



FACULDADE DE TECNOLOGIA DE AMERICANA
Curso Superior de Tecnologia em Jogos Digitais

Ettore Ferreira Bera Fagundes de Lima
Pedro Henrique Fernandes Sanches

Jizlost

Americana, SP
2017



FACULDADE DE TECNOLOGIA DE AMERICANA
Curso Superior de Tecnologia em Jogos Digitais

Ettore Ferreira Bera Fagundes de Lima
Pedro Henrique Fernandes Sanches

Jizlost

Relatório técnico desenvolvido em cumprimento à exigência curricular do Curso Superior de Tecnologia em Jogos Digitais sob a orientação do Prof. Dr. Kleber de Oliveira Andrade

Americana, SP.

2017

FICHA CATALOGRÁFICA – Biblioteca Fatec Americana - CEETEPS
Dados Internacionais de Catalogação-na-fonte

S191j SANCHES, Pedro Henrique Fernandes

Jizlost. / Pedro Henrique Fernandes Sanches ; Ettore Ferreira Bera Fagundes de Lima. – Americana, 2017.

21f.

Monografia (Curso de Tecnologia em Jogos Digitais) - - Faculdade de Tecnologia de Americana – Centro Estadual de Educação Tecnológica Paula Souza

Orientador: Prof. Dr. Kleber de Oliveira Andrade

1 Jogos eletrônicos I. ANDRADE, Kleber de Oliveira II. Centro Estadual de Educação Tecnológica Paula Souza – Faculdade de Tecnologia de Americana

CDU: 681.6

Ettore Ferreira Bera Fagundes de Lima

Pedro Henrique Fernandes Sanches

JIZLOST

Trabalho de graduação apresentado
como exigência parcial para obtenção do
título de Tecnólogo em Jogos Digitais,
pelo CEETEPS/Faculdade de Tecnologia
– Fatec/ Americana.
Área de concentração: Jogos Digitais

Americana, 15 de dezembro de 2017.

Banca Examinadora:



Kléber de Oliveira Andrade (Presidente)
Doutor
Fatec Americana



Benedito Aparecido Cruz (Membro)
Mestre
Fatec Americana



José Mário Frasson Scafi (Membro)
Mestre
Fatec Americana

RESUMO

Para o desenvolvimento em 6 meses, com 3 etapas, o objetivo do grupo foi fazer um jogo simples como de naves, porém desafiar adicionando nele as decisões e administração das peças de Tetris em vez de tiros normais. O jogo criado dessa foi o Jizlost. O jogo evoluiu bem desde sua ideia inicial, tendo 4 tipos diferentes de inimigos, dois modos diferentes de atirar e traz uma experiência divertida e desafiadora exigindo dos jogadores tomada de decisões rápidas e também planejadas para conseguir obter altos placares.

Palavras Chave: rapidez, pensar, desafio, diversão, tetris, canvas, javascript

ABSTRACT

For a development in 6 months, with 3 steps, the objective of the group was to make a simple, yet challenging spaceship game add to it the piece management of Tetris instead of normal bullets. The game created from this idea was Jizlost. The game grown well since it's initial idea, having 4 different types of enemies, 2 different ways to shot and bringing an fun and challenging experience that forces players to make quick and planned decisions in order to obtain high scores.

Keywords: quick, think, challenge, fun, tetris, canvas, javascript

SUMÁRIO

1	INTRODUÇÃO	6
2	METODOLOGIA	7
2.1	ALFA.....	7
2.2	BETA	9
2.3	FINAL	11
3	IMPLEMENTAÇÃO	13
4	RESULTADOS	16
4.1	ALFA.....	16
4.2	BETA	16
4.3	FINAL	16
5	CONSIDERAÇÕES FINAIS	18
6	REFERÊNCIAS BIBLIOGRÁFICAS	19
7	APÊNDICE 1 – CÓDIGO FONTE	20

