

# Real-life Pokémon Game

## 1. Overview

The concept of this game is to combine the Pokémon catching and battling mechanisms from the popular Pokémon game series, with the GPS capability of smartphones to create a competitive team-based, real-life, location-aware Pokémon game.

In the Pokémon games, the various Pokémon species have distinct habitats in which they can only be found. This will be reflected in our game as well, as Pokémon will only roam a given region of the real-life playing field.

Each team will consist of a number of Pokémon trainers and a team commander. The trainers will roam the playing field equipped with smartphones and will encounter the Pokémon. The commander has a computer which shows him an overview of the playing field, with the locations of his team members and the Pokémon. He will instruct his team on where to go and catch specific Pokémon. Two or more teams will compete with each other in this fashion to complete a certain goal.

## 2. Pokémon background

Pokémon, or Pocket Monsters, are creatures caught and trained by Pokémon trainers to fight each other in Pokémon battles.

Throughout the lifetime of the Pokémon franchise, the number of Pokémon has steadily increased, as well as their attributes used in the games. Currently there are 649 Pokémon species. For our game, we will stick to (a subset of) the original 151 Pokémon species (also referred to as "the first generation") and the original set of Pokémon attributes and statistics used in the Pokémon battles.

### 2.1 Pokémon battles

Pokémon battles are battles between either two trainers or a trainer and a wild Pokémon. Battles proceed in a turn based fashion. Each trainer involved brings out his first Pokémon, after which the battle begins.



*Pokémon battle interface from one of the original games*

During his turn, a trainer has 4 options, from which he can only perform one:

- **Fight:** Have the current Pokémon perform a move. A Pokémon can have up to 4 moves to select from. Each move can only be used a certain number of times: stronger moves can be used less often.
- **Change Pokémon:** Call back the current Pokémon and replace him with another one.
- **Use an item:** Use an item on one of the Pokémon involved in the battle. This could for example be a potion to heal your own Pokémon or a PokéBall to try and catch a wild Pokémon. In our game we will probably leave out all items except PokéBalls.
- **Run:** When fighting a wild Pokémon, the trainer can decide to try to run to avoid being defeated. Some formula will decide if the attempt to run was successful or not. If not, the fight continues. Trainers can never run during a duel with another trainer.

The goal of the battle is to defeat all Pokémon of the opponent. A trainer can have up to six Pokémon with him. Pokémon perform moves, which reduce the Health Points of their opponent. When a Pokémon runs out of Health Points, it faints and can no longer partake in the battle.

## 2.2 Catching Pokémon

When a trainer encounters a wild Pokémon, a battle ensues, during which he can try to catch the Pokémon. To do so, he needs to use a PokéBall. A catch formula will decide whether the Pokémon will be caught. Important factors in this formula are the Health Points of the Pokémon and the type of PokéBall used. Therefore, the trainer first weakens the wild Pokémon before attempting the catch.

## 2.3 Training Pokémon

Pokémon that partake in battles receive experience points (unless they faint). After having gained enough experience, they will gain a level, until they reach level 100. Levelling up will increase their statistics and occasionally they will learn a new move when they gain a level. Certain Pokémon will also "evolve" into another Pokémon species, changing their appearance and making them stronger.

## 2.4 Pokémon types

Each Pokémon has one or two types, from a possible range of fifteen types (e.g. Fire, Water,

Grass, Normal, Fighting, Flying, ...). Pokémon moves will also be of a certain type. Types determine the strengths and weaknesses of a species. Types offset each other in a rock-paper-scissors relationship. Attacks of given type will be super effective against Pokémon of a certain type, while inflicting no damage to Pokémon of another type.

## 2.5 Pokémon statistics

Each Pokémon has a number of statistics used in the battle algorithms. These improve when a Pokémon levels up.

- **Health Points:** How much damage a Pokémon can take.
- **Attack:** Strength of "physical" attacks (Moves of type Normal, Fighting, Ground, ...)
- **Defense:** Defense against "physical" attacks
- **Special:** Strength of and defense against "special" attacks (Moves of type Fire, Water, Dragon, ...)
- **Speed:** Determines which Pokémon will attack first after having received their commands during a battle.

Pokémon also have two other statistics: Accuracy and Evasiveness, which together with the Accuracy attribute of the chosen move, will determine if a Pokémon will hit or miss his opponent during a battle.

Certain Pokémon moves used during battles can temporarily increase or decrease any of these statistics.

