**Astro**

Documentation

# Created by

Punya Gunawardana 6331324021

**2110215 Programming Methodology**

**Semester 2 Year 2020**

**Department of Computer Engineering,**

**Faculty of Engineering, Chulalongkorn University**

**Astro**

**Introduction**

Astro is a game based on the arcade game “Asteroids” released in 1979 recreated in Java with soundtrack and effects. The objective of the game is to shoot all the enemies and get the highest score possible.

**Player:**



You are in a ship in the middle of space. You have the ability to steer and thrust your ship forward and you have the ability to shoot bullets. Your base health is 5 HP and every 2 waves you finish will grant you 3 HP more. Your goal is to survive as long as possible.

**Enemies:**





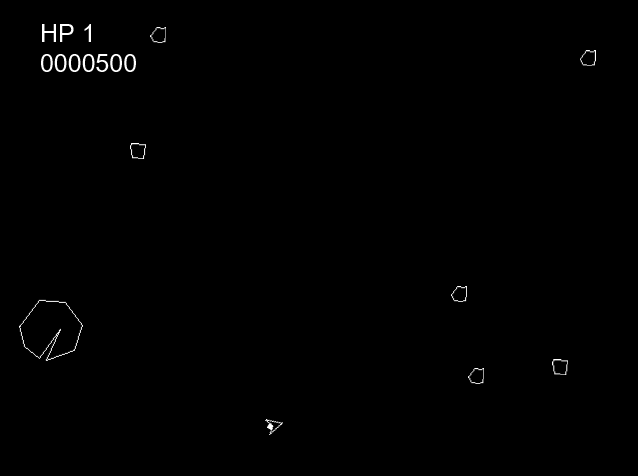


There are 2 types of Enemy in this game Meteoroids and UFO. There are 3 sizes of Meteoroids: big, medium and small. Big Meteoroids have 3 HP while the rest have 1 HP. After you kill a big or medium Meteoroid, 2 of the smaller meteoroids will appear in the same position as where you kill them. All types of meteoroids will travel in a straight line and you will have to try to avoid colliding with them, UFOs on the other hand will move randomly and will try to shoot you.

**Example:**



Player will be presented with 4 buttons when they open the game. Player can press start to enter the game. Pressing the Help button will send you to a screen where it provides you information about how to play the game. Pressing the Credit button will show you the credit of the game. Pressing Exit the program will end.

