### Peripheral awareness

# Atención periférica

- Mientras tenemos el foco en algo puntual, seguimos prestando atención en segundo plano a lo que pasa en nuestra periferia.
- Cómo diseñamos para "la periferia" del foco de atención del usuario?

#### **Sublime**

```
return build (YOUTUBE API SERVICE NAME, YOUTUBE API VERSION,
73
74
              http=credentials.authorize(httplib2.Http()))
75
76
      # Call the API's captions.list method to list the existing caption tracks.
79
      def list_captions(youtube, video_id):
        results = youtube.captions().list(
81
          part="snippet",
82
          videoId=video_id
83
        ).execute()
84
85
        for item in results["items"]:
          id = item["id"]
87
          name = item["snippet"]["name"]
          language = item["snippet"]["language"]
          print("Caption track '%s(%s)' in '%s' language." % (name, id, language))
        return results["items"]
91
92
      # Call the API's captions.insert method to upload a caption track in draft status.
94
      def upload_caption(youtube, video_id, language, name, file):
        current_captions = youtube.captions().list(
          part="snippet",
          videoId=video id
          ) execute()
        # print(f"LIst of captions: {current_captions}")
        if 'Author captions' in json.dumps(current_captions):
101
          print(f"Caption already exists for {video_id}")
102
        else:
103
          print(f"Uploading {file} for {video_id}")
104
          try:
            insert_result = youtube.captions().insert(
              part="snippet",
              body=dict(
107
108
                snippet=dict(
109
                  videoId=video id,
110
                  language=language,
111
                  name=name
112
              ),
113
              media_body=file
114
            ).execute()
115
          except:
            print(f"cap already exists {current_captions}")
116
117
          else:
118
            id = insert_result["id"]
            name = insert_result["snippet"]["name"]
119
            language = insert_result["snippet"]["language"]
120
            status = insert_result["snippet"]["status"]
121
122
            print("Uploaded caption track '%s(%s) in '%s' language, '%s' status." % (name,
                id, language, status))
123
124
```

```
Part Service Company of the Company
```

```
return build (YOUTUBE API SERVICE NAME, YOUTUBE API VERSION,
73
74
              http=credentials.authorize(httplib2.Http()))
75
76
      # Call the API's captions.list method to list the existing caption tracks.
79
      def list_captions(youtube, video_id):
        results = youtube.captions().list(
81
          part="snippet",
82
          videoId=video_id
83
        ).execute()
84
85
        for item in results["items"]:
          id = item["id"]
87
          name = item["snippet"]["name"]
          language = item["snippet"]["language"]
          print("Caption track '%s(%s)' in '%s' language." % (name, id, language))
        return results["items"]
91
92
      # Call the API's captions.insert method to upload a caption track in draft status.
94
      def upload_caption(youtube, video_id, language, name, file):
        current_captions = youtube.captions().list(
          part="snippet",
          videoId=video id
          ) execute()
        # print(f"LIst of captions: {current_captions}")
        if 'Author captions' in json.dumps(current_captions):
101
          print(f"Caption already exists for {video_id}")
102
        else:
103
          print(f"Uploading {file} for {video_id}")
104
          try:
            insert_result = youtube.captions().insert(
              part="snippet",
              body=dict(
107
108
                snippet=dict(
109
                  videoId=video id,
110
                  language=language,
111
                  name=name
112
              ),
113
              media_body=file
114
            ).execute()
115
          except:
            print(f"cap already exists {current_captions}")
116
117
          else:
118
            id = insert_result["id"]
            name = insert_result["snippet"]["name"]
119
            language = insert_result["snippet"]["language"]
120
            status = insert_result["snippet"]["status"]
121
122
            print("Uploaded caption track '%s(%s) in '%s' language, '%s' status." % (name,
                id, language, status))
123
124
```

```
Part Service Company of the Company
```

```
return build(YOUTUBE_API_SERVICE_NAME, YOUTUBE_API_VERSION,
73
74
              http=credentials.authorize(httplib2.Http()))
75
76
      # Call the API's captions.list method to list the existing caption tracks.
      def list_captions(youtube, video_id):
        results = youtube.captions().list(
80
81
          part="snippet",
82
          videoId=video_id
83
        ) execute()
85
        for item in results["items"]:
          id = item["id"]
87
          name = item["snippet"]["name"]
          language = item["snippet"]["language"]
          print("Caption track '%s(%s)' in '%s' language." % (name, id, language))
        return results["items"]
90
91
92
      # Call the API's captions.insert method to upload a caption track in draft status.
      def upload_caption(youtube, video_id, language, name, file):
94
95
        current_captions = youtube.captions().list(
          part="snippet",
          videoId=video id
          ) execute()
        # print(f"LIst of captions: {current_captions}")
99
        if 'Author captions' in json.dumps(current_captions):
100
          print(f"Caption already exists for {video_id}")
101
102
        else:
          print(f"Uploading {file} for {video_id}")
103
104
            insert_result = youtube.captions().insert(
105
106
              part="snippet",
107
              body=dict(
108
                snippet=dict(
109
                  videoId=video_id,
110
                  language=language,
111
                  name=name
112
              ),
113
              media_body=file
114
            ) execute()
115
          except:
116
            print(f"cap already exists {current_captions}")
117
          else:
            id = insert_result["id"]
118
            name = insert_result["snippet"]["name"]
119
            language = insert_result["snippet"]["language"]
120
            status = insert_result["snippet"]["status"]
121
            print("Uploaded caption track '%s(%s) in '%s' language, '%s' status." % (name,
122
123
                id, language, status))
124
```

