Etapas de desenvolvimento: cronograma de atividades

No desenvolvimento de um game, precisamos estabelecer listas de tarefas do que será produzido. Conforme as etapas de desenvolvimento, estabelecemos o que será entregue e com que qualidade os produtos serão liberados. Para isso, iniciamos a organização de um cronograma de atividades, após estabelecer a lista de recursos que estarão no jogo.

Mas o que é um cronograma? Chandler (2012, p. 260) define um cronograma como “uma lista com cada tarefa que deve ser concluída, estimativas de duração das tarefas, quem a está executando e suas interdependências”. Ou seja, o que fazer, em que prazo fazer e quem é o responsável.

Recomendação de *software* para elaboração de cronograma: Microsoft Project ou Microsoft Excel. Pesquisar na internet por ferramentas *on-line* gratuitas do tipo GANTT CHART.

O cronograma inicial começa a ser elaborado na pré-produção. Trata-se de uma tabela contendo o que dever ser feito, divido por área: produção, aprovações necessárias, *design*, arte, engenharia, áudio, localização, QA, cinemática, *marketing*. Este cronograma

ainda precisa ser detalhado conforme forem avançando as definições do GDD e TDD. A responsabilidade por manter o **cronograma do projeto, ou o cronograma-mestre do projeto**, conforme Novak (2008), é do produtor (gerente do projeto).

Veja um exemplo de cronograma inicial no livro *Manual de produção de jogos digitais*, de Heather Maxwell Chandler.

À medida que obtivermos os detalhes das fases e do que deve conter nelas, podemos detalhar as atividades, considerando as fases do jogo e o prazo de entrega. O prazo final do projeto, assim que definido, determinará os prazos das etapas de desenvolvimento (alfa, congelamento de código, beta, código candidato à liberação etc.). Estes prazos são importantes e devem incluir a participação da equipe de *marketing*, pois conhecem o mercado e logo devem influenciar nas entregas. A equipe de *marketing* influencia, ainda, na definição dos *assets* que devem ser incluídos em cada etapa para gerar as promoções do jogo.

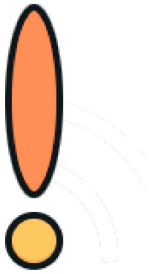
Veja um exemplo de cronograma das etapas de desenvolvimento no *Manual de produção de jogos digitais*, Heather Maxwell Chandler.

Definidas as etapas, cria-se um cronograma detalhado por fase, um WBS (*work brakedown structure* – estrutura de divisão do trabalho), também conhecido como estrutura analítica do projeto. Para cada fase, elenca-se o que deve ser feito (tarefas) em cada uma das seguintes áreas: arte, *design*, som, QA, programação etc. No detalhamento deste cronograma, devem ser indicados para cada tarefa: a data de início e a data de fim, as dependências entre as tarefas e o responsável por executá-las.

Em Chandler (2012, p. 265), é apresentado um exemplo de WBS por fase do jogo.

Cada equipe é responsável por elaborar o cronograma de sua área e manter atualizado o cronograma-mestre para que o produtor tenha o controle sobre o todo de tarefas do projeto. Desta forma, a equipe de programação mantém um cronograma próprio, assim como a equipe de arte. O produtor deve acompanhar o todo para garantir que o projeto se mantenha no caminho até a entrega.

O cronograma de programação que utiliza a metodologia ágil *Scrum* é o Product Backlog. Nele, estão incluídas as tarefas que serão executadas em cada *sprint* de acordo com a prioridade de entrega. As *sprints* são planejadas em função das etapas de desenvolvimento. Para cada etapa, pode ser realizada uma série de *sprints*. A *sprint*, por sua vez, tem o seu cronograma, o Sprint Backlog, planejando o que será elaborado no seu período de duração.



Se você tiver dúvidas quanto ao conceito de *scrum*, retome o material da Unidade Curricular 1 sobre as metodologias de desenvolvimento de *software* ou pesquise na internet sobre a metodologia *Scrum.*

Esses cronogramas não são fechados, são “documentos vivos”, em constante modificação, pois, conforme sabemos, o processo de desenvolvimento sofre mudanças de requisitos, mudanças em função de erros, e atividades devem ser realocadas para um novo período e por causa de novas funcionalidades. Esta característica de constante mudança deve ser controlada pelo gerente de projeto, que é o responsável por garantir o prazo, o custo e a qualidade do projeto. Justamente por isso, ele deve manter o permanente controle sobre as atividades.

Para realizar este rastreamento do progresso das tarefas, Chandler (2012) indica que um membro da equipe de uma área específica pode ficar responsável por alertar os demais sobre os prazos das tarefas. Este rastreamento utilizando as metodologias ágeis pode ser feito com uma ferramenta visual chamada KanBan, que consiste em colocar em um quadro visível para a equipe toda o que tem para ser feito, o que está sendo feito e o que foi concluído. Este quadro é dividido nessas três etapas, e as tarefas são alocadas com Post-It indicando o que é para fazer e quem é o responsável. Esta ferramenta permite ver se há muitas tarefas para fazer ou se a maioria já está concluída.