LISTA DE FIGURAS

Figura 1 - Peças do jogo	7
Figura 2 - Peças sobressalentes	8
Figura 3 - Pré-visualização.....	9
Figura 4 - Inimigo Troll	11
Figura 5 - Pontuação saltando	11
Figura 6 - Telas	13
Figura 7 - Colisão	14

LISTA DE TABELAS

Tabela 1 - Efeitos Sonoros	12
----------------------------------	----

1 INTRODUÇÃO

Em 1984, quando Alexey Pajitnov foi designado a escrever jogos para testar a capacidade de um novo hardware, o mesmo não imaginava o tamanho do sucesso que procederia a sua maior criação: o Tetris. Baseado em um jogo chamado Pentomino, o Tetris conta com sete peças formada por quatro blocos cada, onde o seu objetivo é encaixá-las em uma linha para poder eliminar essa linha e conquistar pontos.

De 1984 para 2017, tetris acumulou algumas conquistas surpreendentes como vender mais de 35 milhões de cópias para o Game Boy, mais de 500 bilhões de downloads em plataformas móveis e quase 23 milhões de partidas de Tetris já foram jogadas no Facebook.

Com essa base, decidiu-se criar o Jizlost como uma inspiração em Tetris, já que este é um jogo muito bem solidificado na sociedade e sem muitas explicações, fica-se óbvio o objetivo do nosso jogo.

O objetivo deste trabalho é desenvolver um jogo de fácil compreensão, com uma jogabilidade simples e manter o desafio sempre em um alto nível. O jogo une elementos de puzzle com peças de Tetris que requer planejamento sobre posição de peças e a rapidez de jogos de naves espaciais com foco em derrotar inimigos o mais rápido possível.

Com esses três pilares, o jogo consegue se sustentar para poder alcançar desde a criança que está aprendendo a usar um computador, como um adulto que tem a alcançar a maior pontuação possível.

Este trabalho possui três capítulos sobre o período de desenvolvimento do jogo à qual o primeiro trata-se da metodologia nas três etapas de desenvolvimento do jogo, o segundo capítulo trata-se da implementação e o terceiro os resultados obtidos durante as etapas de desenvolvimento.

2 METODOLOGIA

O projeto foi idealizado para ser desenvolvido em seis meses, então foi dividido em três etapas de dois meses cada. A primeira etapa foi o desenvolvimento da versão alfa do jogo, a segunda etapa o desenvolvimento da versão beta do jogo e a etapa final foi de encerramento do projeto com o mesmo pronto para publicação.

Os três membros do grupo para desenvolver esse projeto foram os dois autores, Ettore Ferreira Bera Fagundes de Lima e Pedro Henrique Fernandes Sanches, e um outro colega de classe de faculdade, Henrique Lorenzi.

O grupo decidiu por desenvolver um jogo simples de nave, com a mecânica principal em que a munição do personagem eram peças do jogo de Tetris (cada peça sendo formada por 4 blocos individuais), abrangendo assim, um público casual, mas também dedicado a obter uma pontuação alta. A ideia foi de criar um jogo simples como os de Atari, porém com a dinâmica oposta entre agir rapidamente atirando em inimigos e pensar o posicionamento de peças de tetris para desafiar os jogadores. O nome dado ao jogo foi Jizlost que foi criado a partir das peças de tetris como ilustra a figura 1.

Figura 1 - Peças do jogo



Fonte: Próprio autor

2.1 Versão Alpha do jogo

O projeto foi feito em HTML5 Canvas e JavaScript devido a facilidade de se codificar e experiência do grupo com o mesmo em outras aulas da faculdade. O Canvas permite desenhar os quadros do jogo repetidamente e o JavaScript facilita a programação.

Na fase alfa de desenvolvimento o maior foco foi garantir a base de mecânicas do jogo funcionais e um design básico para ser melhor trabalhado futuramente.

