

Игровые сущности

Основные сущности:

1. Ship:

- 1.1. isPlayer bool – тип управления (игрок/AI);
- 1.2. hp int – количество здоровья;
- 1.3. speed float – скорость передвижения по экрану;
- 1.4. dethScore int – количество очков за смерть;
- 1.5. haveCoins int – количество монет при смерти(внутренние монеты);
- 1.6. havePowerUp bool – есть ли внутри павер ап;

2. Gun;

- 2.1. shotType ShotType – тип снарядов;
- 2.2. firerate float – скорострельность (вострелы/мин);

3. Shot;

- 3.1. shotType ShotType – тип снарядов;
- 3.2. damage int – наносимый урон;

4. Coin;

- 4.1. cost int – стоимость одной монеты;

5. Power Up;

- 5.1. fireRateUp float – увеличение скорострельности;

6. Score;

- 6.1. currentScore int – текущий счет;

7. Buff;

- 7.1. hpBuff int – увеличение здоровья;
- 7.2. fireRateBuff float – увеличение скорострельности;
- 7.3. damageBuff int – увеличение урона;

8. Skill;

- 8.1. recoil float – перезарядка (секунды);
- 8.2. shieldHp int – количество здоровья у щита;

Взаимодействие:

Основной класс Ship, в нем присутствует свойство isPlayer которое определяет управляет кораблем игрок или AI. К этому классу прикреплён клас Gun, который создает класс Shot с определённой частотой Gun.firerate, так же присутствует свойство shotType, которое определяет тип создаваемого класса Shot. От свойства shotType зависит параметр damage.

При соприкосновении Ship и Shot происходит влияние параметра damage класса shot на параметр hp класса Ship по формуле $\text{Ship.hp} -= \text{Shot.damage}$.

В том случае если свойство $\text{Ship.hp} \leq 0$ и $\text{Ship.isPlayer} = \text{false}$ то к свойству currentScore класса Score добавляется значение Ship.dethScore по формуле $\text{Scor.currentScore} += \text{Ship.dethScore}$. В дальнейшем есть возможность добавления дополнительного параметра difficult, благодаря которому формула меняется на данную $\text{Scor.currentScore} += \text{Ship.dethScore} * \text{difficult}$.

Так же в случае $\text{Ship.hp} \leq 0$ и $\text{Ship.isPlayer} = \text{false}$ считываются два параметра Ship.haveCoins и Ship.havePowerUp. При значении Ship.haveCoins выше нуля создаются объекты класса Coin в количестве соответствующем значению Ship.haveCoins. При соприкосновении Ship в котором свойство isPlayer = true и Coin к свойству Ship.haveCoins добавляется значение Coin.cost по формуле $\text{Ship.haveCoins} += \text{Coin.cost}$.

При положительном значении Ship.havePowerUp создаются объекты класса Power Up который имеет свойство PowerUp.fireRateUp, которое при взаимодействии с классом Ship в котором свойство isPlayer = true влияет на класс Gun к которому он привязан. В данном классе происходит изменение свойства Gun.firerate по формуле $\text{Gun.firerate} += \text{PowerUp.fireRateUp}$.

При активации класса Skill создается барьер вокруг Ship который имеет определённое количество shieldHp. При работе Skill вместо Ship.hp класс Shot взаимодействует с классом Skill.shieldHp за формулой $\text{Skill.shieldHp} -= \text{Shot.damage}$. При падении Skill.shieldHp до нуля и ниже Skill выключается и идет перезарядка на время Skill.Recoil.

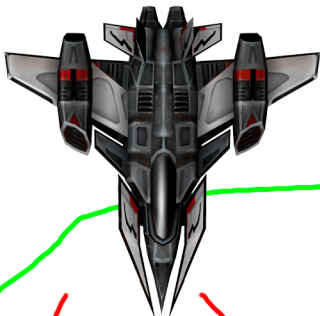
Так же присутствует класс Buff, который взаимодействует с тремя классами Ship, Gun, Shot за формулами $\text{Ship.hp} += \text{Buff.hpBuff}$, $\text{Gun.firerate} += \text{Buff.fireRateBuff}$, $\text{Shot.damage} += \text{Buff.damageBuff}$.



hpBuff 100;
fireRateBuff 4;
damageBuff 10;

00254

currentScore 254;



isPlayer false;
hp 20;
speed 2;
deathScore 100;
haveCoins 2;
havePowerUp true;

Cost 1;



fireRate +1;



shotType simple;
damage 10;

fireRate 3;
shotType simple;

isPlayer true;
hp 100;
speed 3;
deathScore 0;
haveCoins 0;
havePowerUp false;

recoil 20;
shieldHp 40;

