

Luiz Reginaldo Gabriel Guimarães

Animação do jogo Tetris no Latex

São Paulo – Brasil

Junho/2017

Luiz Reginaldo Gabriel Guimarães

Animação do jogo Tetris no Latex

Relatório técnico, em conformidade com as normas ABNT, apresentado como requisito para aprovação na disciplina Projeto Integrador I.

Centro Universitário Senac

Bacharelado em Ciência da Computação

São Paulo – Brasil

Junho/2017

Resumo

Neste Relatório o leitor verá os principais passos para a criação da Animação Tetris no Latex. Através das figuras com trechos do código seguidas de seu resultado, o leitor será capaz de desenvolver a mesma animação sem dificuldades.

Palavras-chaves: latex. abntex. editoração de texto. tetris.

Sumário

1	INTRODUÇÃO	4
2	MATERIAIS E MÉTODOS	5
3	DESENVOLVIMENTO	6
3.1	Animação Tetris: Versão 0	6
3.2	Animação Tetris: Versão 1	6
3.3	Animação Tetris: Versão 2	7
3.4	Animação Tetris: Versão 3	7
3.5	Animação Tetris: Versão 4	8
3.6	Animação Tetris: Versão 5	8
3.7	Animação Tetris: Versão 6	9
3.8	Animação Tetris: Versão 7 - Final	10
4	RESULTADO	12
5	CONCLUSÃO	13
6	CONCLUSÃO	14
	REFERÊNCIAS	15

1 Introdução

Através deste documento o leitor será capaz de criar um animação Tetris sem dificuldades, como um código otimizado.

Através das Figuras ficam claros os comandos que devem ser utilizados para a criação das peças e para posicioná-las de acordo com a necessidade para cada jogador. Houve dificuldades para encontrar a fórmula perfeita para evitar anomalias de dimensionamentos, mas no resultado final eles não estão mais presentes.

Embora o Latex não seja uma linguagem de programação, ele se mostrou bastante maleável para evitar excesso de código e possibilitando uma fácil manutenção do código. Isso é possível através do `\newcommand` que simula o funcionamento de funções, podendo ser chamada a qualquer momento, inclusive recebendo parâmetros, e o comando `\ifthenelse`, que é uma condicional muito útil para tratamento destes parâmetros. Embora não seja possível criar variáveis, os contadores se mostraram capazes de exercer as funções delas.

2 Materiais e Métodos

Para elaboração da Animação Tetris no Latex foi utilizado o programa TexStudio e ocasionalmente o ambiente online overleaf.com para testes de novas funcionalidades. Utilizando o reprodutor de mídias VLC, e diminuindo em 10 vezes a velocidade de reprodução, foi possível registrar o movimento das peças nos primeiros 30 segundos da primeira partida da Final do Campeonato Mundial de Tetris de 2016. Foram utilizados diversos manuais do Latex de diferentes autores para encontrar os melhores comandos para a Animação. O pacote Tikz e animate foram as peças chaves para possibilitar a criação das imagens e animação das peças. Os comandos necessários para a animação foram, em ordem em que aparecem pela primeira vez no código:

newcounter

newcommand

path

ifthenelse

shadedraw

setcounter Foram utilizados os ambientes animateinline, tikzpicture e dentro deste o ambiente scopescope.

3 Desenvolvimento

Desde o início do projeto foi feita a tentativa de reproduzir as mesmas peças e posições da primeira partida da Final do Campeonato Mundial de Tetris 2016. Assim, no primeiro momento deveria ser criada a peça que chamo de Z. A seguir descreverei o processo de criação de cada versão da Animação Tetris, numeradas de 1 a 6.

3.1 Animação Tetris: Versão 0

Foi definido que cada quadrado teria tamanho 1. Então para a criação de cada peça, formadas por quatro quadrados, bastaria desenhar um quadrado e em volta dele os outros três. Já era sabido que para a criação de uma animação seria necessário reproduzir a mesma imagem do frame anterior com a mudança de posição de apenas uma peça.