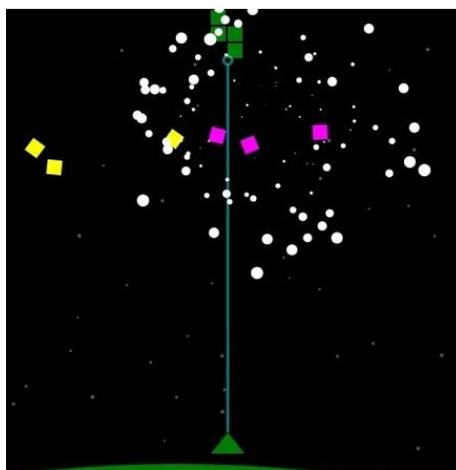
O design inicial da fase foi de ter um fundo preto e adicionado estrelas para representar o espaço, o personagem do jogador que é uma nave verde (representada por um triângulo) e uma semicircunferência no canto inferior da tela representando um planeta.

A primeira mecânica adicionada foi a de movimentação do jogador (para os lados direito e esquerdo usando suas respectivas teclas do teclado). Depois, foi implementada a habilidade de atirar as peças de Tetris. Então, adicionou-se um inimigo cinza com espaço aberto onde as peças encaixariam.

Após isso, foi necessário adicionar a colisão nos inimigos (para poderem causar dano), o planeta (para reconhecimento dos potenciais danos que receberia) e das peças atiradas (para inserir as mesmas no corpo dos inimigos). Após ter colisão completa, foram necessárias funções para completar linhas nos inimigos e logo em seguida destruir o mesmo caso as linhas completadas igualem o valor de vida restante.

O grupo decidiu então que, os blocos restantes (além dos necessários para derrotar o inimigo e ou que estiverem para fora do espaço do corpo do inimigo) cairiam no planeta causando um ponto de dano como se observa na Figura 2. Após isso, decidimos criar uma lista ao lado direito da tela, para pode mostrar ao jogador a peça atual e as três próximas peças.

Figura 2 - Peças sobressalentes



Fonte: Próprio autor

Para os inimigos, toda vez que um deles é derrotado um outro aparece depois de um curto tempo. Os inimigos seguem uma velocidade constante para descerem no eixo y até chegarem e colidirem com o planeta. Foi adicionado a tela de *game over* quando a vida do planeta chega a zero junto com um placar com a quantidade de pontos total do jogador. Além disso, para o jogador não ter apenas ônus durante o jogo, adicionamos que cada inimigo derrotado sem sobrar blocos, recuperava vida do planeta igual a quantidade de blocos usadas para derrotá-lo.

2.2 Versão Beta do jogo

A etapa Beta foi focada em adicionar mais diversidade ao jogo. Foi mapeado mais um botão para a rotação das peças e também adicionado uma pré-visualização de onde a peça será lançada (Figura 3), oferecendo assim melhores meios de fazer com que o jogador encaixe perfeitamente cada peça.

Figura 3 - Pré-visualização



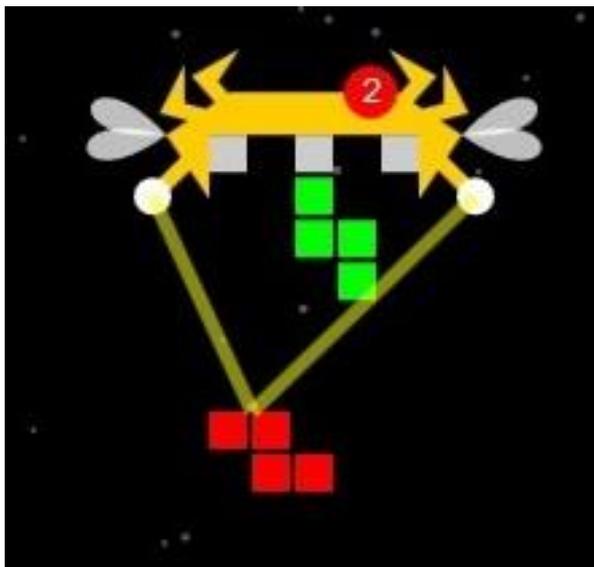
Fonte: Próprio autor

Diferentes pontuações foram criadas para cada linha extra que foi feita ao derrotar um inimigo:

