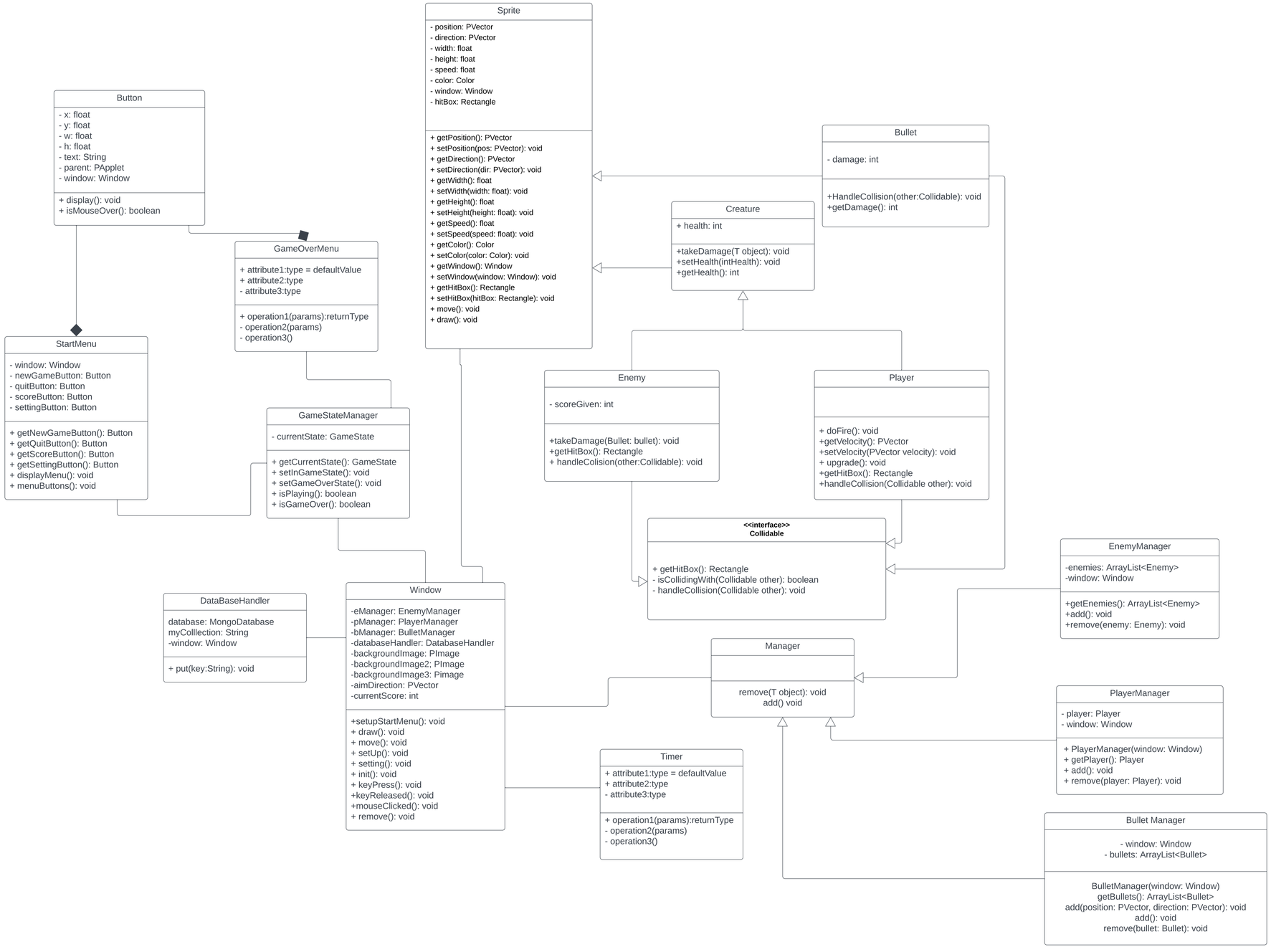
Sprite
- position: PVector
- direction: PVector
- width: float
- height: float
- speed: float
- color: Color
- window: Window
- hitBox: Rectangle
+ getPosition(): PVector
+ setPosition(pos: PVector): void
+ getDirection(): PVector
+ setDirection(dir: PVector): void
+ getWidth(): float
+ setWidth(width: float): void
+ getHeight(): float
+ setHeight(height: float): void
+ getSpeed(): float
+ setSpeed(speed: float): void
+ getColor(): Color
+ setColor(color: Color): void
+ getWindow(): Window
+ setWindow(window: Window): void
+ getHitBox(): Rectangle
+ setHitBox(hitBox: Rectangle): void
+ move(): void
+ draw(): void
Button
- x: float
- y: float
- w: float
- h: float
- text: String
- parent: PApplet
- window: Window
+ display(): void
+ isMouseOver(): boolean
Bullet
- damage: int
+HandleCollision(other:Collidable): void
+getDamage(): int
Creature
+ health: int
+takeDamage(T object): void
+setHealth(intHealth): void
+getHealth(): int
GameOverMenu
+ attribute1:type = defaultValue
+ attribute2:type
- attribute3:type
+ operation1(params):returnType
- operation2(params)
- operation3()
StartMenu
- window: Window
- newGameButton: Button
- quitButton: Button
- scoreButton: Button
- settingButton: Button
+ getNewGameButton(): Button
+ getQuitButton(): Button
+ getScoreButton(): Button
+ getSettingButton(): Button
+ displayMenu(): void
+ menuButtons(): void
GameStateManager
- currentState: GameState
+ getCurrentState(): GameState
+ setInGameState(): void
+ setGameOverState(): void
+ isPlaying(): boolean
+ isGameOver(): boolean
Enemy
- scoreGiven: int
+takeDamage(Bullet: bullet): void
+getHitBox(): Rectangle
+ handleColision(other:Collidable): void
Player
+ doFire(): void
+getVelocity(): PVector
+setVelocity(PVector velocity): void
+ upgrade(): void
+getHitBox(): Rectangle
+handleCollision(Collidable other): void
<<interface>>
Collidable
+ getHitBox(): Rectangle
- isCollidingWith(Collidable other): boolean
- handleCollision(Collidable other): void
EnemyManager
-enemies: ArrayList<Enemy>
-window: Window
+getEnemies(): ArrayList<Enemy>
+add(): void
+remove(enemy: Enemy): void
DataBaseHandler
database: MongoDatabase
myCollection: String
-window: Window
+ put(key:String): void
Window
-eManager: EnemyManager
-pManager: PlayerManager
-bManager: BulletManager
-databaseHandler: DatabaseHandler
-backgroundImage: PImage
-backgroundImage2: PImage
-backgroundImage3: PImage
-aimDirection: PVector
-currentScore: int
+setupStartMenu(): void
+ draw(): void
+ move(): void
+ setUp(): void
+ setting(): void
+ init(): void
+ keyPress(): void
+keyReleased(): void
+mouseClicked(): void
+ remove(): void
Manager
remove(T object): void
add() void
PlayerManager
- player: Player
- window: Window
+ PlayerManager(window: Window)
+ getPlayer(): Player
+ add(): void
+ remove(player: Player): void
Timer
+ attribute1:type = defaultValue
+ attribute2:type
- attribute3:type
+ operation1(params):returnType
- operation2(params)
- operation3()
Bullet Manager
- window: Window
- bullets: ArrayList<Bullet>
BulletManager(window: Window)
getBullets(): ArrayList<Bullet>
add(position: PVector, direction: PVector): void
add(): void
remove(bullet: Bullet): void