Para evitar que o código da animação ficasse muito extenso e impossibilitasse a manutenção foi definido que cada peça seria criada dentro de um 'newcommand'. Nesta primeira tentativa, foi utilizado o comando 'draw':

```
\newcommand{\Z}[1]{%
\begin{tikzpicture}[ultra thick, scale=0.25]%
\draw (0,0) rectangle (1,1);
\begin{scope}[xshift=1cm]\draw[red](0,0)rectangle(1,1);\end{scope}
\begin{scope}[yshift=1cm]\draw[red](0,0)rectangle(1,1);\end{scope}
\begin{scope}[yshift=1cm,xshift=-1cm]\draw[red](0,0)rectangle(1,1);\end{scope}
```




3.2 Animação Tetris: Versão 1

Nesta versão a pretensão era experimentar colocar duas peças Z lado a lado. Para já preencher as peças com cores foi utilizado o comando 'filldraw' ao invés do 'draw'. Porém, como podemos verificar na figura, utilizando ambientes 'tikzpicture' diferentes

para cada peça, há um problema de posição, como se uma quantidade de pixels fosse adicionado nas bordas de cada imagem:

```
\begin{document}
\begin{frame}
| \begin{tikzpicture}[scale=0.25,overlay]
    \filldraw[red,draw=white](0,0) rectangle (1,1);
    \filldraw[red,draw=white,xshift=1cm](0,0)rectangle(1,1);
    \filldraw[red,draw=white,yshift=1cm](0,0)rectangle(1,1);
    \filldraw[red,draw=white,yshift=1cm,xshift=-1cm](0,0)rectangle(1,1);
  \end{tikzpicture}
  \begin{figure}
    \begin{tikzpicture}[scale=0.25,overlay,xshift=1.5cm]
      \filldraw[green,draw=white](0,0) rectangle (1,1);
      \filldraw[green,draw=white,xshift=1cm](0,0)rectangle(1,1);
      \filldraw[green,draw=white,yshift=1cm](0,0)rectangle(1,1);
      \filldraw[green,draw=white,yshift=1cm,xshift=-1cm](0,0)rectangle(1,1);
    \end{tikzpicture}
    \begin{tikzpicture}[scale=0.25,overlay,xshift=1.5cm]
      \filldraw[yellow,draw=white](0,0) rectangle (1,1);
      \filldraw[yellow,draw=white,xshift=1cm](0,0)rectangle(1,1);
      \filldraw[yellow,draw=white,yshift=1cm](0,0)rectangle(1,1);
      \filldraw[yellow,draw=white,yshift=1cm,xshift=-1cm](0,0)rectangle(1,1);
    \end{tikzpicture}
  \end{figure}
\end{document}
```



Estava claro que cada frame deveria ser criado dentro de um único ambiente 'tikzpicture'.

3.3 Animação Tetris: Versão 2

Agora era hora de testar a animação: mudança de frame a frame.

<p>Frame 1</p> 	<pre>\documentclass[twocolumn]{beamer} \usepackage{tikz} \usepackage{animate}</pre>
<p>Frame 2</p> 	<pre>\begin{document} \begin{animateinline}[loop,palindrome]{1} \begin{tikzpicture}[scale=0.25] \draw(0,0)rectangle(1,1); \end{tikzpicture} \newframe \begin{tikzpicture} \draw[red](0,0)rectangle(2,2); \end{tikzpicture} \end{animateinline} \end{document}</pre>

