Gamefication engine

Daniel Specht Menezes - 1010916

O objetivo deste trabalho é conceitualizar e implementar uma engine de gamefication que seja versátil e genérica o suficiente para se ajustar a diferentes modelos de gamefication. Para se chegar a tal produto serão analisados diversos cases de sucesso envolvendo gamefication verificando os recursos e estruturas mais utilizados e os possíveis problemas.

Motivação

Gamefication pode ser traduzido como uma maneira de fabricar e inserir um propósito de competitividade em um contexto que antes não o possuía. Portanto, o próprio conceito em si traz uma ideia de que se trata de algo adicional, que deve ser inserido em um ambiente diferente, que deve ser acoplado.

Percebe-se então que no processo de construção de um sistema as funcionalidades de gamefication podem ser bem separadas e encapsuladas. Afinal, trata-se de um conjunto de estruturas e procedimentos computacionais que são utilizados dentro da aplicação

Elementos Comuns

Pontos

A pontuação é uma maneira de se recompensar o comportamento do usuário quando contribui para o atingimento do objetivo definido para a gamefication implementada.

Implementação

Deve ser possível criar uma estrutura de dados que associe um usuário seus pontos. A princípio verifica-se duas maneiras distintas de se implementar essa estrutura.

1. A própria entidade da conta do usuário pode possuir atributos referentes à pontuação.
2. Uma nova tabela pode ser criada para armazenar as informações referentes à pontuação do usuário e a outros recursos da gamefication, como os badges.

A segunda maneira de se implementar parece ser mais adequada uma vez que encapsula melhor as estruturas utilizadas pela funcionalidade de gamefication.

As funções referentes à manipulação da quantidade de pontos do usuário devem ser encapsuladas e chamadas quando for realizada uma atividade que possui um número de pontos associado. O número de pontos associado a cada atividade pontuada está descrito no arquivo de regras. Os parâmetros desta função são a conta do usuário e a quantidade de pontos definida na regra.

Regras

As regras definem o valor atribuído as ações do usuário que são recompensadas. Isso é definido a partir do que se escolhe para o objetivo da gamefication.

Exemplo tendo em vista um sistema de perguntas e respostas.

Ação	Pontuação
Logar uma vez por dia	+10pt
Responder a uma pergunta	+20pt
Dar a melhor resposta a uma pergunta	+100pt
Fazer uma pergunta	+5pt

Tabela1 - Regras

Existe uma clara vantagem em centralizar as regras para pontuação, afinal é comum que essas regras sofram alterações com a finalidade de balancear melhor as pontuações. Um exemplo aplicado novamente no sistema de perguntas e respostas seria: caso os usuários estejam fazendo muitas perguntas irrelevantes com o intuito de ganhar mais pontos é possível reduzir de maneira ágil a quantidade de pontos oferecida para esta atividade.

Implementação

A maneira mais correta de se implementar essa estrutura seria fazendo com que fosse possível alterar as regras sem que seja necessário recompilar o sistema inteiro.

Exemplo:

Em Java, por exemplo, é possível instanciar em um arquivo XML objetos de determinadas classes conhecidas que são acessados pelo programa. E os objetos definidos no arquivo podem alterados sem que isso implique na necessidade de recompilar o sistema inteiro.

Badges

Badges indicadores colecionáveis de se ter realizado algo, que podem ou não vir acompanhadas de pontos. Cada badge representa um objetivo que deve ser cumprido, por exemplo, visitar o sistema 10 dias seguidos. O tipo do badge define quantas vezes ele pode ser conquistado.

Exemplo:

Nome	Descrição	Pontuação associada	Quantas vezes pode ser obtido
“Produtivo”	Conseguiu fazer +100 pontos em um dia	+30pt	1/dia
“Um de nós”	Membro há um ano	+500pt	1vez
“Funcionário do mês”	Fez mais pontos dentre todos os usuários no mês	+300pt	1vez/mês

Tabela2 - Badges

Implementação

É muito comum que um badge seja atingido a partir do atingimento de um valor por um contador, como o exemplo dado na tabela2 para os badges “Produtivo” e “Um de nós”. Trata-se de uma

lógica de programação comum para a implementação dessa funcionalidade a qual buscarei encapsular.

Portanto, uma estrutura para um contador deverá ser montada.

- O contador poderá estabelecer um tempo de “reset”.
- Funções de incremento do contador.
- Função de reset do contador.
- A todos os incrementos do contador deverão ser checados os badges associados ao mesmo

Entretando, deve ser possível que badges sejam premiados fora das funções de incremento de contadores, uma vez que o trigger para o badge ser premiado pode não ser dependente de um contador.

Leveis

Leveis são indicadores do progresso do usuário. Geralmente os leveis são evoluídos com base em uma quantidade especifica de pontos. Os leveis são importantes no sentido de estimular a competição entre os usuários e definir rankings.

É claro que se organizarmos os rankings apenas pela pontuação total obteremos um efeito bastante semelhante ao que teríamos com os leveis. Entretanto os leveis por precisarem de um número definido de pontos estimula os usuários a irem atrás dessa quantidade como um todo.

