A Player Starts a New Game? Call `player_new_game`.

```
int32_t player_new_game
    (struct player_t* p,
     struct game_t* g)
{
    if (NULL != p->game) {
        return 0;
    }
    p->game = g;
    p->num_games++;
    return 1;
}
```

the player

the new game

Start with error checking: is the player already in a game?

A Player Starts a New Game? Call `player_new_game`.

```
int32_t player_new_game
    (struct player_t* p,
     struct game_t* g)
{
    if (NULL != p->game) {
        return 0;
    }
    p->game = g;
    p->num_games++;
    return 1;
}
```

Update fields as necessary to reflect new game starting.

Return success.

A Player Finishes a Game? Call `player_finish_game`.

```
int32_t player_finish_game
    (struct player_t* p,
     int32_t score)
{
    if (NULL == p->game) {
        return 0;
    }
    p->game = NULL;
    p->score_dist[score_to_bin (score)]++;
    return 1;
}
```

the player

the score of the finished game

A Player Finishes a Game? Call `player_finish_game`.

```
int32_t player_finish_game
    (struct player_t* p,
     int32_t score)
{
    if (NULL == p->game) {
        return 0;
    }
    p->game = NULL;
    p->score_dist[score_to_bin (score)]++;
    return 1;
}
```

Start with error checking: is the player not in a game?

Update **game** field.

Use a helper function to find the right bin.

Return success.

