

# Exposé

## Aufgaben- bzw. Problemstellung

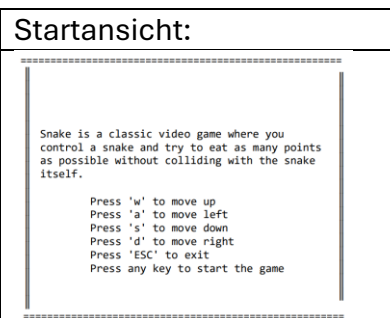
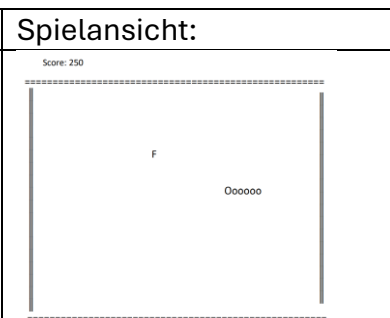
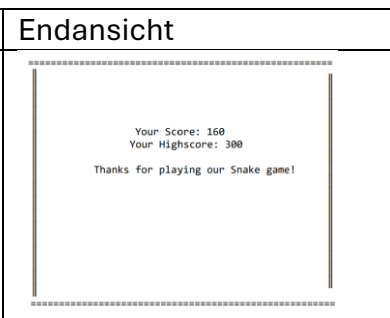
## Snake Programm erstellen.

Eine Schlange bestehend aus einem Kopf und mind. einem Schwanzsegment. Die Richtung lässt sich mit der Tastatur (mit den Tasten WASD oder Pfeiltasten) steuern. Erreicht die Schlange den Rand, kommt sie auf der anderen Seite wieder heraus. Die Ausgabe des Spiels funktioniert im Windows Terminal. Der Bildschirm flackert nicht während der Bewegung. Um das zu erreichen, werden nur einzelne Zeichen auf dem Bildschirm verändert. Es ist immer ein Essen auf dem Spielfeld zu sehen. Frisst die Schlange das Essen, wird sie länger und der Score erhöht sich. Anschließend erscheint auf einer zufälligen Position ein neues Essen. Für jedes Essen erhält die spielende Person einen Punkt. Das Spiel endet, wenn die Schlange gegen ihren eigenen Körper stößt. Am Ende des Spiels wird der Score in einem Textdokument gespeichert und der aktuelle Score und der Highscore ausgegeben.

### Optionales Feature:

Umso länger die Schlange wird, desto schneller wird die Geschwindigkeit der Schlange.

## Benutzeransicht:

Startansicht:	Spielansicht:	Endansicht:
		
<p>In der Startansicht sieht der Spieler eine Erklärung des Spiels.</p> <p>Um das Spiel zu starten, muss er irgendeine Taste drücken.</p>	<p>Der Benutzer sieht folgende Bestandteile:</p> <ul style="list-style-type: none"> <li>● O: Schlangenkopf</li> <li>● o: Schwanz</li> <li>● F : Früchte</li> <li>● =: Spielrand</li> </ul> <p>Mit den Tasten WASD oder den Pfeiltasten, kann er die Schlange bewegen.</p>	<p>Am Ende sieht der Benutzer seinen Score und den besten Score der er jemals erreicht hat.</p>

## Liste des Funktionsumfangs bzw. der Features

- Funktion die die Startansicht ausgibt

- Funktion, die ausliest, ob eine Taste gedrückt, wird und wenn ja welche
- Eine Funktion, die die Schlangenposition in der Oberfläche verändert, überprüft, ob die Schlange eine Frucht berührt und wenn ja eine neue Frucht ausgibt, den Score erhöht und die Schlange verlängert.
- Die Endansicht ausgibt und vergleicht, welcher Wert der Highscore ist, anhand eines Text Dokumentes, wo die vorherigen Scores gespeichert sind.

### Angestrebte Lösung aus technischer Sicht

Die Schlange wird in einem 2-dimensionales-Array gespeichert. Zu jedem einzelnen Schwanzteil wird die x- und y-Koordinate in dem Array gespeichert. Die erste Position ist der Kopf und es gibt noch eine Variable, die die Länge der Schlange angibt.

benötigte Bibliotheken:	Nutzen
time.h	Um Zufallszahlen zu erzeugen mit der Funktion „rand()“.  Wird benötigt um eine zufällige Position für die Frucht zu Erzeugen.
stdio.h conio.h	Benötigt man, um zu prüfen, ob der Benutzer eine Taste gedrückt hat (mit der Funktion „_kbhit()“). Zudem beinhaltet die Bibliothek „conio.h“ die Funktion „_getch()“, in der man sehen kann welche Taste gedrückt wurde.
windows.h	Benötigt man, um die Position in der Konsole zu ändern, um kein neues Spielfeld auszugeben, sondern aktiv die Position der Schlange im Spielfeld zu ändern. Zudem benötigen wir den Befehl „Sleep(...)“ um mit einer zeitlichen Verzögerung die Position der Schlange zu verändern. Damit kann man dann auch die Geschwindigkeit der Schlange steuern.

Die angestrebte Lösung ist speziell für Windows ausgelegt, da die verwendeten Bibliotheken (windows.h und conio.h) Windows-spezifisch sind. Als Compiler wird von uns der MinGW (GCC für Windows) verwendet.