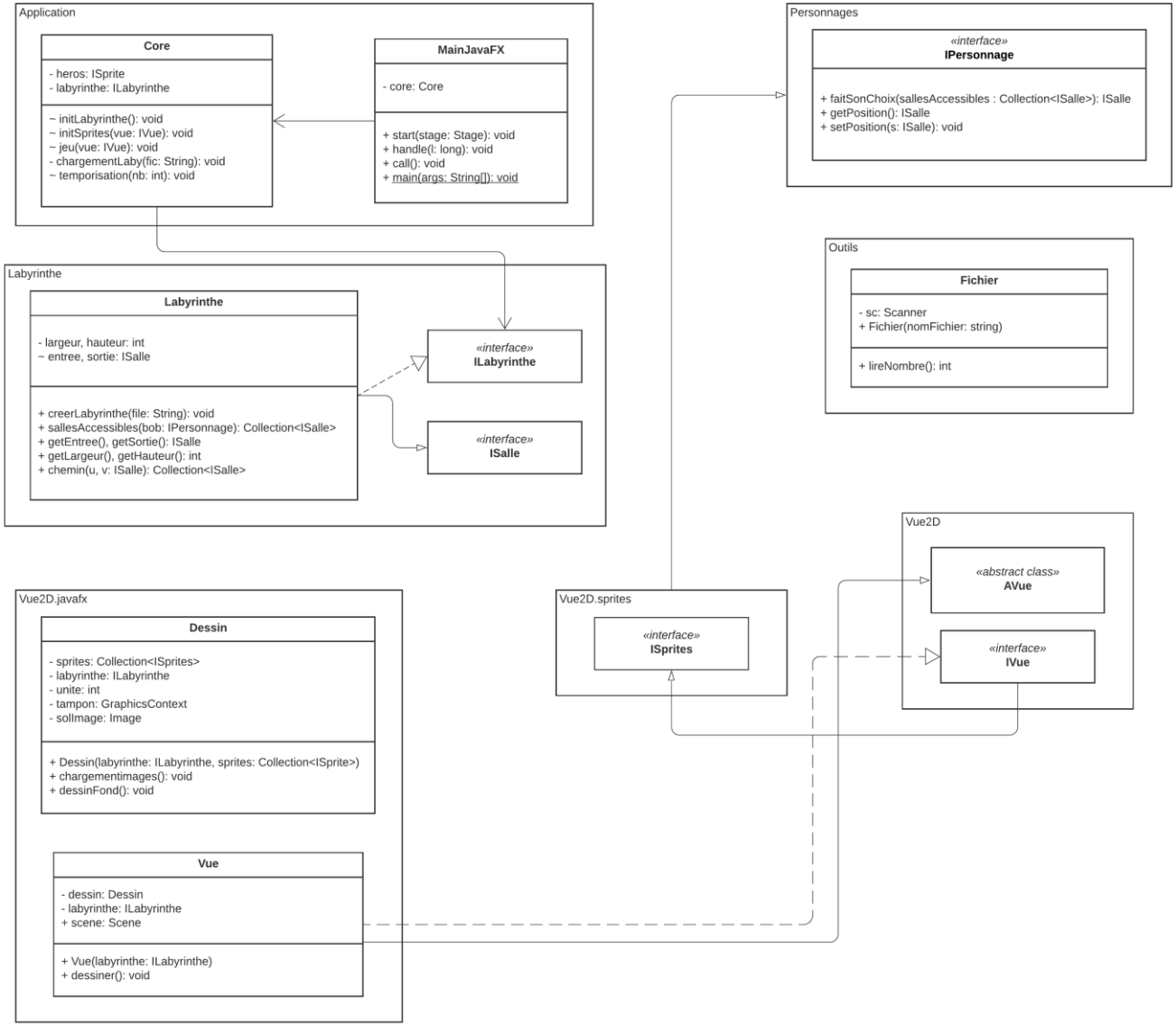
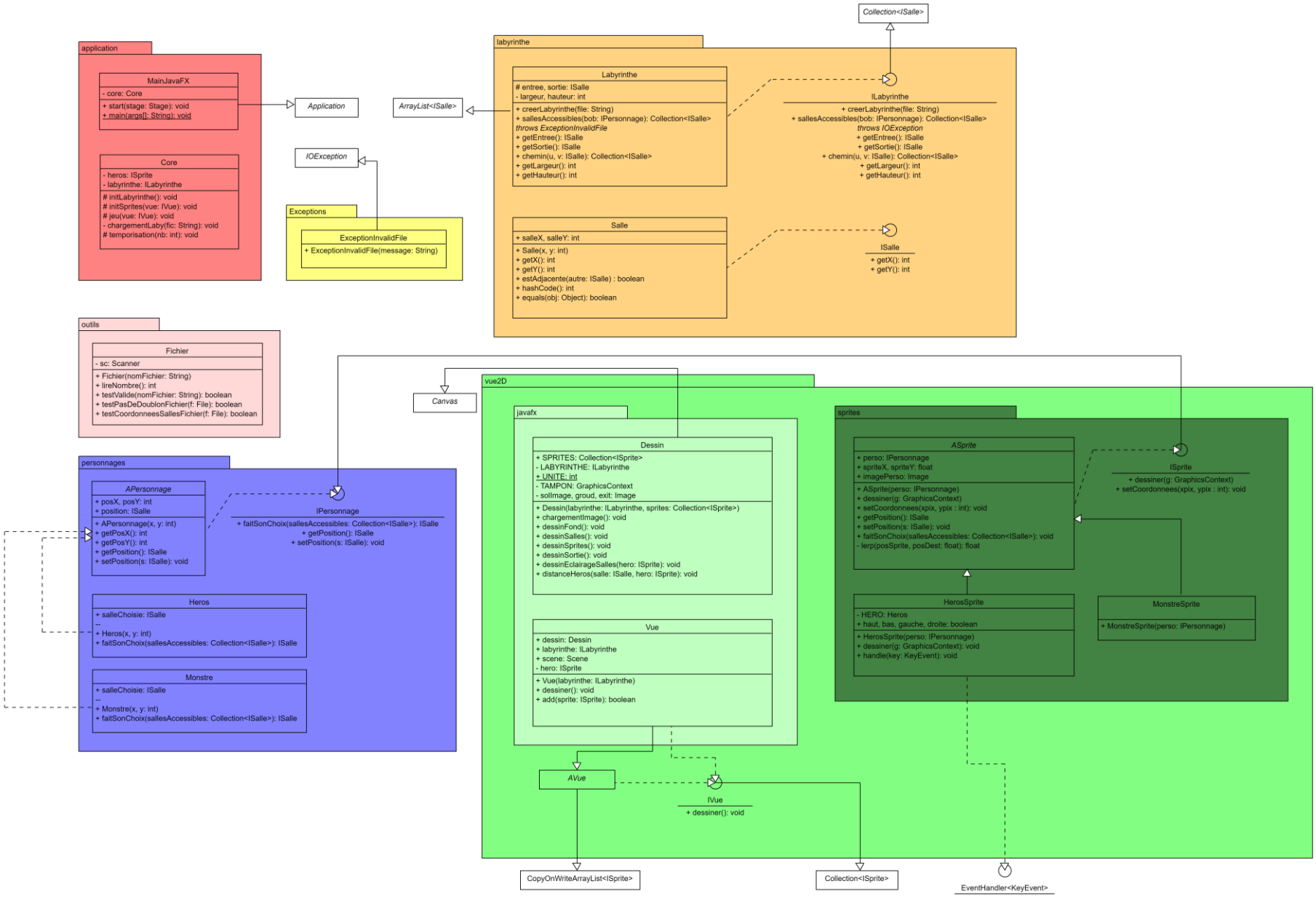

Application
Core
- heros: ISprite
- labyrinthe: ILabyrinthe
~ initLabyrinthe(): void
~ initSprites(vue: IVue): void
~ jeu(vue: IVue): void
- chargementLaby(fic: String): void
~ temporisation(nb: int): void
MainJavaFX
- core: Core
+ start(stage: Stage): void
+ handle(l: long): void
+ call(): void
+ main(args: String[]): void
Personnages
«interface»
IPersonnage
+ faitSonChoix(sallesAccessibles : Collection<ISalle>): ISalle
+ getPosition(): ISalle
+ setPosition(s: ISalle): void
Labyrinthe
Labyrinthe
- largeur, hauteur: int
~ entree, sortie: ISalle
+ creerLabyrinthe(file: String): void
+ sallesAccessibles(bob: IPersonnage): Collection<ISalle>
+ getEntree(), getSortie(): ISalle
+ getLargeur(), getHauteur(): int
+ chemin(u, v: ISalle): Collection<ISalle>
«interface»
ILabyrinthe
«interface»
ISalle
Outils
Fichier
- sc: Scanner
+ Fichier(nomFichier: string)
+ lireNombre(): int
Vue2D
«abstract class»
AVue
«interface»
IVue
Vue2D.sprites
«interface»
ISprites
Vue2D.javafx
Dessin
- sprites: Collection<ISprites>
- labyrinthe: ILabyrinthe
- unite: int
- tampon: GraphicsContext
- solImage: Image
+ Dessin(labyrinthe: ILabyrinthe, sprites: Collection<ISprite>)
+ chargementimages(): void
+ dessinFond(): void
Vue
- dessin: Dessin
- labyrinthe: ILabyrinthe
+ scene: Scene
+ Vue(labyrinthe: ILabyrinthe)
+ dessiner(): void


application
MainJavaFX
- core: Core
+ start(stage: Stage): void
+ main(args[]: String): void
Core
- heros: ISprite
- labyrinthe: ILabyrinthe
