

# TARZAN & JANE

## Teste und schärfe Deine Intuition unter Stressbedingungen!

---

Tarzan verpflegt sich in Ufernähe gerade mit einer köstlichen Kokosnuss, derweil seine liebe Freundin Jane als Nichtschwimmerin im Einbaum auf dem nahen Fluss herumpaddelt.

Plötzlich kippt das Boot und die gute Jane muss gegen das Ertrinken ankämpfen. Tarzan eilt selbstverständlich sofort zu Hilfe. Nur: Welchen Weg zum Ufer soll er einschlagen, um sie innert möglichst kurzer Zeit retten zu können?

Hier ist die Intuition des Naturburschen gefragt, denn: Der kürzeste Weg ist nicht der schnellste!

**Der Test weist vier Schwierigkeitsstufen (LEVEL 1 bis 4) auf:**

LEVEL 1: Jane kämpft im ruhenden Gewässer ums Überleben!

LEVEL 2: Die Bedauernswerte wird von der Strömung des fließenden Gewässers erfasst und von der Unglücksstelle weggetrieben!

LEVEL 3 und 4: Die Ertrinkende treibt mit der Strömung einem Wasserfall oder gefährlichen Stromschnellen zu, so dass sich ihr Retter – auch zu seiner eigenen Sicherheit – keine grossen Abweichungen vom optimalen Weg leisten kann!

In allen Fällen variieren die Standorte von Tarzan und Jane zueinander sowie bezüglich der Entfernungen zum Ufer nach dem Zufallsprinzip. Desgleichen sind das zufällig vorgegebene Laufvermögen sowie die Schwimmgeschwindigkeit von Tarzan, aber auch die variierende Strömung des fließenden Gewässers zu berücksichtigen.

Als Ergebnis wird auf dem Bildschirm der optimale sowie der gewählte Weg eingeblendet und die „gehabte Intuition“ als Fehlerabweichung in Prozent zwischen erzielter und theoretisch minimaler Rettungszeit (ideal = 0%) mitgeteilt.

Für jeden der vier wählbaren Levels wird eine kleine Statistik über die bisher besten zehn Ergebnisse geführt, so dass die Fortschritte aus häufigem Üben verfolgt werden können.

Und nun: **ab in den Urwald!**

Was traue ich mir zu? Wird ein LEVEL antippt, so geht es sofort los!

Doch Halt! Was passiert dann?

Mit antippen von LEVEL 1 zeigt der Bildschirm die Standorte von Tarzan (T) und Jane (J) sowie den Uferverlauf. Ausserdem weiten sich vom Standort Tarzans mit zufälliger Geschwindigkeit konzentrische Kreisbögen aus, wobei der vorderste ab einem bestimmten Moment die Uferlinie (beidseitig der Lotrechten zum Ufer) zu schneiden beginnt. Genau dann wenn man glaubt, einer der beiden sich entfernenden Schnittpunkte sei „gerade jetzt“ an der idealen Stelle für den Sprung ins Wasser, drückt man auf den entsprechenden Button unmittelbar links  $\oplus$  bzw. rechts  $\ominus$  der Startposition (T). Dabei ist vorher zu beachten, mit welcher Vergleichsgeschwindigkeit sich ab gleichem Startpunkt (T) eine zweite Kreisbogenschar konzentrisch ausweitet. Diese symbolisiert die bevorstehende Schwimm – Geschwindigkeit unseres Helden (Schnittpunkte mit der Uferlinie müssen hier nicht beachtet werden).

Falls LEVEL 2 angetippt wird, ist auf dem Bildschirm zusätzlich zu sehen, wie Jane (J) mit zufälliger Strömungsgeschwindigkeit nach links abdriftet. Diesem Einfluss wird sich auch der schwimmende Tarzan nicht entziehen können! Auch hier wird der „Sprung ins Wasser“ beim momentanen Schnittpunkt von Kreisbogen und Uferlinie einfach durch antippen des Plus – oder des Minusknopfes (getätigte Laufrichtung!) ausgelöst.

