Classes:

1. Game Controller
2. Atom
3. Player
4. Henchmen
5. Boss
6. Enemy
7. Prime
8. Personality Test
9. Questions
10. Obstacles
11. Weapons
12. Bullets
13. Food
14. Orbs
15. Cacti
16. Texture
17. Node(Struct)
18. Queue
19. Screen
20. Buttons

Description:

1) *GameController*:
   The GameController manages everything that is happening in the game behind the scenes. This includes instantiating, screens, reloading save data, keeping track of active objects, check if game over, if restart is pressed, if game is paused. (Game States)
2) *Atom*:
   Abstract class. Only useful to unify all objects so they can be of one parent class and hence can be grouped.
3) *Player:*
   This is the main player character. It depicts the player, and its stats.
4) *Henchmen:*
   These are the henchmen characters in the game. Depicting their stats, collider, and position. The player will interact with them and fight them.
5) *Boss:*
   This class refers to the two boss primes i.e. Sulfur and Chlorine, that the player will fight at the end of the game. The
6) *Enemy:*
   All objects of this class are hostiles and to be attacked. They will also attack. The class itself will not be implementing
7) *Prime:*
   Abstract class. All the classes inheriting from this class will be the prime elements.

8) *Personality Test:*
The game has a personality test that assigns the player a character based on their answers to the personality test. This class creates 4 questions objects, and depending on the score resulting from the answer, assigns one of the primes as the player.

9) *Questions:*
This class contains text for each question, and the points for each choice. The encapsulation makes it easy to read and understand structure of code.

10) *Obstacles:*
(This is an abstract class.) All objects deriving from this class will have collider and will damage.

11) *Weapons:*
Class that handles the attributes, damage, and sprite of weapons; Ordinary and special weapons both.

12) *Bullets:*
Electron gun is the standard weapon for all primes. The bullets that are fired at the henchmen are represented by this class. It includes a collision detector.

13) *Food:*
Details and attributes of food objects. This includes details of how much health a food object restores, and the food sprite.

14) *Orbs:*
These orbs act two-way. One is that each orb is assigned a random score, and collecting them will add to the player's score. And two that

15) *Cacti:*
These are obstacles. If the player comes in contact with them, they get hurt.

16) *Texture:*
Texture class for SDL.

17) *Screen:*
These are the main-menu, pause menu, and game over screens. Also the background for the game.

18) *Buttons:*
This class represents the on-screen buttons for the main-menu, pause menu, and the game over screen.

Removed classes:

Aren't required since using std::vector.

1) *Node(Struct)*
2) *Queue*