

Peripheral nerves

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?

subline

```

73     return build(YOUTUBE_API_SERVICE_NAME, YOUTUBE_API_VERSION,
74                 http=credentials.authorize(httplib2.Http()))
75
76
77
78 # Call the API's captions.list method to list the existing caption tracks.
79 def list_captions(youtube, video_id):
80     results = youtube.captions().list(
81         part="snippet",
82         videoId=video_id
83     ).execute()
84
85     for item in results["items"]:
86         id = item["id"]
87         name = item["snippet"]["name"]
88         language = item["snippet"]["language"]
89         print("Caption track '%s(%s)' in '%s' language." % (name, id, language))
90     return results["items"]
91
92
93 # 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):
95     current_captions = youtube.captions().list(
96         part="snippet",
97         videoId=video_id
98     ).execute()
99     # print(f"List of captions: {current_captions}")
100     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:
105             insert_result = youtube.captions().insert(
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:
118             id = insert_result["id"]
119             name = insert_result["snippet"]["name"]
120             language = 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))

```



```

73     return build(YOUTUBE_API_SERVICE_NAME, YOUTUBE_API_VERSION,
74                 http=credentials.authorize(httplib2.Http()))
75
76
77
78 # Call the API's captions.list method to list the existing caption tracks.
79 def list_captions(youtube, video_id):
80     results = youtube.captions().list(
81         part="snippet",
82         videoId=video_id
83     ).execute()
84
85     for item in results["items"]:
86         id = item["id"]
87         name = item["snippet"]["name"]
88         language = item["snippet"]["language"]
89         print("Caption track '%s(%s)' in '%s' language." % (name, id, language))
90     return results["items"]
91
92
93 # 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):
95     current_captions = youtube.captions().list(
96         part="snippet",
97         videoId=video_id
98     ).execute()
99     # print(f"List of captions: {current_captions}")
100     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:
105             insert_result = youtube.captions().insert(
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:
118             id = insert_result["id"]
119             name = insert_result["snippet"]["name"]
120             language = 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))

```

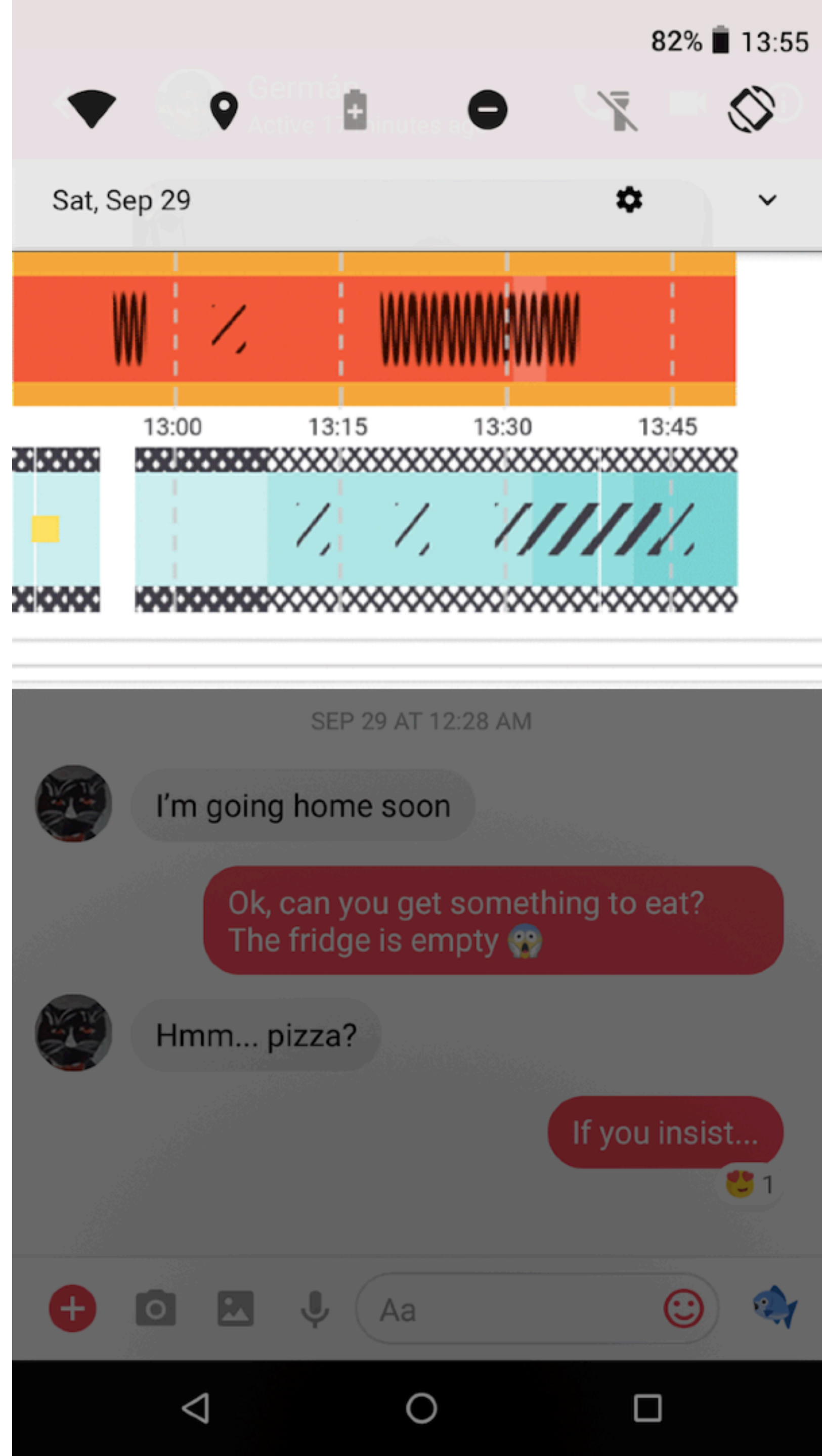


Atención periférica

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?

```
73     return build(YOUTUBE_API_SERVICE_NAME, YOUTUBE_API_VERSION,
74                  http=credentials.authorize(httplib2.Http()))
75
76
77
78 # Call the API's captions.list method to list the existing caption tracks.
79 def list_captions(youtube, video_id):
80     results = youtube.captions().list(
81         part="snippet",
82         videoId=video_id
83     ).execute()
84
85     for item in results["items"]:
86         id = item["id"]
87         name = item["snippet"]["name"]
88         language = item["snippet"]["language"]
89         print("Caption track '%s(%s)' in '%s' language." % (name, id, language))
90     return results["items"]
91
92
93 # 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):
95     current_captions = youtube.captions().list(
96         part="snippet",
97         videoId=video_id
98     ).execute()
99     # print(f"List of captions: {current_captions}")
100     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:
105             insert_result = youtube.captions().insert(
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:
118             id = insert_result["id"]
119             name = insert_result["snippet"]["name"]
120             language = 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
125
```



Visualizaciones de “información contextual”
para parejas

Lifelines

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