- **Pontuação *Awesome* (incrível):** mil e duzentos pontos recebidos ao derrotar inimigos no estado invulnerável de *power up*.
- **Pontuação *Nice* (legal):** um mil pontos são recebidos ao derrotar inimigos fazendo duas ou mais linhas com uma peça sendo um valor igual à vida restante do inimigo.
- **Pontuação *Excellent* (excelente):** um mil e quinhentos pontos são recebidos ao derrotar inimigos fazendo duas ou mais linhas com uma peça sendo um valor maior à vida restante do inimigo.
- **Pontuação *Perfect* (perfeito):** três mil pontos recebidos ao derrotar inimigos fazendo duas ou mais linhas com uma peça sendo um valor igual à vida restante do inimigo e sem deixar nenhum bloco restante no inimigo.
- **Pontuação *Ultra-Perfect* (ultra perfeito):** quatro mil e quinhentos pontos recebidos ao derrotar inimigos fazendo duas ou mais linhas com uma peça sendo um valor maior à vida restante do inimigo e sem deixar nenhum bloco restante no inimigo.

Alterações na estrutura do código foram necessárias para facilitar implementação de mais inimigos, então criou-se três novos inimigos para fazer o jogador prestar mais atenção durante o jogo: o inimigo troll (Figura 4) na cor amarela e com duas antenas que soltam um raio amarelo em peças atiradas no mesmo rotacionando-as, o inimigo bônusde cor laranja com três blocos de tamanho com uma estrela, que ao derrotar, altera temporariamente o tiro do jogador para tiros rápidos de um bloco e cria mais inimigos que o normal (criando um estado de invulnerabilidade) e por fim, um terceiro inimigo curade cor rosa com três blocos de tamanho com um símbolo de uma cruz que, ao derrotar sem peças restantes, recupera cinquenta pontos de vida do planeta. Foi então removida a mecânica de ganhar vida da fase alfa já que esta estava muito fácil para os jogadores recuperarem vida.

Figura 4 - Inimigo Troll



Fonte: Próprio autor

Recentemente foi adicionado um reconhecimento de *input* de controles de Xbox 360, um menu para começar o jogo e a tela de como jogar.

2.3 Versão Release do jogo

A etapa final foi de focar no polimento do jogo. Foi adicionado pernas e asas para os inimigos parecerem mais como aliens. Depois criou-se a animação de morte dos inimigos e do planeta com o *game over*. Adicionou-se o efeito do contador de pontuação saltar com as diferentes quantidades de pontos conforme a Figura 5.

Figura 5 - Pontuação saltando



Fonte: Próprio autor

Criamos também a pontuação *Awesome* para inimigos derrotados durante a invulnerabilidade e a pontuação *Ultra-Perfect* para quando inimigos são derrotados sem sobrar blocos de peça sobrando no inimigo e tendo feito mais linhas do que o total de vida do inimigo.

Por fim adicionamos os efeitos sonoros ao jogocomo segue a Tabela 1.

