Github com os arquivos base e algumas instruções <https://github.com/udacity/frontend-nanodegree-arcade-game>

Guia para iniciar: <https://docs.google.com/document/d/1v01aScPjSWCCWQLIpFqvg3-vXLH2e8_SZQKC8jNO0Dc/pub?embedded=true>

Atentar às regras do jogo: <https://review.udacity.com/#!/rubrics/15/view>

Atentar ao guia de estilo <http://udacity.github.io/frontend-nanodegree-styleguide/javascript.html>

Página da Udemy sobre o projeto: <https://classroom.udacity.com/nanodegrees/nd042/parts/078cd281-9326-4d25-8234-97b5bf1486ef/modules/269645859775463/lessons/2696458597239847/concepts/bf9f084e-9a20-4d40-a69c-9c180748275b>

**Fazer tudo isso com orientação a objeto -> pensar sempre se o que está fazendo é pra ter uma nova função/classe e na outra função chamar esta!**

**REVER TODAS AS VARIÁVEIS DECLARADAS COMO VAR**

**REVER ORIENTAÇÃO A OBJETO => Scope, closure, encapsulamento, this, etc.**

**Checar todos os TODOs antes de terminar o projeto!**

Distâncias jogador:

//distância lados direito e esquerdo: 17px;

//distância baixo até sombra: 23

//distância baixo até o jogador: 32

//The player's dimensions are 67x84 with the shadow below the character, but the char-boy is 67x75

Distâncias inimigo:

//The enemy's dimensions are 101x69

//\*\*It works without circular reference, testes with the values above. The values below will

//be keeped only for reference if decided to change for some reason (maybe if the player's move become more fluid or something like that)

//Enemy's dimmensions are about an ellipsis with 70px in the Y axis and 100px in the X axis

//Player's dimmensions are about an ellipsis with 72px in the Y axis and 64px in the X axis

Pendencias do projeto:

1. **Fazer a função inimigo**
   1. Set the enemy initial location; -> **Revisar ao final, parece que não é exatamente 101 o tamanho pq a linha de baixo tem também parte que não é grama -> No eixo x parece estar certo, pois não tem esse ‘excedente’**
   2. Set the enemy speed
2. **Fazer a função Update**
   1. Update the Enemy location
   2. Handle collision with the Player
3. **Fazer a função player**
   1. Carregar a imagem do player
   2. Setar a posição inicial do player **Revisar ao final, parece que não é exatamente 101 o tamanho pq a linha de baixo tem também parte que não é grama -> No eixo x parece estar certo, pois não tem esse ‘excedente’**
   3. Criar a classe update() **Do jeito que fiz não precisa do update, ver se está certo**
   4. Criar a classe render()
   5. Criar a classe handleInput()
4. **Instanciar**
   1. O jogador
   2. Os inimigos (array)
5. Funções de fim de jogo
   1. Analisar onde cada função deve ficar, se no app Player, app Enemy, ou no Engine sendo chamado direto pelo Main durante o loop (lembrar que main roda a cada x milissegundos).
   2. Ver se era interessante fazer uma tela de fim de jogo, e se coloca um botão de reset ou se faz automático => talvez no winGame, rodar reset e colocar um alerta ou texto de parabéns vc ganhou o jogo
   3. Criar a função de winGame (está em Player, ver se é o melhor lugar)
   4. Criar a função reset (está em engine de padrão – deixar lá)

1. Ajustes finais
   1. Verificar todos os TODOS que ainda existem
   2. Testes finais:
      1. Ver se ficou legal o início e final do jogo
      2. Reiniciar, ver se não ficou elemento sobrando dos jogos anteriores
   3. Checar orientação a objetos, ver onde é necessário ter globais, e onde pode ter locais (onde pode ter encapsulamento -> onde a função é chamada só pelo mesmo objeto e onde é chamado por outros objetos) – vou deixar sem arrumar e enviar, esperar a resposta pra ver o que dá pra fazer
   4. Checar comentários