/---------o---------\

Basic Unity **Tactical** Tool

Save a lot of time to **make your own tactical** game!
Create a tile board with holes and obstacles, easily **customize your tile board**, **add characters** on, create **teams**, **NPCs**, **add behaviors** and **let's fight**!

Enjoy your **Basic Unity Tactical Tool!**

***NB**: I advise you to read the first two categories: "Feature list" explains what exactly is in the project, and "How to use it in Unity?" is a tutorial of these different features. Everything is done to be the most simple to understand and use.*

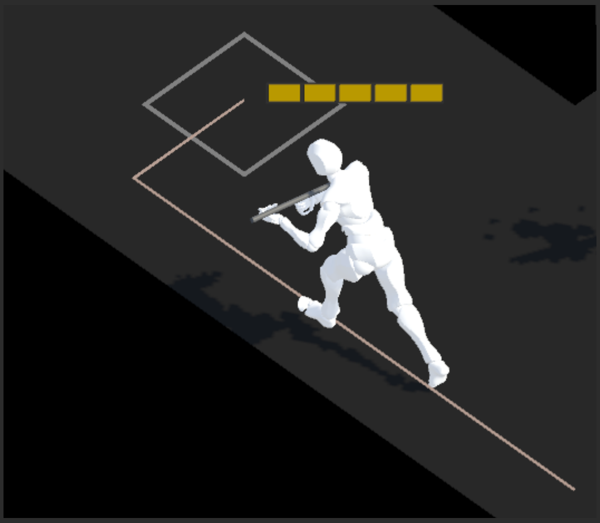
SUMMARY

\---------o---------/

/---------o---------\

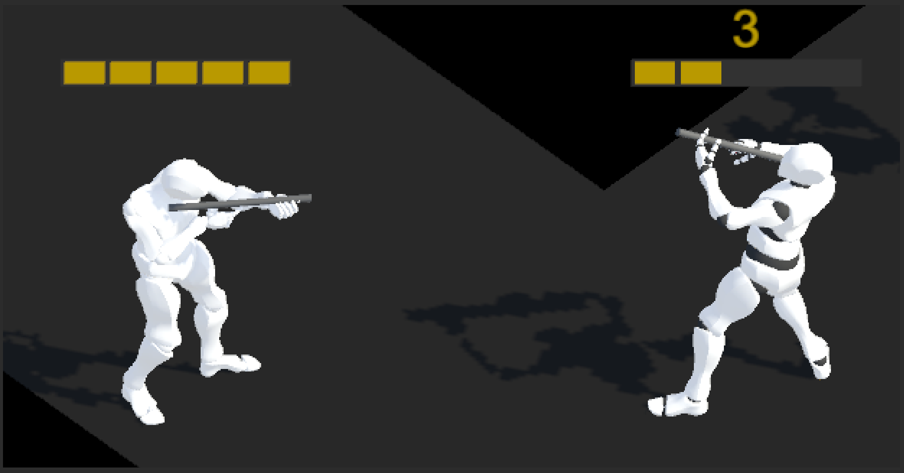
- FEATURE LIST

Movement along the grid



Pathfinding on 2D grid (A* algorithm) using diagonals or not, through other characters or not, with holes and obstacles.
Movement feedback with character animation, movement line, allowed/forbidden tiles.