# initLabyrinthe(): void
# initSprites(vue: IVue): void
# jeu(vue: IVue): void
- chargementLaby(fic: String): void
# temporisation(nb: int): void
Application
ArrayList<ISalle>
IOException
Exceptions
ExceptionInvalidFile
+ ExceptionInvalidFile(message: String)
Collection<ISalle>
labyrinthe
Labyrinthe
# entree, sortie: ISalle
- largeur, hauteur: int
+ creerLabyrinthe(file: String)
+ sallesAccessibles(bob: IPersonnage): Collection<ISalle>
throws ExceptionInvalidFile
+ getEntree(): ISalle
+ getSortie(): ISalle
+ chemin(u, v: ISalle): Collection<ISalle>
+ getLargeur(): int
+ getHauteur(): int
ILabyrinthe
+ creerLabyrinthe(file: String)
+ sallesAccessibles(bob: IPersonnage): Collection<ISalle>
throws IOException
+ getEntree(): ISalle
+ getSortie(): ISalle
+ chemin(u, v: ISalle): Collection<ISalle>
+ getLargeur(): int
+ getHauteur(): int
Salle
+ salleX, salleY: int
+ Salle(x, y: int)
+ getX(): int
+ getY(): int
+ estAdjacente(autre: ISalle) : boolean
+ hashCode(): int
+ equals(obj: Object): boolean
ISalle
+ getX(): int
+ getY(): int
outils
Fichier
- sc: Scanner
+ Fichier(nomFichier: String)
+ lireNombre(): int
+ testValide(nomFichier: String): boolean
+ testPasDeDoublonFichier(f: File): boolean
+ testCoordonneesSallesFichier(f: File): boolean
Canvas
vue2D
personnages
APersonnage
+ posX, posY: int
+ position: ISalle
+ APersonnage(x, y: int)
+ getPosX(): int
+ getPosY(): int
+ getPosition(): ISalle
+ setPosition(s: ISalle): void
IPersonnage
+ faitSonChoix(sallesAccessibles: Collection<ISalle>): ISalle
+ getPosition(): ISalle
+ setPosition(s: ISalle): void