3.4 Animação Tetris: Versão 3

Verificado a estrutura do código para uma animação, usei o 'whiledo' para automatizar o movimento de uma peça Z. Mas no último frame, uma anomalia acontecia, onde a largura do path diminuía alguns pixels.


```

\begin{document}
\newcounter{moveY}
\setcounter{moveY}{0}
\newcounter{maxY}
\setcounter{maxY}{20}
\newcounter{iniY}
\begin{animateinline}[autoplay]{2}
\while{\themoveY < \themaxY}{
\begin{tikzpicture}[scale=0.25]
\path[fill=black](0,0) rectangle (3,\themaxY+1);
\setcounter{iniY}{\themaxY-\themoveY-1}
\begin{scope}[yshift=\theiniY cm,xshift=1cm]
\shadedraw[ball color=red,draw=white](0,0)rectangle+(1,1);
\shadedraw[ball color=red,draw=white](1,0)rectangle+(1,1);
\shadedraw[ball color=red,draw=white](0,1)rectangle+(1,1);
\shadedraw[ball color=red,draw=white](-1,1)rectangle+(1,1);
\end{scope}
\shadedraw[ball color=green,draw=white](0,\themaxY-\themoveY) rectangle
\end{tikzpicture}
\stepcounter{moveY}
\ifthenelse{\themoveY < \themaxY}{\newframe}{}
}
\end{animateinline}
\end{document}

```



Posteriormente, cancelei o uso do `while` pois não haveria uma matemática possível que simulasse o movimento das peças, visto que estas não somente desciam, como poderiam se movimentar para a direita e esquerda em momentos diferentes. O `while` se mostraria útil apenasse fossem necessários repetir frames para sincronismo com o vídeo.

3.5 Animação Tetris: Versão 4

Agora era necessário criar as demais peças. A exemplo da peça 'Tetris' abaixo (que viria a se chamar Y), foram criadas as peças L, Linverse, Q, I e newS e a já vista anteriormente, Z.



```

\newcommand{\newtetris}[3]{
\ifthenelse{#3>0}{
\ifthenelse{#3>1}{
\ifthenelse{#3>2}{
\setcounter{x}{#1}
\setcounter{y}{#2+2}
\begin{scope}[xshift=\theX*-1 cm,yshift=\theY cm]
\begin{rotate}{270}
{
\setcounter{x}{#1+2}
\setcounter{y}{#2+2}
\begin{scope}[xshift=\theX*-1 cm,yshift=\theY*-1 cm]
\begin{rotate}{180}
{
\setcounter{x}{#1+2}
\setcounter{y}{#2+1}
\begin{scope}[xshift=\theX cm,yshift=\theY*-1 cm]
\begin{rotate}{90}
{
\setcounter{x}{#1+1}
\begin{scope}[xshift=\theX cm,yshift=#2 cm]
\begin{rotate}{0}
{
\shadedraw[ball color=pink,draw=white](0,0)rectangle+(1,1);
\shadedraw[ball color=pink,draw=white](1,0)rectangle+(1,1);
\shadedraw[ball color=pink,draw=white](0,1)rectangle+(1,1);
\shadedraw[ball color=pink,draw=white](-1,0)rectangle+(1,1);
\end{rotate}
\end{scope}
\end{rotate}
\end{scope}
}
}
}
}
}
}
}
}
}
}
\end{document}

```

3.6 Animação Tetris: Versão 5

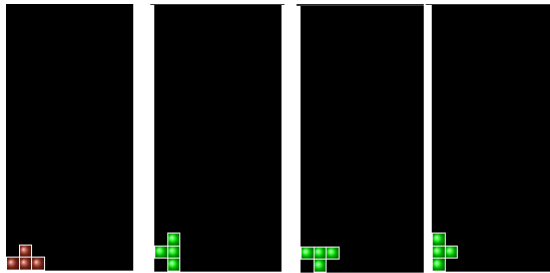
Após verificar que o uso do `rotate` para girar as peças de acordo com os parâmetros passados para cada peça não era viável, sendo que sempre que a peça girava, o path de cada jogador se achatava, diminuindo consideravelmente sua largura. Foi então definido

