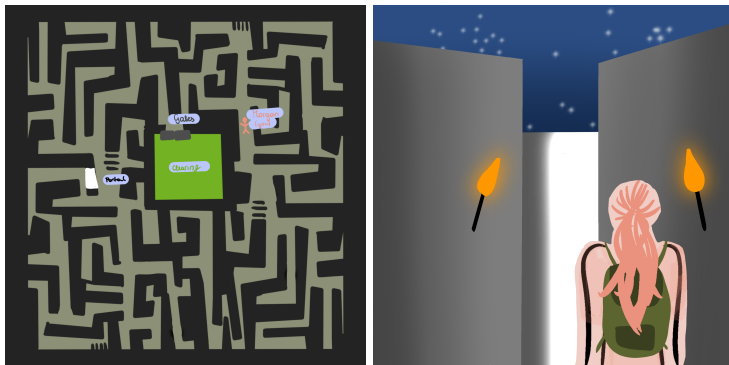


Submission 2:

Administrative		
Group Name / Game Name	Morgan Maze — are you in the clear yet?	
GitHub Link	https://github.com/christinatuechler/cgue22-morganmaze	
Students	Christina Tüchler, 11908107 Alexandra Irger, 11912783	
Genre	Maze	
Goal	Find the portal to win the game.	
Game Idea and Content		
Story (max. 4 sentences)	Morgan finds herself in the middle of a maze. In order to make win the game,, she has to find the magic portal. However, if she does not find the portal before dawn, she'll be stuck in the maze forever. Can you help Morgan find the magic portal?	
Gameplay (max. 4 sentences)	Morgan starts by standing in the middle of the maze. She has to navigate through the walls and memorize where she has been already. To win the game, she has to find & walk through the magic portal in a specified timeframe.	
User interaction	<ul style="list-style-type: none"> • WASD: Movement of Morgan (main character) • Mouse movement: Camera view direction • Enter: starting/pausing the game • Space: Jump • ESC: Quit • ↑/↓: Increase/decrease illumination multiplier • H: turn HUD on/off • C: toggle camera (free flying/third person) 	
3D Objects	Static: <ul style="list-style-type: none"> • Maze • Obstacles Dynamic: <ul style="list-style-type: none"> • Morgan (main character) • Portal 	
Scene lighting	Point Light in the middle of the maze. Directional light in an angular direction from above(sun).	
Features		
Category	Feature	Description (Usage)
Optional Gameplay 14	Collision Detection (Basic Physics) (4 Points)	Player colliding with walls, the obstacles and the portal.
	Advanced Physics (6 Points)	Simulate the physics of the player character e.g. running, jumping and going through the

		portal as well as the collapse of the portal when touching it.
	Heads-Up Display (4 Points)	Displaying the countdown until the dawn and the current brightness.
Effects 48	<u>Animation:</u> GPU Vertex Skinning (20 Points)	Skeleton of the main character.
	<u>Texturing:</u> Specular Map (4 Points)	Applied to the walls to make them more realistic.
	<u>Post processing:</u> Bloom/Glow (8 Points)	Glow effect of the portal and on the player.
	<u>Lighting:</u> Shadow Map with PCF (16 Points)	Calculate the shadows from the directional light (sun).

Sketches



Implementation Details	
Playable, Advanced Gameplay, Win/Lose Condition, Intuitive Controls, Intuitive Camera, Framerate Independence and min. 60 FPS	<p>When you start the project, the "welcome" text is presented, which has been implemented using FreeType. If you press the "Enter" key, the game can be started (or paused if you continue to press the key). From this point on, a timer runs, which is visible in the HUD and represents the win/lose condition. The player must now run through the maze within this time (mouse to rotate the player, WASD to move the player) and find the portal. To reach the portal the player has to overcome obstacles by jumping. If he succeeds, he has won in this state of the game. However, if the timer runs out without the player reaching the portal, he has lost the game. The third-person camera is activated by default, but this can be replaced by the free-flying camera by pressing the "C" key. Moving the player is only possible within the game time (while the timer is running), and not before starting or after winning/losing the game.</p> <p>Our game is also framerate independent and runs with 60 FPS depending on the used computer.</p> <p>Therefore the "Game" class is the most important component of the project.</p>