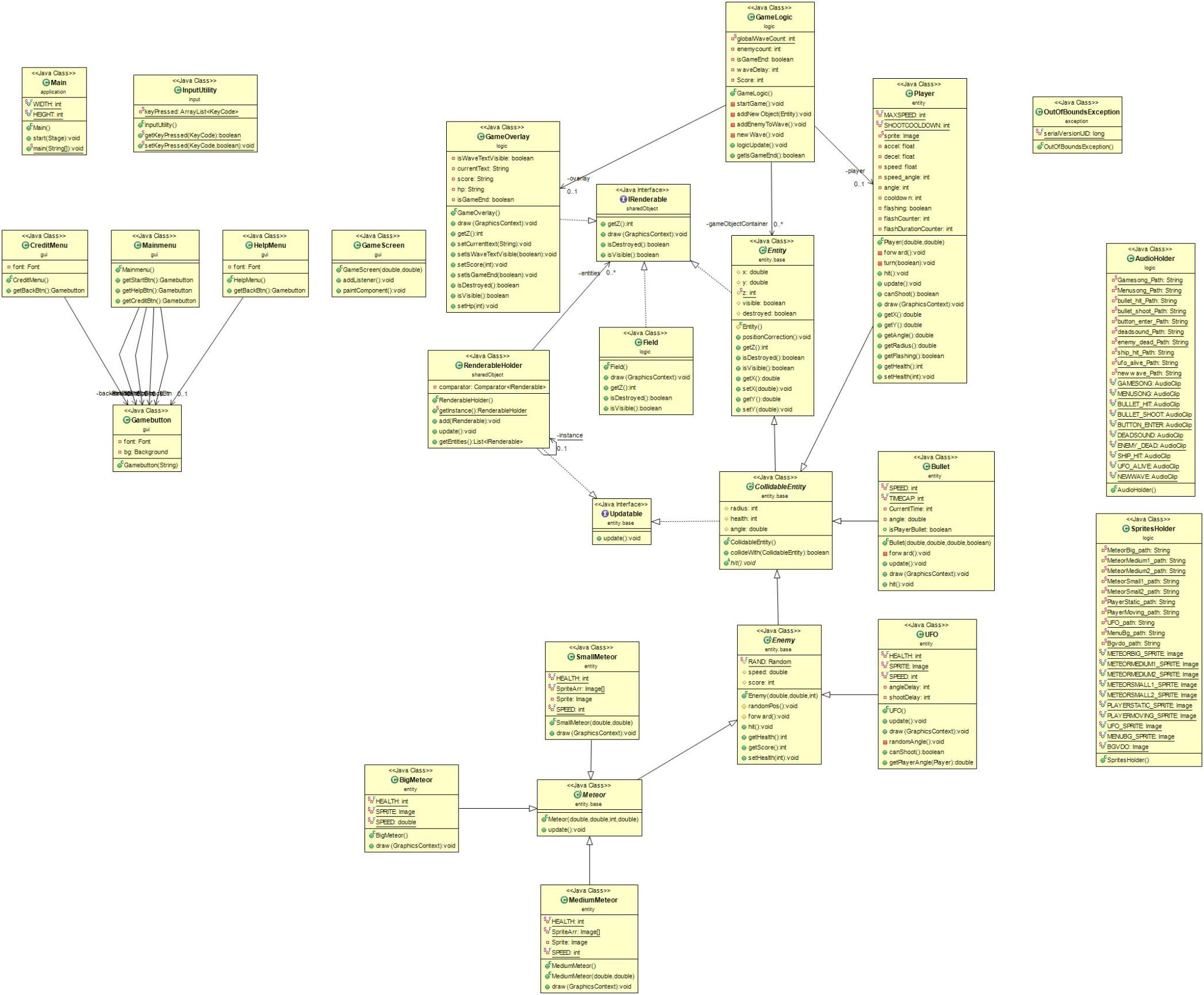


In the game you can see you Hp and your current score on the top left of the screen.

To control the ship, Player can press ‘A’ to steer left and press ‘D’ to steer right. Pressing ‘W’ will thrust the ship forward and pressing ‘J’ will make the ship shoot bullets.

****

When your HP reaches 0 the game will be over. Player can click anywhere to exit the game.

Class Diagram:****

**1. Package application**

**(1.1) Class Main extends Application**

**(1.1.1) Fields**

|  |  |
| --- | --- |
| + final static int WIDTH | Constant for application window width equals 640 |
| + final static int HEIGHT | Constant for application window height equals 480 |

**(1.1.2) Methods**

|  |  |
| --- | --- |
| +void start(Stage primaryStage) throws Exception | Create all the scenes  Set action to all the button  Show the stage |
| +static void main(String[] args) | Launch an application |

**2.Package entity.base**

**(2.1) Abstract Class Entity implements IRenderable**

**(2.1.1) Fields**

|  |  |
| --- | --- |
| # double x | X position of the entity |
| # double y | Y position of the entity |
| # static int z | Render order of the entity |
| # boolean visible | Visibility of the entity |
| # boolean destroyed | Status of the entity |

**(2.1.2) Constructor**

|  |  |
| --- | --- |
| # Entity() | Set visible to true  Set destroyed to false |

**(2.1.3) Methods**

|  |  |
| --- | --- |
| + void positionCorrection() | Correct the position if entity is out of bounds |
| + setter/getters for fields |  |

**(2.2) Abstract Class CollidableEntity extends Entity implements Updatable**

**(2.2.1) Fields**

|  |  |
| --- | --- |
| # int radius | Radius of the Collidable entity |
| # int health | HP of the Colidable entity |
| # double angle | Angle of the Colidable entity |

**(2.2.2) Methods**

|  |  |
| --- | --- |
| + boolean collideWith(CollidableEntity o) | Returns the Collision with the other Colidable entity |
| + abstract void hit() | Abstract void for getting hit |

**(2.3)Abstract class Enemy extends CollidableEntity**

**(2.3.1) Fields**

|  |  |
| --- | --- |
| # final static Random RAND | Static variable for randomizing data |
| # double speed | Speed of the enemy |
| # int score | Score gain from killing that enemy |

**(2.3.2) Constructor**

|  |  |
| --- | --- |
| +Enemy(double x,double y, int health) | Initialize all fields |

**(2.3.3) Methods**

|  |  |
| --- | --- |
| # void randomPos() | Random spawn position of the enemy |
| # void forward() | Move the enemy forward in the facing direction |
| + void hit() | Deduct hp by 1 and check if the enemy HP equals 0, If so set destroyed to true |
| getters/setters for fields |  |