## 3. Game play

### 3.1 Core game

The players are divided into teams (at least 2, but in theory this game could scale to more teams). Within each team (2 to 5 players), one team commander is chosen. The other players will be the Pokémon trainers.

The Pokémon trainers will have a smartphone and will choose a starting Pokémon. During the course of the game, they will roam a predefined playing field, a part of campus 200. The team commanders will have a stationary position.

Pokémon will also roam this playing field, controlled by a central game server. Team commanders will have a map in their client software, showing the locations of these Pokémon, as well as the locations of their team members, which are obtained through the GPS of the trainer's smartphone.

Team commanders can communicate with their team members through messages and send them instructions on where to find particular Pokémon.

When a Pokémon trainer comes in the vicinity (5m) of one of the wild Pokémon, the central game server will initiate a Pokémon battle, similar to those from the original Pokémon games. Pokémon trainers can then try to catch the encountered Pokémon. Caught Pokémon can then be used by the trainer.

Different parts of the playing field will be populated by Pokémon of different species and different levels.

A designated position on the playing field will serve as a “PokéCenter”, where trainers Pokémon will be healed if they enter its vicinity.

In its most basic form, the goal of the game would be to catch a given number of Pokémon. The first team to succeed would win. To add a bit of strategy, the Pokémon that need to be caught could be chosen as such that they are of a certain type or level that would require the players to first level up their Pokémon, or catch Pokémon of a particular type that is effective against the target Pokémon, because they would otherwise not be capable of defeating that Pokémon.

## **3.2 Possible extensions**

These extensions to the core game could be considered if time permits and/or if the core game is not compelling enough.

### **3.2.1 Trainer duels**

Another key aspect of the Pokémon universe are the duels between Pokémon trainers. These could be added to the game as duels between trainers of opposing teams. These could then result in bonus points for the winner’s team, or in the winner stealing (one of) the Pokémon caught by the loser, or setting them free instead of stealing them.

### **3.2.2 Quests**

Instead of each having each team compete to be the first to catch a designated set of Pokémon, the central game server could issue a number of quests. These quests would require the team to catch some Pokémon, defeat a particularly strong Pokémon, ... Successfully completing a quest would be rewarded with victory points. Quests are not issued to a specific team, so the first team to complete it would get the points. Harder quests result in more points.

The first team to gather a target number of victory points wins. At all times there will be a number of quests available. Team commanders have to decide which quests they will try to compete, not knowing which quests the other team is trying to fulfill.

## **4. Game components**

### **4.1 Client for Pokémon trainers**

The trainer client needs to provide the following functionality:

- An overview of the trainer’s Pokémon.

- A map showing the trainers position and the location he needs to go (which is provided by the team commander).
- Capable of receiving instructions from the commander (text messages and locations)
- Periodically send GPS location data to the server.
- Taking care of encounters with Pokémon: providing a battle UI, evaluating battle algorithms and informing the server of the battle results.

## 4.2 Client for team commanders

The team commander clients needs to provide the following functionality:

- An overview of the team's Pokémon.
- The team's progress towards the victory conditions.
- A map showing the locations of the Pokémon and the team's members.
- A way to send instructions to team members (text messages and locations)
- A way to set up the game

## 4.3 Central server for game logic and Pokémon control

The central server will be responsible for:

- Controlling the Pokémon movement.
- Collecting trainer location data and relaying it to the commanders.
- Initiating battles between trainers and Pokémon.
- Tracking the progress of each team and determining when the game should end.

# 5. Technology

## 5.1 Android for trainer client

For the trainer's smartphones we chose the Android platform. The Android client will use the built-in GPS of the smartphone to send location data to the central server. It will then receive a response indicating if a Pokémon has been encountered. The battle itself will be taken care of locally on the Android client, the results will be sent to the server. The server will notify the trainer client if a new instruction from the commander is available, after which the client will fetch it from the server.

Resolving the Pokémon encounters locally on the client allows us to make the battle mechanisms as complex as we desire. Initially this could be limited to only a simple "Attack" option instead of the Pokémon's various moves, which should enable us to get things up and running quickly for a first prototype.

The various algorithms used in the original Pokémon games are available online. But whereas

the original game is intended to be played over a prolonged period of time, ours is not, so we might need to tweak for example the levelling algorithm, to allow Pokémon to level up faster.

