Classes:

Game, GameObject, Player, Asteroid, UFO, Projectile, Map, Items, UI, ID, GameManager

Conventions:

Method – verb – tick, render, isColliding()

Variable – noun = health , xPos, yPos

Boolean functions should ask question – isAlive()

Classes – ExampleClass

Getters/setters - getHealth(), setHealth()

All numbers are floats

Width/height are ints

xPosition, xVector

Class UFO, Player, Asteroid, Player, Projectile,

ID -> enumeration from tutorial

Player Health = 3

Obstacle/UFO health/projectile health = 1

Damage = 1

Score + 1 asteroid, + 2 ufo

**Game - high priority - Kiana**

* + main game loop
* + render
* + tick

**GameObject- Lars**

* -xPosition, yPosition
* -width, height
* -speed
* -xVector, yVector
* -xPosition += xVector \* speed
* -yPosition += yVector \* speed
* -scale
* +constructor, getter, setters, tick, render
* - texture -> string -> playerTexture
* damage

**Spaceships – UFOs, player – fire function**

**Obstacle– projectile, asteroid- Raechel inherit from gameobject**

* -xPosition, yPosition
  + ID
  + -xVector, yVector
* -width, height
* -Speed
* -Direction
* -xPosition += xVector \* speed
* -yPosition += yVector \* speed
* - damage = default something
* -Health
* -scale
* +constructor, getters, setters
* + render, tick
* + isAlive() if(object.health <=0 return false)

+ getters, setters

**SpaceShip: high priority - inherits from gameobject - Jason**

* -xPosition, yPosition
* -xDirection, yDirection
  + ID
* -width, height
* -Speed
* -Direction
* -position.x += direction.x \* speed
* -position.y += direction.y \* speed
* - damage
* -Health
* -scale
* +constructor, getters, setters
* + render, tick
* + isAlive() if(object.health <=0 return false)

+ getters, setters

Assets - texture

* FireProjectile();

Inherits from GameObject:

Player, UFO, Object, Projectile, items

**Player**:  **high priority -inherits from SpaceShip - Jason**

* , KeyInput
* +constructor, getters, setters

+

**GameManager - high priority - David**

* - Linked List<GameObject>
* + render, tick
* Loop through list, if(isAlive) do render, or tick or update, or delete
* + spawn – create objects when needed - check collision, generate random number
* + CheckCollision()

**UI – Medium Priority - Kiana**

Main Menu, health bar, score/time, Button

* Main menu, in-game, death screen

**Map – low priority - Raechel**

Xpos, ypos, speed, direction, scrolling – low priority

**Art – spaceships, asteroids, ufo – high priority - Lars**

**Sound – low priority freesound.org?**

**Items/powerups**

**Effects – low priority**

* Animations
* Stationary, moveLeft, moveRight

Dividing Work

Jason: Player, UFO

Lars: Art, GameObject

David: GameManager, projectile

Kiana: Game, UI

Raechel: Asteroid, Map

Made all gameobject variables protected, added constructor

Added direction to gameobjecxt

Player should be -1, ufo should be 1 – direction to launch projectriles