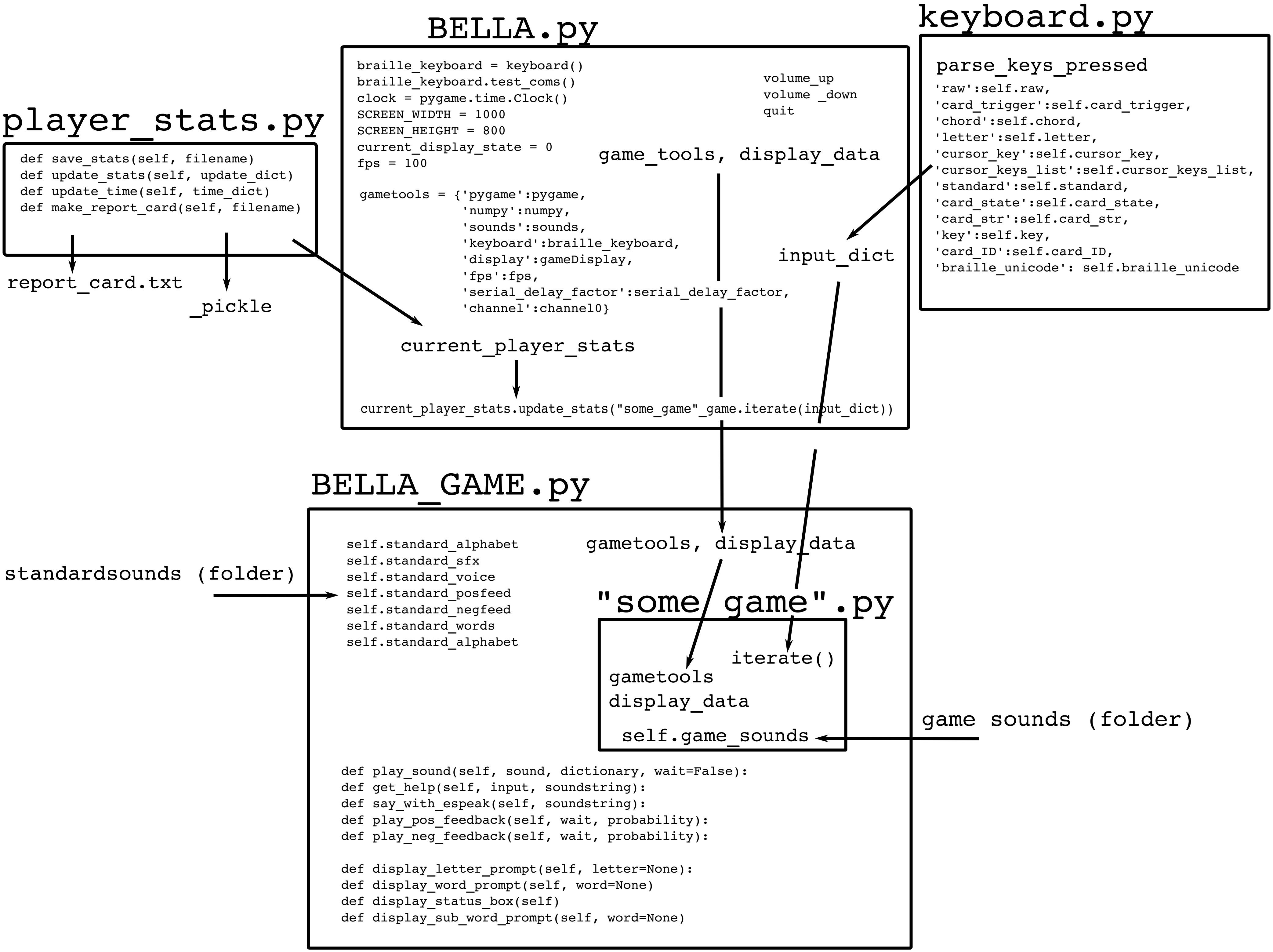

BELLA.py

```
braille_keyboard = keyboard()
braille_keyboard.test_coms()
clock = pygame.time.Clock()
SCREEN_WIDTH = 1000
SCREEN_HEIGHT = 800
current_display_state = 0
fps = 100

gametools = {'pygame':pygame,
             'numpy':numpy,
             'sounds':sounds,
             'keyboard':braille_keyboard,
             'display':gameDisplay,
             'fps':fps,
             'serial_delay_factor':serial_delay_factor,
             'channel':channel0}
```

```
volume_up
volume _down
quit
```

game_tools, display_data

input_dict

current_player_stats

```
current_player_stats.update_stats("some_game"_game.iterate(input_dict))
```

keyboard.py

parse_keys_pressed

```
'raw':self.raw,
'card_trigger':self.card_trigger,
'chord':self.chord,
'letter':self.letter,
'cursor_key':self.cursor_key,
'cursor_keys_list':self.cursor_keys_list,
'standard':self.standard,
'card_state':self.card_state,
'card_str':self.card_str,
'key':self.key,
'card_ID':self.card_ID,
'braille_unicode': self.braille_unicode
```

player_stats.py

```
def save_stats(self, filename)
def update_stats(self, update_dict)
def update_time(self, time_dict)
def make_report_card(self, filename)
```

report_card.txt

_pickle

BELLA_GAME.py

standardsounds (folder)

```
self.standard_alphabet
self.standard_sfx
self.standard_voice
self.standard_posfeed
self.standard_negfeed
self.standard_words
self.standard_alphabet
```

gametools, display_data

"some game".py

iterate()

gametools

display_data

self.game_sounds

game sounds (folder)

```
def play_sound(self, sound, dictionary, wait=False):
def get_help(self, input, soundstring):
def say_with_espeak(self, soundstring):
def play_pos_feedback(self, wait, probability):
def play_neg_feedback(self, wait, probability):

def display_letter_prompt(self, letter=None):
def display_word_prompt(self, word=None)
def display_status_box(self)
def display_sub_word_prompt(self, word=None)
```