Ranged attacks

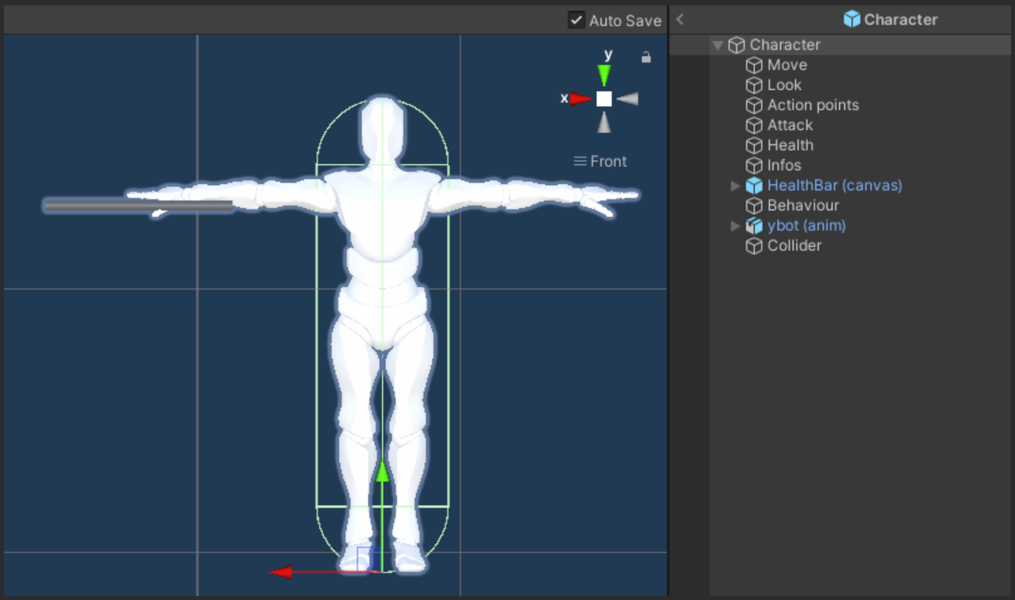


Line of sight with or without obstacles.
Shooting range.
Damage range to set damages between two values.
Percent of chance to touch the enemy.

\---------o---------/

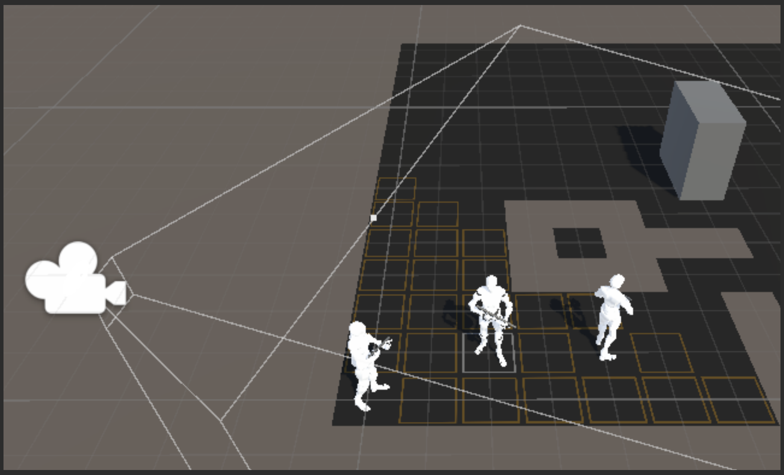
/---------o---------\

Characters



Action points, usable and tweakable for movement and attack
Heath points with hurt and death animations.
Teams and team color system.
AI Behavior with some basic behaviors (follow, attack, attack once).

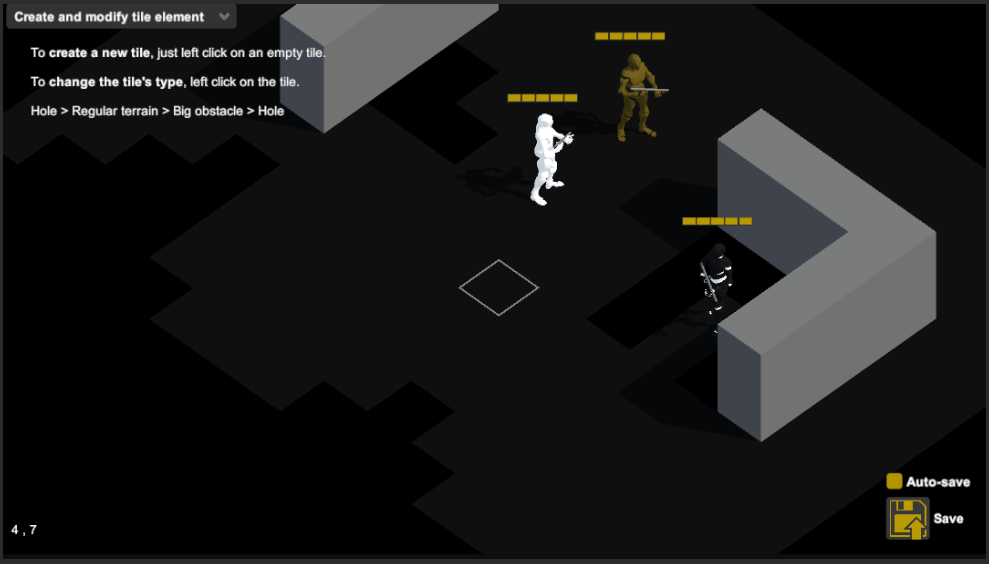
Camera



Mouse movement on the borders of the screen.
Center position on current character.