Bei der Wahl von LEVEL 3 taucht gegenüber Bildschirm LEVEL 2 zusätzlich noch ein Wasserfall oder eine Stelle mit tödlichen Stromschnellen auf. Das Hindernis ist so angelegt, dass das rettende Ufer für Tarzan und Jane bei nur noch halber Schwimmgeschwindigkeit und mit eingerechnetem Sicherheitszuschlag (20% Zuschlag auf den Horizontalabstand zwischen „Wasserfall“ und „optimaler Einsprungstelle ins Wasser“) gerade noch vor dem „Absturz“ zu erreichen ist, wenn hierfür unbesehen der Strömung der Weg mit der kürzesten Aufenthaltsdauer im eisigen Wasser, d.h. eine Schwimmrichtung rechtwinklig zum Ufer gewählt wird. Durch dieses Hindernis mit Absturzgefahr wird zwar nicht das „Erahnen“ der optimalen Einsprungstelle ins Wasser nach der Laufstrecke erschwert, wohl aber die Gefahr geschaffen, dass bei zu grosser Fehlerabweichung die verzweifelte Jane gar nicht mehr gerettet werden kann und zusammen mit Tarzan beim Rettungsversuch abstürzt.

Mit der modellmässigen Festlegung, dass Tarzan nach Erreichen von Jane immer den „rechtwinkligen Weg“ zum Ufer zurück wählt, ist „Rettung“ oder „Absturz“ allein vom gewählten „Sprung ins Wasser“ bestimmt. Weil also hier – im Gegensatz zu LEVEL 4 – auf dem Rückweg keine Korrektur in Form von „Gegensteuer“ (Schwimmwinkel zur Strömung  $< 90^\circ$ ) möglich sein soll, stehen dem Rettungs - schwimmer bei LEVEL 3 stattdessen die folgenden zwei Optionen zur Verfügung:

Option 1: Auf seiner Schwimmstrecke hin zu Jane kann Tarzan abschätzen, ob ihm die Rettung gemeinsam zurück zum Ufer unter den bekannten Bedingungen wohl gelingen wird. Beurteilt er die Situation auf seiner Schwimmstrecke zunehmend pessimistisch, so ist es ihm (bzw. seinem Charakter) überlassen, die Ertrinkende im Stich zu lassen und sich selber durch rechtzeitiges Zurückschwimmen zum Ufer vor dem Tod zu retten. Den allfälligen Rückzug tätigt er bei unveränderter Schwimmleistung und unbesehen der Abdrift stets mit Körperlage rechtwinklig zu Strömung und Uferverlauf.

Option 2: Auch wenn Tarzan bis zur ertrinkenden Jane geschwommen ist und sich mit ihr, bei jetzt nur noch halber Schwimmleistung, auf dem Rückweg zum Ufer befindet, kann seine Einschätzung der Rettungschance u. U. zunehmend pessimistisch ausfallen. Er hat dann die Möglichkeit, Jane „fallen zu lassen“ und zu versuchen, auf der verbleibenden Reststrecke zum Ufer – nun wieder mit voller Schwimmleistung – wenigstens sich selber zu retten. Verpasst er allerdings (bei Option 1 oder 2) den spätesten Moment zur vorzeitigen Umkehr bzw. zum „Loslassen“, so ist es trotz seines Egoismus’ auch um ihn geschehen! Ist die Rettung als Ganzes (von Tarzan und Jane) dagegen erfolgreich, so wird der Spielerfolg wie bei LEVEL 1 und LEVEL 2 über den Zeitbedarf bis zum Zusammentreffen im Wasser ausgedrückt.

