Programmaontwerp

Casusgroep 1 – Finn Alberts, Maud Derhaag, Peter Derks en Brent Vliex

# Strategie

Voor het spel te maken hebben we gekozen voor Pygame. **Pygame kan geïnstalleerd worden via het bijgevoegde document install\_pygame.bat.**

Bij het opstarten van de applicatie zal er door een vragen bestand geïtereerd worden. Voor elke vraag zal er een vraag object worden aangemaakt met de vraag, antwoord a t/m d en een int met het juiste antwoord. Deze objecten worden binnen een lijst gezet. Het programma gaat verlopen volgens 3 while-loops: de menu loop, de main loop en de game over loop. Tijdens de menu loop zal de speler instructies kunnen opvragen. Na het invullen van een naam kan de gebruiker het spel starten waarna de gebruiker de main loop ingaat. Tijdens de main loop zal er een timer lopen de timer zal bijhouden wanneer het tijd is om obstakels te ontwijken en wanneer er een vraag gesteld moet worden. Tijdens de main loop zal de gebruiker het speler object kunnen besturen. Het speler object bevat een x en y coördinaat, een naam, snelheid en score. Ook zal de achtergrond van het scherm langzaam naar beneden bewegen om de illusie van beweging te creëren. Er wordt telkens een tweede afbeelding aan de bovenkant van het scherm geplakt zodat de achtergrond kan blijven bewegen. Wanneer er geen vraag gesteld wordt zullen er obstakel objecten binnen een random range boven het scherm geplaatst worden. De objecten worden binnen een lijst geplaatst. Deze objecten zullen naar beneden bewegen en verdwijnen na met een speler in contact te komen of aan het eind van het scherm komen. Wanneer de speler met het obstakel in contact komt dan wordt er 10 punten van de score afgetrokken. Nadat de obstakel lijst leeg is zullen er nieuwe obstakels boven het scherm geplaatst worden. Na verloop van een tijdsinterval zal er een vraag gesteld worden, de obstakel lijst zal leeg gemaakt worden. De vraag en de antwoorden zullen aan de bovenkant van het scherm verschijnen. Na het verlopen van de antwoordtijd zal gecheckt worden of de x positie van het midden van het speler object binnen de staartgrens van het juiste antwoord bevindt. Zo ja dan krijgt de speler 200 punten. Vervolgens wordt gecheckt of het einde van de vragen lijst bereikt is. Zo ja dan wordt de score van de speler opgeslagen in een tekstbestand. Vervolgens wordt er door elke regel van het tekstbestand geïtereerd en elke score en naam in een 2d-lijst gestopt. Deze lijst wordt gesorteerd op score. Hierna verlaat de gebruiker de main loop en betreed hij de game over loop. Binnen de game over loop verschijnen de behaalde score van de speler en de top 3 hoogste scores op het scherm. Vervolgens kan de applicatie afgesloten worden door op de escape knop te drukken of door op het kruisje aan de bovenkant van het scherm te klikken.

# Datacontainers

Int screen\_width, hierin wordt de breedte van het scherm bijgehouden.

Int screen\_height, hierin wordt de hoogte van het scherm bijgehouden.

Boolean menu, dit is de bool die bijhoudt of de gebruiker zich binnen het menu bevindt.

Boolean main, deze bool zal bijhouden of de gebruiker zich binnen de main loop bevindt.

Boolean game\_over, deze bool zal bijhouden of de gebruiker zich binnen de game over loop bevindt.

Int frames\_per\_second, met deze variabele zal de interne clock van het spel tikken waarmee bepaald wordt hoeveel frames per seconden de mainloop verloopt.

Boolean bInstruction, hiermee wordt bijgehouden of de speler zich op het instructie scherm binnen het menu bevindt.

Boolean bInputName, hiermee wordt bijgehouden of de speler zich binnen het naam input scherm binnen het menu bevindt.

String user\_input, hiermee wordt de speler input opgeslagen tijdens het invullen van de naam.

Int question\_number, hiermee wordt bijgehouden hoeveel vragen er al gesteld zijn.

Int start\_time\_in\_seconds, hiermee wordt bijgehouden op welk tijdstip de speler de main loop betreedt.

Int answer\_time, de hoeveelheid tijd in secondes die de gebruiker heeft om de vragen te beantwoorden.

Int obstacle\_time, de hoeveelheid tijd in secondes waarmee wordt bijgehouden hoelang er obstakels op het scherm verschijnen.

List scores, in deze lijst worden de scores van de speler en de scores van het highscore tekstbestand opgeslagen.

List question\_list, in deze lijst worden alle question objecten opgeslagen.

List obstacles, in deze lijst worden alle obstakel objecten opgeslagen.

