# ${\bf Exploration DuMonde}$

- carte\_: std::shared\_ptr<Carte>
- carteLue\_: bool

- paysDestination\_: std::string
- paysOrigine\_: std::string

- Carte sommets\_: std::map<std::string, std::shared\_ptr<Sommet>>
- addSommet(Sommet&): void
- addSommet(std::shared\_ptr<Sommet>): void
- Carte()
- Carte(Carte&)
- colorierGraphe(): void
- creerGraphe(std::string&): bool extractionGraphe(char): Carte
- lireGraphe(): void {query}
- plusCourtChemin(std::string, std::string): void

- couleurAEviter\_: char
- frontieresDeterminees : bool
- nomFichier\_: std::string
- afficherCarte(): void choisirOptionMenu(): char
- determiner Frontieres (): void
- determiner Plus Court Chemin (): void
- ExplorationDuMonde()
- lancer(): void
- lireCarte(): void
- reinitialiserJeu(): void

# Sommet

- arretes\_: std::vector<std::pair<std::shared\_ptr<Sommet>, int>>
- couleur\_: char

0..\*

- nom\_: std::string
- + addNeighbor(std::shared\_ptr<Sommet>&, int): bool
- + adjacentACouleur(char): bool
- + afficher(std::ostream&): void {query}
- + getArretes(): std::vector<std::pair<std::shared\_ptr<Sommet>, int>> {query}
- + getCouleur(): char {query}
- + getNom(): std::string {query}
- + getSize(): int {query}
- + removeNeighbor(char): std::shared\_ptr<Sommet>
- setCouleur(char): void
- setNom(std::string): void
- Sommet(std::string)
- ~Sommet()

# Jeulnstructif

- lexiqueLu\_: bool
- choisirOptionMenu(): char
- entrerLettre(std::string&): void
- initialiserJeu(): void
- JeuInstructif()
- lancer(): void
- saisirTexte(bool, bool): void
- verifierMotFini(): bool

### Automate

- currState\_: std::shared\_ptr<Etat>
- lexique\_: std::vector<std::shared\_ptr<std::string>>
- startState\_: std::shared\_ptr<Etat>
- ajouterMot(std::string): void
- Automate(std::vector<std::shared\_ptr<std::string>>&)
- corrigerMot(std::string&): std::vector<std::shared\_ptr<std::string>>
- suggererMots(std::string&): std::vector<std::string>
- suggererMots(std::shared\_ptr<Etat>&, std::vector<std::string>&): void transition(char): bool

Etat

 $is Terminal\_: bool$ 

name\_: std::string

0..\* transitions\_: std::map <char, std::shared\_ptr<Etat>>

addTransition(char, bool): std::shared\_ptr<Etat>

addTransition(char, std::string, bool): void

addTransition(char, std::shared\_ptr<Etat>): void

+ Etat(std::string, bool)

getNom(): std::string {query}

getTransitions(): std::map <char, std::shared\_ptr<Etat>>&

hasTransition(char&): bool {query}

isTerminal(): bool {query}

transition(char&): std::shared\_ptr<Etat>&