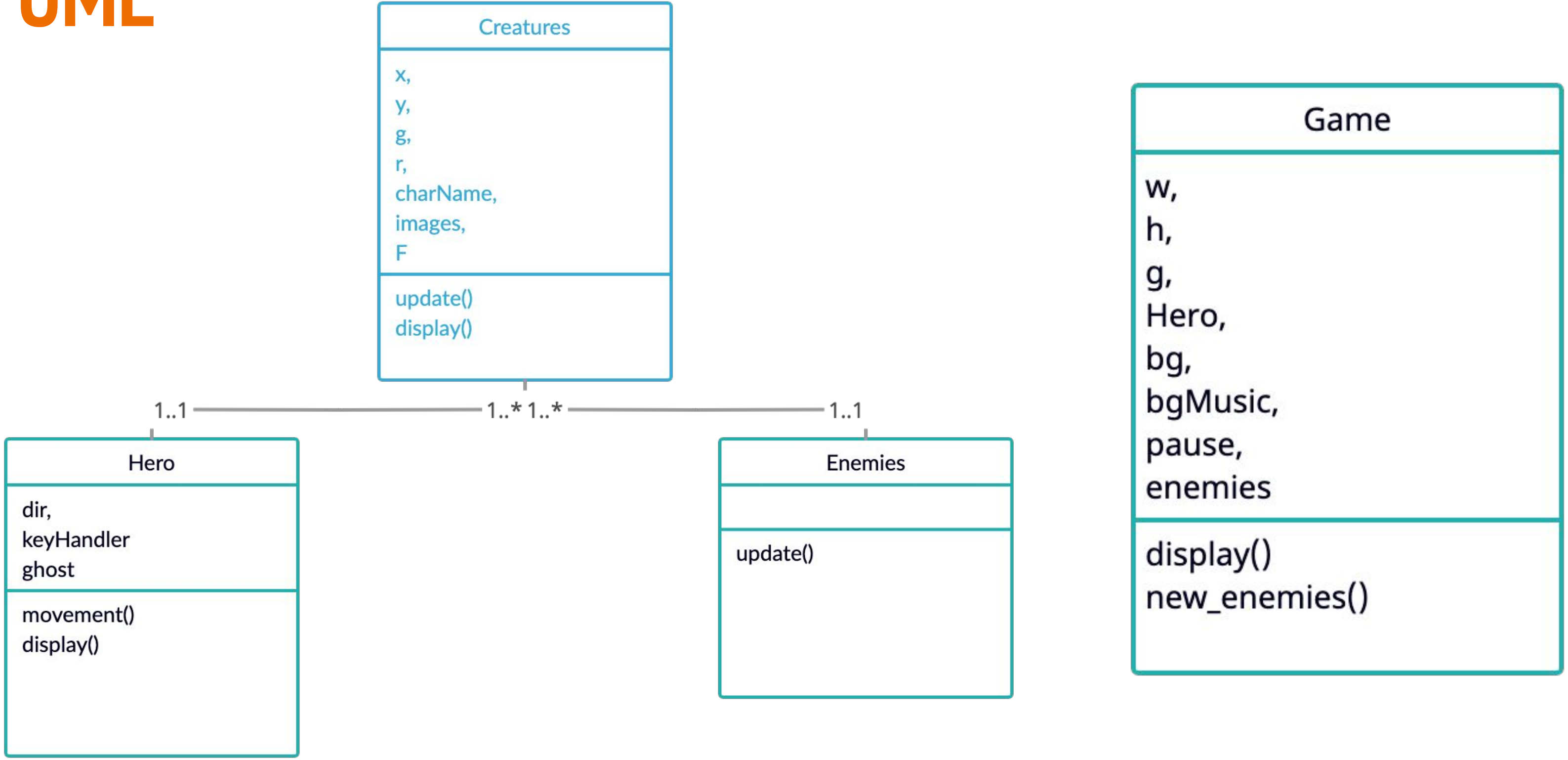
Journey Of A Medieval Knight

What is the Game About

- A knight gets spawned in a wasteland
- fights all the enemies that approaches him
- has two modes:
  - a knight mode which throws a axe and does heavy damage to enemies
  - a ghost/Mage mode which does not have a physical attack but makes the enemies take damages by poisoning them with in a certain radius of the hero (class interchangeable by pressing T).

