

The diagram illustrates the architecture of a 2D game project, organized into several interconnected packages and classes.

- BaseScreen Class:**
 - Attributes: `Canvas`, `Age: GraphicsContext`.
 - Operations: `+BaseScreen(Canvas: Canvas)`, `+start(): void`, `+onKeyUp: KeyEvent() void`, `+onKeyPressed: KeyEvent() void`.
- GameScreen Class:**
 - Attributes: `NUM_ENEMIES: int`, `NUM_ROOMS: int`, `LEVEL: int`, `background: Image`, `shootRight: boolean`, `shootUp: boolean`, `playing: boolean`, `bulletTimer: int`, `score: int`, `scoreLabel: Label`, `speedX: double`.
 - Operations: `+GameScreen(Canvas: Canvas, scoreLabel: Label, transitionScreen: TransitionScreen, speedX: int)`, `+startBulletTimer(): void`, `+resetGame(): void`, `+generateEnemies(speedX: double): void`, `+paint(): void`, `+checkCollision(): void`, `+checkBullets(): void`, `+onKeyUp: KeyEvent() void`, `+onKeyPressed: KeyEvent() void`, `+onScreenChange(): void`.
- TransitionScreen Class:**
 - Attributes: `Label: String`, `win: boolean`, `currentScore: String`, `highScore: String`.
 - Operations: `+TransitionScreen(Canvas: Canvas)`, `+paint(): void`, `+onKeyUp: KeyEvent() void`, `+onKeyPressed: KeyEvent() void`.
- PauseScreen Class:**
 - Attributes: `Canvas: Canvas`.
 - Operations: `+PauseScreen(Canvas: Canvas)`, `+paint(): void`, `+onKeyUp: KeyEvent() void`, `+onKeyPressed: KeyEvent() void`.
- ScoreData Class:**
 - Attributes: `highScore: int`.
 - Operations: `+updateScoreOnNewScore(int): void`, `+retrieveScore(): void`, `+loadScore(): void`.
- Energy Class:**
 - Attributes: `speedX: double`, `speedY: int`, `image: Image`, `active: boolean`, `moveRight: boolean`, `shoot: int`.
 - Operations: `+Energy(Canvas: Canvas, x: int, y: int, width: int, height: int, impulse: int, speedX: double)`, `+relative(): boolean`, `+setRelative(boolean): void`, `+paint(): void`, `+changeDirection(): void`, `+run(): void`.
- HBBox Class:**
 - Attributes: `Canvas: Canvas`, `age: GraphicsContext`, `x: double`, `y: double`, `width: int`, `height: int`, `centerX: double`, `centerY: double`.
 - Operations: `+HBBox(Canvas: Canvas, x: double, y: double, width: int, height: int)`, `+paint(): void`.
- Bullet Class:**
 - Attributes: `image: Image`, `shootRight: boolean`, `moveLeft: boolean`, `moveUp: boolean`, `moveDown: boolean`, `active: boolean`, `move: boolean`, `alive: boolean`.
 - Operations: `+Bullet(Canvas: Canvas, x: double, y: double, width: int, height: int, adjustment: int, enemyBullet: boolean)`, `+paint(): void`.
- Avatar Class:**
 - Attributes: `Canvas: Canvas`, `x: int`, `y: int`, `width: int`, `height: int`.
 - Operations: `+Avatar(Canvas: Canvas, x: int, y: int, width: int, height: int)`, `+paint(): void`, `+moveX(): int: void`, `+moveY(): int: void`, `+addBullet(Bullet: Bullet): void`, `+removeBullet(Bullet: Bullet): void`, `+resetBullet(): void`.
- MainWindow Class (Main Package):**
 - Operations: `+startGame(): URL, arg1: ResourceBundle() void`, `+onEvents(): void`, `+onEvents(): void`, `+startGame(): ActionEvent() void`, `+continueGame(): ActionEvent() void`.
- Main Class (Main Package):**
 - Operations: `+startGame(): String: void`, `+main(): void`.
- Control Package:** Contains the `MainWindow` class.
- Model Package:** Contains the `HBBox` class.
- View Package:** Contains the `Avatar` class.
- Enemies Package:** Contains the `Energy` class.
- ScoreData Package:** Contains the `ScoreData` class.
- GameScreen Package:** Contains the `GameScreen` class.
- TransitionScreen Package:** Contains the `TransitionScreen` class.
- PauseScreen Package:** Contains the `PauseScreen` class.
- BaseScreen Package:** Contains the `BaseScreen` class.

The diagram shows a complex network of dependencies and associations between these classes, reflecting the game's logic and user interface components.