Schliesslich kann sich der Spieler auch an LEVEL 4 heranwagen. Im Gegensatz zu LEVEL 3 hat Tarzan hier keine vorzeitige Rückzugsoption. Hingegen kann er nun nach Erreichen von Jane entscheiden, mit welcher Schwimmrichtung (Gegensteuer durch Körperlage mit Winkel  $< 90^\circ$  zur Strömung) die gemeinsame Rettung zum Ufer (bei halber Schwimmleistung) angegangen werden soll. Wählt er einen sehr spitzen Winkel (fast im Gegenstrom) so steigen seine Chancen auf Rettung – allerdings benötigt er dann sehr viel Schwimm – und damit Gesamtzeit. Der Spielerfolg wird daher bei LEVEL 4 über den Zeitbedarf für den gesamten Rettungsweg vom Start bis zurück ans Ufer ausgedrückt.

Bei jedem der vier Levels kann die Unglücksstelle zufälligerweise rechts oder links der Lotrechten zum Uferverlauf liegen und dem Retter die Fluss - Strömung demnach entgegenkommen oder „davonschwimmen“. Überdies verläuft auch die Uferlinie - bezogen auf den Bildschirmrand - in stets neuer Ausprägung nach dem Zufallsprinzip.

Aber jetzt geht’s richtig los!

LEVEL 1

LEVEL 2

LEVEL 3

LEVEL 4

## Meine TOP 10 – Resultate (ideal = 0%):

LEVEL 1		LEVEL 2		LEVEL 3			LEVEL 4	
Datum	Ergebnis	Datum	Ergebnis	Datum	Ergebnis		Datum	Ergebnis
	+57.2%							
			+171.3%					
					+14.3%	J√ T√		
					∞	J♣ T♣		
					∞	J♣ T♣		
					∞	J♣ T√		
					∞	J♣ T♣		
					∞	J♣ T√		
								+78.9%
löschen		löschen		löschen			löschen	

Die Fehlerprozentage (+%) bei LEVEL 1 bis 3 beziehen sich immer nur auf die kürzest mögliche Zeitspanne bis zum Erreichen von Jane! Die Zeitdauer ab hier bis zum rettenden Ufer ist jeweils eine Konstante und daher aus der Berechnung ausgeklammert. Wird nach dem Zusammentreffen das rettende Ufer gemäss LEVEL 3 nicht erreicht (Absturz von Tarzan und / oder von Jane), so wird unbeschadet der Zeitdauer bis zum Zusammentreffen der Wert unendlich (∞) als Fehlerprozentage angegeben. Die Resultatstatistik zu LEVEL 3 liest sich hierbei wie folgt:

+14.3%	J√	T√	gemeinsame Rettung, mit Fehlerabweichung (+ %) zum Optimum
∞	J♣	T♣	gemeinsamer Absturz
∞	J♣	T♣	ungleichzeitige Abstürze, trotz vorzeitiger Umkehr von Tarzan
∞	J♣	T√	Absturz von Jane, aber Rettung von Tarzan dank vorzeitiger Umkehr
∞	J♣	T♣	ungleichzeitige Abstürze, trotz „Loslassen“ von Jane während der Rückkehr
∞	J♣	T√	Absturz von Jane, aber Rettung von Tarzan dank „Loslassen“ von Jane während der Rückkehr

Bei LEVEL 4 beziehen sich bei erfolgreicher Rettung die Fehlerprozentage (+%) auf die gesamte Zeitspanne vom Start bis zurück ans Ufer.

## Hinweise und Ansprüche betreffend Programmierung

Vor dem Hintergrund der nachstehenden Anforderungen an die Programmierung ist zu überlegen, ob das Spiel „Tarzan & Jane“ als Online – Programm via Host / Server (wie die PCE – Programme) entwickelt werden soll – oder ob eine auf PC via CD installierbare Lösung „entwicklungstechnisch“ geeigneter wäre. Als Fernziel aber wäre eine Lösung als App (z.B. auf iPhone via Apple Store o. ä.) allem Anderen vorzuziehen! Benötigte „Klick – Buttons“ sollten (später?) daher auch auf / als „touchscreen“ funktionieren!

