

INDIVIDUAL PROJECT FOR

ARTIFICIAL INTELLIGENCE FOR VIDEO GAMES

made by: Federico Porteri
mat: 945812

INDEX

● PREMISE	3
● PLAYER ACTIONS	4
● THE CREATION OF THE MAP	5
● KILLER BEHAVIOR	7
● KILLER STATES	8
● A*	9
● GAME'S AGENTS	10

PREMISE:

Unity version: 2018.4.9f1

The project is a demo of a survival turn based single player game, with a procedural generated map.

After loading and generating the play-board, the player will roam a grid-like environment, where she/he will be able to move to adjacent tiles by clicking on them with the left button of the mouse. For every step taken by the player, the “killer” takes a step too, following its own path.

Around the map the player will encounter lockers and broken engines. By standing in the same spot of one of the two types of props, the player can interact with them, by clicking on them with the left mouse button.


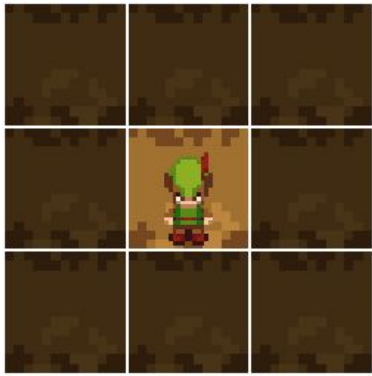


By clicking on a locker, the player will enter a hidden state that will allow her/him to let the killer pass by without being noticed. Clicking on the locker a second time will make the player exit the locker and end the hidden state.

By clicking on an engine, a minigame will commence. The player has to enter in a text box the same code that the new window will display in maximum 7 seconds. When the player has successfully done 5 minigames on the same engine (not necessarily in a streak) the engine will be considered done. Once all 3 engines are done, the player wins.

By clicking on an empty tile, the player will simply pass the turn to the killer.

If it happens that the player gets caught two times by the killer, the player loses.

PLAYER ACTIONS:

	MOVE	Moves the player in an adjacent square.
	UNCANNY DODGE	For 5 turns, the player can move diagonally 1 square. The player starts the game with 1 activation and can refresh it every time she/he completes an engine. At any time, the player can have a maximum of 1 activation stored.
	UNCANNY DODGE BUTTON	Activates the uncanny dodge.
	KILLER CAMERA BUTTON	Moves the camera to the killer location. In a game, the player can do so a maximum of 3 times.

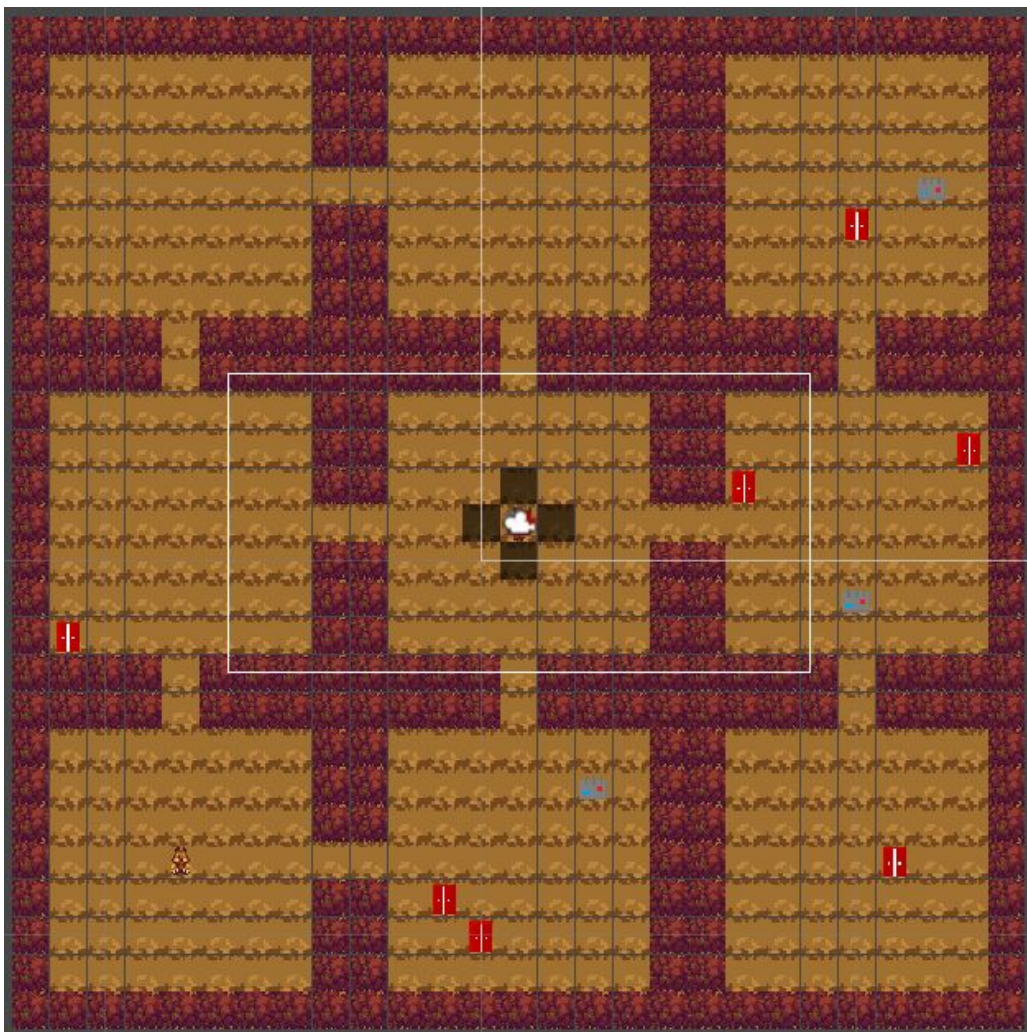
THE CREATION OF THE MAP:

