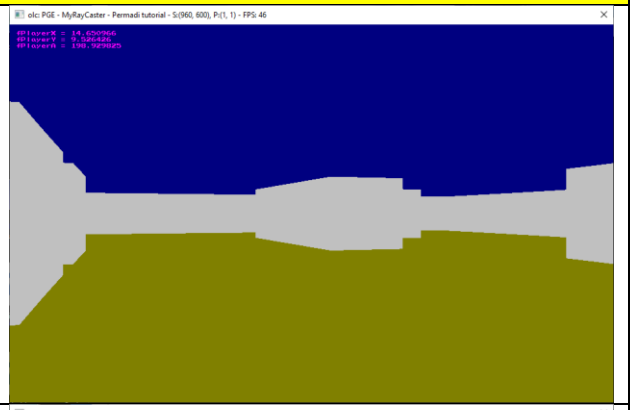
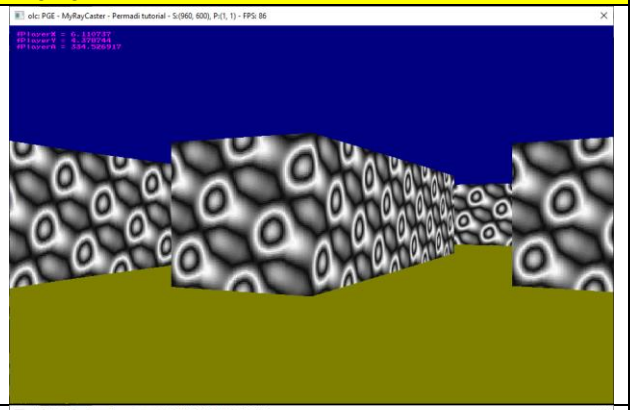
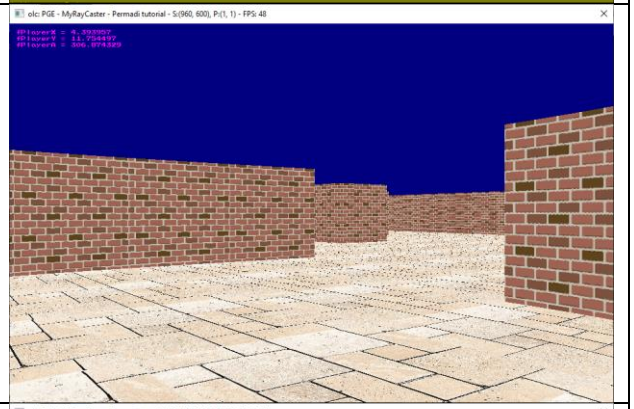
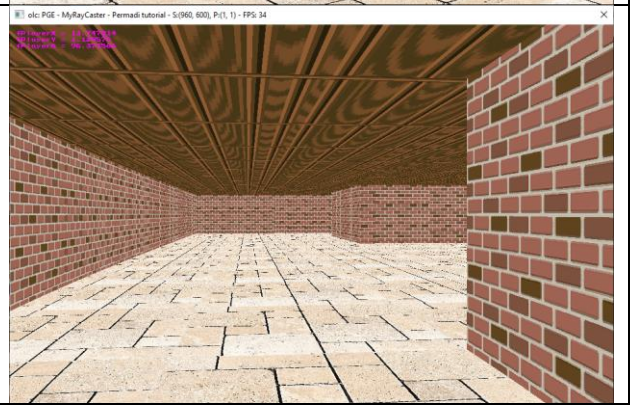
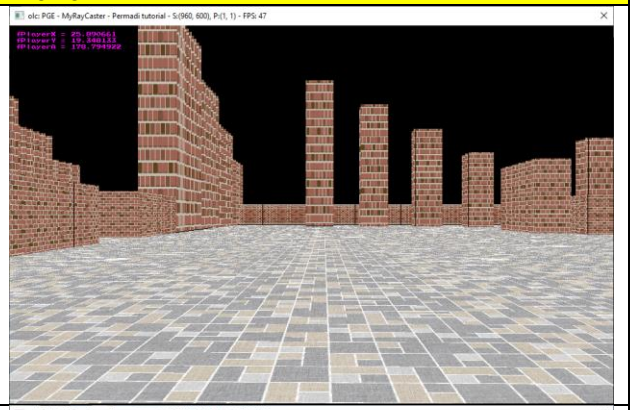
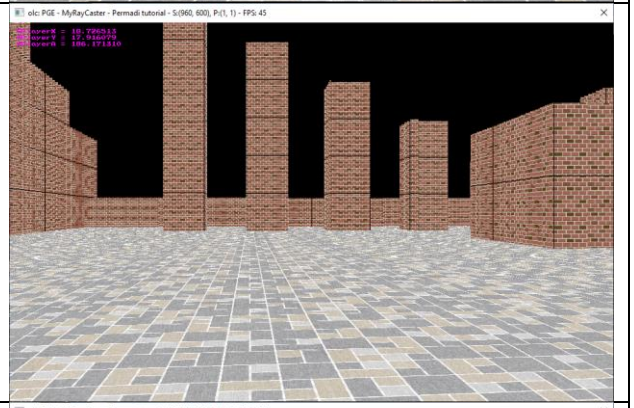
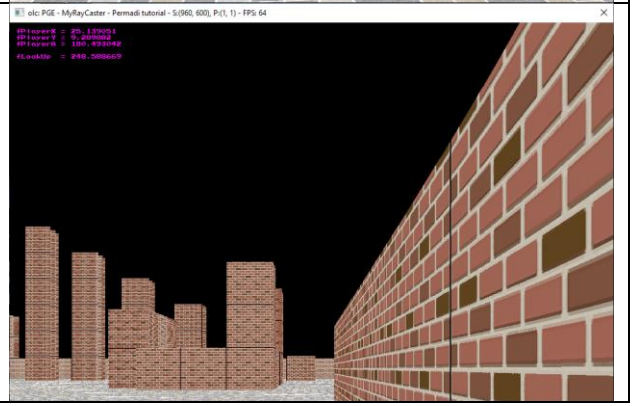
Raycasting implementations – Permadi tutorial

