

Gabriel Ligoski — Eduardo Xavier Dantas — Rodrigo Mamédio Arrelaro

GLEX: Um acrônimo para nada

Tema Proposto

Nossa proposta é solucionar um problema que é o ócio ruim, criando uma recreação, por meio de uma aplicação utilizando arte pixelada. Em outras palavras um Jogo eletrônico.

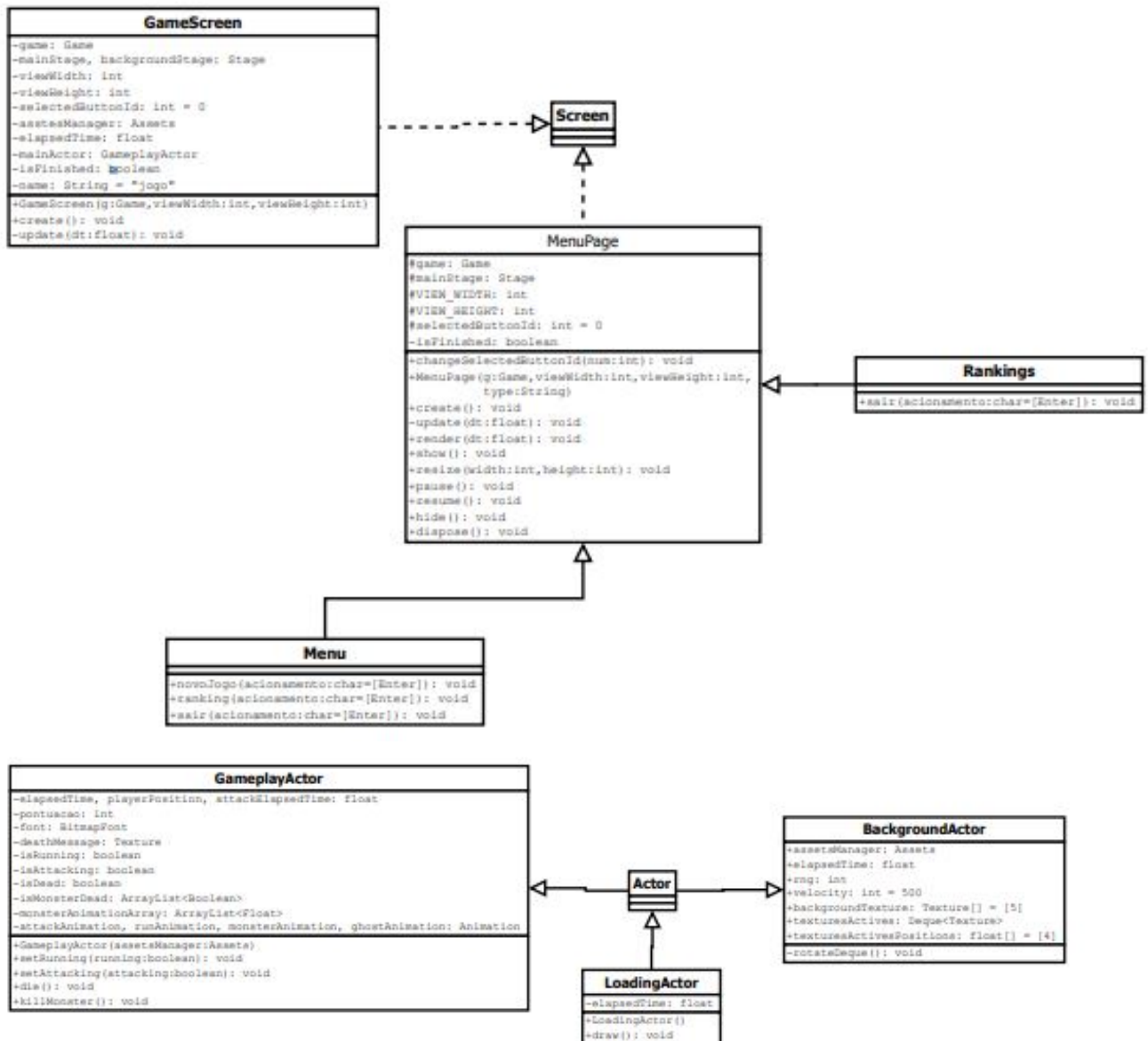
Descrição do problema

Durante a pandemia a saúde mental dos indivíduos teve uma queda significativa, que ocasionou na formação de um ócio negativo, sendo este o tempo livre que as pessoas não tinham e agora tem, é negativo em vista do aumento do tédio relatado em redes sociais. Por isso decidimos criar uma forma de ocupar este ócio de forma criativa e inovadora.