UML



Creatures

x,
y,
g,
r,
charName,
images,
F

update()
display()

Hero

dir,
keyHandler
ghost

movement()
display()

Enemies

update()

Game

w,
h,
g,
Hero,
bg,
bgMusic,
pause,
enemies

display()
new_enemies()

Screen Shots

Hero:

Weapon:

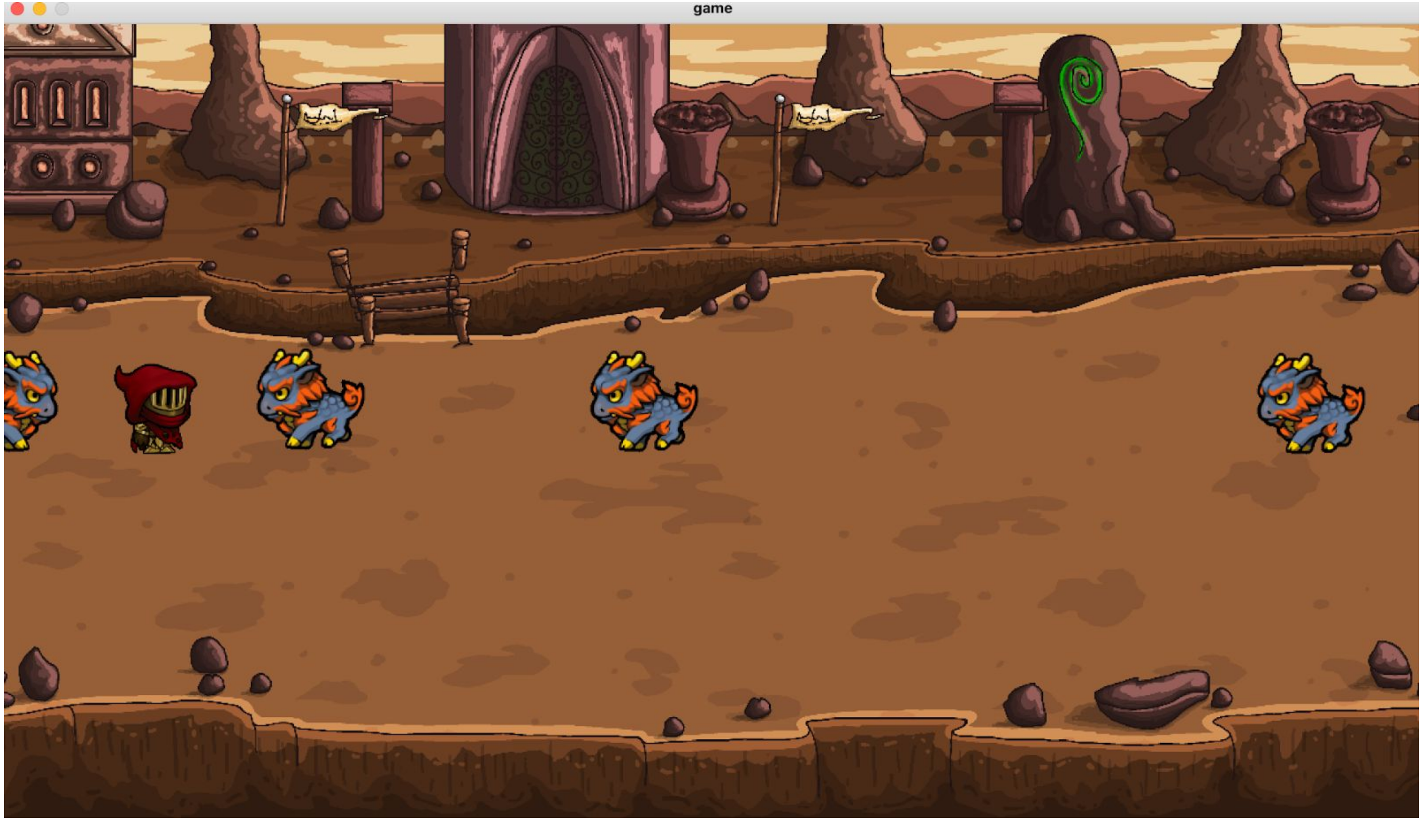
Ghost Mode:

