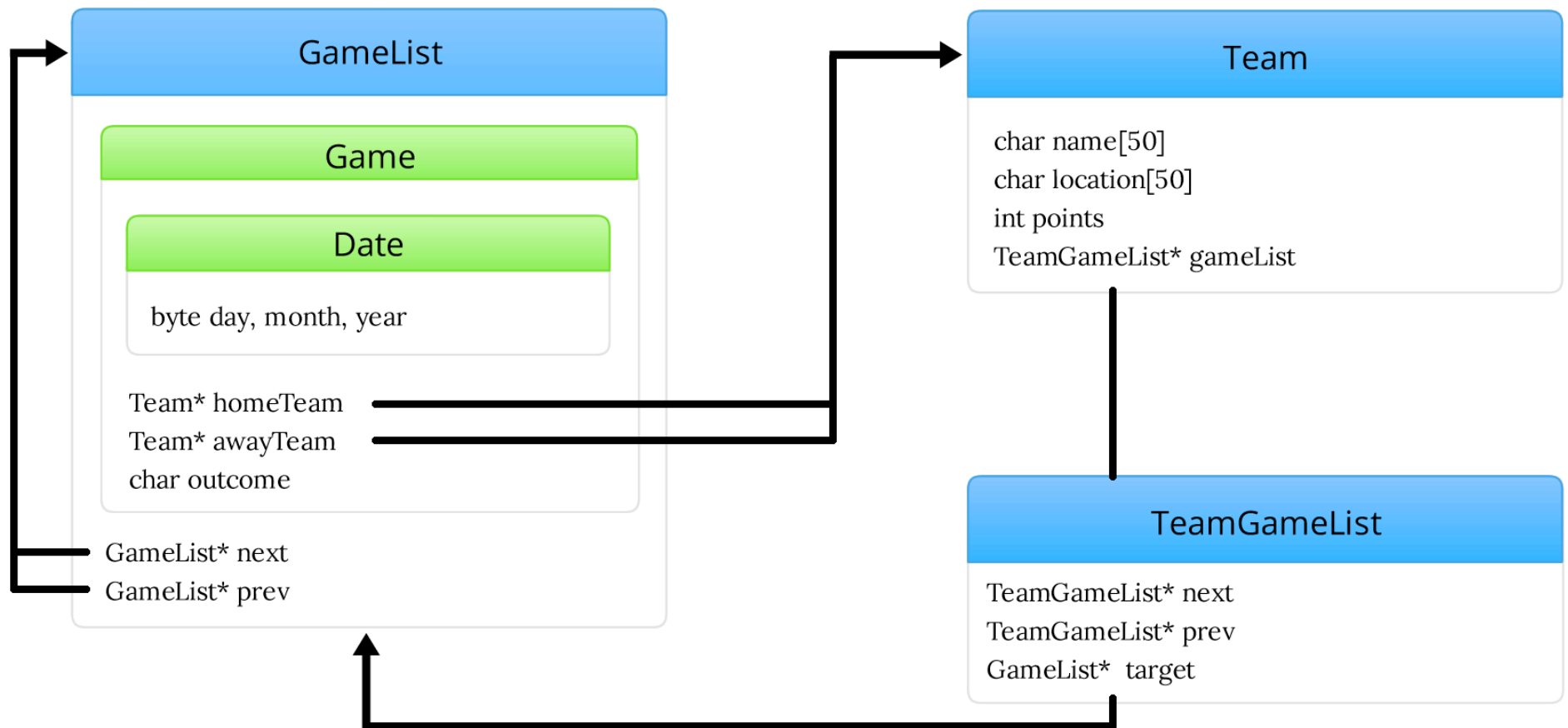


## Gráfico



## Descrição

### Date

Estrutura para armazenamento de datas.

byte day, month, year	Inteiros necessários para especificar uma data
-----------------------	--

### Game

Estrutura para armazenamento de toda a informação relativa a um jogo.

char outcome	Valor dependente do resultado (0 - vitória casa, 1 - empate, 2 derrota casa).
Team* homeTeam	Ponteiro para a equipa a jogar em casa.
Team* awayTeam	Ponteiro para a equipa a jogar fora.
Date date	Data do encontro.

### GameList

Vulgar lista duplamente ligada para variáveis do tipo *Game*. Uma única lista deste tipo será criada no início de execução do programa.

Game game	Jogo pertencente à “node”.
GameList* next	Ponteiro para a próxima “node”.
GameList* prev	Ponteiro para a “node” anterior.

### TeamGameList

Lista duplamente ligada para variáveis do tipo *GameList*. Utilizada para cada equipa, ao invés de *GameList*, para mais fácil remoção de jogos.

GameList* target	Ponteiro para o jogo pertencente à “node”.
TeamGameList* next	Ponteiro para a próxima “node”.
TeamGameList* prev	Ponteiro para a “node” anterior.

## Team

Estrutura para armazenamento de toda a informação relativa a uma equipa, bem como jogos que tenha disputado.

char name[50]	Nome da equipa (máximo 50 caracteres).
char location[50]	Localidade da equipa (máximo 50 caracteres).
int points	Total de pontos da equipa. Calculado <i>on demand</i> .
TeamGameList* gameList	Ponteiro para a lista dos jogos disputados pela equipa. Actualizado <i>on demand</i> .

## Actualização

Em vários elementos das estruturas é mencionada uma actualização *on demand* dos dados. Isto significa que, ao invés de serem actualizados sempre que ocorre uma alteração noutra estrutura (que afecte a em questão), só o sejam após a sua criação inicial, que só ocorrerá para dar resposta a um pedido do utilizador.

Isto trás vários benefícios para o tempo de execução do programa, pois certas instruções só serão efectuadas quando inevitavelmente necessárias. Por outro lado, pode aumentar o tempo de resposta em determinadas condições aos pedidos do utilizador, pois alguns dados não estarão previamente calculados.

## Ideias em Discussão

Foi discutida a possibilidade da utilização de uma lista duplamente ligada genérica (fazendo uso de void pointers), permitindo evitar a criação de vários tipos de listas ligadas e repetição de código.

A sua implementação ou não ainda está em discussão.

## Trabalho Realizado Por

João Afonso Libório Cardoso, 2011151968

João Ricardo Lourenço, 2011151194