Regras de Negócio

1. Deve ser iniciado com um menu com as opções: "Novo jogo", "Rankings" e "Sair".
2. O jogo é localizado dentro dos corredores de um castelo abandonado, com a aparição de inimigos sendo realizada de forma aleatória.
3. Arte do jogo como: herói, chefão e inimigos deve ser feita utilizando artes pixeladas.
4. Caso o usuário toque no chefão ou em um inimigo, ele deve perder o jogo e voltar ao menu inicial.
5. existem 3 tipos de personagens como, jogador, inimigo, e chefe.
 - (a) O jogador deve ser capaz de executar as seguintes ações:
 - i. mover utilizando o [SPACE] (para acelerar ou desacelerar).
 - ii. atacar utilizando [CTRL].
 - (b) Chefe deve ser um tipo de Inimigo único, diferenciado dos demais, podendo ser atribuído comportamentos únicos.
6. Ao eliminar o Chefe o jogo é finalizado e uma tela de Vitória deve ser mostrada, com opção de voltar para o menu.

Diagrama UML



Funções auxiliares do programa (nem todos os métodos foram declarados)

Assets
+manager: AssetManager
+load(type:String): void
-loadMenu(): void
-loadRankings(): void

RenderFunctions
+<<static>> drawInPlace(batch:Batch,texture:Texture, xPos:float,yPos:float): void

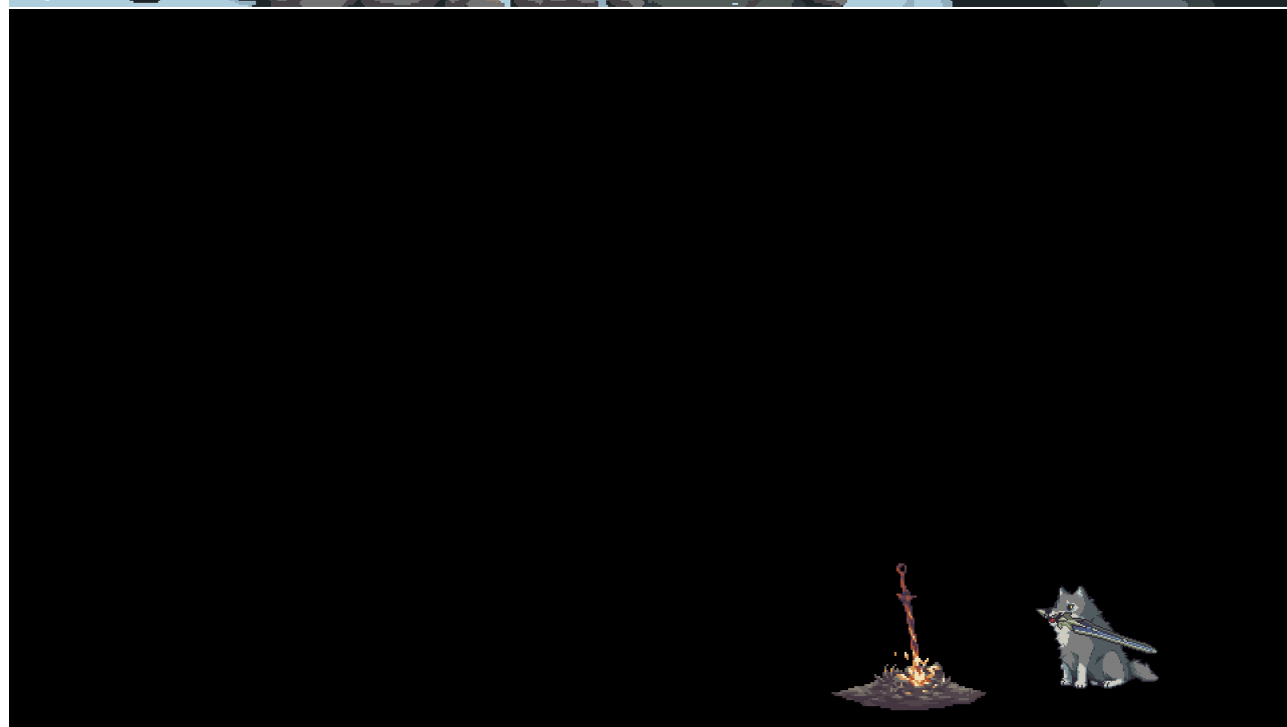
AnimationFunctions
+<<static>> png2Animation(path:String,width:int, height:int,padding:int, frameNumber:int, fps:float,offset:int): Animation
+<<static>> animateInPlace(batch:Batch,animation:Animation, elapsedTime:float, xPos:float,yPos:float, looping:float): void
+<<static>> drawInPlace(batch:Batch,animation:Animation, elapsedTime:float, xPos:float,yPos:float): void