Ferner ist zu entscheiden, ob die Darstellung auf dem Bildschirm eine „naturnah – realistische“ Flusslandschaft mit beweglichen Figuren (T und J, quasi als Videobild) abbilden soll – oder ob man sich mit einer „nüchtern – geometrischen“ Darstellung begnügen muss (Programmieraufwand vs. Eindruck/Wirkung). Zweckmässig wäre es wohl, zunächst die „nüchtern – geometrische“ Version zu entwickeln, um diese dann später auf einen „naturnah – realistischen“ Hintergrund zu übertragen.

## Zu den Besonderheiten des Programms

Grundsätzlich ist mit den entwickelten Formeln der schnellste Weg zu Rettung auch ohne Grafik schon bestimmt. Die Grafik wird aber benötigt, um den intuitiv günstigsten Standort für den Sprung ins Wasser festzulegen – womit auch die aus diesem Entscheid erforderliche Zeit zur Rettung, und daraus die Fehlerabweichung zum optimalen Weg ermittelt wird. Die grafische Darstellung am Bildschirm (insbesondere der gezeigte Verlauf der beiden Wegstrecken) soll den Nutzer des Programms aber auch bildlich ansprechen und „in Spannung halten“.

Die Umsetzung / Programmierung der Aufgabe und Lösungsdarstellung weist dazu ein paar spezielle Charakteristiken auf:

1) Die rein formelmäßige, an sich problemlose Umsetzung bedingt einen iterativen Prozess, da wegen der etwas komplizierten Formeln der optimale Punkt für den „Sprung ins Wasser“ nicht über eine Differenzialgleichung gefunden werden kann. Eine konventionelle Gleichung ist statt dessen mit variabler Kenngröße ( $\alpha$ ) sooft durchzurechnen, bis sich daraus ein Minimum im Resultat, das heißt, in der gesuchten Rettungszeit ( $T_{min.}$ ) ergibt.

2) Die Lage des Standortes von Tarzan (T = Start) und der Unglücksstelle von Jane (J = kippen des Bootes) zueinander sowie bezüglich der Uferabstände werden durch drei Zufallsgeneratoren (in Formeln eingebaute Zufallszahlen) bestimmt. Desgleichen werden Laufgeschwindigkeit sowie Schwimmgeschwindigkeit und (allfällige) Flussströmung von drei weiteren Zufallsgeneratoren beeinflusst. Und schliesslich wird – wenn über alle Einflüsse der Rechnungsgang vorgegeben ist – die Grafik auf dem Bildschirm durch einen siebten Zufallsgenerator in einen willkürlichen Winkel zum rechteckigen Bildschirm – Format gedreht. Damit ändert sich zwar nichts am Rechnungsgang resp. Resultat, erschwert aber den Entscheid über den besten „Einsprungpunkt“ ins Wasser. In die Programmierung müssen demnach sieben unabhängige Zufallsgeneratoren eingebaut sein (Beispiel Excel - Programm).

3) Die drei Geschwindigkeiten für Laufen, Schwimmen und Strömung müssen auf dem Bildschirm sichtbar gemacht werden. Für Laufen und Schwimmen sind zur Visualisierung konzentrische Kreisbögen ab Startpunkt T (= Nullpunkt auf der x – Achse) angedacht, welche sich im Verhältnis der (zufällig) vorgegebenen Geschwindigkeiten ausbreiten. Die Strömungsgeschwindigkeit hingegen muss durch entsprechend schnelles Verschieben der Unglücksstelle (Punkt J, von rechts nach links) sichtbar gemacht werden. Für die Programmierung ist dabei nur wichtig, dass sich die drei Geschwindigkeiten im vorgegebenen Verhältnis zueinander darstellen. Die absoluten Geschwindigkeiten, wie in den Vorgaben beziffert, müssen auf dem Bildschirm natürlich nicht eingehalten werden.

