

Java 2D – Jogo do Arpão

Francisco Monteiro

Mecânica do jogo e objetivos

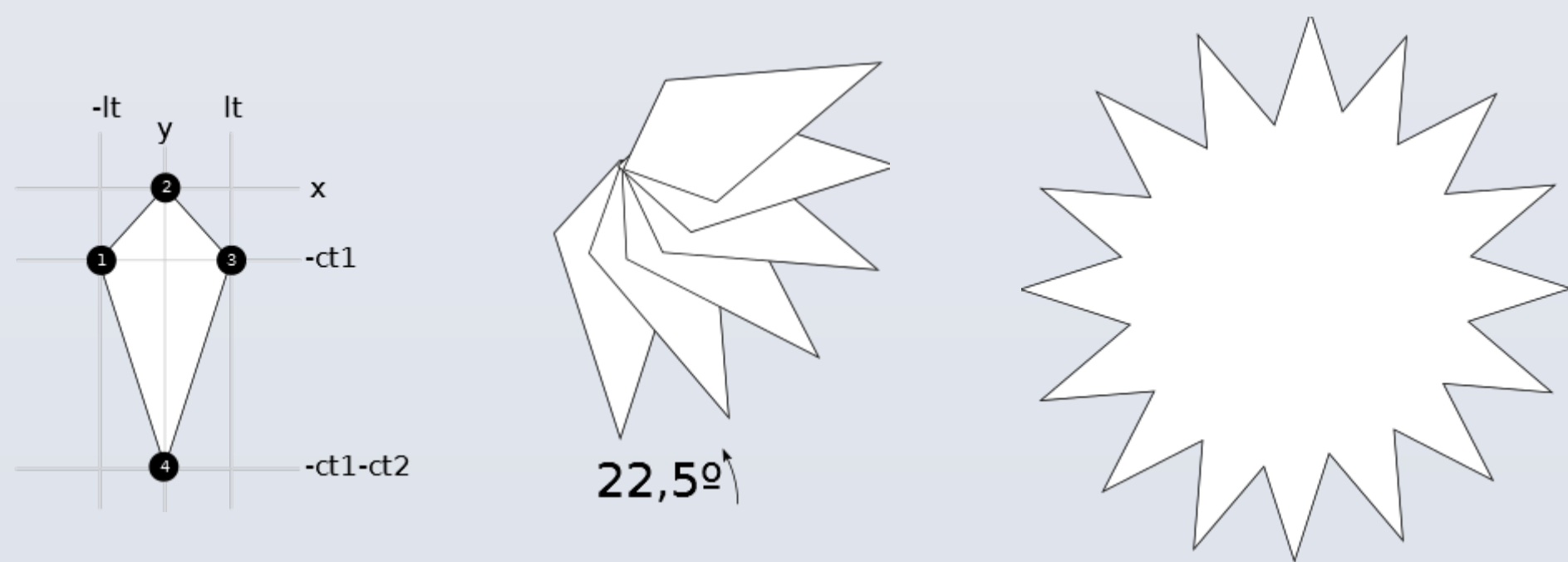
- O Jogo tem como objetivo controlar um personagem que se desloca no cenário através de inputs do teclado dado pelo utilizador.
- O personagem pode mover-se na horizontal e na vertical sendo que na vertical sofre a influência de aceleração.
- O jogo é dividido em níveis que podem ser editáveis pelo utilizador, cada nível além da personagem tem bolas e estrelas que podem ser destruídas pelo arpão do personagem.
- O arpão do personagem é disparado pela tecla espaço e faz um movimento vertical crescente, desde a posição inicial do personagem, até à ao topo do cenário, até à próxima colisão com uma bola ou estrela, ou colidindo com a plataforma central.
- Cada nível tem um tempo limitado de 60 segundos o qual debitará uma vida ao jogador caso termine e neste caso o nível reiniciará.
- Cada colisão de uma bola ou estrela com o arpão farão dividir-se em duas bolas ou estrelas respetivamente mais pequenas até um limite mínimo.
- No caso de as bolas ou estrelas estarem no seu limite mínimo de tamanho, após a colisão com o arpão desaparecerão.
- Cada nível tem por objetivo destruir a totalidade das bolas e estrelas, no tempo limite de 60 segundos após estas estarem no tamanho mínimo.

Técnicas de conceção dos níveis

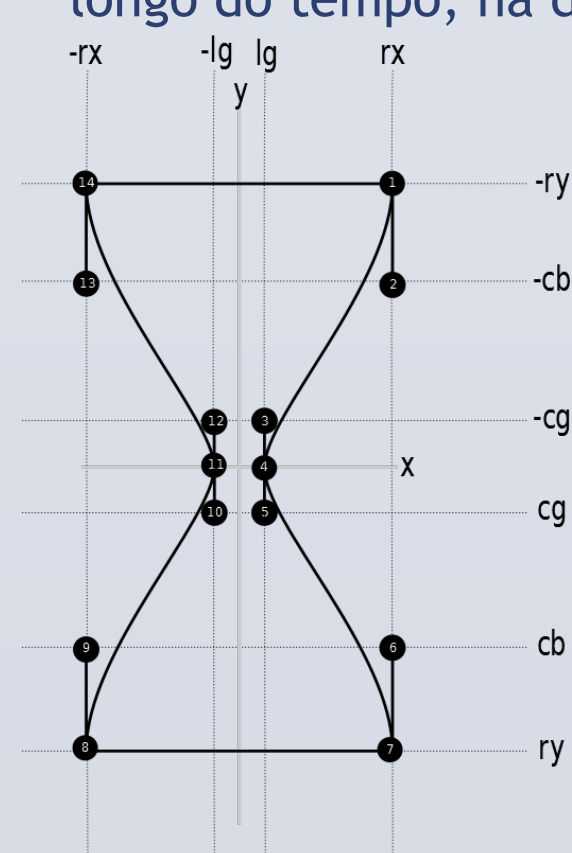
- Para a conceção do jogo foi usada a ferramenta Java2D para o desenho das animações, com uma frequência 60 fps.
- A animação do personagem é feita através da utilização de sprites.
 - Sprites são fragmentos de uma imagem que são recortados e usados alternadamente para simular o movimento do personagem.
- A animação das bolas é feita através do recorte da bola respetiva, dependendo do tamanho, na posição correspondente.