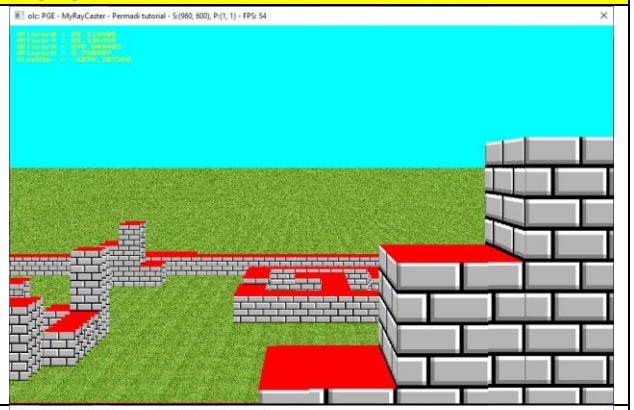
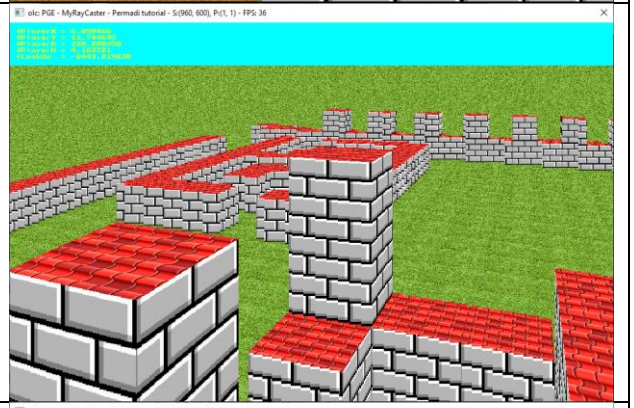
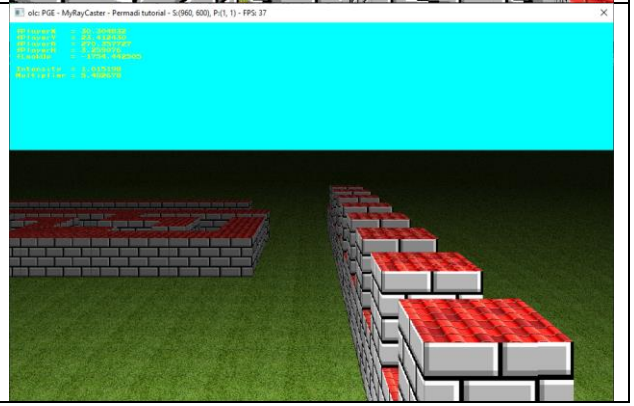
Joseph21
February 1, 2023

All source files on: https://github.com/Joseph21-6147/Raycasting-tutorial-series---Permadi-inspired
Note – in this phase of the project all code is organized into one single implementation file.