\---------o---------/

Board creation



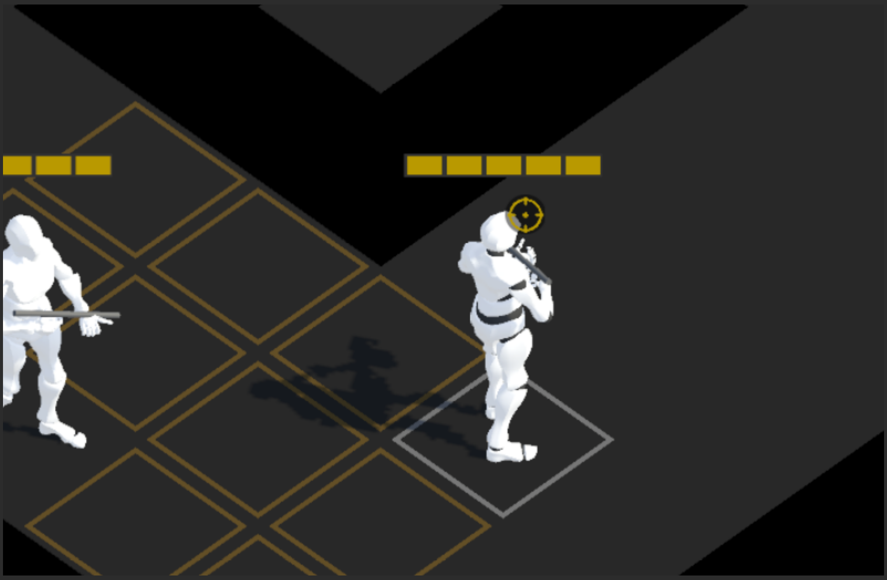
To **create a new tile**, just left click on an empty tile.

To **change the tile's type**, left click on the tile.

Hole > Regular terrain > Big obstacle > Hole

Create your tile boards and save them.
Integrated tutorials are here to help you.

Others

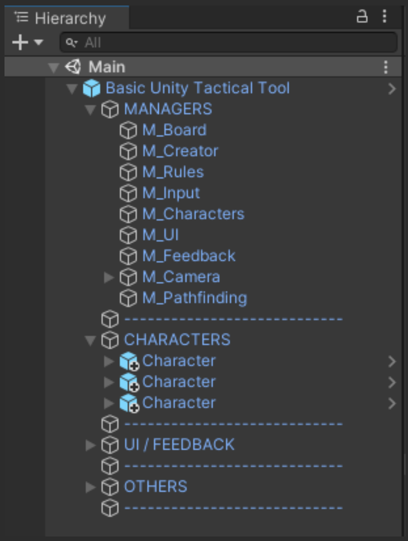


Custom mouse cursors, easy to modify and adapt to your game.
Fog of war showing your range view.
Victory screen for a fancy reward!

- HOW TO USE IT IN UNITY?

The project is based around :

- **Big managers**, each one managing a part of the game.
- **Characters**, each one managing itself.