Os leveis também podem ser associados com títulos que simbolizam status.

Exemplo:

Level	Ranking	Pontos
1	Novato	+50pt
5	Veterano	+150pt
10	Mestre	+300pt
15	Guru	+500pt

Tabela3 - Leveis

Implementação

Deve existir uma lista como a acima para descrever como se dá a evolução dos níveis.

Uma maneira de se implementar essa estrutura seria fazendo com que fosse possível alterar as regras sem que seja necessário recompilar o sistema inteiro. Ou seja, definindo objetos em arquivos XML como o descrito para as regras. Se possível seria interessante colocar ambos no mesmo arquivo.

Diagrama

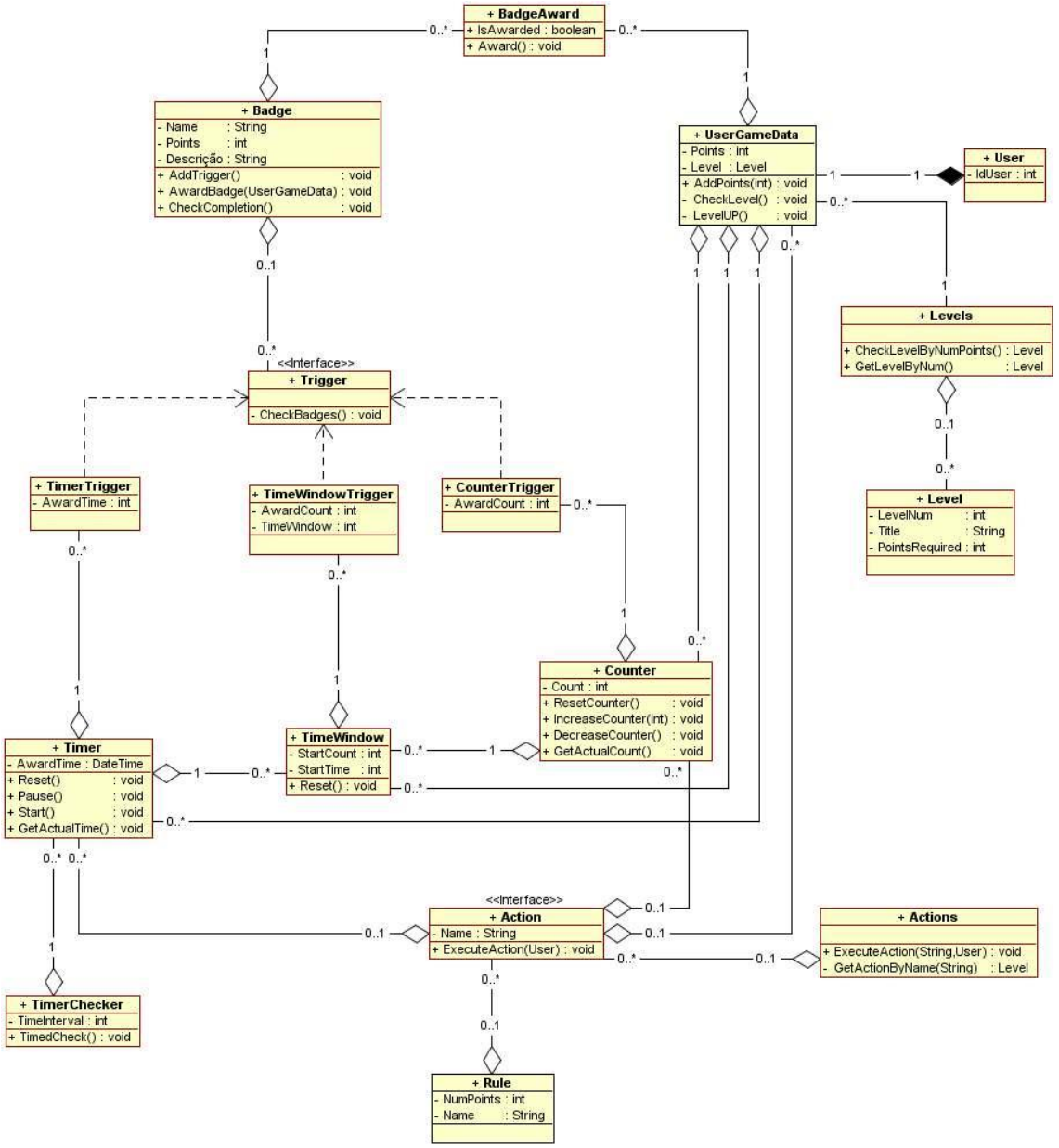

Fig 1 – Diagrama versão 2.0

Tabela – User

Representa apenas um usuário genérico que varia de aplicação para aplicação.

Tabela – UserGameData

Armazena as informações referentes ao usuário, ou seja, a pontuação, os “Badges” e os cronômetros e contadores do usuário, que são representados pelas tabelas Counter, Timer e TimeWindow.