4) In Ergänzung zu Ziffer 3) muss es für / mit dem sich ausweitenden Kreisbogen für die Laufgeschwindigkeit möglich sein, dessen Schnittstelle mit der Uferlinie – wo immer diese durch „Klick“ die Einsprungstelle markieren soll – „geometrisch zu lokalisieren“ und in die Berechnung einzubringen. Durch „Klick“ muss also erstens ein bleibender Punkt auf der Uferlinie entstehen, und zweitens muss dessen gewählte Lage „formalisiert“ werden können. Letzteres scheint mindestens auf zwei Arten machbar zu sein:

4a) Entweder die tatsächliche „Fortbewegungsgeschwindigkeit“ des Lauf – Kreisbogens auf dem Bildschirm ist bekannt (z.B. in mm/sec)<sup>\*\*\*</sup> und die Zeitdauer ab Start bis zum „Klick“ wird mit einer eingebauten Uhr gestoppt. Durch Umrechnung auf die vorgegebene Geschwindigkeit laut Zufallsgenerator kann dann die wahre Länge des Radius' zur Schnittstelle bestimmt, und der horizontale Abstich gegenüber dem Startpunkt für die Programmierung berechnet werden.

\*\*\* Da nur die Relativgeschwindigkeiten zueinander massgebend sind, könnte die fragliche Lauf - geschwindigkeit an sich stets mit „1“ angesetzt werden. Dies hätte allerdings den unerwünschten Effekt, dass sich der Lauf – Kreisbogen von Spiel zu Spiel nicht in variablem, sondern in stets gleichem Tempo ausbreiten würde. Um dies zu umgehen, muss also tatsächlich nicht nur eine Uhr zur Messung der Laufzeit, sondern auch ein Geschwindigkeitsmesser für „das Laufen auf Bildschirm“ eingebaut sein (aufwändig??).

4b) Oder es befindet sich auf der gesamten Uferlinie (im Versteckten) ein Massstab, auf welchem mit dem „Klick“ an der gewählten Schnittstelle eine Ablesung erfolgt. (Prinzip Zeichnungsprogramme). Bezogen auf den Startpunkt resultiert so ein Streckenmass, welches – umgerechnet aus dem definierten Bildschirm - Massstab – die reale Lage der Einspungstelle für die Programmierung festhält. Diese zweite Vorgehensweise ist vermutlich der einfachere Weg .....

5) Wie eben erwähnt, muss die Bildschirmgrösse – bezogen auf die generierten Abmessungen resp. errechneten Wegstrecken – so „formatiert“ werden, dass alle denkbaren Ausgangslagen / Darstellungen in der Resultatgrafik erfasst werden können. In den Angaben zur Programmierung ist dies berücksichtigt.

6) Bei LEVEL 3 findet der „Absturz“ von Jane dann statt, wenn sie (mit Strömungs - geschwindigkeit) die zufällige Wasserfall – Linie überquert, bevor sie vom heran schwimmenden Tarzan erreicht wird. Gelangt der Rettungsschwimmer vorher zu ihr, muss er aber mit ihr zusammen auch noch (bei nur noch halber Schwimm – geschwindigkeit!) das Ufer erreichen, bevor sie beide bis zum Wasserfall abgedriftet sind. Bedingt durch die bei LEVEL 3 gegebenen Optionen (vorzeitige Umkehr von Tarzan, oder „Loslassen von Jane während der Rückkehr zum Ufer), muss dieser „point of return“ resp. „des Loslassens“ auf der Schwimmstrecke durch „Klick“ oder „touch“ fixiert werden können. Dazu muss „im Versteckten“ ein orthogonaler Bildschirm – Massstab (analog Zeichnungsprogramme) eingebaut sein. Die Rückkehr ab „Klickpunkt“ zum rettenden Ufer ist in der Spielanlage definiert (siehe Einleitung) und erfordert vom Spieler keine Entscheidung mehr. Für den fraglichen „Klick“ oder „touch“ ist ein entsprechender Button <STOP> vorzusehen.

