



Nathan Versavel
Erwann Duvillard
Enzo Silvestre

SUBJECT OF FREE CHOICE
SOLDADOSS
AI in Aviation Engineering



Overview

- OUR TEAM
- INTRODUCTION
- WEBSITE
- CODE
- CONCLUSION

01

02

03

04

05



OUR TEAM



Nathan

Project Manager



Erwann

Technical Developer



Enzo

Data Analyst

OUR ROLES



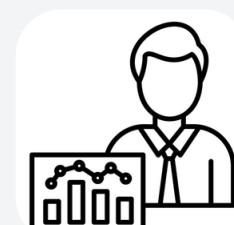
Project Manager

plans tasks, manages deadlines, coordinates the team, writes the introduction and organises the portfolio/README



Technical Developer

writes code, builds models, runs experiments, explains the code and its purpose



Data analyst

collects, cleans, analyses data, creates graphs, writes the conclusion and organises the portfolio/README



Introduction

Task: create a system that uses AI
in our code and that answers to a
current problematic



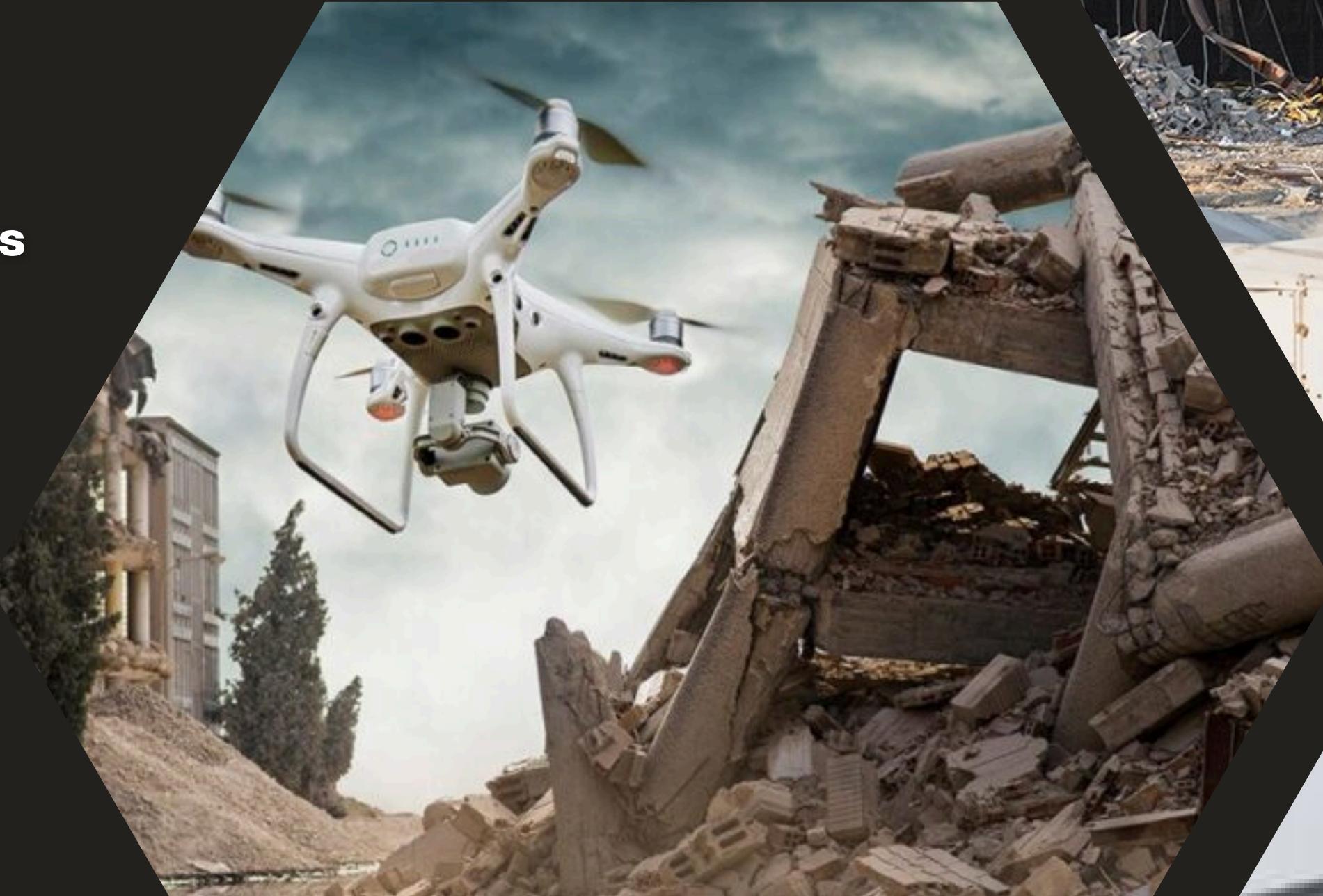
Our Project

SOLDADOSS:

A drone that will save lives using AI in order to access complicated places.

Places where we would use this drone:

- natural disasters
(Tsunamis, Earthquakes, Floods,...)
- human disasters
(Fires, Wars/armed conflicts,...)