Nr	Permadi tutorial	Source file name	Subject	Preview
1	Parts 03-09 (&15)	• `main - part 09a (plain rendering, hor. motion, naive distance finding algo).cpp`	Non textured rendering, horizontal motion, naive distance finding (ray marching)	
2	Parts 03-09 (&15)	• `main - part 09b (plain rendering, hor. motion, DDA algo).cpp`	DDA implementation (instead of ray marching)	

Nr	Permadi tutorial	Source file name	Subject	Preview
3	Part 10	• `main - part 10 (textured walls).cpp`	Added: Wall texturing	
4	Parts 11-12	• `main - part 12 (textured floor).cpp`	Added: Floor texturing	
5	Part 13	• `main - part 13 (textured ceiling).cpp`	Added: Ceiling texturing	

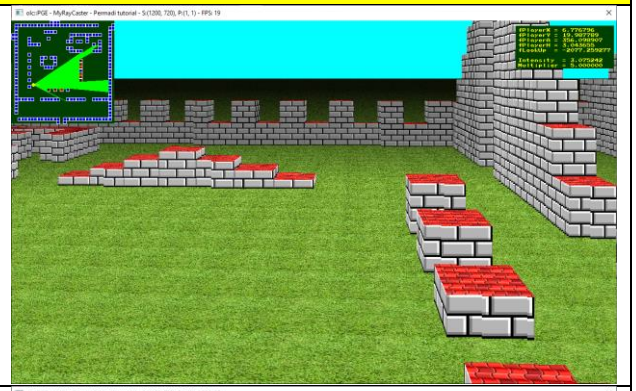
Nr	Permadi tutorial	Source file name	Subject	Preview
6	Part 14	• `main - part 14a (variable height walls).cpp`	Added: Variable height walls	
7	Part 14	• `main - part 14b (variable height walls - improved texturing).cpp`	Added: Improved wall texturing for variable height walls	
8	Part 16	• `main - part 16 (vertical motion - looking up and down).cpp`	Added: Effect to simulate looking up or down	

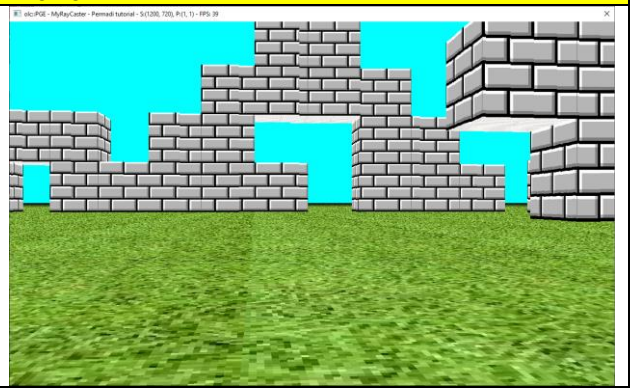
Nr	Permadi tutorial	Source file name	Subject	Preview
9	Part 17	• `main - part 17a (flying and crouching).cpp`	Added: Code for flying and crouching of player, in combination with variable height walls.	
10	Part 17	• `main - part 17b (textured roofs, optional mouse control).cpp`	Added: Roof texturing and optional mouse control	
11	Part 19	• `main - part 19 (shading - night effect).cpp`	Added: Simple form of distance shading	

Elaborations on the Permadi tutorial

Joseph21
April 22, 2023

I implemented and posted the Permadi based tutorial series in spring 2022. Currently I decided to elaborate on that series with some of my own creations:

Nr	Source file name	Subject	Preview
12	• main - part 20 (fractional wall heights).cpp	Experiment with walls that are ¼, ½, ¾ unit high – it's straightforward to create walls with other fractions as well	
13	• main - part 21a (sprites - basic rendering).cpp • main - part 21b (sprites - with column based depth buffer).cpp • main - part 21c (sprites - painters algo).cpp • main - part 21d (sprites - looking and moving up and down).cpp • main - part 21e (sprites - randomly initialized).cpp • main - part 21f (demo version with 2D depthbuffer).cpp	Introduction of objects (sprites) using the technique of billboarding. These parts build up the functionality so that looking and moving up and down are supported in combination with (scaled) billboard rendering	