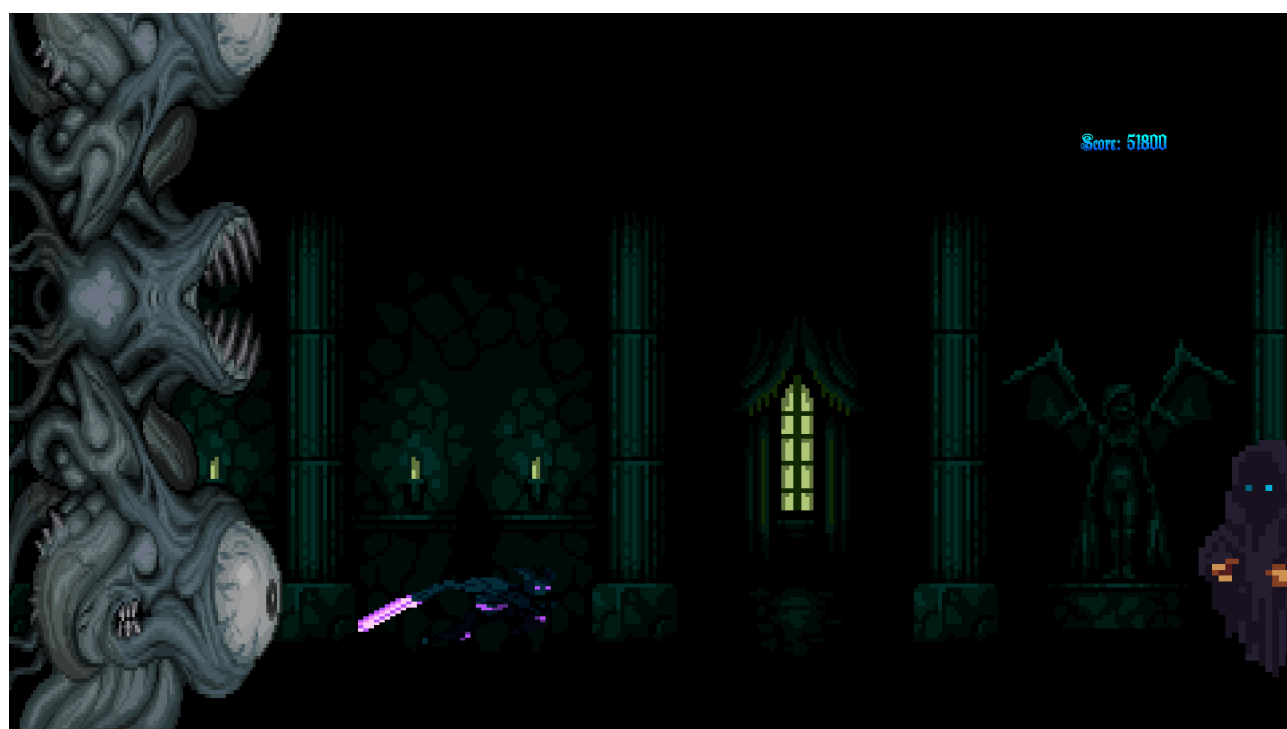


GameplayFunctions
+<<static>> spawnMonster(getMonstersPositions:ArrayList<Float>, monsterAnimationArray:ArrayList<Float>, isMonsterDead:ArrayList<Boolean>): void
+<<static>> drawMonsters(actor:GameplayActor, batch:Batch,monsterAnimation:Animation): void

GifDecoder
+STATUS_OK: int = 0
+STATUS_FORMAT_ERROR: int = 1
+STATUS_OPEN_ERROR: int = 2
#MAX_STACK_SIZE: int = 4096
#in: InputStream
#status: int
#width: int
#height: int
#gctFlag: boolean
#gctSize: int
#loopCount: int = 1
#gct: int[]
#lct: int[]
#act: int[]
#bgIndex: int
#bgColor: int
#lastBgColor: int
#pixelAspect: int
#lctFlag: boolean
#interlace: boolean
#lctSize: int
#ix, iy, iw, ih: int
#lrx, lry, lrw, lrh: int
#image: DixieMap
#lastPmap: DixieMap
#block: byte[]
#blockSize: int = 0
#dispose: int = 0
#lastDispose: int = 0
#transparency: boolean = false
#delay: int = 0
#transIndex: int
#prefix: short[]
#suffix: byte[]
#pixelStack: byte[]
#pixels: byte[]
#frames: Vector<GifFrame>
#frameCount: int

Capturas de tela do projeto:





Você Morreu