Tabela 1 - Efeitos Sonoros

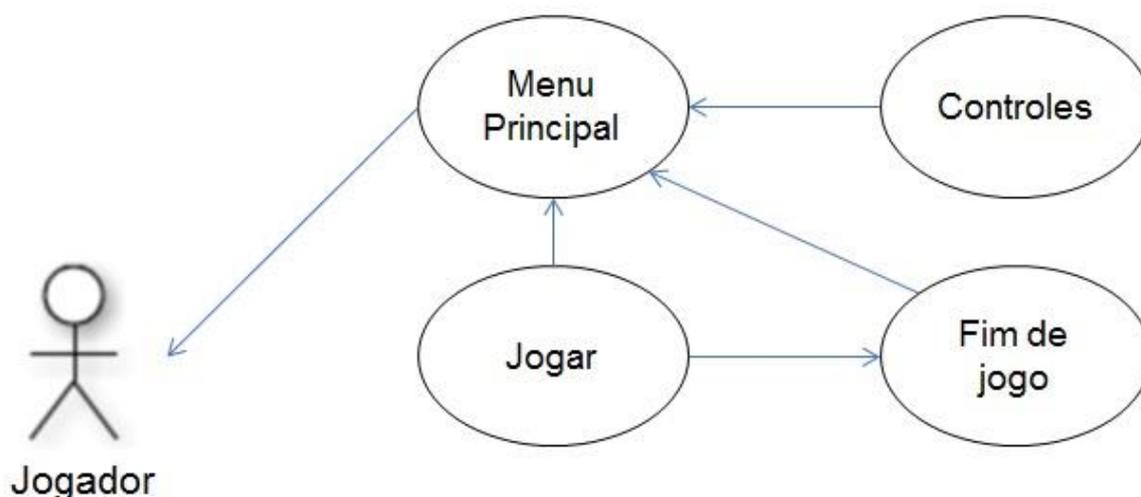
Nome do Arquivo	Função
mscGame	Música tocada durante o jogo
mscPowerup	Tocado durante o período de <i>power up</i>
mscMenu	Transição do menu para o jogo
mscGameOver	Transição do jogo para o <i>game over</i>
sndPieceBulletShoot	Som do tiro da peça
sndPieceBulletRotate	Som da peça sendo rodada
sndPieceBulletImpact	Som da peça atingindo o inimigo
sndLineClear	Usado quando o jogador completa uma linha
sndLaser	Usado pelo inimigo trol quando rotaciona as peças
sndPlanetDamage	Usado para o dano que o planeta recebe
sndEnemyDeath	Som do inimigo ao ser destruído pelo jogador
sndEnemyExplosion	Som do inimigo atingindo o planeta
SndPerfect	Bonus <i>perfect</i>
sndExcellent	Bonus <i>excellent</i>
sndNice	Bonus <i>nice</i>
sndUltraPerfect	Bonus <i>ultra perfect</i>
sndAwesome	Bonus <i>awesome</i>
sndPowerup	Transição para a tela de <i>power up</i>
sndHealthup	Vida sendo regenerada
sndNoBonus	Jogador não consegue o bonus

3 IMPLEMENTAÇÃO

O jogo foi desenhado no Canvas do HTML5 com loop constante e cada um destes chamando a função de *update* e *draw*. O Javascript *main* possui várias funções de auxílio para lógica e desenho, como por exemplo: *handlersKeyDown* e *handlerKeyUp* para botões apertados; *SoundPlay*, *SoundSet* e *SoundStop* para os efeitos sonoros; *Approach* para movimentação de objetos e cálculo de distância.

O jogo possui quatro telas em que são possíveis serem navegadas entre elas com diferentes opções: menu principal; controles do jogo e como jogar; jogo executado e fim de jogo (Figura 6).