Nela está presente o método responsável pela manipulação da pontuação do usuário e o cálculo do nível do usuário é realizado a cada manipulação dos pontos.

A cada alteração da pontuação do usuário que é realizada através de “AddPoints” o método “CheckLevel” deve ser chamado para verificar se houve aumento do nível do usuário a partir da lista de leveis na tabela “Levels”. Se houver, chama o método “LevelUp”.

O número de pontos passado como parâmetro para tornar o código mais legível deve ser o representado por um dos objetos do tipo “Rule” contido em “Rules”.

Points

Número de pontos do usuário.

Level

Representa o level atual do usuário.

AddPoints

Função responsável por modificar o número de pontos do usuário.

Tabela – Rule

NumPoints

Número de pontos para a regra.

Name

Nome da regra, remete a ação que é acompanhada de premiação de pontos. Exemplo: “Responder a uma pergunta”.

Tabela – Levels

Coleção dos leveis.

Tabela – Level

LevelNum

Número do nível

PointsRequired

Quantidade de pontos requerida para Se alcançar o nível

Title

Título conquistado com o nível

Tabela – Badge

Os objetos de badge e seus triggers não são duplicados para cada usuário, eles existem como uma definição de seus detalhes e todos os triggers que devem ser satisfeitos para a conquista do mesmo. O processo de premiação de um badge para um determinado usuário é realizado através dos contadores e timers instanciados para o usuário que são associados aos triggers dos badges. Ou seja, o processo de premiação ocorre da seguinte maneira:

Premiação de badges através de contadores.

1 - Um contador é incrementado.

2 - Verifica em todos os "CounterTrigger" associados ao contador se foram satisfeitos.

> **Para todos os "**CounterTrigger" **que foram satisfeitos.**
>
> 1 - Checa todos os triggers dos badges associados aos triggers satisfeitos para o usuário em questão.
>
> > **Para os badges que tenham todos os triggers associados satisfeitos.**
> >
> > 1 - O badge é premiado ao usuário.

3 – Verifica todos os "TimeWindowTrigger" associados aos "Counter" se foram satisfeitos.

> **Para todos os** "TimeWindowTrigger" **que foram satisfeitos.**
>
> 1 - Checa todos os triggers dos badges associados aos triggers satisfeitos para o usuário em questão.
>
> > **Para os badges que tenham todos os triggers associados satisfeitos.**
> >
> > 1 - O badge é premiado ao usuário.

Premiação de badges através de Timers.

1 - Um "TimerChecker" detecta que chegou ao seu intervalo estipulado de checagem

2 - Verifica em todos os "TimerTrigger" associados aos "Timer" associados ao "TimerChecker" se foram satisfeitos.

> **Para todos os** "TimerTrigger" **que foram satisfeitos:**
>
> 1 - Checa todos os triggers dos badges associados aos triggers satisfeitos para o usuário em questão.
>
> > **Para os badges que tenham todos os triggers associados satisfeitos.**
> >
> > 1 - O badge é premiado ao usuário.

3 - Verifica em todos os "TimeWindowTrigger" associados aos "Timer" associados ao "TimerChecker" se chegou o momento de resetar.

> **Para todos os** "TimeWindowTrigger" **que extrapolaram sua janela de tempo.**
>
> 1 – Resetar o "TimeWindowTrigger"

Name
Nome do badge.

Points
Quantidade de pontos conquistada com o badge.

Description
Descrição do badge.

AddTrigger
Adiciona um trigger ao badge.

CheckCompletion
Checa se todos os triggers estão satisfeitos.

Tabela – TimerTrigger
O funcionamento dos triggers está descrito na secção de Badges.

AwardTime
É simplesmente o tempo que o timer deve passar para o badge ser premiado.

Tabela – TimeWindowTrigger
O funcionamento dos triggers está descrito na secção de Badges.

AwardCount
É o valor que o contador associado deve alcançar dentro da janela de tempo.

TimeWindow
É o tamanho da janela de tempo.

Tabela – CountTrigger
O funcionamento dos triggers está descrito na secção de Badges.

AwardCount
É o valor que o contador associado deve alcançar.

Tabela – Action
É uma interface que define que características uma atividade deve possuir. Possúi acesso a todos os contadores, timers, regras e as informações de usuário.

Exemplo de Uso:

Em um sistema de perguntas e respostas toda vez que uma pergunta é respondida pelo usuário ele ganha 10 pontos e para as primeiras 20 perguntas respondidas ganha um badge.

Uma “Action” pode ser implementada de tal forma que ao executar o método “ExecuteAction” o contador criado para armazenar a quantidade de vezes que o usuário respondeu perguntas é incrementado e a pontuação definida para a regra “Responder pergunta” é recompensado ao usuário.

ExecuteAction

É o método que é chamado quando determinada atividade é realizada. É implementado a fim de executar as operações pertinentes envolvendo as estruturas de gamification.

Tabela – Actions

Lista de todas as “Action” do sistema.