



TABLE OF CONTENT

APPROVAL PAGE	2
DECLARATION	3
LIST OF TABLES	7
LIST OF FIGURES	8
LIST OF CODE	9
ABSTRACT	10
INTISARI	11
INTRODUCTION	1
1.1. Research Background.....	1
1.2. Research Problem.....	3
1.3. Research Scope.....	3
1.4. Research Objective.....	4
1.5. Benefits of the Study.....	4
LITERATURE REVIEW	6
THEORETICAL BASIS	9
3.1. Video Games and Game Maps.....	9
3.2. Video Game Testing.....	9
3.3. Reinforcement Learning.....	10
3.4. Markov Decision Process (MDP).....	12
3.5. Soft Actor-Critic (SAC) Algorithm.....	13
METHODOLOGY	17
4.1. Map Creation.....	17
4.2. Environment Initialization.....	18
4.2.1. Gymnasium Interface.....	18
4.2.2. Observation Space and Action Space.....	19
4.2.3. Game Mechanics: Agent Movement.....	20
4.3. Reward Function and Termination Conditions.....	21
4.4. Custom Callbacks.....	21
4.5. Training and Policy Update.....	22
4.6. Evaluation Scenario.....	26
IMPLEMENTATION	28
5.1. Installing and importing libraries.....	28
5.2. Pygame integration and Game State initialization.....	28
5.3. Map Creations.....	30



5.4. RL Environment Creation.....	32
5.4.1. Observation space.....	32
5.4.2. Local Map.....	32
5.4.3. Action Space.....	33
5.5. Gymnasium Interface.....	34
5.5.1. step() method.....	34
5.5.2. reset() method.....	36
5.5.3. render() method.....	37
5.5.4. close() method.....	38
5.6. Game Mechanics.....	38
5.6.1. Move method.....	39
5.6.2. Check collision.....	40
5.7. Reward Function and Termination Conditions.....	41
5.7.1. Exploration Reward.....	41
5.7.2. Penalties.....	42
5.7.3. Termination Conditions.....	43
5.8. Stable Baselines 3 and Callback Classes.....	43
5.8.1. Heatmap Logger.....	43
5.8.2. Episode Reward Logger.....	45
5.9. Training using SAC.....	46
5.10. Testing on a Test Map Layout.....	47
5.11. Random movement implementation.....	48
RESULTS & ANALYSIS.....	50
6.1. Exploit Detection Through Heatmap.....	50
6.2. Game States Visited.....	53
6.3. Training Performance Analysis.....	54
CONCLUSION.....	57
REFERENCES.....	59



LIST OF TABLES

Table 2.1 Research comparison.....	8
Table 4.1 Evaluation scenario of model performance.....	27
Table 6.1 Amount of unique grids visited per episode.....	53
Table 6.2 Training process development over time.....	55



LIST OF FIGURES

Figure 3.1 Flowchart of RL System Processes.....	11
Figure 3.2 Reinforcement Learning loop.....	13
Figure 4.1 Research methodology.....	17
Figure 4.2 Map creation in Tiled.....	18
Figure 4.3 Rendered Pygame environment.....	20
Figure 4.4 Reward function structure.....	21
Figure 4.5 Example of 3x3x2 local map observation.....	23
Figure 4.6 Example reward for each grid in the local map range.....	24
Figure 5.1 Training map.....	30
Figure 5.2 Testing map.....	31
Figure 6.1 Test map with exploits.....	50
Figure 6.2 Agent exploration heatmap episode 2.....	51
Figure 6.3 Agent exploration heatmap episode 4.....	51
Figure 6.4 Random exploration heatmap episode 2.....	52
Figure 6.5 Random exploration heatmap episode 4.....	52
Figure 6.6 Reward per Episode graph of the training process.....	56



LIST OF CODE

Code 5.1 Library and package installations.....	28
Code 5.2 Initialization of GameTesting class parameters.....	29
Code 5.3 Sprite group initialization with Pygame.....	30
Code 5.5 Observation space dictionary.....	32
Code 5.6 Local map implementation.....	33
Code 5.7 Action space.....	34
Code 5.8 Step method implementation.....	35
Code 5.9 Step method dictionary and info updates.....	36
Code 5.10 Reset method implementation.....	36
Code 5.11 Reset method dictionary and info updates.....	37
Code 5.12 Render method implementation.....	38
Code 5.13 Player class initialization.....	39
Code 5.14 Move method of the player class.....	40
Code 5.15 Method to check player collision in player class.....	41
Code 5.16 Positive reward calculation.....	42
Code 5.17 Negative reward calculation.....	43
Code 5.18 Heatmap Logger callback.....	44
Code 5.19 Log heatmap method.....	45
Code 5.20 Episode Reward Logger callback.....	46
Code 5.21 Training model with SAC.....	47
Code 5.22 Testing model on test map.....	48
Code 5.23 Random movement implementation.....	49