At the start of the game, the player has to pick the settings for the creation of the playing board that dictate:

- the size of the map
- the level of saturation of the map (how many unpassable squares)

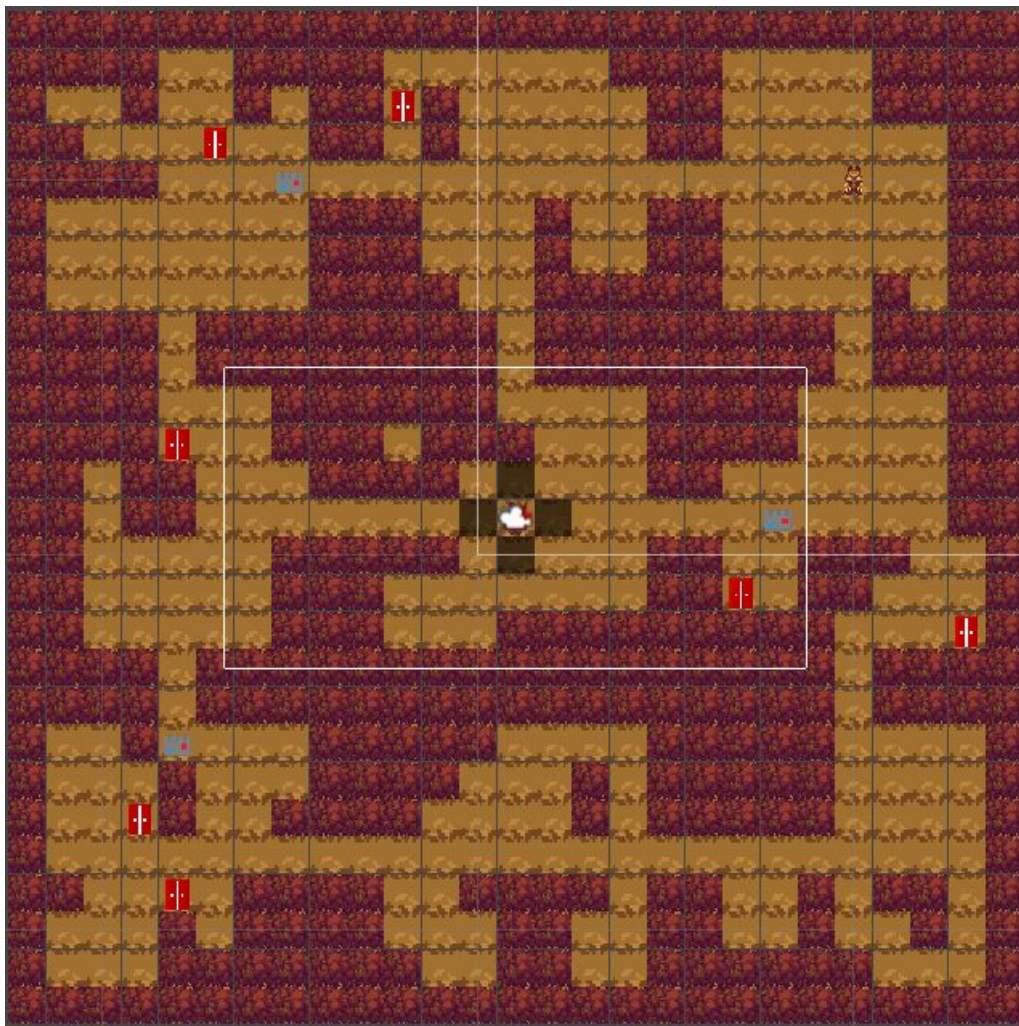
Once the player has selected both statistics, the map will be created, following this rules:

- 9 rooms, in a 3x3 matrix
- the central room is the player spawning room
- the kill will start in one of the four corner rooms
- the rooms are connected via 2x1 squares corridors
- the central room has 3 or 4 corridors, connecting to random (or all) adjacent rooms
- the central-adjacent rooms have at least one connection to their respective adjacent corner rooms
- if the player can't reach all 9 rooms by moving from the central room, the corridors layout is discarded and redone



example of a “empty” map

- the rooms are filled with impassable walls, following the saturation chosen by the player
- little gaps between walls are filled and isolated once are deleted, following the Conway game of life
- 7 lockers are placed around the map
- one engine is placed in one corner room
- at least 1 locker will be present in every engine room
- another is placed in one room adjacent to the opposite corner-room of the last one
- the last engine is placed randomly in a non-central room that doesn't already contains an engine
- to control that every room is accessible and all connected rooms are accessible, an a* algorithm runs from every corridor to every other corridor, from the engine (if present) to every corridor and (if only one corridor is present) from the central square to the only corridor
- the same a* runs from the central square of the central room (player spawning point) to a random corridor
- the same is done for the killer
- if one a* fails in a room, the room is cleared from all walls, and repopulated. The a* runs again until it doesn't fail
- the a* algorithm uses a Manhattan heuristic



example of a “dense” map

KILLER BEHAVIOR:

The agent of the killer is set to roam the map in search of the player.

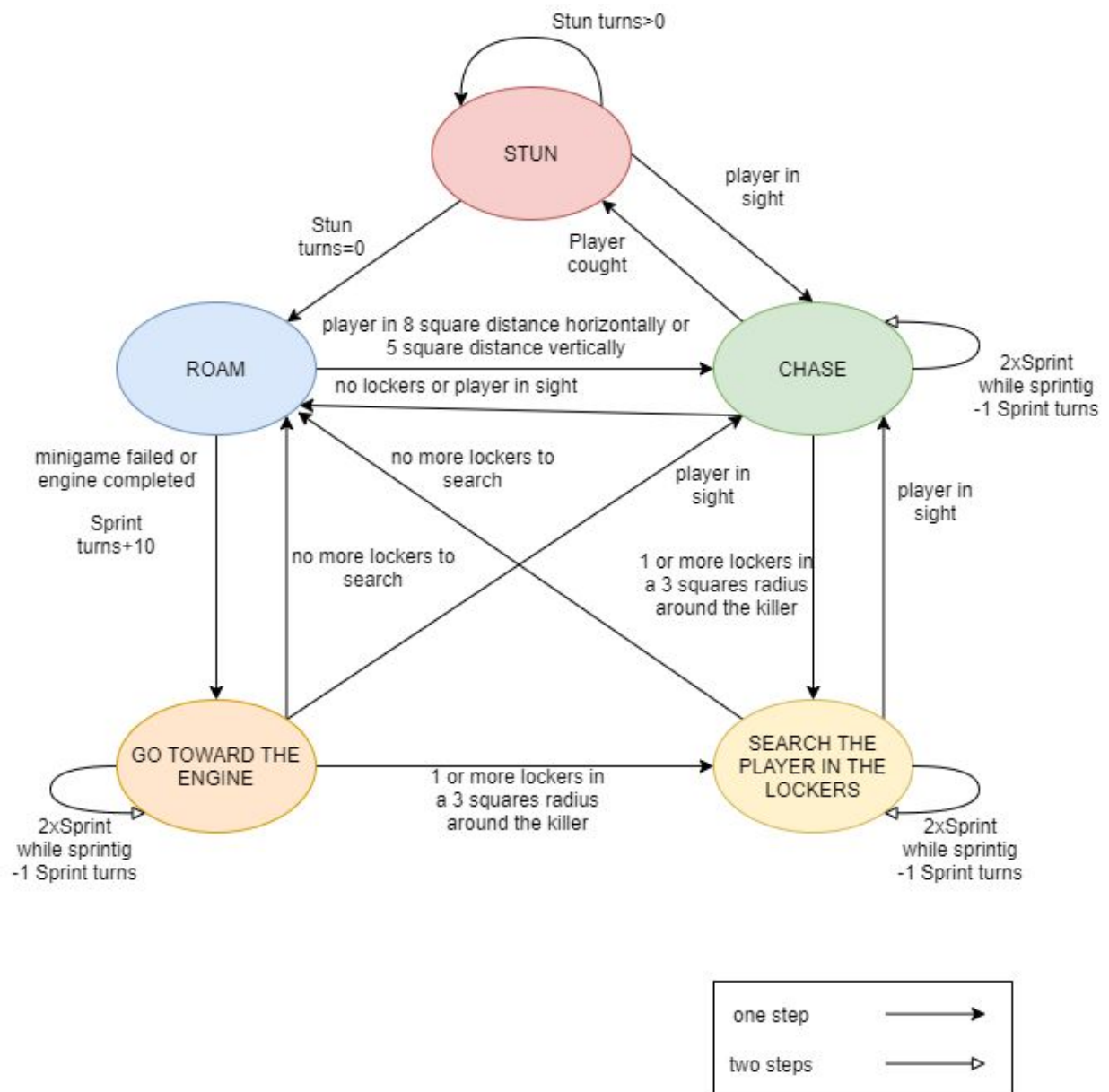
Starting from its spawning point, the killer will move to one adjacent room that it didn't already visit, if there are none, it will visit one unvisited room at random, following the shortest path from one room to another. Once it has no more unvisited rooms, it will repeat the process.