7) LEVEL 4 unterscheidet sich gegenüber LEVEL 3 insofern, als hier für die gemeinsame Rückkehr zum Ufer (ohne Option auf vorzeitigen Rückzug) die Schwimmrichtung, bzw. Körperlage zum Ufer frei gewählt, und so der herrschenden Strömung entgegen gewirkt werden kann. Nach Zusammentreffen von Tarzan und Jane wird das System kurz angehalten. Es erscheint ein sich um den gewählten Punkt „Sprung ins Wasser“ ausweitender Halbkreisbogen (in beliebiger Geschwindigkeit), welcher in wachsendem Abstand je einen Schnittpunkt mit der Uferlinie bildet. Durch „Klick“ oder „touch“ auf einen Button (links und rechts der Lotrechten zur Uferlinie) kann damit (analog zum „Sprung ins Wasser“) bestimmt werden welcher Schnittpunkt mit der Uferlinie die Schwimmrichtung der Beiden zum rettenden Ufer vorgeben soll. Rechnerisch muss dessen Lage auch hier (analog dem „Sprung ins Wasser“) durch die versteckte Ablesung auf der x – Achse des Bildschirms erfasst und übertragen werden. Als Erfolgsmassstab (Fehlerprozente) wird bei LEVEL 4 die Zeitspanne vom Start bis zum Ufer zurück herangezogen.

## Zusammenfassung

Die Darstellungen auf dem Bildschirm müssen sowohl bezüglich der drei Geschwindigkeiten als auch in der Anordnung gegenüber dem Bildschirmrand „beweglich“ sein. Der günstigste Weg ( $\rightarrow$  der mit der kürzesten Rettungszeit  $T_{min}$ ) ist dabei durch die zufallsgenerierten Vorgaben automatisch festgelegt und wird iterativ berechnet. Für den „Sprung ins Wasser“ hingegen verfolgt der Spieler die auf dem Bildschirm sich darstellenden (Relativ -) Geschwindigkeiten für Laufen, Schwimmen und Strömung. An der mutmasslich günstigsten Uferstelle löst er durch „Klick“ (bzw. durch „touch“) den Sprung aus. Diese Stelle muss auf dem Bildschirm festgehalten / dargestellt, und als „wahrer Abstand zu Ausgangspunkt / Start (T)“ in die Berechnung der benötigten Rettungszeit ( $T_{eff}$ ) eingebracht werden.

Mit der Wahl dieses Punktes ist bei LEVEL 1 bis 3 Erfolg oder Misserfolg der Rettung, bzw. die bei Erfolg dazu beanspruchte Zeit bestimmt, wobei für LEVEL 3 die zwei Optionen bestehen, entweder die Rettung vorzeitig abubrechen oder Jane auf dem Rückweg zum Ufer „los zu lassen“, um sich dadurch (vielleicht) selber noch in Sicherheit zu bringen. Der „point of return“ resp. „des Loslassens“ muss hier über einen versteckten, orthogonalen Bildschirm – Massstab erfasst werden.

Bei LEVEL 4 besteht diese Option nicht, wohl aber die Möglichkeit (bzw. Aufgabe), den Rückzugsweg unter Berücksichtigung der herrschenden Strömung (d.h. mehr oder weniger gegen diese gerichtet) so festzulegen, dass ein Absturz beim Wasserfall vermieden wird. Hierbei wird die gewählte Schwimmrichtung „zum Ufer zurück“ in die Berechnung eingefügt und die benötigte Rettungszeit mit der theoretischen Minimalzeit verglichen. Dazu ist wiederum die Formalisierung der auf der x – Achse des Bildschirms durch „Klick“ markierten Schwimmrichtung erforderlich.

Ein für dieses Spiel entwickeltes Programm auf der Basis „Excel“ ist bereits funktionstauglich, wobei allerdings das Ergebnis  $T_{min}$  aus der Iteration „von Hand“ übernommen und „weitergegeben“ werden muss. Auch der „Sprung ins Wasser“ sowie die Richtung für das Zurückschwimmen ans Ufer bei LEVEL 4 können mangels Grafik nur von Hand (oder mittels Zufallsgenerator) festgelegt werden. Dies gilt ebenso für den „point of return“ bei LEVEL 3.

Das verfügbare Excelprogramm kann somit nur „simulieren“, nicht aber „testen“.