IPPAS rapportage \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Naam student: Bram van Leusden

Student nummer: 1779320

**Probleemomschrijving**

Mijn ouders hebben tijdens de corona Lock down erg weinig te doen. Het maken van kruiswoordwoordpuzzels vinden zij erg leuk alleen de krant heeft er niet superveel. Mij ouders willen een graag een oplossing voor de schaarste aan kruiswoordpuzzels.

**Eisen Product**

* **Puzzel word binnen 10 seconden ingeladen**
* **De puzzel word weergegeven in een overzichtelijke user interface**
* **Iedere kruiwoord puzzel word random gegenereerd**

**Gekozen (eigen)algoritme**

Ik genereer een pseudorandom kruiswoordpuzzel die ik vul met woorden die kunnen kruisen op de gegeven indexen. Eerst genereer ik coördinaten voor één woord met een random lengte en check ik of deze geldig zijn. Als de coördinaten geldig zoek ik naar een word om te plaatsen op de coördinaten. Als er geen woord is gevonden probeert de code het opnieuw met andere coördinaten en ga ik zo door totdat de puzzel is gegenereerd met het gevraagde aantal woorden. Als dit proces te lang duurt haal ik alle geplaatste woorden weg en probeer het hele voorheen genoemde proces opnieuw.

**Toelichting afwijkingen**

Tijdens het testen van het originele algoritme kwam ik er achter dat het niet ideaal werkte. In plaats van het originele geplande algoritme te gebruiken besloot ik het algoritme dat hierboven beschreven is te gebruiken. Het nieuwe algoritme geeft de gebruiker sneller kruiswoord puzzel en heeft een grotere kans het in één keer goed te doen. Het oude algoritme genereerde alle coördinaten in één keer. Dit zorgde er voor dat er op één of meerdere coördinaten geen word kon worden geplaats en dat het hele proces opnieuw kon beginnen, hierdoor was de code heel of zelfs oneindig lang bezig met het generen van een puzzel. Desondanks het wisselen van algoritme kon ik veel code hergebruiken en kun ik mijn PVA planning normaal volgen sinds de stappen van het algoritme over het algemeen in ander volgorde staan.

**Planning in PVA**

1. **Woorden uit het woordenboek krijgen (5 uur)**
2. **Kruiswoordpuzzel genereren (12 uur tot)**
   1. Individuele cellen genereren (4 uur)
   2. Kruisingen tussen cellen maken (8 uur)
3. **Kruiswoordpuzzel vullen met woorden (17 uur tot)**
   1. Woord en cel met overeenkomende lengte koppelen (5 uur)
   2. Woorden kunnen met elkaar kruisen waar nodig is (12 uur)
4. **Grafische weergave van kruiswoordpuzzel (12 uur)**
5. **Gebruiker kan de kruiswoordpuzzel invullen via de Gui (19 uur tot)**
   1. De gebruiker individuele cellen selecteren (12 uur)
   2. De gebruiker kan woorden bij de cellen invullen (7 uur)
6. **Rapportage (3,5 uur)**
   1. Presentatie voorbereiden (1,5 uur)
   2. Poster maken (1 uur)
   3. Samenvatting schrijven(1 uur)
7. **Reflectie schrijven (1 uur)**

**Resultaat**

Ik ben tevreden over mijn resultaat voor de tijd die ik heb benut, maar ik had veel meer uit het project kunnen halen als ik er meer tijd in had gestoken. Ik heb wel iets werkends opgeleverd maar de werking van de GUI is gelimiteerd en het gebruik is niet optimaal. Over het algoritme ben ik wel tevreden het is na mijn mening herbruikbaar en voldoende efficiënt/snel.

**Bronnen**

Inspiratie voor de Gui : <https://github.com/PySimpleGUI/PySimpleGUI/blob/master/DemoPrograms/Demo_Crossword_Puzzle.py>

Artikel: Cheng, D., & Dhulekar, N. (2009). Crossword puzzle generator. *CS-44: Artificial Intelligence*, 1–10. <https://www.cs.rpi.edu/~dhulena/CS44FinalProjectReport.pdf>

Woordenboek: <https://github.com/OpenTaal/opentaal-wordlist>