

Cahier des charges *Terminal 20*

Auteur	Date	Version
Romain MELLAZA	10/02/2025	v0.1

Sommaire :

- Nom du jeu
- Système de simulation 3D dans le terminal UNIX
- Raycasting
- Principe du jeu

Nom du jeu

Le jeu s'appelle **Terminal 20**, cela fait référence au fait que le jeu se déroule directement dans le terminal, le 20 rappelle la limite de temps qu'aura le joueur.

Système de simulation 3D dans le terminal UNIX

Le projet est basé sur le principe d'*ASCII art*, on affiche seulement des symboles présents dans la table ASCII.

ASCII TABLE

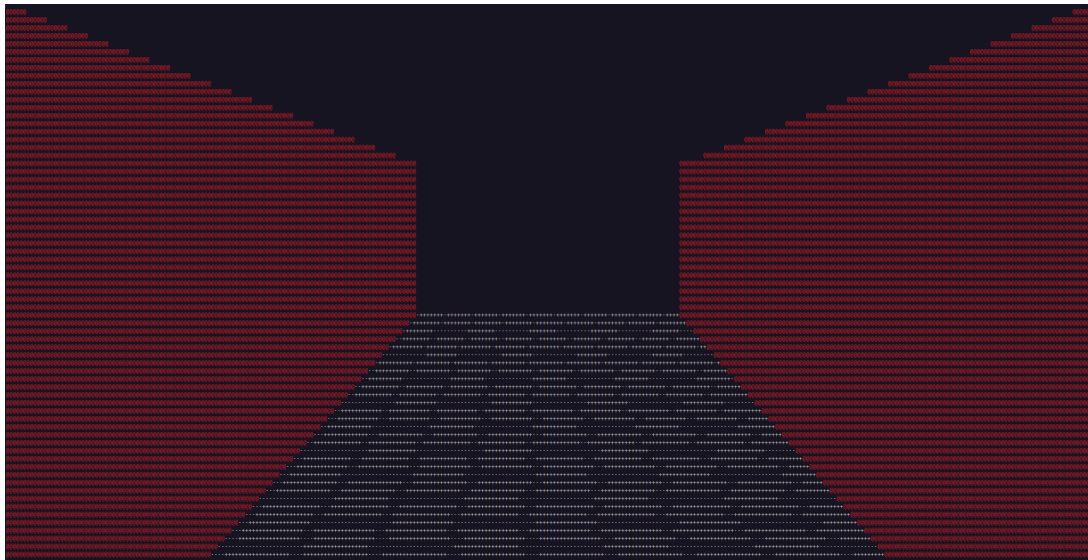
Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[END OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

J'ai déjà codé un exemple basique pour que je puisse me représenter concrètement le niveau de difficulté d'un tel système graphique. Dans les paragraphes suivants j'explique ce que j'ai réalisé.

Pour simuler un effet de profondeur 3D, je joue sur la perspective du rendu. Un moyen simple est de générer un couloir dont le sol et les murs se rapprochent du centre de l'image (fixé à une certaine profondeur de champ). Cette profondeur a été choisie arbitrairement et je réfléchis à la faire varier en fonction de futures contraintes inhérentes au *gameplay*.

Pour donner à l'utilisateur l'impression d'avancer ou de reculer, l'algorithme génère continuellement une texture pavée au sol, les interstices entre les différentes pierres reculent dès lors que le joueur avance, et inversement ils avancent dès lors que le joueur recule. Pour accentuer cette sensation, je considère l'idée d'ajouter des repères fixes sur les murs, tels que des fenêtres ou des tableaux.

En choisissant des paramètres proportionnels, on obtient finalement le résultat suivant :



Le couloir généré à ce stade du projet sert en premier lieu à prouver la faisabilité d'un tel système. Il est évident qu'un couloir rectiligne et froid ne permettra une bonne expérience utilisateur. La suite s'oriente donc vers des embrochements, où l'utilisateur pourra faire des choix de circuit. De même, des environnements plus complexes sont à l'étude, il est nécessaire d'introduire des personnages.