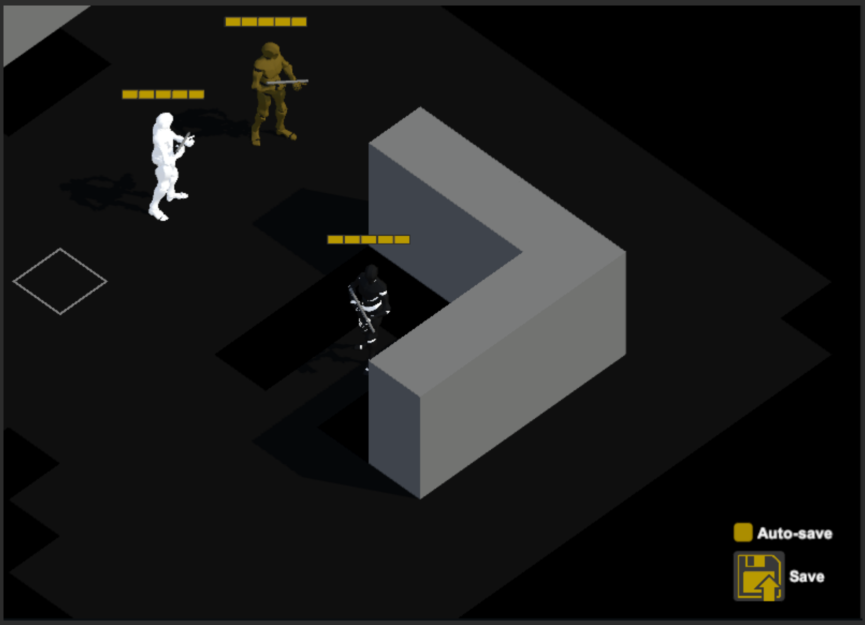
```
Main
  Basic Unity Tactical Tool
    MANAGERS
      M_Board
      M_Creator
      M_Rules
      M_Input
      M_Characters
      M_UI
      M_Feedback
      M_Camera
      M_Pathfinding
    ----------------------------
    CHARACTERS
      Character
      Character
      Character
    ----------------------------
    UI / FEEDBACK
    ----------------------------
    OTHERS
    ----------------------------
```

If you want to **create a new scene**, that's quite simple:

- **Drag and drop the Basic Unity Tactical Tool** prefab.
- **Drag and drop characters** and parameter them.
- Customize your **game rules**.
- Start to play.

How to customize your tile board and characters is explained under.

Tile board



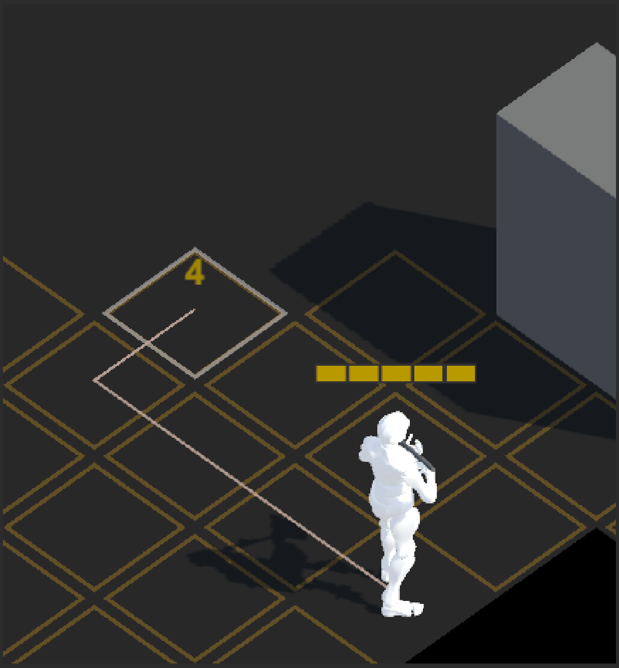
I want to **create a tile board** and customize it.

- Simply **check Edit mode** in M_Creator (in hierarchy).
- Click on Unity's Play mode.
- Everything you want to do is **explained in tutorials** (in play mode mode). They are **quite simple**.
- *NB : Character management in play mode is not included yet, how to parameter them is explained in the Characters section (just under).*

I want **to play**.

- Simply **uncheck Edit mode** in M_Creator (in hierarchy).
- Click on Unity's Play mode.

Characters



I want to **create a new character**.

- Just drag and drop the prefab Character in the scene (or duplicate a current one). That's quite simple.
- *NB : A parent empty gameobject named CHARACTERS has been created to store them.*

I want to change the **starting coordinates** of a character.

- In the hierarchy, in your character, in its child Move, change X and Y coordinates (in Move script).