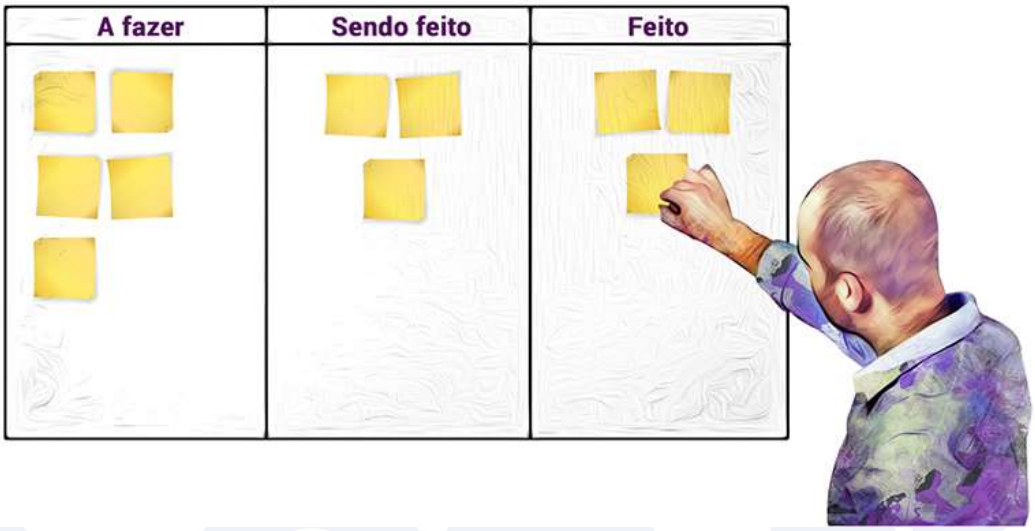


Figura 1 – Quadro de tarefas dividido em três etapas

SAIBA +

Dica: pesquise sobre SCRUM-BAN na internet para descobrir mais sobre a utilização de KanBan no acompanhamento de projetos ágeis com a metodologia *Scrum.*

Vamos à prática!

Como identificar ou dividir as tarefas? Podemos partir de uma abordagem de baixo para cima, ou de uma abordagem de cima para baixo, bem como de uma abordagem mista. Por exemplo, estabeleceremos as tarefas de programação para a jogabilidade “impedir abdução”, do Alien City, na abordagem de cima para baixo(*top-down*). Conforme a figura 2, partimos de Impedir abdução, quebramos na programação dos eventos de início da abdução e salvamento; quebramos Salvamento nas seguintes tarefas: Gatilho em função da localização do jogador, Sensor de proximidade Ray 360°, Controles do

jogador, Salvamento bem-sucedido e Salvamento malsucedido. Em Salvamento bem-sucedido, temos o Disparo de animações de sucesso e a programação da Lógica para que ele ocorra. Em Salvamento malsucedido, temos as mesmas tarefas que no bem-sucedido, porém para a lógica e a animação correspondentes.

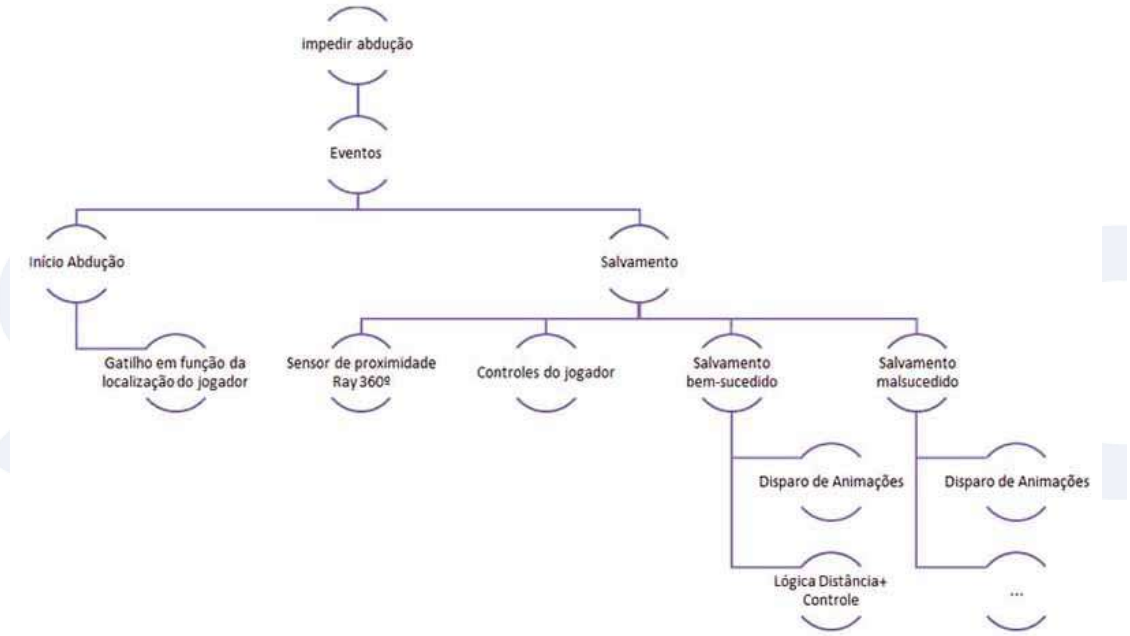


Figura 2 – Diagrama da abordagem top-down para a elaboração da jogabilidade “impedir abdução”