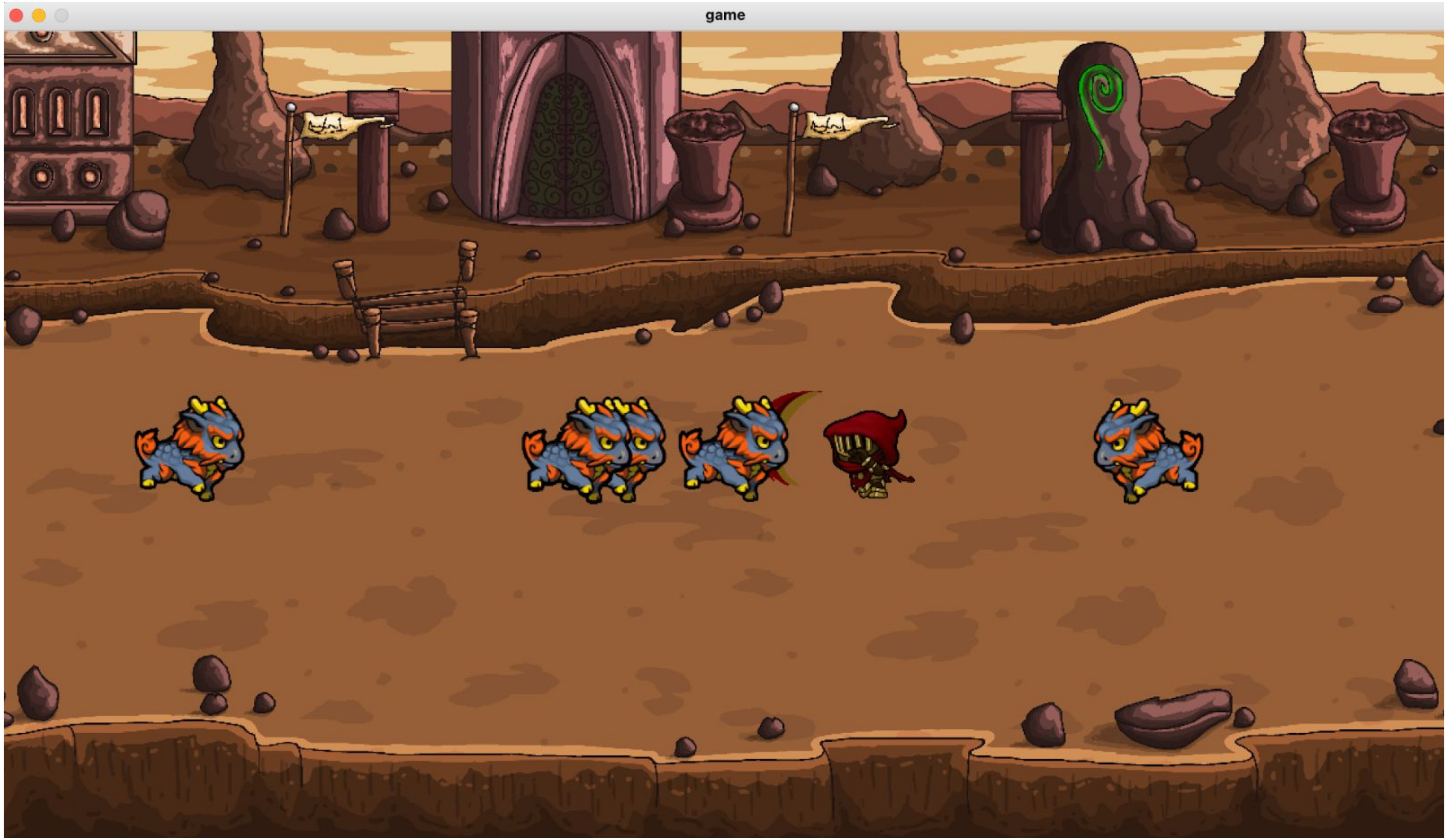
Enemies

Implemented

Yet to be Implemented

Some Screen Shots





game

Features

BG music
Hero Spawn
Hero Movement(left/right/down/up)
Hero Attack animation
Hero ghost mode
Enemies Spawn
Enemies Follow Hero
Pause Screen
Game Over
Mute
Collision Detection
Score
New Game

Hero tracking

Version 1:

```
if self.y < self.hero.y:
        self.vy = 3
elif self.y > self.hero.y:
        self.vy = -3
else:
        self.vy = 0
```

```
if self.x < self.hero.x:
        self.vx = 3
elif self.x > self.hero.x:
        self.vx = -3
else:
        self.vx = 0
```

Hero tracking

Version 2:

```
if self.y < self.hero.y and (self.hero.x-self.x)**2 + (self.hero.y-self.y)**2 < 40000:
        self.vy = 3
elif self.y > self.hero.y and (self.hero.x-self.x)**2 + (self.hero.y-self.y)**2 < 40000:
        self.vy = -3
else:
        self.vy = 0
```

Hero tracking

Version 3:

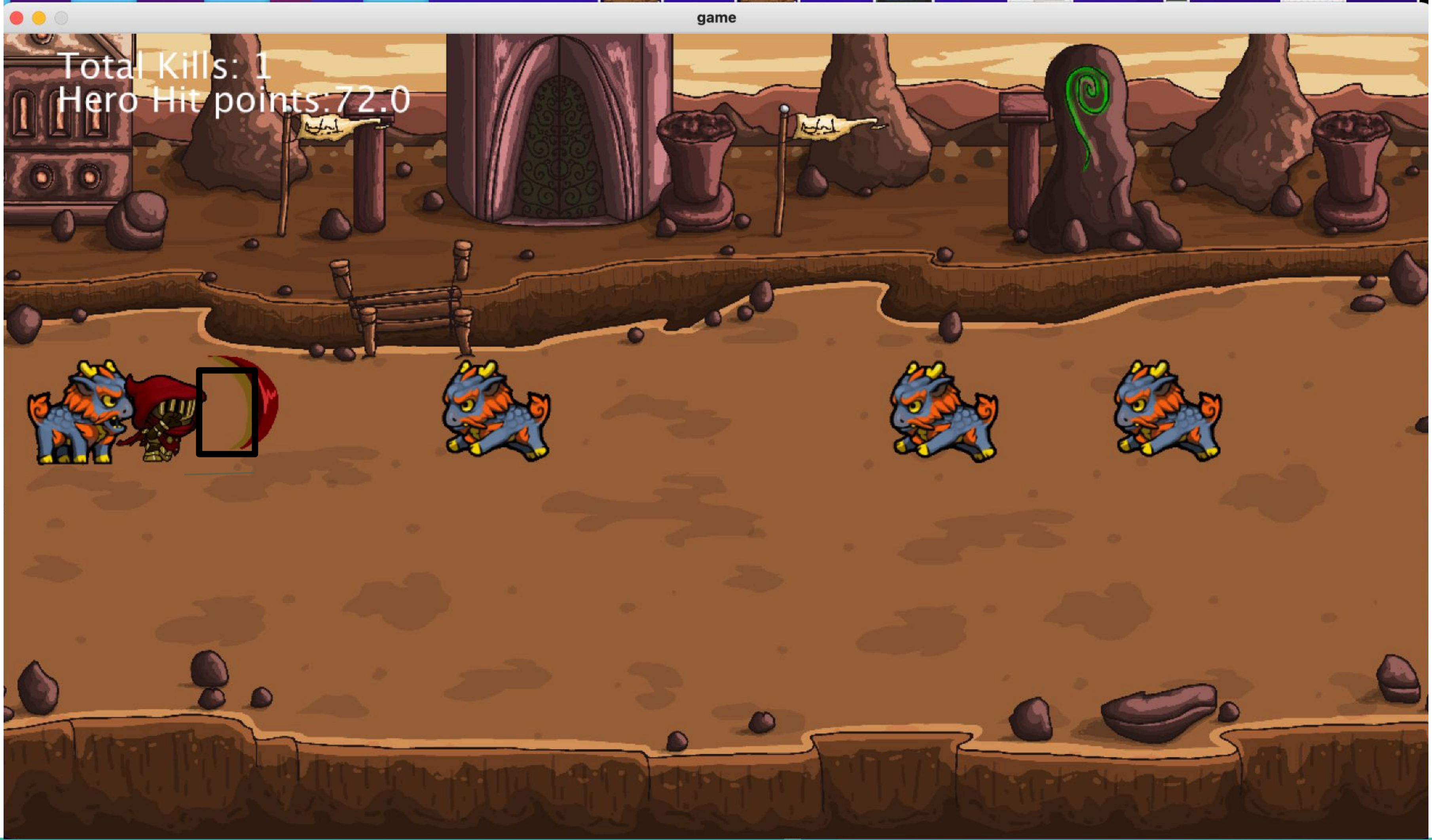
```
if self.y < self.hero.y and (self.hero.x-self.x)**2 <= 3600:
        self.vy = 3
elif self.y > self.hero.y and (self.x - self.hero.x)**2 < 3600:
        self.vy = -3
else:
        self.vy = 0
```