Furthermore, we have also found two other Android Pokémon projects which implement the battle component. These could perhaps provide reusable code (if this is allowed at least):

- Pokédroid (<http://sourceforge.net/projects/pokedroid/>)
  - Standalone Pokémon game
  - Useable algorithms, although with some bugs. Can be used as a basis for developing our own algorithms.
  - Parts of UI / graphics can perhaps be reused.
- Pokémon Online Android client (<http://code.google.com/p/pokemon-online-android/>)
  - Client for an online multiplayer Pokémon game
  - Has an SQLite database with Pokémon information
  - Parts of UI could perhaps be reused.

Should we wish to extend the game with trainer duels, these battles will require interaction between two Android clients and thus will need to pass information to each other via the server (An alternative would be a peer-to-peer connection between the smartphones, which would complicate things).

## 5.2 Web app for commander client

For the commander client we chose to develop a web app, as all the data for the commander will need to be fetched from the server anyway. This also allows us to use it on any computer with an internet connection. Instead of running it on a stationary computer/laptop, we could also use it on a tablet, allowing the commander to be mobile as well.

For the map part of the commander client, the Google Maps API can be used. New location data will be pushed to the client by the server. Instructions for the trainer clients will be sent to the server, which relays them to the trainers.

## 5.3 App Engine for central server

For the central game server we chose Google App Engine, which allows to easily create a Java backend. The App Engine server will expose a REST API to the clients, through which they can push and pull data using HTTP requests. Data will be exchanged between server and client using JSON. The trainer client will use this API to push location data. The commander client can use it to send instructions to the trainer clients.

Google App Engine also provides something called the Channel API. This allows to set up a persistent connection between the server and the commander web app, useful to push Pokémon and player location data in real-time to the web app.

The Channel API can not be used to communicate with the Android clients (at least not without a hack), but Google's Cloud 2 Device Messaging framework provides an alternative. It can be used to push notifications to Android clients, which can then pull new data from the server. This is useful to inform the trainer clients of new instructions sent by the commander client.

The server maintains a database of all Pokémon with their locations and all trainers with their Pokémon. It will implement the rules of the game and check whether any of the victory conditions have been met. Using cron jobs, the Pokémon will periodically be moved about the playing field.

## 6. Planning

The three game components will be developed in parallel. The idea will be to incrementally add basic functionality to each of the components until a first prototype has been created. This can then be tested, after which features can be expanded (or if time permits, new features can be added).

How the work will be divided amongst the four team members still needs to be decided.

WEEK	ANDROID CLIENT	WEB APP	SERVER
13/02 - 19/02	Overall UI structure	Overall UI structure	Database for Pokémon
20/02 - 26/02	UI for Pokémon encounters	Map view with fake location data	Pokémon movement
27/02 - 04/03	UI for Pokémon encounters + basic battle algorithms	Map view with Pokémon location data from server	MILESTONE: Pokémon move around ----- Push Pokémon location data to Web App
05/03 - 11/03	MILESTONE: Basic Pokémon battles possible ----- Collecting location data	MILESTONE: Pokémon movement can be visualized ----- UI for sending instructions to trainers	Infrastructure for receiving location data from Android, for receiving instructions from web app and for Android to retrieve messages
12/03 - 18/03	Sending location data to server + receiving Pokémon encounters from server	Trainer positions on map view	Pushing trainer location data to web app and sending notifications to Android
19/03 - 25/03	Retrieve messages sent by commander from server + UI for instructions	UI for game setup + game progress	Game logic (team setup, goal setup)

26/03 - 01/04	UI for instructions	UI for game setup / progress + hook it up to server	Game logic (checking victory conditions)
02/04 - 08/04	MILESTONE: 1st basic working prototype ----- Test the prototype	MILESTONE: 1st basic working prototype ----- Test the prototype	MILESTONE: 1st basic working prototype ----- Test the prototype
09/04 - 15/04	Implement Pokémon levelling up + advanced battle algorithm	Optionally: add UI for quests	Optionally: more advanced game logic (quests)
16/04 - 22/04	Implement Pokémon levelling up + advanced battle algorithm	Optionally: add UI for quests + hook it up to server	Optionally: more advanced game logic (quests) + trainer duels
23/04 - 29/04	Implement trainer duels		Optionally: trainer duels
30/04 - 06/05	MILESTONE: 2nd more advanced prototype ----- Test the prototype	MILESTONE: 2nd more advanced prototype ----- Test the prototype	MILESTONE: 2nd more advanced prototype ----- Test the prototype
07/05 - 13/05	Finetuning and more testing	Finetuning and more testing	Finetuning and more testing
14/05 - 20/05	BUFFER	BUFFER	BUFFER
21/05 - 27/05	BUFFER	BUFFER	BUFFER