**(2.4) Abstract class Meteor extends Enemy**

**(2.4.1) Constructor**

|  |  |
| --- | --- |
| +Meteor(double x, double y, int health,double speed) | Call the super() methods  randomize the angle  Initialize speed |

**(2.4.2) Methods**

|  |  |
| --- | --- |
| +void update() throws OutOfBoundsException | move the meteor forward  throw OutofBondsException if meteor goes out of bounds |

**(2.5)Interface Updatable**

**(2.5.1) Methods**

|  |  |
| --- | --- |
| + Abstract void update() throws OutOfBoundsException; | Abstract void for update() |

**3.Package entity**

**(3.1)Class BigMeteor extends Meteor**

**(3.1.1) Fields**

|  |  |
| --- | --- |
| -final static int HEALTH | Health constant equals 3 |
| -final static Image SPRITE | Sprite of Big Meteorite |
| -static final double SPEED | Speed constant equals 1 |

**(3.1.2) Constructor**

|  |  |
| --- | --- |
| +BigMeteor() | Calls Super()  Initialize score  Initialize radius  Random position |

**(3.1.3) Methods**

|  |  |
| --- | --- |
| + void draw(GraphicsContext gc) | Draw big meteor |

**(3.2)Class Bullet extends CollidableEntity**

**(3.2.1) Fields**

|  |  |
| --- | --- |
| +final static int SPEED | Speed constant equals 3 |
| +final static int TIMECAP | Maximum bullet traveling time constant equals 200 |
| -int CurrentTime | Current travel time |
| -double angle | Bullet angle |
| +boolean isPlayerBullet | Boolean to check is bullet shoot by player |

**(3.2.2) Constructor**

|  |  |
| --- | --- |
| +Bullet(double x, double y,double d,boolean playerBullet) | Initialize fields |

**(3.2.3) Methods**

|  |  |
| --- | --- |
| +void forward() | Move bullet forward |
| +void update() | Update bullet position and travelling time |
| +void draw(GraphicsContext gc) | Draw Bullet |
| +void hit() | set destroyed to true  set visible to false |

**(3.3)Class MediumMeteor extends Meteor**

**(3.3.1) Fields**

|  |  |
| --- | --- |
| -final static int HEALTH | Health constant equals 1 |
| -final static Image[] SpriteArr | Array containing image sets for the Medium Meteor |
| -Image Sprite | Current Meteor Sprite |
| -final static int SPEED | Speed constant equals 2 |

**(3.3.2) Constructor**

|  |  |
| --- | --- |
| +MediumMeteor() | Calls Super()  Initialize Random Sprite  Random position |
| +MediumMeteor(double x, double y) | Calls Super()  Initialize Positions  Initialize Random Sprite |

**(3.3.3) Methods**

|  |  |
| --- | --- |
| +void draw(GraphicsContext gc) | Draw Medium Meteor |

**(3.4)Class Player extends CollidableEntity**

**(3.4.1) Fields**

|  |  |
| --- | --- |
| -final static int MAXSPEED | Speed constant equals 2 |
| -final static int SHOOTCOOLDOWN | Bullet Shoot cooldown constant equals 40 |
| -static Image sprite | Player Sprite |
| -float accel | Player acceleration speed |
| -float decel | Player deceleration speed |
| -float speed | Player current speed |
| -int speed\_angle | Player current velocity angle |
| -int angle | Player current angle |
| -int cooldown | Player current shoot cooldown |
| -boolean flashing | Player flashing status |
| -int flashCounter | Player flashing counter |
| -int flashDurationCounter | Player flashing duration counter |

**(3.4.2) Constructor**

|  |  |
| --- | --- |
| +Player(double x, double y) | Initialize fields |

**(3.4.3) Methods**

|  |  |
| --- | --- |
| -void forward() | Move player forward |
| -void turn(boolean left) | Turn player |
| +void hit() | Deduct 1 HP from player and set Flashing status |
| +boolean canShoot() | Boolean telling can player shoot |
| +void draw(GraphicsContext gc) | Draw Player |
| getters/setters for fields |  |

**(3.5)Class SmallMeteor extends Meteor**

**(3.5.1)Fields**

|  |  |
| --- | --- |
| -final static int HEALTH | Health constant equals 1 |
| -final static Image[] SpriteArr | Array containing image sets for the Small Meteor |
| -Image Sprite | Current Meteor Sprite |
| -final static int SPEED | Speed constant equals 2 |

**(3.5.2)Constructor**

|  |  |
| --- | --- |
| +SmallMeteor(double x, double y) | Calls Super()  Initialize Positions  Initialize Random Sprite |