Display

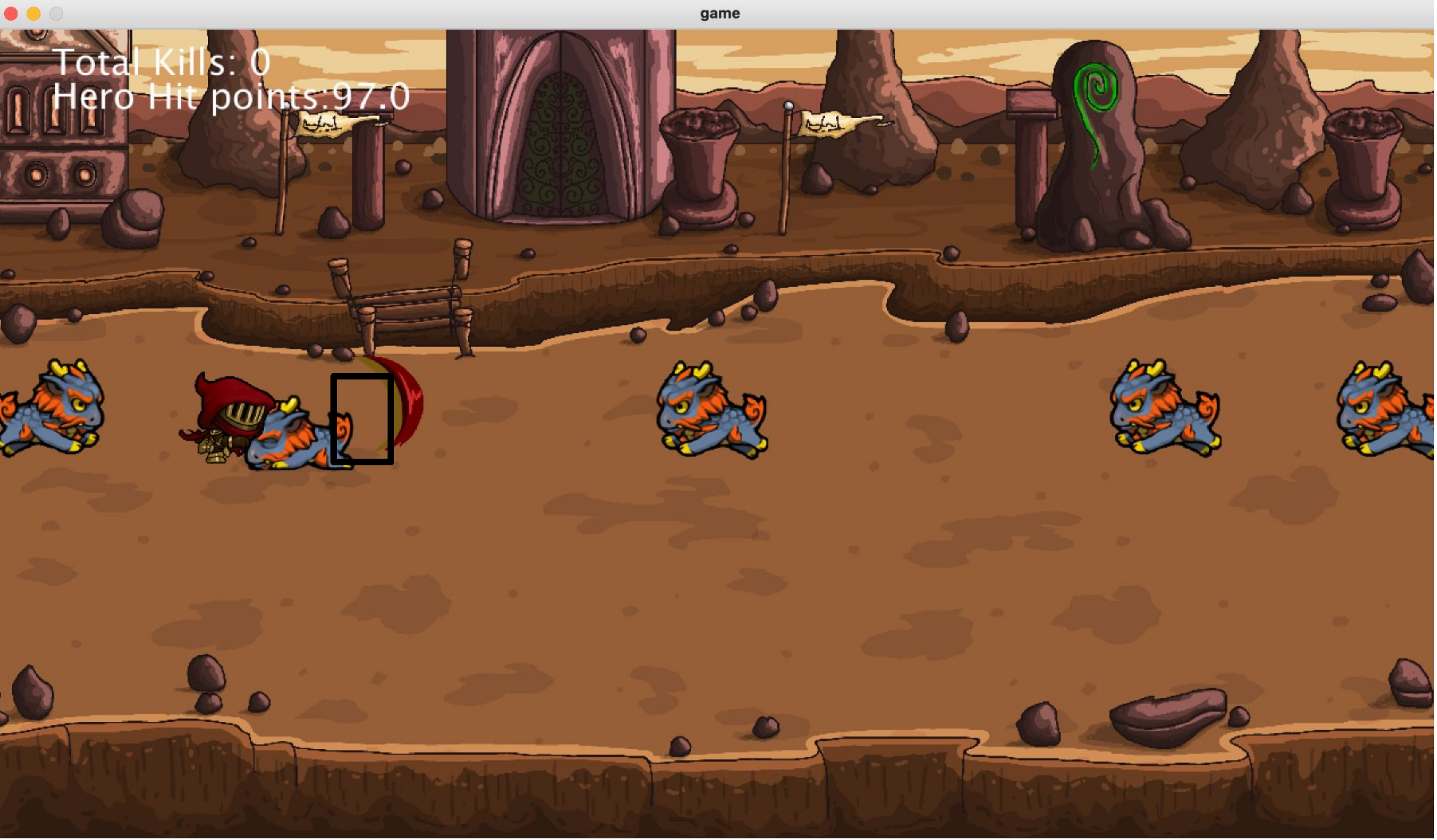
```
objects = []

objects.extend(game.enemies)

objects.append(game.Hero)

objects.sort(key=lambda x: x.y, reverse=False)
```