# Pseudo Code

main -> False  
menu -> True   
frames\_per\_second -> 60   
start\_time\_in\_seconds -> 0   
answer\_time -> 15   
obstacle\_time -> 10   
bQuestion -> False   
bGameOver -> False   
bInstruction -> False  
bInputName -> False   
user\_text -> ""  
question\_number -> 0   
  
scores -> []   
question\_list -> []   
obstacles -> []   
background\_y -> 0

f -> open('gts\_questions.txt', 'r')  
  
Voor elke line in f:  
 line -> split line op in delen met als splitsing het &-teken  
 het \n karakter weg aan het eind van het laatste woord in de regel  
 Maak een nieuw question object aan met als parameters de gesplitste regels in lijn.  
 Stop het question object in de question\_list  
  
f.close()  
  
randomize de volgorde van de question\_list

Terwijl menu:   
  
 Als niet bInstruction en niet bInputName:  
 Teken de menu tekst op het scherm  
 Kijk of de speler het spel afsluit

Als de speler op de “Play” knop klikt:

bInputName -> True

Anders als de speler op de “Instruction” knop klikt:   
 bInstruction -> True  
  
 Anders als bInstruction:  
 Teken de instructie tekst op het scherm  
 Kijk of de speler het spel afsluit

Als de speler op de “back” knop klikt:  
 bInstruction -> False  
 Anders als bInputName:  
 Teken de play tekst op het scherm  
 Teken de input box op het scherm

Als de speler op de “back” knop klikt:

bInputName -> False

Kijk of de speler het spel afsluit

Als de speler een toets indrukt:  
 Als de toets backspace is:  
 Verwijder de laatse letter van user\_text  
 Anders als de toets enter is en user\_text is geen lege string:  
 Maak een speler object aan en geef de user\_text mee als naam

Start\_time\_in\_seconds -> wordt het huidige tijdstip

Menu -> False

Main -> True

Anders als de lengte van user\_text minder of gelijk is aan 11 karakters:   
 user\_text -> user\_text + het karakter van de gebruikte knop

Terwijl main:  
 clock.tick(frames\_per\_second)   
 time\_in\_seconds -> pygame.time.get\_ticks() // 1000 - start\_time\_in\_seconds  
  
 Verander de x en y positie van de speler aan de hand van de ingedrukte pijltjes toetsen   
   
 Als niet bQuestion:

Als time\_in\_seconds % obstacle\_time gelijk is aan 0 en time\_in\_seconds is niet 0:  
 start\_timer -> time\_in\_seconds  
 bQuestion -> True  
   
 Als de lengte van de obstacles lijst gelijk is aan 0:  
 Voor i in range(5):  
 Creëer een nieuw obstakel object met een random x en y  
 stop het obstakel object in de obstacles lijst  
  
 Voor elk obstakel in de obstacles lijst  
 beweeg het obstakel naar beneden aan de hand van de velocity van het obstakel object  
  
 Als het obstakel object in aanraking komt met de speler:  
 player.score -> player.score -10  
 verwijder het obstakel uit de obstacles lijst  
 Anders als het obstakel onder aan het scherm komt:  
 verwijder het obstakel uit de obstacles lijst  
 Anders:   
 Maak de obstacles lijst leeg  
 Als time\_in\_seconds gelijk is aan start\_timer + answer\_time:  
 Als het correcte antwoord A is en het midden van het speler object bevindt zich binnen de grenzen van de eerste straat:  
 player.score += 200  
 Anders als het correcte antwoord B is en het midden van het speler object bevindt zich binnen de grenzen van de tweede straat:  
 player.score += 200  
 Anders als het correcte antwoord C is en het midden van het speler object bevindt zich binnen de grenzen van de derde straat:  
 player.score += 200  
 Anders als als het correcte antwoord D is en het midden van het speler object bevindt zich binnen de grenzen van de vierde straat:  
 player.score += 200  
  
 question\_number += 1  
 bQuestion -> False  
   
 Als question\_number groter of gelijk is aan 1:  
 bGameOver -> True  
 main -> False  
   
 with open('highscores.txt', 'a') as f:  
 f.write(player.name + '&' + str(player.score) + '\n')  
   
 f -> open('highscores.txt', 'r')  
  
 Voor elke line in f:  
 stop de naam en de score als lijst in scores   
 Als het niet de eerste regel van het score document is:  
 haal het /n karakter weg aan het einde van line  
  
 Als de lengte van de score lijst minder dan 3 is:  
 Voor i in range(0, 3 - len(scores)):  
 stop een lijst met none en 0 in de score lijst  
  
 f.close()  
  
 sorteer de score lijst op score  
  
   
 Update alle objecten in het scherm  
  
 quit\_check()

# PSD

Omdat het PSD erg groot is, is deze niet in het document geplakt. Het PSD is te zien in 20201104stappenplan7c\_psd.nsd

# Testen

Zie voor de uitgevoerde tests het testrapport.