SAIBA +

Pesquise sobre as abordagens de baixo para cima e mista associadas à gestão de projetos.

A seguir, veremos como organizar o cronograma de atividades da equipe de programação. Estão registradas as dependências das tarefas de programação com as de animação.

Task Name	Duration	Start	Finish	Predecessors	Resource Names
Produção da Fase 1 Alien City					
Programação					
Implementar a Jogabilidade "Impedir Abdução"					
Programar Eventos					
Início Abdução					
Programar Gatilho em função da localização					
Salvamento					
Sensor de proximidade Ray 360°					
Controles do Jogador					
Salvamento Bem Sucedido					
Disparo de Animação				18	
Lógica de Avaliação de Distância+Evento Controle					
Salvamento Mal Sucedido					
Disparo de Animação				19	
Lógica de Avaliação de Distância+Evento Controle					
Arte					
Animação de Abdução					
Animação Salvamento Bem-Sucedido					
Animação Salvamento Malsucedido					

Figura 3 – Cronograma de produção de nível

A figura 3 apresenta um cronograma de produção de nível com recursos atribuídos. Podemos ver na tabela, na primeira coluna (a mais à esquerda), o nome da tarefa (Task Name), seguido de uma coluna com a duração (Duration), da coluna para a data de início (Start) e para a de fim (Finish). Mais à direita, vemos a predecessora (Predecessor) para registrar as dependências, e por último, a coluna contendo os nomes dos recursos responsáveis (Resource Name). Este cronograma serve para o acompanhamento de todas as tarefas para produção da fase 1 do Alien City. Nele, o gerente de projeto deve estabelecer os prazos e os recursos responsáveis pela implementação. No exemplo acima, foram estabelecidas apenas as tarefas e as dependências entre elas. Os prazos e a duração serão trabalhados no tópico sobre estimativas.

As tarefas no cronograma devem conter um nome, uma descrição (se necessário) em uma coluna adicional, a duração, sua data de início e de fim, as dependências e o recurso responsável. Se considerarmos a metodologia ágil *Scrum*, as tarefas do *backlog* são escritas como histórias, elas contêm as informações a seguir.

- O papel do usuário: aquele que se beneficia com a história, o *player* ou o usuário de alguma ferramenta da linha de produção (artista, programador, *tester*).
- O objetivo: o objetivo da história, uma característica, uma funcionalidade do jogo ou uma ferramenta para produção.
- A justificativa: o benefício para o usuário ou o jogador com a implementação da história.

Veja exemplos de histórias.

Vocês podem perceber que todas as histórias apresentadas estão em alto nível, ou seja não trazem detalhes de como fazer. Este detalhamento será feito pela equipe responsável que dividirá essas histórias, conhecidas como épicas, em histórias menores, que serão estimadas conforme seu tamanho e sua prioridade. É muito importante cuidar o nível de detalhamento das tarefas e das histórias no cronograma, pois, se forem muito detalhadas, o cronograma fica grande demais para ser acompanhado. Posto isso, delega-se para a equipe responsável a verificação de algum detalhamento para a realização da tarefa.

Por fim, devemos nos atentar para a utilização do *Scrum*, que pode ser concomitante ao cronograma de atividades. Por exemplo, o *Scrum* pode ser utilizado apenas pela equipe de programação, que é alimentada a partir das tarefas registradas no cronograma-mestre do projeto, mantido pelo produtor.

Quando o desenvolvimento do jogo digital for multiplataforma, Chandler (2012) recomenda que 95% da equipe se foque na plataforma principal e 5% nas plataformas adicionais.

Lembre-se de buscar mais detalhes sobre esta parte na internet e na bibliografia sugerida para esta unidade. Em Rabin (2013), Chandler (2012) e Novak (2008), você tem informações e exemplos valiosos à sua disposição. Não se esqueça da vasta fonte de informações disponível pela internet.

Bibliografia

CHANDLER, Heather Maxwell. **Manual de produção de jogos digitais.** 2ª ed. Porto Alegre: Bookman, 2012.

NOVAK, Jeannie; HIGHT, John. **Game Development Essentials:** Game Project Management. New York: Cengage Learning, 2008.

RABIN, Steve. **Introdução ao desenvolvimento de games:** v. 4. São Paulo: Cengage Learning, 2013.