que a peça deveria ser desenhada para cada direção possível, como por exemplo a peça Y, 0, 90, 180 e 270 graus.

```
\newcommand{\newtetristwo}[3]{
  \ifthenelse{#3>0}{
    \ifthenelse{#3>1}{
      \ifthenelse{#3>2}{%3 - 270°
        \setcounter{x}{#1}\setcounter{y}{#2+1}
      }{%2 - 180°
        \setcounter{x}{#1+1}\setcounter{y}{#2+1}
      }
    }{%1 - 90°
      \setcounter{x}{#1+1}\setcounter{y}{#2+1}
    }
  }{%=0
    \setcounter{x}{#1+1}\setcounter{y}{#2}
  }
}

\begin{scope}[xshift=\thex cm,yshift=\they cm]
\ifthenelse{#3>0}{
  \ifthenelse{#3>1}{
    \ifthenelse{#3>2}{%3 - 270°
      \shadedraw[ball color=green,draw=white](0,0)rectangle+(1,1);
      \shadedraw[ball color=green,draw=white](0,-1)rectangle+(1,1);
      \shadedraw[ball color=green,draw=white](1,0)rectangle+(1,1);
      \shadedraw[ball color=green,draw=white](0,1)rectangle+(1,1);
    }{%2 - 180°
      \shadedraw[ball color=green,draw=white](0,0)rectangle+(1,1);
      \shadedraw[ball color=green,draw=white](0,-1)rectangle+(1,1);
      \shadedraw[ball color=green,draw=white](1,0)rectangle+(1,1);
      \shadedraw[ball color=green,draw=white](-1,0)rectangle+(1,1);
    }
  }{%1 - 90°
    \shadedraw[ball color=green,draw=white](0,0)rectangle+(1,1);
    \shadedraw[ball color=green,draw=white](0,-1)rectangle+(1,1);
    \shadedraw[ball color=green,draw=white](0,1)rectangle+(1,1);
    \shadedraw[ball color=green,draw=white](-1,0)rectangle+(1,1);
  }
}
\shadedraw[ball color=orange,draw=white](0,0)rectangle+(1,1);
\shadedraw[ball color=orange,draw=white](1,0)rectangle+(1,1);
\shadedraw[ball color=orange,draw=white](0,1)rectangle+(1,1);
\shadedraw[ball color=orange,draw=white](-1,0)rectangle+(1,1);
}

\begin{document}
\begin{animateinline}[autoplay]{0.5}
\begin{tikzpicture}[scale=0.25]
\newpath\newtetristwo{0}{0}{0}
\end{tikzpicture}\newframe
\begin{tikzpicture}[scale=0.25]
\newpath\newtetristwo{0}{0}{1}
\end{tikzpicture}\newframe
\begin{tikzpicture}[scale=0.25]
\newpath\newtetristwo{0}{0}{2}
\end{tikzpicture}\newframe
\begin{tikzpicture}[scale=0.25]
\newpath\newtetristwo{0}{0}{3}
\end{tikzpicture}\newframe
}
```



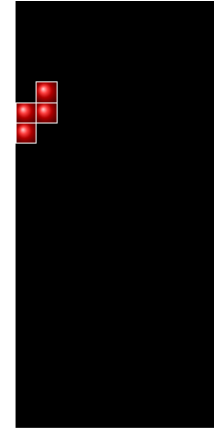
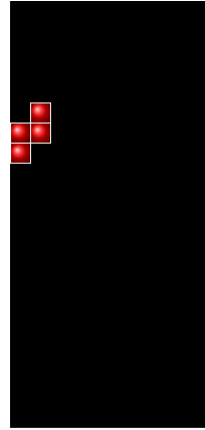
3.7 Animação Tetris: Versão 6