- Esta animação é feita com uma velocidade linear segundo o eixo dos x's e acelerada segundo o eixo dos y's, à aceleração é reduzida uma unidade a cada colisão no eixo dos y's.
- As estrelas são construídas por adição de geometrias rodando-as em torno de um eixo.



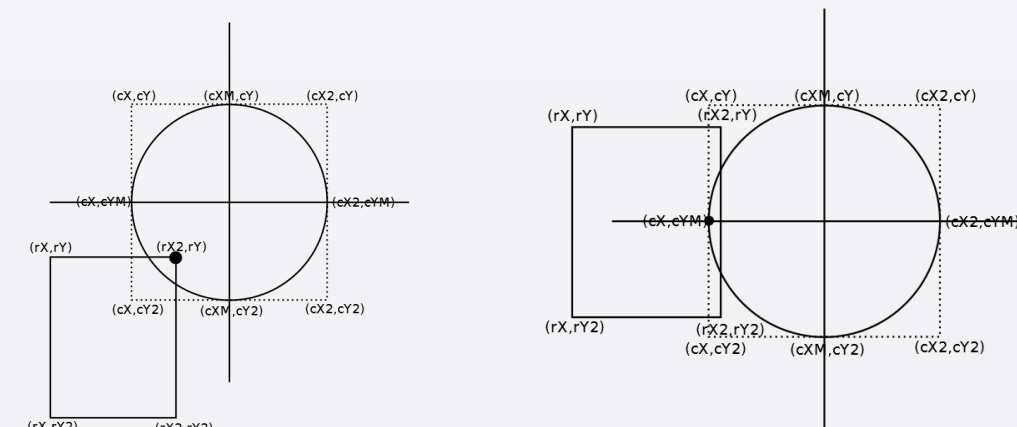
- O preenchimento da geometria com cor, é feito com um gradiente radial de dentro para fora, iniciando em amarelo a terminar em vermelho nas extremidades.
- A animação das estrelas é constituída por um movimento de translação linear ao longo do tempo e um movimento de rotação, também linear ao longo do tempo, na direção dos ponteiros do relógio.



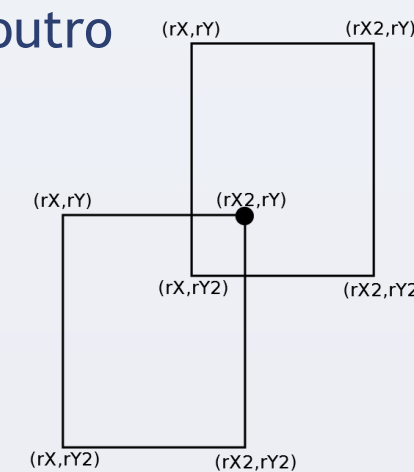
- A ampulheta, construída com *GeneralPath* em representação da passagem do tempo.
- Preenchido com uma textura arenosa através da técnica de clipping, a textura é deslocada na vertical para simular a queda da areia.

Colisões e consequências

- As colisões que fazem a interação do jogo, são as das bolas e estrelas com os limites do cenário, as das bolas e estrelas com a plataforma central, as das bolas e estrelas com o personagem, a do personagem com os limites do cenário, a do personagem com a plataforma central.
- As colisões são feitas manualmente de duas formas distintas, retângulos com retângulos e retângulos com círculos



- As colisões entre dois retângulos são definidas quando um dos pontos de um retângulo está dentro de outro



- Há colisão entre um círculo e um retângulo, se um dos vértices do retângulo estiver dentro do círculo.
- Também há colisão caso um dos pontos do círculo esteja dentro do retângulo, para isso, existem apenas 4 possibilidades, que é quando uma das coordenadas de um ponto do círculo corresponda a uma das coordenadas do seu centro.

Editor de níveis

- O editor de níveis guarda os níveis criados pelo utilizador
- Através do JComponentes personalizados foi possível criar botões recortando cada uma das bolas para os botões respetivos.
- Também com Jcomponents personalizados e o método utilizado para a geração das estrelas foram criados os botões correspondentes.
- Com os listeners do mouse, foram feitas num Jcomponent personalizado correspondente ao tamanho do Jcomponent do jogo, foi possível capturar as posições de todos os objetos dinâmicos do jogo, personagem, estrelas e bolas, assim como criar novas.
- Para estas ações foi necessário criar dois modos de funcionamento
 - Inserção
Neste modo o utilizador seleciona um dos botão entre estrelas ou bolas e clica no ponto correspondente ao qual a bola/estrela irá ficar posicionada.
 - Seleção
Este modo serve para a correção da posição, é possível selecionar para mudar de posição o personagem as bolas ou as estrelas.
- Após as alterações efetuadas é possível persistir os dados numa base de dados local SQLITE

Referência

Para a elaboração deste trabalho foram usados os diapositivos das aulas.

Site <http://www.pontov.com.br/site/index.php/java/48-java2d>