**(3.5.3)Methods**

|  |  |
| --- | --- |
| +void draw(GraphicsContext gc) | Draw Small Meteor |

**(3.6)Class UFO extends Enemy**

**(3.6.1)Fields**

|  |  |
| --- | --- |
| -final static int HEALTH | Health constant equals 1 |
| -final static Image SPRITE | UFO Sprite |
| -final static int SPEED | Speed constant equals 2 |
| -int angleDelay | Current angle change cooldown delay |
| -int shootDelay | Current shoot cooldown delay |

**(3.6.2)Constructor**

|  |  |
| --- | --- |
| +UFO() | Calls Super()  Initialize Random Sprite  Random position |

**(3.6.3)Methods**

|  |  |
| --- | --- |
| + void update() throws OutOfBoundsException | move the UFO forward  throw OutofBondsException if meteor goes out of bounds |
| + void draw(GraphicsContext gc) | Draw UFO |
| -void randomAngle() | Random angle for the ufo |
| +boolean canShoot() | Boolean returning status of Can the UFO Shoot |
| getter for fields |  |

**4.Package exception**

**(4.1) Exception OutOfBoundsException**

**(4.1.1) Fields**

|  |  |
| --- | --- |
| - static final long serialVersionUID |  |

**(4.1.2) Constructor**

|  |  |
| --- | --- |
| + OutOfBoundsException() | Call Super()  Set Message |

**5.Package gui**

**(5.1) Class CreditMenu extends StackPane**

**(5.1.1)Fields**

|  |  |
| --- | --- |
| -Font font | font for credit text |
| -Gamebutton backBtn | Game button for going back to Mainmenu |

**(5.1.2)Constructor**

|  |  |
| --- | --- |
| +CreditMenu() | Set credit text  add to self credit text and back button  Set Background |

**(5.1.3)Methods**

|  |  |
| --- | --- |
| +Gamebutton getBackBtn() | return back button object |

**(5.2) Class Gamebutton extends Button**

**(5.2.1)Fields**

|  |  |
| --- | --- |
| - Font font | font for button text |
| - Background bg | Background for button |

**(5.2.2)Constructor**

|  |  |
| --- | --- |
| +Gamebutton(String name) | set Button text to string  set Button font  set Action on Mouse enter to play a sound |

**(5.3) Class GameScreen extends Canvas**

**(5.3.1) Constructor**

|  |  |
| --- | --- |
| +GameScreen(double width, double height) | Calls Super()  SetVisible  Call addListener |

**(5.3.2) Methods**

|  |  |
| --- | --- |
| + void addListener() | Capture user input |
| + void paintComponent() | Paint everything in renderable holder to the screen |

**(5.4) Class HelpMenu extends StackPane**

**(5.4.1)Fields**

|  |  |
| --- | --- |
| -Font font | Font for help text |
| -Gamebutton backBtn | Game button for going back to Mainmenu |

**(5.4.2)Constructor**

|  |  |
| --- | --- |
| + HelpMenu() | Set help text  Set Background  add to self help text and back button |

**(5.4.3)Methods**

|  |  |
| --- | --- |
| +Gamebutton getBackBtn() | return back button object |

**(5.5) Class Mainmenu extends StackPane**

**(5.5.1)Fields**

|  |  |
| --- | --- |
| - Gamebutton startBtn | Game button for Starting game |
| - Gamebutton helpBtn | Game button for going to help screen |
| - Gamebutton creditBtn | Game button for going to credit screen |
| - Gamebutton exitBtn | Game button for exiting program |

**(5.5.2)Constructor**

|  |  |
| --- | --- |
| + Mainmenu() | Set Background  Add all button to self  Set action for exit button |

**(5.5.3)Methods**

|  |  |
| --- | --- |
| getter for fields |  |

**6.Package input**

**(6.1) Class InputUtility**

**(6.1.1) Fields**

|  |  |
| --- | --- |
| -static ArrayList<KeyCode> keyPressed | Arraylist of key being pressed currently |

**(6.1.2)Methods**

|  |  |
| --- | --- |
| +static boolean getKeyPressed(KeyCode keycode) | Return if Keycode are being pressed |
| +static void setKeyPressed(KeyCode keycode,boolean pressed) | Set keycode that are currently being pressed to the array |