3D Geometry, Textures, Moving Objects	<p>Furthermore, there are various 3D objects in the game, such as the player, the walls of the labyrinth, the portal, etc.</p> <p>All these loaded objects are provided with textures.</p> <p>The player is the most complex model. It was loaded with Assimp and can be moved forward, left, back, right with the WASD-Keys and rotated with the mouse as described above.</p> <p>Our game also has "moving objects": On the one hand, the player can move in any direction in the game (x, z to move and y to jump) and our portal also consists of many small "moving objects".</p>
Illumination Model	<p>In our game there are two light sources, on the one hand a point light in the middle of the labyrinth and on the other hand a directional light above the labyrinth which imitates the sun.</p>
Adjustable Parameters	<p>The parameters "Screen Resolution", "Fullscreen-Mode", "Refresh-Rate" and "Brightness" can be adjusted in the settings.ini file (in the assets folder).</p> <p>The "Brightness" parameter can also be increased/decreased in the game by using the up/down arrow and the current brightness-value is shown in the HUD.</p>
Heads-Up-Display	<p>When pressing the "H" key, the heads-up display can be activated/deactivated. Here the player can see certain statistics, such as the time remaining until game over or the frame rate and the brightness value.</p> <p>This is implemented with the FreeType-Library.</p>
Collision Detection, Advanced Physics	<p>The character as well as the maze is rigid and implemented with physx. Therefore it can be moved around within the maze. The maze is roughly based on a pattern. The maze was then hardcoded consisting of boxgeometries & later the boxes were added to the physics scene as rigid bodies. It was tested via the Physx Visual Debugger.</p> <p>The character consists of a Character Controller Element based on the PhysX sdk and then linked to the Animated Player Model.</p> <p>The player not only collides with the labyrinth walls, but also with the obstacles in the labyrinth; for example, he has to jump to overcome them. The player also collides with the portal and it breaks into many pieces as soon as it is touched. These parts also collide with each other and with the maze.</p>
Animation - GPU Vertex Skinning	<p>Our submission also includes the gpu vertex skinning implementation of the player. This was implemented using assimp, just like the model loading mentioned above, but here bone information is also loaded.</p> <p>In our case, we have used a Collada model (.dae) because it can be used to create an animation. In our case we did not create the model ourselves, but it was already delivered with a skeleton and an animation in the download.</p> <p>In general, it is important to understand that in skeletal animation the entire (weighted) hierarchical bone structure is very important. For example, if the shoulder is to be rotated, not only the shoulder bones move, but also the connected child bones in the upper and lower arm.</p> <p>An animation is defined by the movement of bones over a certain time and is interpolated, calculated and represented by the animator.</p> <p>All the relevant code for this feature can be found in the project in the folder "Model loading".</p>
Shadow Map with PCF	<p>Our submission includes separately rendered shadow map. In the first rendering pass, a depth map of the scene is generated, in the second pass the depth map is incorporated into the illumination model. The shadow map is improved with a bias calculation, oversampling and PCF. It is applied analogue to the directional sunlight, in the main scene shader "bloom".</p>
Specular Map	<p>A specular map was added to the maze texture and taken into account for rendering in the "bloom" shader.</p>

Bloom / Glow	The bloom effect is generated by the bloom shader, which has two render outputs: the regular output with all objects rendered with their respective shaders combined, as well as a lightMap, which only renders the parts which should glow (in our case, the player, and the portal). The lightMap is then blurred using two pingpongFBOs which alternate in blurring the image horizontally and vertically. The finished blurred output is then combined with the regular output to create the final image.
Other Special Features	
Skybox	To make the game a bit more interesting and real, a textured cube was rendered over the whole scene, which is used as a skybox.
Libraries and Sources	
Assimp	https://github.com/assimp/assimp
PhysX	https://github.com/NVIDIAGameWorks/PhysX
FreeType	https://www.freetype.org/download.html https://www.youtube.com/watch?v=qW_8Dyq2asc https://learnopengl.com/code_viewer_gh.php?code=src/7.in_practice/2.text_rendering/text.vs https://learnopengl.com/code_viewer_gh.php?code=src/7.in_practice/2.text_rendering/text.fs https://learnopengl.com/In-Practice/2D-Game/Render-text
Player Model	https://www.mixamo.com/#/?page=1&query=walking&type=Motion%2CMotionPack
Sources concerning the code	see sources.txt and in the code