Figura 6 - Telas

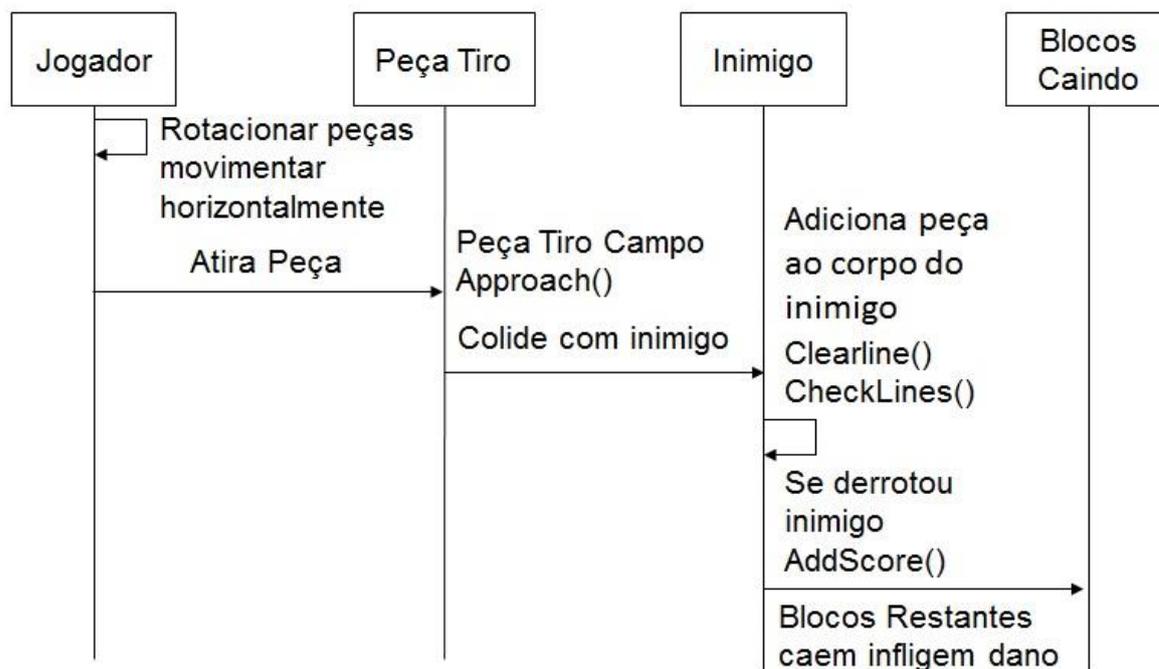


Fonte: Próprio autor

O jogador consegue mover o personagem horizontalmente pelas setas direcionais do teclado e atirar peças de tetris com a tecla espaço. Pressionando a tecla R ou a tecla para cima, a peça atual é rotacionada.

As peças possuem colisão com os inimigos e as peças deles (peça-campo). Caso colida com um inimigo, a peça do jogador torna-se parte do inimigo: preenchendo seu corpo de acordo com o formato e blocos da peça (blocos de peças que fiquem fora são quebrados e caem). Quando uma peça chega no limite vertical do Canvas, ela quebra em blocos e os mesmos caem infligindo dano, vide Figura 7.

Figura 7 - Colisão



Fonte: Próprio autor

Inimigos normais são gerados no topo da tela e vão se aproximando do planeta na parte de baixo do canvas de acordo com a dificuldade. Os inimigos são destruídos do canvas de duas formas: a primeira é quando são completadas as linhas exatas ou superiores ao número de vida do mesmo, ao qual o jogador é recompensado com pontos e a segunda é quando a posição do inimigo alcança o planeta para colisão, onde o inimigo é destruído e a vida do planeta é reduzida por cinquenta pontos acrescidos em um ponto para cada bloco restante no inimigo.

Os inimigos bônus funcionam da mesma maneira que normais, porém eles se movimentam para outra posição x e y no canvas depois de determinado período de tempo. Esse processo sempre é repetido duas vezes, mas na terceira repetição, eles movimentam-se verticalmente ao topo do canvas e são descarregados.

O inimigo troll funciona majoritariamente igual ao inimigo normal, contudo ele rotaciona uma vez no sentido horário as peças que forem atiradas no seu campo de espaço horizontal x.

A pontuação do jogo é gerada a cada vez que uma linha é completada em um inimigo. Por completar somente a linha pedida pelo inimigo o jogador recebe mil menos o número de blocos restantes no inimigo multiplicado por dez ($1000 - \text{blocksRemaining} * 10$) pontos, porém pontuações especiais podem ser recebidas

caso o jogador consiga também derrotar o inimigo ao completar a linha e seguir as condições já citadas em Metodologia.

4 RESULTADOS

4.1 ALFA

O primeiro *feedback* foi dos colegas de sala e alguns amigos próximos (totalizando 20 pessoas). Concluiu-se que as ideias e mecânicas do jogo eram boas, porém a maioria jogadores demoraram para entender os controles por irem diretamente à opção de “começar o jogo” e não em “como jogar”. Além disso, os testes apontaram que o jogo estava entediante e com um baixo desafio.

