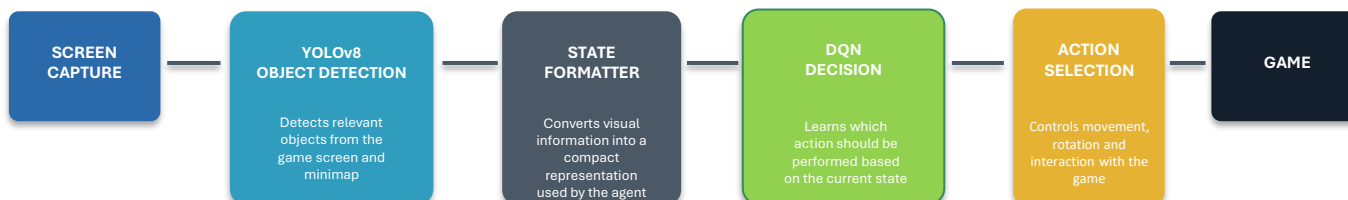
PROBLEM

- Resources appear at predefined spawn points, but only a limited number are active at a time.
- The agent cannot directly access the game's internal state or current resource positions.
- Traditional scripted automation relies on static routes and can get stuck in unexpected situations.

Goal

- Develop an autonomous game agent that can perceive the environment, navigate to resources and collect them without direct access to the game's internal state.

PROPOSED SOLUTION



Global view (minimap) → makro-navigation (Broad search) | Local view (3D screen) → mikro-navigation (Specific localization)

1 - Object detection



YOLOv8 model

- A fast one-stage object detector used to transform raw game-screen pixels into meaningful information about the environment.
- An almost instantaneous response is essential for smooth control of the in-game agent.

Object classes

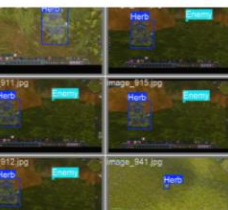
- Herb
- Ore
- Enemy
- Neutral
- Node

Dataset

- 600+ manually annotated images

Data augmentation (600+ → 2000+)

- Geometric shifts
- Brightness changes
- Gaussian noise
- Blur



1.5 - Dual visual input

Global View – Minimap

- Wide-area perception
- Resource detection
- Macro-navigation



Local View – 3D Screen

- Precise positioning
- Threat detection
- Resource interaction



2 - Reinforcement learning

DQN Agent

- Instead of processing millions of raw pixels, the agent receives a compact state representation containing 11 important numerical values together with visual information.

State includes

- Relative position of the nearest resource
- Distance to target
- Detected enemies and obstacles
- Visual information from minimap and 3D view

The reward function encourages:

- Moving towards resources
- Successful resource collection
- Avoiding threats
- Reducing unnecessary movement and rotation

Wandering → Hunting → Collecting → Reset → Wandering

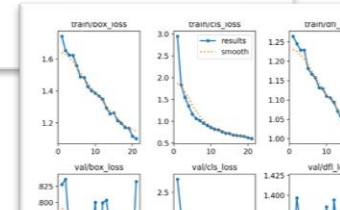
3 - Results

Object Detection

- Relevant game objects successfully detected
- Herb, Ore, Enemy, Neutral and Node classified
- Bounding boxes aligned well with ground truth
- Detection remained effective under partial occlusion

Reinforcement Learning

- Episodic reward became consistently positive
- Agent learned to navigate towards resource nodes
- Unnecessary rotations were reduced
- Successful resource collection was achieved
- Episodes regularly reached the 500-step limit



CONCLUSION : The proposed system combines computer vision and reinforcement learning to create an autonomous game agent capable of:
Perceiving → Deciding → Navigating → Collecting

The modular architecture separates visual perception (YOLOv8) from decision-making (DQN), allowing the individual components to be adapted independently to new environments