**7.Package logic**

**(7.1) Class AudioHolder**

**(7.1.1) Fields**

|  |  |
| --- | --- |
| -static String Gamesong\_Path  -static String Menusong\_Path  -static String bullet\_hit\_Path  -static String bullet\_shoot\_Path  -static String button\_enter\_Path  -static String deadsound\_Path  -static String enemy\_dead\_Path  -static String ship\_hit\_Path  -static String ufo\_alive\_Path  -static String newwave\_Path | Path to Sprites files |
| +final static AudioClip GAMESONG  +final static AudioClip MENUSONG  +final static AudioClip BULLET\_HIT  +final static AudioClip BULLET\_SHOOT  +final static AudioClip BUTTON\_ENTER  +final static AudioClip DEADSOUND  +final static AudioClip ENEMY\_DEAD  +final static AudioClip SHIP\_HIT  +final static AudioClip UFO\_ALIVE  +final static AudioClip NEWWAVE | Sprite images |

**(7.2) Class Field implements IRenderable**

**(7.2.1) Methods**

|  |  |
| --- | --- |
| + void draw(GraphicsContext gc) | Draw blank black screen in the background |
| getters/setters for fields |  |

**(7.3) Class GameLogic**

**(7.3.1) Fields**

|  |  |
| --- | --- |
| - List<Entity> gameObjectContainer | List of all gameObject |
| - Player player | Player object |
| - static int globalWaveCount | Current wave |
| - int enemycount | Current enemy on the screen |
| - boolean isGameEnd | Game status |
| - int waveDelay | Time delay after each wave have ended |
| - int Score | Current Score |
| - GameOverlay overlay | Gameoverlay object |

**(7.3.2)Constructor**

|  |  |
| --- | --- |
| + GameLogic() | Initialize fields  Calls startGame() |

**(7.3.3)Method**

|  |  |
| --- | --- |
| - void startGame() | Add Player,Field,Gameoverlay to GameObjectContainer |
| - void addNewObject(Entity entity) | Method for adding new Game object |
| - void addEnemyToWave() | Add enemy to each wave |
| - void newWave() | Add wave count  Set Overlay Text |
| + void logicUpdate() | Update all game logic |
| + boolean getIsGameEnd() | Getter for current game status |

**(7.4) Class GameOverlay implements IRenderable**

**(7.4.1)Fields**

|  |  |
| --- | --- |
| - boolean isWaveTextVisible | Boolean for telling is Wave text going to be visible |
| - String currentText | String for current Wave text |
| - String score | String for current Score |
| - String hp | String for player HP |
| - boolean isGameEnd; | Boolean for telling has the game ended |

**(7.4.2)Methods**

|  |  |
| --- | --- |
| + void draw(GraphicsContext gc) | Draw Game Overlay |
| getters/setters for fields |  |

**(7.5) Class SpritesHolder**

**(7.5.1) Fields**

|  |  |
| --- | --- |
| -static String MeteorBig\_path  -static String MeteorMedium1\_path  -static String MeteorMedium2\_path  -static String MeteorSmall1\_path  -static String MeteorSmall2\_path  -static String PlayerStatic\_path  -static String PlayerMoving\_path  -static String UFO\_path  -static String MenuBg\_path  -static String Bgvdo\_path | Path to Sprites files |
| +final static Image METEORBIG\_SPRITE  +final static Image METEORMEDIUM1\_SPRITE  +final static Image METEORMEDIUM2\_SPRITE  +final static Image METEORSMALL1\_SPIRTE  +final static Image METEORSMALL2\_SPRITE  +final static Image PLAYERSTATIC\_SPRITE  +final static Image PLAYERMOVING\_SPRITE  +final static Image UFO\_SPRITE  +final static Image MENUBG\_SPRITE  +final static Image BGVDO | Sprite images |

**8.Package sharedObject**

**(8.1) Interface IRenderable**

**(8.1.1) Fields**

|  |  |
| --- | --- |
| + int getZ() | abstract int for GetZ() |
| + void draw(GraphicsContext gc) | abstract void for draw(GraphicsContext gc) |
| + boolean isDestroyed() | abstract boolean for isDestroyed() |
| + boolean isVisible() | abstract boolean for isVisible() |

**(8.2) Class RederableHolder implements Updatable**

**(8.2.1) Fields**

|  |  |
| --- | --- |
| - static final RenderableHolder instance | Current RenderableHolder instance |
| - List<IRenderable> entities | List of all entities in the game |
| - Comparator<IRenderable> comparator | Comparator for 2 object |

**(8.2.1)Constructor**

|  |  |
| --- | --- |
| +RenderableHolder() | Initialized fields |

**(8.2.2)Methods**

|  |  |
| --- | --- |
| +void add(IRenderable entity) | Add new renderable object |
| +void update() | Update renderable object in entities |
| getter for fields |  |