**Introduction**

* We are going to improve what we have done in project-1. At project-1, we added 1 player and object collisions. In this project, we need to add 2nd player (also effected by collisions) and bullet for warzone environment.

**Modifications made to the existing classes**

* Common attributes and methods of all game objects (except Game and BulletList) are moved to the base Object class. Using inheritance prevents from repeating the same data and code over and over again.
* Texture loading is moved to the init() method to be executed only once at the game start without slowing down window drawing.

**Collisions between sprites to prevent walking through sandbags and barrels**

* The base Object class has the collideObject() method that uses a simple distance-based check for collision detection. It is called by Player and BulletList classes to check collisions with various game objects.
* On collision with the edge of the screen or with sandbags and barrels, the previous position of the player is restored. This prevents leaving the game window or walking through sandbags and barrels.
* sprite.setOrigin() is used to set the reference point of each sprite at its center. This ensures that bullets fire from the center of the player’s sprite instead of its corner. For sandbags and barrels, the sprite center is set somewhat higher because their images are not symmetric – this improves visual collision with players.

**Linked list of bullets**

* Adding a new bullet to the list: for a singly linked list, each new bullet is prepended to the start of the list, updating the list pointer.
* Removing the bullet from the list: for a singly linked list, the next pointer of the previous bullet in the list is needed to properly relink it after deletion. When removing the first bullet in the list, the list pointer is updated accordingly.
* Performing operations on all bullets is done by traversing the list using their next pointers, starting from the first bullet until the null pointer is encountered.

**Bullet collision**

* The checkCollision() method of the BulletList class uses the insideWindow() method of the base Object class to collide bullets with the edge of the screen. Also, it uses the collideObject() method of the base Object class to collide bullets with players, sandbags, and visible barrels. When collided, the bullet is removed from the list and the current pointer is set to the next one. The collided player is respawned and the collided barrel is hidden.

**Sticky keys to handle multiple presses simultaneously**

* The game class contains the stickyKeys attribute, storing the current sticky keys for each player.
* When walking keys are pressed, their codes are stored in the stickyKeys array for each player.
* When walking keys are released, the corresponding stickyKeys entries are reset for each player.
* The players’ positions are updated according to the current values stores in stickyKeys. This allows controlling each player independently from each other.

**The problems faced in the implementation of the project**

* Sprite images are not symmetric around their center and required some adjustments.
* The function sprite.setRotation() expects angles in degrees unlike all functions from the math.h header that expect angles in radians.
* After window resizing, the function window.getSize() returns changed window dimensions but the game world uses width and height of the initial window.
* The Player class has 14 textures for sprite animation but the inherited Object class also has a texture attribute. This requires reassigning the texture object in the init() method and passing the empty texturePath to initialize the base Object class.
* Singly linked list of bullets is harder to maintain than doubly linked list. During traversal, it requires maintaining the previous bullet for updating its next pointer to delete the current bullet.

**How the project could be improved**

* Walking sequence doesn’t look perfect in all cases.
* Collision detection between players could be added.
* Arbitrary rotation and shooting at any angle would look more natural.