I want to change the **starting orientation** of a character.

- In the hierarchy, in your character, in its child Move, change Orientation (in Move script).

I want to change the **team** of a character.

- In the hierarchy, in your character, in its child Info, change its Team value: all the characters with the same value are in the same team.
- *NB : To create a new team, read the Teams section (just under).*

I want to **set the character playable** / **set it as a NPC with a behavior**.

- In the hierarchy, in your character, in its child Behavior:
    - If you want it playable, check the Playable checkbox (the other parameters under doesn't matter).
    - If you want it as a NPC, uncheck the Playable checkbox and select a Behaviour under:
        - None: does nothing, simply passes its turn.

        - Follower: follows the target (parameter under this one).
        - Attacker once: find the closest enemy and attack it only once.
        - Offensive: find the closest enemy and attack it until the end of its action points.

I want to **use diagonals** or not for moving characters.

- In the hierarchy, in M_Rules, check Use diagonals if you want characters to use diagonals for their movement.

I want my character to be able to **pass through other ones** or not.

- In the hierarchy, in M_Rules, select Can Pass Across:
    - Everybody: character pass through allies and enemies
    - Nobody: character consider other one like obstacles
    - Allies only: character pass through allies only

I want to change the **name** of a character.

- In the hierarchy, in your character, in its child Info, change its Designation.

I want to change the **health points** of a character.

- In the hierarchy, in your character, in its child Health, you can define health (the starting health) and max health (the maximum health a character can have, *for example if a character begins hurted, health and max health could be different*).

I want to change the **action points** of a character.

- ***Game rule****: Every* ***movement costs 1 action point****. Each* ***attack costs 3 action points*** *(you can change it, cf. next point).*
- In the hierarchy, in your character, in its child Action points, you can define action points (the starting action points) and max action points (the maximum action points a character can have, *for example if a character begins without all its action points*).

I want to change the **action point cost of a character's attack** .

- In the hierarchy, in your character, in its child Attack, set the value Action Points Cost.

I want to change the **attack damage** of a character.

- In the hierarchy, in your character, in its child Attack, set the damages between X and Y. Damages are between these values (both values included). If you want a fixed damage value, set both to the same value.

I want to change the **attack range** of a character.

- In the hierarchy, in your character, in its child Look, you can change the value of Range.

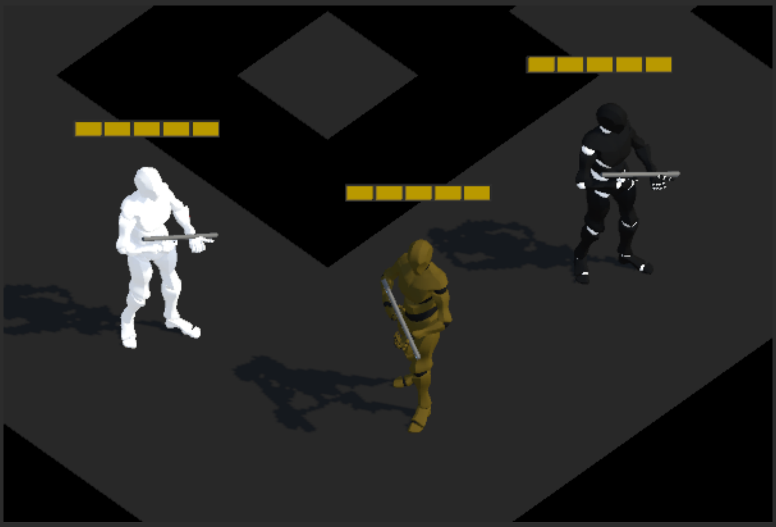
I want to change the **percentage of chance to touch** (depending on the distance with the target).

- In the hierarchy, in M_Rules, change **Percent reduction by distance** value : for each tile of distance with its target, a character have this value in malus (*ex: if a character haves 3 tiles between him and its target, and a Percent reduction by distance equals to 5, he haves 100% - (3 x 5%) =* ***85% chances*** *to reach his target*).

I want to **choose which one is the first character to play**.