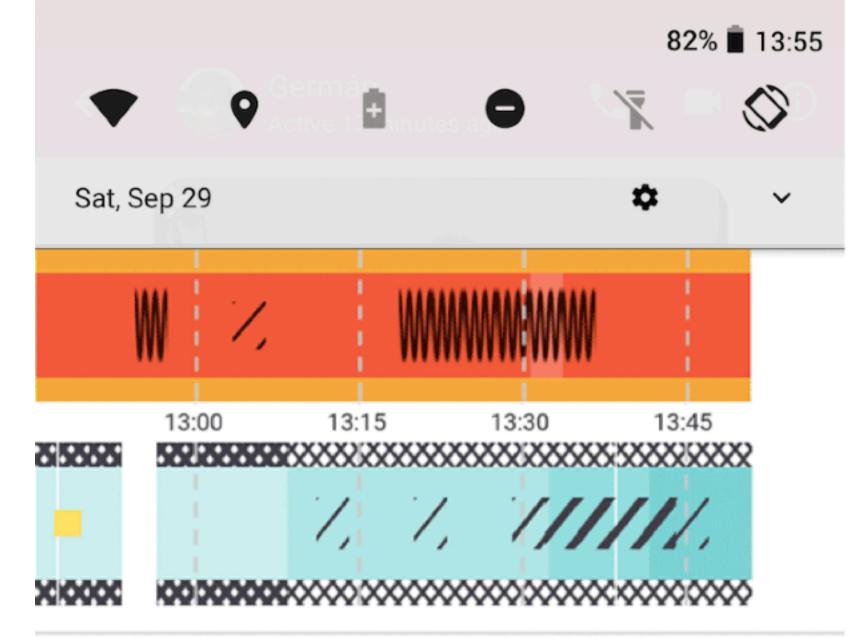
```
de name manage de la company d
```

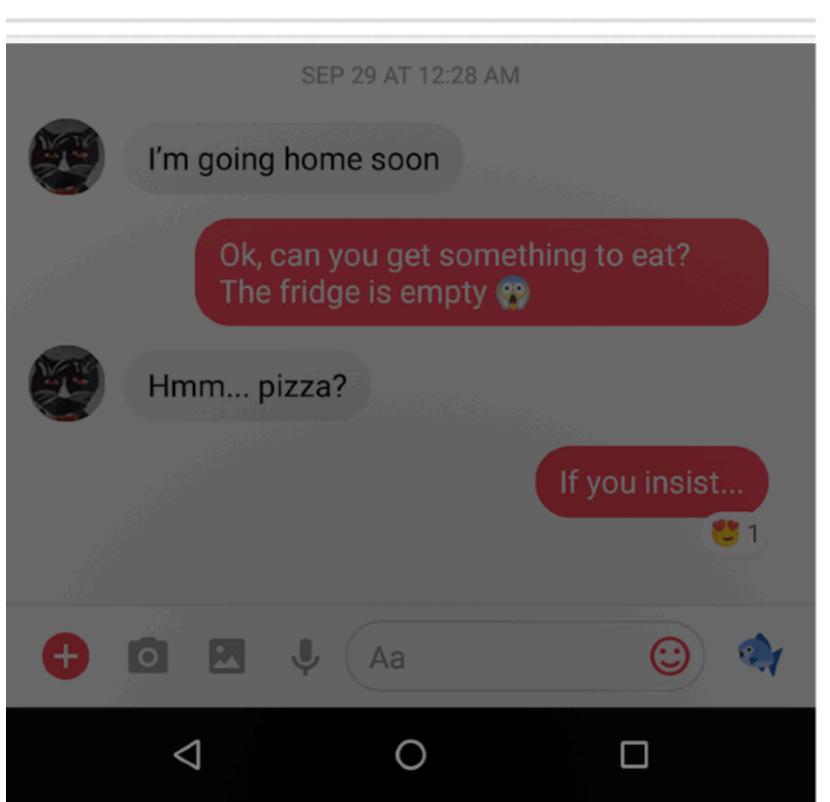
# Atención periférica

#### Peripheral awareness

- Mientras tenemos el foco en algo puntual, seguimos prestando atención en segundo plano a lo que pasa en nuestra periferia.
- Cómo diseñamos para "la periferia" del foco de atención del usuario?

```
http=credentials.authorize(httplib2.Http()))
      # Call the API's captions.list method to list the existing caption tracks.
      def list_captions(youtube, video_id):
        results = youtube.captions().list(
          videoId=video_id
        ) execute()
        for item in results["items"]:
           name = item("snippet")("name")
           language = item["snippet"]["language"]
          print("Caption track '%s(%s)' in '%s' language." % (name, id, language))
      # Call the API's captions.insert method to upload a caption track in draft status.
      def upload_caption(youtube, video_id, language, name, file):
        current_captions = youtube.captions().list(
          part="snippet",
          videoId=video_id
         # print(f"LIst of captions: {current_captions}")
        if 'Author captions' in json.dumps(current_captions):
          print(f"Caption already exists for {video_id}")
          print(f"Uploading {file} for {video_id}")
            insert_result = youtube.captions().insert(
              part="snippet",
                   videoId=video_id,
                   language=language
                   name=name
            ).execute()
            print(f"cap already exists {current_captions}")
            id = insert_result["id"]
            name = insert_result["snippet"]["name"]
                  age = insert_result["snippet"]["language"]
121
            status = insert_result["snippet"]["status"]
122
            print("Uploaded caption track '%s(%s) in '%s' language, '%s' status." % (name,
123
                id, language, status))
124
```





Visualizaciones de "información contextual" para parejas

## Lifelines

Griggio, Nouwens, McGrenere and Mackay. CHI 2019. <a href="https://doi.org/10.1145/3290605.3300853">https://doi.org/10.1145/3290605.3300853</a>