Collision Detection

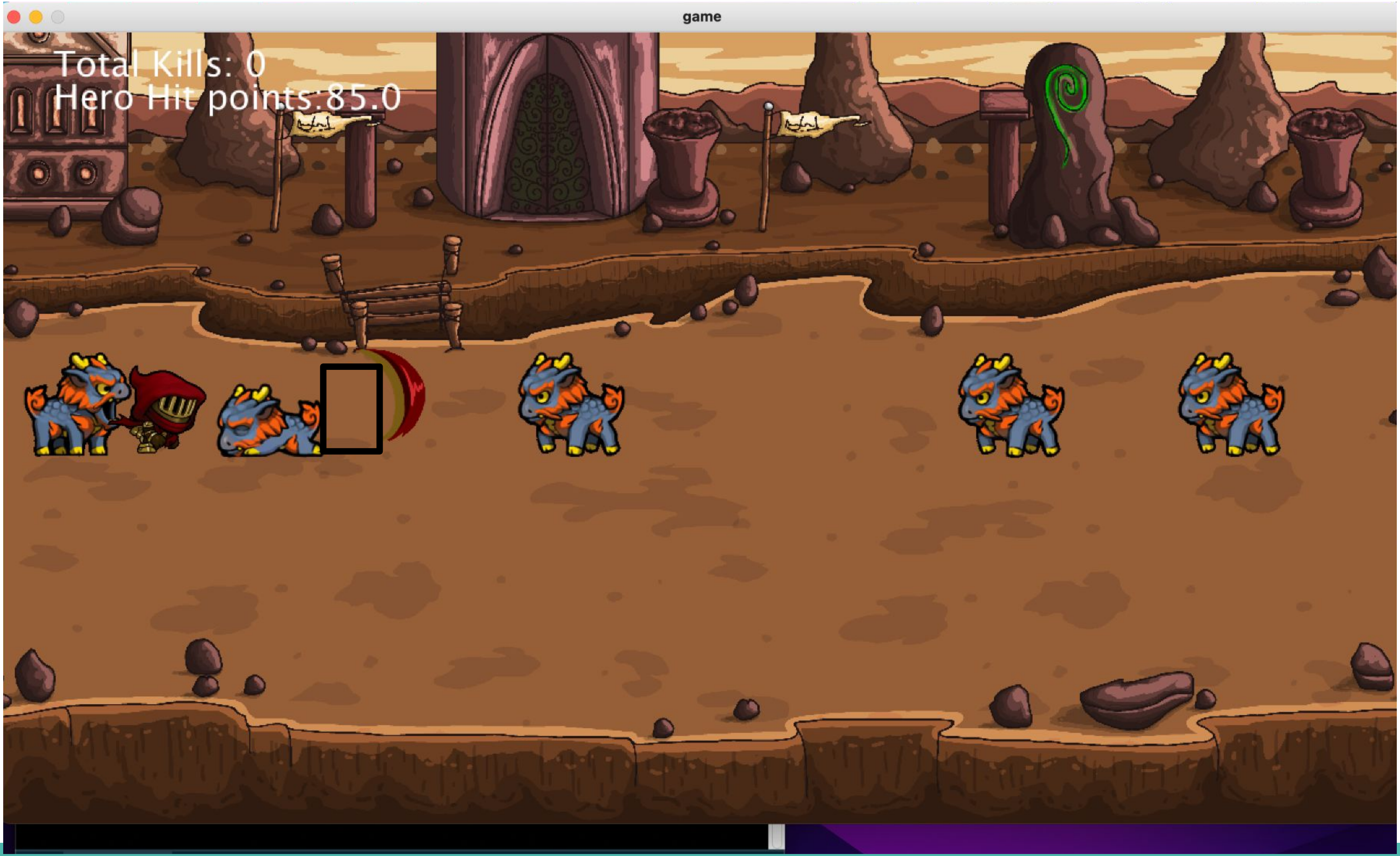
```
for e in self.enemies:
       if e.dir ==0 and e.x <= self.Hero.hit[0]-35 and e.x > self.Hero.hit[0]-100 and e.y < self.Hero.hit[1] + 35 and e.y > self.Hero.hit[1] - 35:
          e.hitpoints -= 1
       if e.dir ==1 and e.x >= self.Hero.hit[0]+35 and e.x < self.Hero.hit[0]+100 and e.y > self.Hero.hit[1] - 35 and e.y < self.Hero.hit[1] + 35:
          e.hitpoints -= 1

 if e.hero.ghost == True and (e.hero.x-e.x)**2 + (e.hero.y-e.y)**2 < 24000 and (e.hero.keyHandler[RIGHT] or e.hero.keyHandler[LEFT]or
e.hero.keyHandler[UP] or e.hero.keyHandler[DOWN]):
          e.hitpoints -=0.3
```



Total Kills: 1
Hero Hit points:72.0



Total Kills: 0
Hero Hit points:97.0



Total Kills: 0
Hero Hit points:85.0

Mostly Soft Code

```
self.animations_enemies = ['_walking', '_attacking', '_dead']

..........

self.frames_enemies = {"enemy_1": [9, 11, 1], "enemy_2": [9, 8, 1]............}

...........

enemyType = random.choice(["enemy_1", "enemy_2"...............................])

...........

self.enemies.append(Enemies(1300+200*(i)*((-1)**i), 300+i*10, self.g,10,
enemyType, self.animations_enemies, self.frames_enemies[enemyType], 0,
self.Hero, 3))
```

Features we plan to add

More enemies
Moving background
Or locking Hero
Different spawn pattern
Potions
Damage Overtime
Levels
More Weapons
Two Player Mode
increase/decrease sound rather than only Mute
Levels with flying gameplay
More frames for existing animation
Lag optimization

Thank you so much. Hope you liked it :)