Before taking any steps though, the killer will control if the player is in sight (8 squares horizontally and 5 square vertically). If not, the killer will continue as usual, otherwise it will enter the "chase state", leaving any task behind, focusing only on the player until she/he gets out of sight or the killer reaches the player. The killer, not like the player, can move diagonally. While chasing the player, the killer can destroy any lockers that he comes across, rendering it unusable until the player repairs it, by standing on it and clicking on it 5 times. If the killer sees the player hiding in a locker, it will follow until the player comes out and resumes the chase or the killer comes in contact with the hiding locker, revealing and hurting the player and breaking the locker. If the player runs out of sight, the killer will always move to the last known player position, checks if there are any nearby lockers where the player can be hiding (3 square radius from the last known player location), breaks all the lockers and lastly resumes the roam.

If the killer reaches the player, it will enter the "stunned state", where the killer must wait 5 turn, letting the player hide again. While stunned the killer can't hurt the player, move or see the player hiding.

If the player fails a minigame or completes an engine, the killer will enter the "sprint state" and will start to run towards the engine where the player was working. The killer in this state will move twice as fast for 10 turns. When it arrives at the engine, the killer will start to search for the player in the nearby lockers (if there are any), breaking all of them. Once there are no more lockers to destroy, the killer will resume the roam.

KILLER STATES:



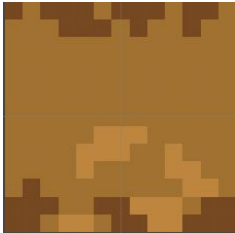
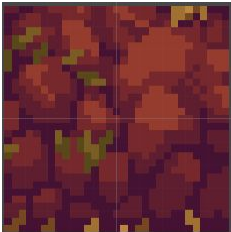
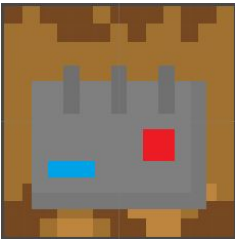
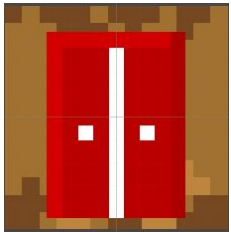


A*:

The A* algorithm follows a Manhattan heuristic, with every square of walkable terrain representing one node of the graph and every side that faces another walkable terrain as an edge (for the killer pathfinding the corners are considered edges too, as long as the corner doesn't touch a wall). The graph is not weighted and it's undirected. The graph is also divided in clusters depending on the situation:

- for calculating the solvability of the map, every room is clustered since there's no need to calculate every time a clear path from the player starting position to the target
- for the movement of the killer the hole board is considered

GAME'S AGENTS:

Player	
Killer	
Walkable surface	
Wall	
Engine	
Locker	

player, killer, wall and walkable surface -> asset: Tiny RPG - Forest (Ansimuz)