Nesta versão foi inserido um quarto parâmetro para cada peça que indicaria em qual path (jogador) a peça deveria ser criada. Assim, passei a observar numa velocidade de reprodução baixíssima cada movimento no vídeo do campeonato mundial de tetris.

```

\newslide\z{0}{19}{1}{0}\x{0}{19}{1}{1}
\newslide\z{0}{18}{1}{0}\x{0}{18}{1}{1}
\newslide\z{0}{17}{1}{0}\x{0}{17}{1}{1}
\newslide\z{0}{16}{1}{0}\x{0}{16}{1}{1}
\newslide\z{0}{15}{1}{0}\x{0}{15}{1}{1}
\newslide\z{0}{14}{1}{0}\x{0}{14}{1}{1}
\newslide\z{0}{13}{1}{0}\x{0}{13}{1}{1}
\newslide\z{0}{13}{1}{0}\x{0}{14}{1}{1}
\newslide\z{0}{13}{1}{0}\x{0}{13}{1}{1}
\newslide\z{0}{12}{1}{0}\x{0}{13}{1}{1}
\newslide\z{0}{11}{1}{0}\x{0}{13}{1}{1}
\newslide\z{0}{11}{1}{0}\x{0}{13}{1}{1}
\newslide\z{0}{11}{1}{0}\x{0}{13}{1}{1}
\newslide\z{0}{11}{1}{0}\x{0}{12}{1}{1}
\newslide\z{0}{11}{1}{0}\x{0}{11}{1}{1}%Jonas alcança
\newslide\z{0}{10}{1}{0}\x{0}{11}{1}{1}
\newslide\z{0}{09}{1}{0}\x{0}{11}{1}{1}%Jeff se adianta 2
\newslide\z{0}{09}{1}{0}\x{0}{10}{1}{1}
\newslide\z{0}{09}{1}{0}\x{0}{09}{1}{1}%Jonas alcança
\newslide\z{0}{08}{1}{0}\x{0}{09}{1}{1}
\newslide\z{0}{07}{1}{0}\x{0}{09}{1}{1}%Jeff se adianta 2
\newslide\z{0}{07}{1}{0}\x{0}{08}{1}{1}
\newslide\z{0}{07}{1}{0}\x{0}{07}{1}{1}%Jonas alcança
\newslide\z{0}{06}{1}{0}\x{0}{07}{1}{1}
\newslide\z{0}{05}{1}{0}\x{0}{07}{1}{1}%Jeff se adianta 2
\newslide\z{0}{05}{1}{0}\x{0}{06}{1}{1}
\newslide\z{0}{05}{1}{0}\x{0}{05}{1}{1}%Jonas alcança
\newslide\z{0}{04}{1}{0}\x{0}{05}{1}{1}
\newslide\z{0}{03}{1}{0}\x{0}{05}{1}{1}%Jeff se adianta 2
\newslide\z{0}{03}{1}{0}\x{0}{04}{1}{1}
\newslide\z{0}{03}{1}{0}\x{0}{03}{1}{1}%Jonas alcança
\newslide\z{0}{02}{1}{0}\x{0}{03}{1}{1}

```



3.8 Animação Tetris: Versão 7 - Final

Esta é a versão final. Foram adicionados alguns newcommands novos para efeitos especiais, como o fundo piscante e a função eraser para apagar as linhas completas.

```

\newcommand{\fixborderbug}{
  \draw[line width=1pt,white](25,0)--+(0,21);
  \draw[line width=1pt,white](0,-0.04)--+(25,0);
  \draw[line width=1pt,white](14.95,0)--+(0,21);
}

\newcommand{\newpath}{
  \path[fill=black](0,0) rectangle +(10,21);
  \path[fill=black](15,0) rectangle +(10,21);
}

\newcommand{\newpathcombo}{%fundo branco para o primeiro jogador - efeito
piscante
  \path[fill=white](0,0) rectangle +(10,21);
  \path[fill=black](15,0) rectangle +(10,21);
}

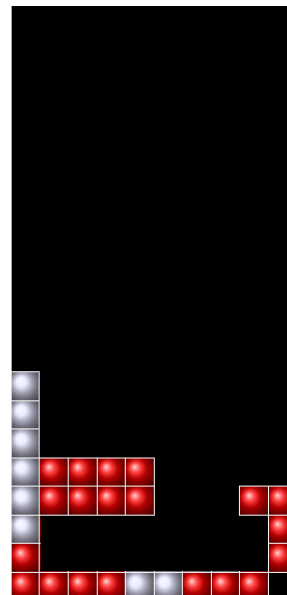
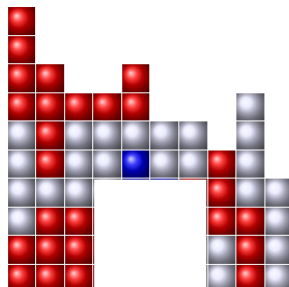
\newcommand{\newsli}{
  \end{tikzpicture}\newframe\begin{tikzpicture}[scale=0.4]
  \newpath
}

