**------------------------------------------------------**

**Requirement Analysis**

1. Main Menue

* UserName and Password Registration system.
* High Scores
* Game Customization (Options).
  + Changing Sound level.
  + Changing Player Controls.

1. GamePlay

* Players
  + Two Players First With Keybaord and Second with Mouse.
  + Second player can enter the game at any point during the game.
  + The two Players Motion are adjusted to be equal in difficulty.
  + Players Panel that contain number of lives, collected Chickens, number of rockets, and weapon current power.
* Weapons
  + Rocket, that can destroy all Chickens at once
    - A rocket is gained when collecting 50 Chicken Parts.
  + 3 Weapons (Red, Green, Yellow)
  + Every Weapon has its pattern and powers.
  + There is maximum power for each weapon
* Pickups
  + PowerUp which increase the power of the weapon
    - Type that increase any type of weapon
    - 3 Types each increase the weapon power if it is the current and change the weapon if the current one is different.
  + Sheild that protect you for 10 Seconds.
  + Chicken Parts which drop when a chicken dies
    - Full Chicken (Give you 10 Parts)
    - Sandwitch (Give you 5 Parts)
    - Chicken Leg (Give you 1 Part)
* Enemies
  + Chickens
    - Big Chickens that Drop Eggs.
    - Small Chickens
  + Rocks
    - Astroids, rocks goes in diagonals.
    - Rocks that comes vertically and fastly.
    - Stubborn rock that does not exist the screen.
  + Beast
    - One Big chicken with high strength and features.

**------------------------------------------------------**

**Classes**

**player**

**Class Methods:**

int getPlayerNumber();

void drawPlayer();

void fireRocket();

void move();

void explode();

void reborn();

void gainlife();

void loselife();

void gainrocket();

void increaseScore();

void collectChickenPart();

void setPosition();

void powerUp();

void setPlayerState();

playerState getPlayerState();

Position getPosition();

void fire();

Sprite getSprite();

**Chicken**

**Class Methods:**

void move();

void setPosition();

Position getPosition();

void drawSprite();

void die();

Sprite getSprite();

bool damage();

void setStartingFrame();

**SmallChicken (Inherited from Chicken)**

**Class Methods:**

void animate();

void hit();

**BigChicken (inherited from Chicken)**

**Class Methods:**

void animate();

void hit();

void throwEgg();

void throwChickenLeg();

**ChickenPart**

**Class Methods:**

void setPosition(int x, int y);

Position getPosition();

void drawSprite();

void go();

void reverseX(); //reverse direction when hitting a wall

void reverseY(); //reverse direction when hitting the gorund

Sprite getSprite();

ChickenLegType getType(); //different types (sandwich/ chicken leg/ full chicken)

**Egg**

//may be an instance in this class will be implemented

//in the chicken classes

**Class methods:**

void move();

void setPosition();

Position getPosition();

void drawSprite();

void fall();

Sprite getSprite();

void die();

void animate();

**baseBullet**

//different types of bullets will inherit from this class

**Class methods:**

void move();

void setPosition();

Position getPosition();

Void fire(); //to be fired and move up the screen

Sprite getSprite();

void drawSprite();

void accelerate(); //the bullet will be accelerating

**Weapon**

**Class methods:**

void changeWeapon(); //Changing the maxPower also

void powerUp(); //power++

void powerToHalf(); //Power to its half

void fire();

void draw();

int getPower();

WeaponType getCurrentWeapon(); //return the type of current weapon

**------------------------------------------------------**

**Code Structure**

• A class chicken with the functions that draw and position it.

• A class small chicken to inherit the properties of chicken: a new function is added here to rotate the chicken randomly and change the sprite so that it gives a sense of appealing animation.

• A function hit for the chicken so that once it is hit.

• Class chicken parts to have private properties including the position and leg; we will use the chrono library so it makes a delay in the way the chicken parts fall on the player. The chicken parts are considered as parts of the rewards the player could get.

• In the class of what the chicken lays and increment of the sprite so that it gives an appealing sense of animation.

• In the function of processing the background, there will be two BGs so that both moves in a sense that would make them look animated.

• A template vector with a loop that receives anything related to the chicken to draw: draw.explosion, draw.

• In the function observe, we should detect any intersection between anything in the game, player and chickens for example.

• In the function animate, any animation related in the game is done through it like changing the frames of the sprites.