WEBSITE



Window)



Hyperparameters & Algorithm

Learning rate α : 0.8

Discount γ : 0.95

Greedy ϵ (Exploration): 0.25

Epsilon decay per episode: 0.995

Environment & Execution

Nb. Episodes: 1000

Speed (ms / step): 40

Rows10: 10

Cols10: 10

Start | Pause | Reset

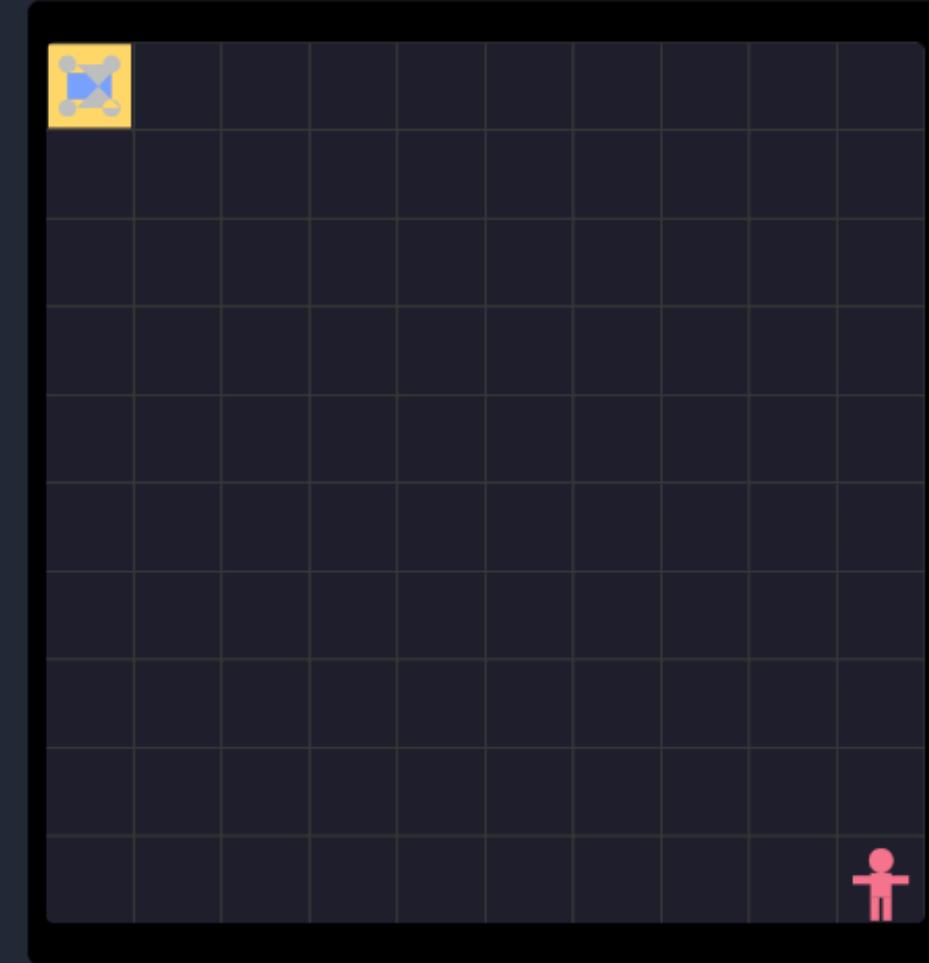
Load Example | Clear Obstacles

Export Map | Import Map | Choisir un fichier Aucun f

Show Q-Heatmap

Show Policy Arrows

Click: Toggle obstacle.
Ctrl/Cmd + click: Set Start. Shift + click: Set Goal.



Episode: 0 Step: 0 Reward: 0 TD Error: 0.0000 ϵ : 0.250

Watch the agent explore and optimize its path. Adjust ϵ , α , γ parameters in real time!



Code

```
426     Q = Array.from({length:r}, ()=>Array.from({length:c}, ()=>[0,0,0,0]));
```

- Agent uses a Q-table to store what it learns.
- To choose an action, the agent uses an epsilon-greedy strategy.

```
797 // Q-Learning helpers
798 function chooseAction(r,c, eps){
799   if(Math.random() < eps) return Math.floor(Math.random()*4);
800   const q = Q[r][c];
801   let best = -Infinity; let bestActs = [];
802   for(let a=0;a<4;a++){
803     if(q[a] > best){ best = q[a]; bestActs=[a]; }
804     else if(q[a] === best) bestActs.push(a);
805   }
806   return bestActs[Math.floor(Math.random()*bestActs.length)];
807 }
```

```
873
874 // Q-Learning Update
875 const tdError = res.reward + gamma*bestNext - Q[prevR][prevC][a];
876 Q[prevR][prevC][a] += alpha * tdError;
877
```

- After each action, the agent updates its Q-value.

Conclusion

- AI improves over time → faster, safer
- Adapts to many or single victims
- Cost-effective: cheaper than helicopters
- Life-saving with thermal/infrared sensors





Recommendations

- Autonomous navigation using Reinforcement Learning
- Better sensors: infrared, thermal, ultrasonic
- Physical prototype for real-life tests
- GPS-free communication
- Advanced vision: movement, gestures, vital signs

Reflection & Impact

- AI has real-world potential and limits
- Reinforcement Learning makes drones safer and faster
- SOLDADOSS could become a life-saving tool



SOLDADOSS

<https://github.com/erwann74800-cmd/SOLDADOSS/tree/main>