```

```

\newcommand{\eraser}[6]{
  \ifthenelse{#5>0}{
    \setcounter{x}{15}
  }{
    \setcounter{x}{0}
  }
  % \setcounter{x}{\thex+#1}\setcounter{y}{#2}
  \ifthenelse{#6>0}{
    \filldraw[debug]{\thex+#1+0.04,#2+0.05}rectangle(\thex+#1+3-0.04,#2+#4-0.05);
  }{
    \filldraw[debug]{\thex+#1+0.04,#2+0.05}rectangle(\thex+#1+3-0.04,#2+#4-0.05);
  }
  \filldraw[debug]{\thex+#1+0.04,#2+0.05}rectangle(\thex+#1+3-0.04,#2+#4-0.05);
}

```



O código das peças fixas dos frames anteriores eram incluídos em newcommands chamados safe(de a a r). Isso possibilitou que o tamanho do código diminuísse consideravelmente.

```
\setcounter{aux}{0}\whiledo{\theaux<40}{\newslide
\safer
\I{3}{10}{1}{0}
\Q{4}{10}{0}{0}
\Q{4}{13}{0}{0}
\I{3}{11}{1}{0}
\newlinverse{-4}{10}{2}{1}
\newL{4}{13}{0}{1}
\newlinverse{19}{10}{2}{0}
\newL{-4}{13}{0}{1}
\I{11}{11}{1}{0}
\I{4}{11}{1}{1}
\eraser{4}{10.7}{1.2}{0.6}{0}{0}
\eraser{4}{13.7}{1.2}{0.6}{0}{0}
\stepcounter{aux}}

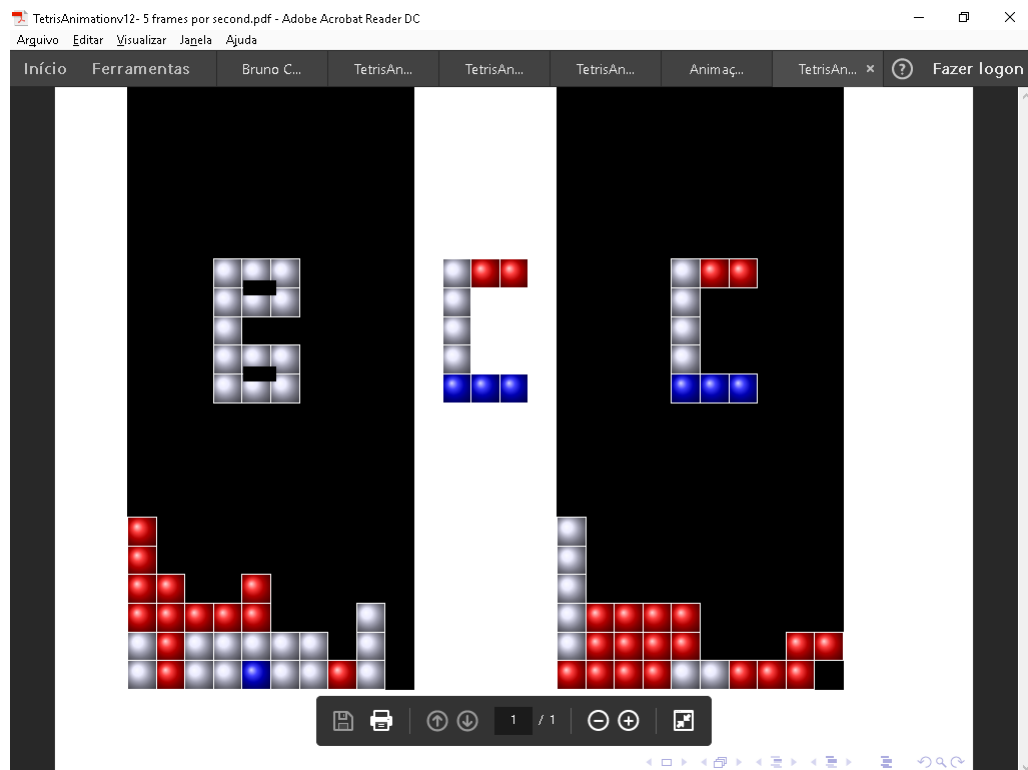
\newcommand{\safer}{\newL{2}{06}{2}{0}\y{2}{04}{2}{0}\safey{6}{02}{2}{1}\eraser
{0}{0}{10}{14}{1}{0}\fixborderbug
\I{0}{0}{1}{1}\I{0}{4}{1}{1}\z{0}{0}{0}{1}\z{2}{0}{0}{1}\z{4}{0}{0}{1}\z{6}{0}{0}
{1}\newL{0}{0}{0}{1}\Q{7}{1}{0}{1}\newL{6}{0}{0}{1}\Q{6}{1}{0}{1}\Q{4}{0}{0}{1}\I
{3}{2}{0}{1}\z{3}{2}{0}{1}\newL{1}{2}{1}{1}
}
\newcommand{\safey}{\newL{0}{07}{1}{0}\safer\newL{2}{03}{0}{1}}
\newcommand{\saferq}{
\z{5}{0}{0}{0}
\Q{0}{0}{0}{0}\Q{2}{0}{0}\Q{4}{0}{0}\Q{5}{0}{0}{0}
\I{8}{0}{1}{0}\newL{0}{0}{3}{0}\newL{0}{3}{1}{0}
\newS{3}{0}{1}{0}\newL{2}{2}{2}{0}\I{3}{1}{0}{0}
\newL{0}{0}{1}{1}\newL{1}{0}{1}{1}\newL{2}{0}{1}{1}\newL{3}{0}{1}{1}\newL{7}{
0}{0}{1}\newL{4}{0}{1}{1}\I{4}{0}{0}{1}\newL{6}{0}{2}{1}\I{0}{1}{1}{1}\I{0}{2}
{1}{1}
\eraser{6}{1}{2}{2}{1}{0}
}
\newcommand{\safer}{\saferq\eraser{7}{3}{3}{1.5}{0}{0}}
```