4.2 BETA

Por possuir mais inimigos diferentes, criou-se mais diversidade e interesse no jogo. Os jogadores tiveram uma maior dificuldade contra o inimigo troll, porém, depois de alguns tiros eles entenderam as mecânicas. A alternância entre este e os inimigos normais gerou a dinâmica desejada de confundir o jogador e fazê-lo prestar mais atenção com cada tipo de inimigo que está na tela. O inimigo de cura fez com que o jogador mantivesse um maior cuidado com a sua barra de vida, assim se permitindo tomar mais dano, mas recuperando logo em seguida. O inimigo bonus permitiu uma situação para pontos de graça e invencibilidade contra um inimigo que estivesse muito complicado de derrotar.

Apesar dos novos inimigos, o *feedback* dos jogadores apontou que o jogo ainda continuava no mesmo passo e gerava desinteresse em várias jogadas pois o jogo não possuía nenhuma progressão de dificuldade.

4.3 FINAL

Para tratar o tédio gerado pelo jogo, adicionamos a mecânica de que cada inimigo derrotado, a velocidade no eixo y aumentava para os próximos inimigos e não só no número de linhas como antes. Aumentamos também, a velocidade da música conforme o jogador perdia vida. Com isso, percebemos que a dinâmica do jogo aumentou, já que inimigos mais rápidos exigem maior estratégia para eliminar os inimigos rapidamente.

Foi percebido também que, os jogadores se arriscavam mais no começo do jogo para acumular a maior quantidade de pontos para que em um nível avançado do jogo, eles pudessem apenas eliminar os inimigos com a quantidade perfeita de peças.

Criou-se uma página oficial do jogo no Facebook para uma maior divulgação e contato com os jogadores finais.

5 CONSIDERAÇÕES FINAIS

Quando um projeto de desenvolvimento nasce, uma variedade infinita de ideias são escritas e criadas pelos desenvolvedores. Porém, ao longo do projeto, essas ideias poderão ser mantidas ou até mesmo modificadas e descartadas para que a jogabilidade flua da maneira em que os desenvolvedores desejarem. Com o Jizlost, não foi diferente: mecânicas que criamos foram deixadas para trás para que o foco do jogo não se perdesse.

Com todo o feedback recebido, percebemos que o jogo estava com uma jogabilidade e o polimento além do esperado inicialmente. A mecânica de interação dos blocos foi uma das surpresas que mais animaram a equipe e fizeram o foco do jogo mudar bastante. Apesar de satisfeitos, outras mecânicas não conseguiram ser implementadas e ficaram para um futuro desenvolvimento, como um inimigo chefe e um novo modo de jogo.

Com todas as críticas e elogios recebidos, percebemos que a dificuldade do desenvolvimento de um jogo vai além do dia a dia da equipe e está muito mais na dificuldade em agradar o jogador final. Muitos desenvolvedores esquecem que o ponto final do seu jogo é dado pelo público e que são esses que vão decidir se seu jogo é bom ou não. Por isso é sempre importante receber todo o tipo de feedback possível para manter a qualidade em um alto patamar.

6 REFERÊNCIAS BIBLIOGRÁFICAS

Tetris, 2017. Disponível em: <https://en.wikipedia.org/wiki/Tetris> acesso em 11 nov2017

Pentominoes, 2017. Disponível em: <http://www.ericharshbarger.org/pentominoes/> acesso em 12 nov2017

Tetris Story, 2017. Disponível em: <https://web.archive.org/web/20060901232913/http://vadim.oversigma.com:80/Tetris.htm> acesso em 12 nov2017

Tetris by the numbers, 2017. Disponível em: <https://tetris.com/by-the-numbers> acesso em 13 nov 2017

Jizlost, 2017. Disponível em: <https://ebera.itch.io/jizlost> Acesso 27 nov 2017

Git Hub, 2017. Disponível em: <https://github.com/bentodiz/jizlost> Acesso em 20 dez 2017

Facebook. 2017. Disponível em: <https://www.facebook.com/jizlost/> Acesso em 18 dez 2017

7 APÊNDICE 1 – CÓDIGO FONTE

Código do jogo disponível no repositório do Git Hub.