Je vais donc me tourner vers le système de Raycasting qui est à l'origine de *Wolfstein 3D* ou encore de *DOOM*, je fais ce choix car ce type de moteur graphique est largement documenté sur Internet, à plusieurs niveaux d'approfondissement et dans différents langages. De plus, le Raycasting permet de générer des images en continu, en ayant pour seule base une matrice binaire représentant la carte du niveau, cela va donc me permettre une plus grande versatilité. Mon principe initial n'est pas foncièrement mauvais mais très peu malléable dans les faits.

Cependant une chose est sûre, avec cette technique initiale ou celle du Raycasting (explicitée ci-dessous) je ne pourrais simuler des changements de hauteur : pas d'escaliers ou de saut.

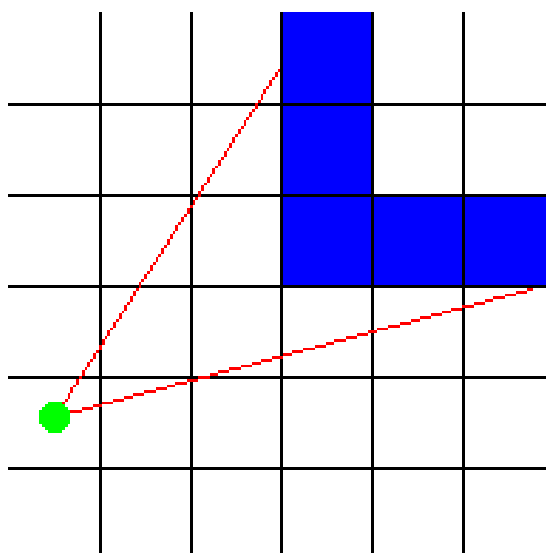
Migration vers le **Raycasting**

Une grande partie des informations sont tirées de ces documents ou projets déjà réalisés :

- *Lode's Computer Graphics Tutorial*
- *PyRay - Python Raycasting Engine*
- *Terminal Dungeon*

Je n'ai pas encore étudié en profondeur ce moteur graphique, ma compréhension reste largement superficiel, je compte coder en Python prochainement la base du Raycasting, ainsi qu'ajouter une grande quantité d'informations ci-dessous.

Le Raycasting est basé sur le calcul constant des collisions entre les rayons émanant de la vision du joueur et les murs présents sur la carte. La distance mesurée entre le personnage et le mur permet de connaître la hauteur du mur affiché sur l'écran du joueur. En d'autres termes, plus le mur est proche plus il apparaît grand sur l'écran du joueur.



Une bonne gestion du degré de champ de vision est primordiale car ce dernier influence directement le calcul des rayons.

Principe du jeu

Je compte faire un jeu avec plusieurs cartes (avec des textures/couleurs différentes), des ennemis, des quêtes et des énigmes. Des objets seraient disposés sur la carte, une *mini-map* serait disponible dans le coin inférieur gauche pour permettre à l'utilisateur de se repérer correctement dans l'espace.

Je détaillerais prochainement les différents niveaux.

Le jeu sera basé sur l'univers science-fiction, en profitant notamment de l'ASCII-Art pour réaliser des effets à la *Matrix*.

Le personnage incarné par le joueur sera un agent de la DGSV (*Direction Générale de la Sécurité Virtuelle*). Un matin, il ouvre sa messagerie et tombe sur un mail menaçant l'incitant à installer un jeu, le jeu se lance, il a 20 minutes pour le finir sous peine qu'une bombe explose en plein centre-ville de Brest.

Le joueur aura réellement un compteur rouge dans le coin supérieur gauche de son écran. S'il finit le jeu en moins de 20 minutes il a gagné, sinon c'est perdu.

Concernant l'introduction, je vais réaliser un boot fictif de système d'exploitation, puis une interface très simplifiée de messagerie, elle se commandera entièrement au clavier, l'utilisateur aura alors l'impression de ne pas être totalement guidé dans ses choix. Pour augmenter l'immersion, je récupérerai le véritable nom d'utilisateur de la machine pour l'ajouter au jeu.