- In the hierarchy, in **M_Rules**, select in First character:
  - Random, if you want a random character begins to play.
  - First of hierarchy, if you want the first character in hierarchy begins to play.
  - Current character if you want the Current character in M_Characters begins to play (it can't be null!).

Teams



I want to **create a new team**.

- In the hierarchy, in M_Rules, this is a list of teams named Team infos.
- Add a new entry in the list, choose a name and two materials for the team members' color.

I want to change the **name or color** of a team.

- In the hierarchy, in M_Rules, this is a list of teams named Team infos.
- You can change the team name (this name will appear if the team wins the fight).
- You can change the materials of the team (currently, there are the two materials of the character's model, first one for the skin and second one for the joints).

Camera

I want to **change the distance** between characters and the camera.

- In the hierarchy, in M_GameCamera, you can set the X, Y and Z Offset values to set the offset between the camera and the current character.

I want to make the camera **quicker**/**slower**.

- In the hierarchy, in M_GameCamera, you can change the **Moving time** (the time the camera needs to reach its destination). The higher this value is, the slower the camera is.
- The **Moving Time Curve** between this value smoothes the movement (basically a logarithmic curve, so quick movement at the beginning and slow in the end).
- If you want to **change the speed multiplier** when the pointer overlaps a screen border extremity, go in the hierarchy, in M_Input and change the Border Multiplier to the desired value.

I want to change the **size of the detection zone** on the screen borders (which moves the camera).

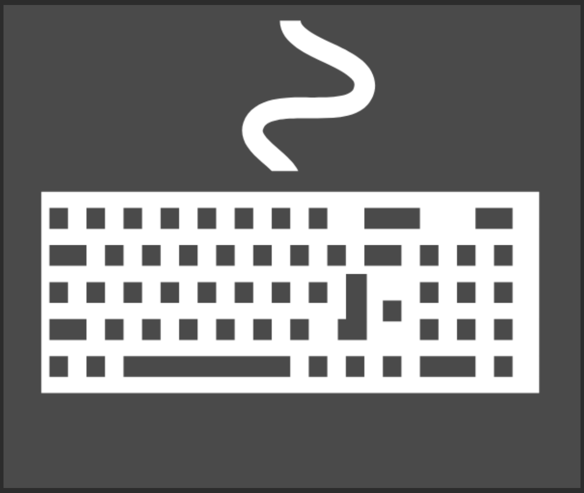
- In the hierarchy, in M_Input, set the Screen Percent value to set the percent of your screen (width and height) considering the screen border (and starting the movement on this border).

Pointers

I want to change the **special pointers sprite**.

- In the hierarchy, in M_Feedback, you can change the sprite of:
    - **Aim Cursor** appears when the mouse overlaps an enemy and the player has action points enough to attack it.
    - **Out Aim Cursor** appears when the mouse overlaps an enemy and the player hasn't line of sight on it.
    - **Out Action Points Cursor** appears when the mouse overlaps an enemy and the player hasn't enough action points to attack it.

Shortcuts



I want to change the **keyboard player's shortcuts**.

- In the hierarchy, in M_Input, you can change the key used for:
    - **Recenter camera** on the current character.
    - **Change character** in the team.
    - **End your turn** and pass to the new team turn.

- TECHNICAL DOCUMENTATION

How works a game turn

Game initialization : Tile board construction

In this order :

- M_Board **creates the board** on Awake with the given data.
- M_Characters :
    - **finds all the characters on the scene** and puts them in the character’s list.
    - **starts the turn of the first character** chosen by M_Rules
    - passes to the next step.

Beginning of a turn

This is the **initialization of all the elements of the turn**. It’s managed by **M_Characters**, who manages the characters’ turn.
NB : The method who initializes the situation is NewCurrentCharacter().

- It **clears all the feedback**.
- It gives the **camera** a new target and recenters it.
- It **fills the action points** of the character.
- The next step depends if the character is a playable character (PC) or a non playable character (NPC).
    - If the **character is a PC**, it enables the player's input, player’s UI and enables the tiles feedback of the character (go to the step Playable character’s turn).
    - If the **character is a NPC**, it disables the player’s input, disables the player’s UI and plays the NPC’s behavior (go to the step Non playable character’s turn).
    - *NB : To know if a character is a PC or a NPC, uncheck or check the Playable parameter on C_Behavior, on the character (in the scene).*

Playable character’s turn : Player’s action choice

A player plays his character's turn.
M_Input checks for the player’s input.

