Feature Number: 1

Feature Name: Physics-Based Driving

Programmer: Collin Hughes

Tester: Brian Bennett

- **GameObject** – Abstract base class for playerObjects, colliderObjects
    - Methods for Initialization, Update and Draw
        - GameObject(Vector2 _position, Texture2D _sprite, Orientation _orientation) | Constructor for GameObject
            - Position is location on screen.
            - Sprite is the texture displayed at its location.
            - Orientation is the direction that the sprite is facing.
        - OnUpdate() | Does nothing, only a placeholder for child classes. Should be called in game class Update method. Should be overridden by child class.
        - OnDraw(SpriteBatch _spriteBatch) | Draws the sprite at the position. Should be overridden by child class.
            - SpriteBatch is the sprite batch for the game.

- **ColliderObject** – Base collider class, can be used for non controlled physics objects
    - Methods for Initialization, Update and Draw
        - BoxColliderObject(World _world, Texture2D _sprite, Vector2 _position, BodyType _bodyType, Orientation _orientation) | Constructor for BoxColliderObject
            - World is the game world where the physics body is located
            - Sprite is the texture displayed at its location.
            - Position is the location on screen.
            - BodyType is the type of physics body of the object
                - Static: Unmoving, Unaffected by Physics
                - Kinematic: Moving, Unaffected by Physics and Collision
                - Dynamic: Moving, Affected by Physics and Collision
            - Orientation is the direction that the sprite is facing.
                - For BoxColliderObjects it defaults to “north”. Creates a physics body
        - OnUpdate() | Does nothing, only a placeholder for child classes. Should be called in game class Update method. Should be overridden by child class.

- OnDraw(SpriteBatch _spriteBatch) | Draws the sprite at the position. Should be overridden by child class.
  - SpriteBatch is the sprite batch for the game.

- **PlayerObject** – Main Player Class. Player is a dynamic physics body, controlled by forces and collisions. Eventually it will be controlled by the Interpreter to allow for programmable cars.
  - Has specific stats for movement
    - Handling:
      - How tight the car turns, and how much the car rotates when the wheel is turned.
      - Higher values -> Tighter turns, less rotation
    - Mass:
      - How much force is required to move the car.
      - Higher values -> More force required
    - Acceleration:
      - How much extra power the engine gives the car to accelerate forward.
      - Higher values -> Faster car
  - Methods for Initialization, Update and Draw
    - PlayerObject(World _world, Texture2D _sprite, Vector2 _position, Orientation _orientation, float _braking, float _handling, float _mass, float _acceleration) | Constructor For PlayerObject. Calls constructor for BoxColliderObject using a dynamic body. Calls SetStats with car data.
      - World is the game world where the physics body is located
      - Sprite is the texture displayed at its location.
      - Position is location on screen.
      - Orientation is the direction that the sprite is facing.
      - Braking is how long it takes the car to slow when no longer accelerating or when the brake is pressed.
        - Higher values -> Faster stopping
      - Handling is how tight the car turns, and how much the car rotates when the wheel is turned.
        - Higher values -> Tighter turns, less rotation
      - Mass is how much force is required to move the car.
        - Higher values -> More force required
      - Acceleration is how much extra power the engine gives the car to accelerate forward.
        - Higher values -> Faster car

  - OnUpdate(Vector2 moveInput, float brakeInput) | Calls steer and accelerate with moveInput data. Should be called in Update method of main game class
    - MoveInput is a vector representing what percentage of movement the car should do.
      - X component is used for steering.
      - Y component is used for acceleration.
  - OnDraw(SpriteBatch _spriteBatch) | Draws the sprite using the position, rotation, center origin and scale of the physics body.
    - SpriteBatch is the sprite batch for the game.
- Methods for steering, braking, accelerating and setting stats
  - Steer(float direction) | Steer to the left or right various amounts
    - Direction is a float between -1 and 1.
  - Accelerate(float direction) | Accelerate positively or negatively various amounts
    - Direction is a float between -1 and 1
  - Brake(float brakeInput) | applies force in the negative of the current velocity multiplied by a braking value until the velocity is 0
  - SetStats(float handling, float steering, float acceleration, float mass) | controls the stats of the car, changing how it interacts in the world
    - Braking is how long it takes the car to slow when no longer accelerating or when the brake is pressed.
      - Higher values -> Faster stopping
    - Handling is how tight the car turns, and how much the car rotates when the wheel is turned.
      - Higher values -> Tighter turns, less rotation
    - Mass is how much force is required to move the car.
      - Higher values -> More force required
    - Acceleration is how much extra power the engine gives the car to accelerate forward.
      - Higher values -> Faster car