4 Resultado

Foram escritas 1209 linhas de código nos mesmos moldes que descritas anteriormente, resultando numa simulação de 16 peças de cada jogador, num período de animação de 40 segundos quando configurada com 20 frames por segundo (totalizando aproximadamente 800 frames).

Nestes 40 segundos 5 foram destinados a peças que surgem aleatoriamente, simultaneamente e em locais diferentes, que se juntam formando a inscrição BCC (Bacharelado em Ciência da Computação).

Foram necessárias muitas horas de testes até que experimentos a versão 5 se concretizasse, e a partir da versão 6 dias inteiros para construir cada frame e não deixar sequer um movimento diferente do vídeo original. Porém houve um momento que se mostrou coerente forçar um combo para que a animação finalizasse e não ficasse muito extensa.



5 Conclusão

Foi muito prazeroso chegar numa versão final que fizesse com que meus colegas de turma, após a apresentação em sala de aula, pedissem bis. A Animação Tetris nos mostra mais uma vez que assim como o próprio Tetris, vale apenas sempre fazer algo mais do que pode ser feito.

6 Referências

Referências

- [1] CREMER, Jacques A very minimal introduction to TikZ, 2011.
- [2] OETIKER, Tobias et. al. Introdução ao \LaTeX 2 ϵ , 2001. Nenhuma citação no texto.
- [3] TANTAU, Till. The TikZ and PGF Packages, <http://www.texample.net/tikz/>, 2014. Nenhuma citação no texto.
- [4] RICHTER, Pascal et. al. The TikZposter class, <http://www.ctan.org/pkg/tikzposter/>, 2014. Nenhuma citação no texto.

Nenhuma citação no texto.