- He can **put his pointer over tiles** :
    - If the tile is **not allowed** (hole, an obstacle or if the pointer is out of the board), it clears feedback (line, cursor, pathfinding values, etc.).
    - If the **tile is occupied** by an enemy, it sets the cursor depending on the line of sight on this enemy and the action points.
    - If the **tile is free** and accessible (no matter the action points), it sets moving feedback.
    - *NB : M_Feedback manages feedback and the cursor and M_Pathfinding returns the path and the line of sight.*
- He can **click on a tile** :

  - If this **tile is occupied** by an enemy, it allows the character to attack the target (the character will block the attack if the enemy is out of sight or the character hasn’t enough action points). The attack is managed by the C_Attack script of the character.
  - If the **tile is free** and the character can move on (matter its action points), it allows the character to move. The movement is managed by the C_Move script of the character.
- He can press the shortcut to **pass to his next playable character**, managed by M_Characters (back to the Beginning of a turn).
- He can press the shortcut to **pass to the next player’s turn**, managed by M_Characters (back to the Beginning of a turn).
- He can press the shortcut to **recenter the camera** on the current character, managed by M_Camera.
- He can put his pointer over the borders of the screen, so it will **start the camera movement** (managed by M_Camera).

Non playable character’s turn : Behavior execution

If the character is a NPC :

- M_Characters disables the UI and starts the NPC’s behavior (on NPC’s C_Behavior script).
- C_Behaviour checks the current behavior :
  - None : character waits 1 second and passes to the next turn.
  - Follower : if the character has a target parameter in C_Behavior component, it goes the closer it can to the target. Else, it passes to the next turn.
  - AttackerOnce : character finds the closest enemy on sight and range and attacks it once.
  - Offensive : like AttackerOnce, but the character attacks until it has no Action points.

Team victory

At the end of a character’s turn or action, **C_Characters checks if the team wins**.
In this case, it shows the victory screen.

Managers

All the big parts of the game are managed by a **manager class**, named *M_PartOfTheGameToManage*.
They are all **singleton**.
They all begin by **M_** to signify they are managers.

M_Board creates the tile board and contains the tiles list.
M_Creator manages the tile board creation and contains Edit mode (or play mode) choice.
M_Rules contains the different rules of the game, to change them easily and make quick tests.
M_Input checks the player's controls.
M_Characters manages characters list and turns (next turn, next character, victory, etc.).
M_UI manages all actions made by UI elements (and UI elements themselves).

M_Feedback manages feedback and pointers.
M_Camera manages the camera of the game.
M_Pathfinding manages pathfinding and line of sight methods.

NB : **M__Managers** is used to reference all the managers and simplify their call in the code.

Character components

The character contains **different components** working together (health, movement, attack, behavior, etc.).
C__Character contains references of all these components, and all these components contain a reference to C__Characters to make easy code references.
They all begin by **C_** to signify they are character components.

C_ActionPoints manages the action points of the character.
C_AnimatorScript manages the animations of the character (with events on the animations).
C_Attack manages the attack.
C_Behavior manages the behavior of non playable characters.
C_Health manages health and death.
C_Infos contains the name, the team and gives the team's colors to the character.
C_Look manages line of sight, range and target acquisition.
C_Move contains coordinates and manages movement.

Other scripts

Feedback scripts begin by a **F_** to signify they are feedbacks.
UI scripts begin by a **UI_** to signify they are UI.

NB : Some scripts have no letter on the beginning, they are misc scripts.

/---------o---------\

- CREDITS

Icons from **game-icons.net**.
Character and animations from **mixamo.com**.
Thanks to **@henriforshort** for his help.
Made by **@dracot** for fun.

\---------o---------/

/---------o---------\

- LICENSE

This is an **open source** project.
You are **free to use it** for every usage (including commercial) **without citation**.
But you are free to let me know if you use this project, I'll be happy to know it.

\---------o---------/