

```

249  /// <summary>
250  /// get BattleAction choice from the Dalek and the Traveler
251  /// calculate the BattleResult based on the BattleActions and
252  /// return it
253  /// </summary>
254  /// <returns>Battle Result</returns>
255  private BattleResult Battle()
256  {
257      BattleResult battleResult = BattleResult.None;
258      Dalek dalekToBattle;
259      BattleAction travelerBattleActionChoice;
260      BattleAction dalekBattleActionChoice;
261      Random random = new Random();
262
263      dalekToBattle = _gameConsoleView.DisplayGetDalekToBattle();
264
265      if (dalekToBattle != null)
266      {
267          travelerBattleActionChoice = _gameConsoleView.DisplayGetBattleActionChoice();
268          dalekBattleActionChoice = dalekToBattle.GetBattleAction();
269
270          if (travelerBattleActionChoice == BattleAction.Retreat
271              & dalekBattleActionChoice == BattleAction.Retreat)
272          {
273              battleResult = BattleResult.BothRetreat;
274          }
275          else if (travelerBattleActionChoice == BattleAction.Attack
276                  & dalekBattleActionChoice == BattleAction.Retreat)
277          {
278              battleResult = BattleResult.NPCRetreats;
279          }
280          else if (travelerBattleActionChoice == BattleAction.Retreat
281                  & dalekBattleActionChoice == BattleAction.Attack)
282          {
283              battleResult = BattleResult.TravelerRetreats;
284          }
285          else if (travelerBattleActionChoice == BattleAction.Attack
286                  & dalekBattleActionChoice == BattleAction.Attack)
287          {
288              int travelerBattleNumber = random.Next(1, 100) * _gameTraveler.BattleIndex;
289              int dalekBattleNumber = random.Next(1, 100) * dalekToBattle.BattleIndex;
290
291              if (travelerBattleNumber > dalekBattleNumber)

```

```

292         {
293             battleResult = BattleResult.TravelerWins;
294         }
295         else
296         {
297             battleResult = BattleResult.NPCWins;
298         }
299     }
300 }
301
302 ProcessBattleResult(battleResult, daleckToBattle);
303
304 return battleResult;
305 }
306
307 /// <summary>
308 /// perform the required tasks based on the battle results
309 /// </summary>
310 /// <param name="battleResult">result of the battle</param>
311 /// <param name="daleckToBattle">dalek battled</param>
312 private void ProcessBattleResult(BattleResult battleResult, Dalek daleckToBattle)
313 {
314     switch (battleResult)
315     {
316         case BattleResult.TravelerWins:
317             daleckToBattle.SpaceTimeLocationID = 0; // Daleck removed from game
318             break;
319         case BattleResult.NPCWins:
320             _gameTraveler.Lives--;
321             break;
322         case BattleResult.TravelerRetreats:
323         case BattleResult.NPCRetreats:
324         case BattleResult.Draw:
325         case BattleResult.BothRetreat:
326             break;
327         default:
328             break;
329     }
330 }

```