Heros
+ salleChoisie: ISalle
--
+ Heros(x, y: int)
+ faitSonChoix(sallesAccessibles: Collection<ISalle>): ISalle
Monstre
+ salleChoisie: ISalle
--
+ Monstre(x, y: int)
+ faitSonChoix(sallesAccessibles: Collection<ISalle>): ISalle
javafx
Dessin
+ SPRITES: Collection<ISprite>
- LABYRINTHE: ILabyrinthe
+ UNITE: int
- TAMPON: GraphicsContext
- sollmage, groud, exit: Image
+ Dessin(labyrinthe: ILabyrinthe, sprites: Collection<ISprite>)
+ chargementImage(): void
+ dessinFond(): void
+ dessinSalles(): void
+ dessinSprites(): void
+ dessinSortie(): void
+ dessinEclairageSalles(hero: ISprite): void
+ distanceHeros(salle: ISalle, hero: ISprite): void
Vue
+ dessin: Dessin
+ labyrinthe: ILabyrinthe
+ scene: Scene
- hero: ISprite
+ Vue(labyrinthe: ILabyrinthe)
+ dessiner(): void
+ add(sprite: ISprite): boolean
sprites
ASprite
+ perso: IPersonnage
+ spriteX, spriteY: float
+ imagePerso: Image
+ ASprite(perso: IPersonnage)
+ dessiner(g: GraphicsContext)
+ setCoordonnees(xpix, ypix : int): void
+ getPosition(): ISalle
+ setPosition(s: ISalle): void
+ faitSonChoix(sallesAccessibles: Collection<ISalle>): void
- lerp(posSprite, posDest: float): float
ISprite
+ dessiner(g: GraphicsContext)
+ setCoordonnees(xpix, ypix : int): void
HerosSprite
- HERO: Heros
+ haut, bas, gauche, droite: boolean
+ HerosSprite(perso: IPersonnage)
+ dessiner(g: GraphicsContext): void
+ handle(key: KeyEvent): void
MonstreSprite
+ MonstreSprite(perso: IPersonnage)
AVue
IVue
+ dessiner(): void
CopyOnWriteArrayList<ISprite>
Collection<ISprite>
EventHandler<KeyEvent>