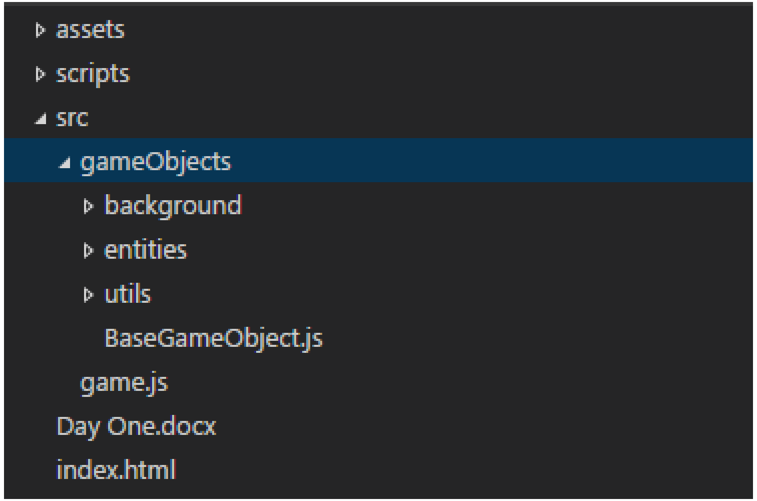
# Uvod igre uz Phaser.js

Ljetna škola programiranja

mono

# Project folder structure



```
▷ assets
▷ scripts
◢ src
  ◢ gameObjects
    ▷ background
    ▷ entities
    ▷ utils
      BaseGameObject.js
    game.js
  Day One.docx
  index.html
```

# Scrolling background

1. In game.js file add HEIGHT property.
2. Add update property to mainPhaserMethods object.
3. In BaseGameObject.js file add sprite property.
4. Add update property to BaseGameObject prototype

5. In ScrollingBackground.js file, pass correct arguments to BaseGameObject.call
6. In BaseGameObject.js file pass sprite to enable physics for that sprite.
7. In game.js file, initalize background as array object.
8. Create background instance.
9. In update method, create loop through background array and call update method for each.

10. In ScrollingBackground.js file, update method check if background position is at negative WIDTH. If so , set it to WIDTH. Try to run game now.

# Player

11. In Player.js file, add parameters to player constructor.
12. Call BaseGameObject constructor.
13. Create player prototype.
14. Make sprite body collidable with world.
15. In Update method, add checks for left and right key.
16. In game.js file, create player instance. Try running game to see player.

# Enemies

17. In BaseEnemy.js file, add check for enemy position. If sprite has passed left side of screen , set active to false.
18. Create Asteroid class, add appropriate parameters. Inherit from BaseEnemy. Set negative velocity and random rotation for Asteroid. Random rotation should be property which is set to some random number, between -1 and 1 for example.
19. In game.js file, add enemies property of array type.
20. In update method, add loop for enemies. Enemies array should loop from end of array to begining of array.

21. Update enemies in for loop. If enemies active property is set to false, destroy it's sprite with sprite.destroy() and remove enemy from array. Use array splice method to remove enemy.

22. Uncomment asteroid spawn method and asteroidSpawnTimer

23. Create random int ( whole number) between 0 and 7 ( 0 and 7 included) and use it to create asteorid sprite. Use parseInt and Math.random functions. See asteorid sprite naming in preload function. Create random y position for sprite that goes from offset to HEIGHT - offset. X position should be set to WIDTH + offset.Best number for offset would be sprite.height / 2 (since scaling is set to 0.5 for most sprites)

24. Create asteorid and push it to enemies array.
25. Uncomment collision code and add index
26. In BaseGameObject.js file, add object property to sprite and set it to this.
27. In Player.js file add health property and set it to 100
28. In Asteroid.js file add damage property and set it to some number less then 100.

29. In game.js file, playerEnemyCollision method, decrease player health on collision. Use enemy damage property.
30. If player health is less then 0, set active property to false. Destroy player sprite. Try to run game after this step. Player should collide with enemies. Try running game after this step.

# Bullets

31. In BaseBullet.js file , create BaseBullet constructor and set appropriate properties.
32. In RegularBullet.js file , call base bullet update method
33. In player.js file add bind key method to player prototype, this way we can add new keys to player.
34. Add check key down method, pass key as parameter.
35. Add getPosition method which will return sprite position

36. In game.js file create spawnRegularBullet method, which will just add new RegularBullet. Use player position for bullet sprite position and set bullet direction.
37. Bind shooting key to player, space for example.
38. Check if shooting key is pressed. If so, call spawnRegularBullet. Try running game after this step.

# Controlling bullets

39. Add COOLDOWN property to window object. Next to WIDTH and HEIGHT
40. Add cooldown property in create method and set it to 0
41. Decrease cooldown by time.physicsElapsed
42. Add check for cooldown <= 0 and reset this.cooldown to COOLDOWN inside if statement.

43. Add for loop for bullets starting from bullets end, update bullet, also add check for inactive bullets. If bullet is not active, remove it from array and destroy bullet sprite.
44. Add for loop for bullets and check for collision between bullet and enemy. Pass bulletEnemyCollision to collide call.
45. In bulletEnemyCollision method, set sprite objects active property to false
46. Finally add offset to in bullets spawn function to make bullets spawn at correct point. Try running game after this step.

Hvala ☺

Luka Erkapić: lerkapic@mono.hr

http://phaser.io/

https://github.com/luka712/PhaserDemo/tree/dayone

https://gamedevacademy.org/free-ebook-game-development-for-human-beings/?a=13

https://gamedevacademy.org/category/tutorials/html5-gamedev-phaser/