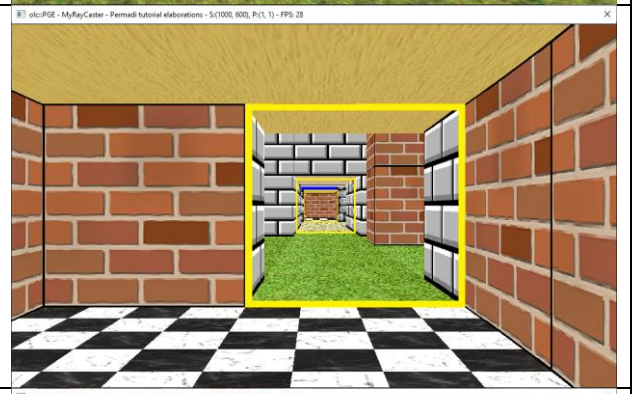
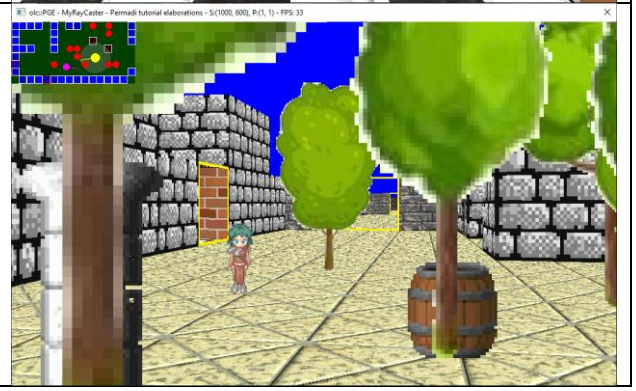
Nr	Source file name	Subject	Preview
14	• main - part 22a (class RC_Map introduced).cpp • main - part 22b (map representation adapted).cpp • main - part 22c (working version, bugs in roof ceil texturing).cpp • main - part 22d (texturing and CD fixed).cpp	Introduction of gaps/holes in the walls, overhanging and floating blocks. An alternative method of representing the map is introduced to this end.	

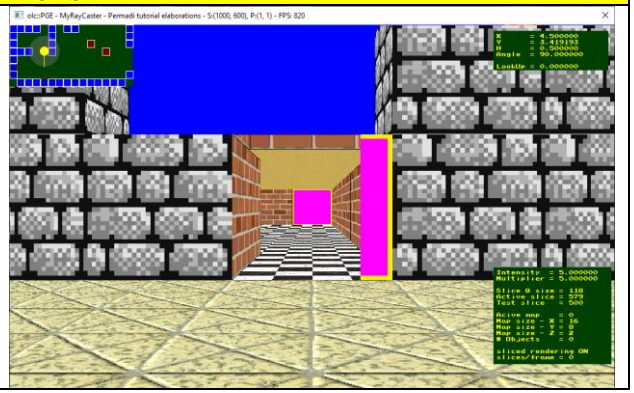
Note – in this phase of the project I started to implement the map definition data as a separate header file.

Nr	Source file name	Subject	Preview
15	• main - part 23a (class RC_Objects introduced).cpp • map_demo - part 23a.h • main - part 23b (block and face differentiated texturing).cpp • map_10x10 - part 23b.h • main - part 23c (face hit detection added in DDA algo).cpp • map_16x16 - part 23c.h • main - part 23d (see-through windows and doors).cpp • map_16x16 - part 23d.h • main - part 23e (door gate animation).cpp • map_16x16 - part 23e.h • main - part 23f (refactored block structure in classes).cpp • main - part 23g (refactored blocks in faces as well).cpp • map_16x16 - part 23g.h	Added see through (transparent) texturing (to be applied in combination with windows, doors, openings in roofs, etc). In addition I added animated textures (for doors etc.), and I implemented texturing differentiated per block face (east, north, west, south, top, bottom)	

Note – in this phase of the project I started organizing the code into multiple implementation units.

Nr	Source file name	Subject	Preview
16	• part 24a (isolated RC_Misc) • part 24b (isolated RC_Face) • part 24c (isolated RC_MapCell) • part 24d (isolated RC_Map) • part 24e (isolated blueprints data and functions) • part 24f (added DepthDrawer) • part 24g (isolated RC_Object)	Consider this a maintenance release. The code is split up into smaller modules to make it more manageable. Some improvements were made in the process, very little functionality is added.	

Nr	Source file name	Subject	Preview
	• part 24h (refactoring finalized, small changes and improvements)		
17	• part 25a (dynamic map cells)	Created a type of map cell (previously called “block”) that moves and shrinks dynamically and periodically. It’s just a prototype, could be extended later on.	
18	• part 25b (multiple maps) • part 25c (portals - first version, no portal rendering yet) • part 25d (portals - recursive sub slice rendering)	These implementation versions lead to the first / rudimentary portal rendering, where two or more different maps are visible in one view, and the player can move seamlessly from one map to the other through the portal. The user has no notion that under the hood another map is activated.	
19	• part 25e (extended input checking) • part 25f (objects organized per map)	Since the file with the map definition becomes increasingly important, I put some effort into checking on the correctness of the input data, and on the integrity of the maps that are created from it. Additionally I organized the objects per map.	

Nr	Source file name	Subject	Preview
20	• part 25g (face portals) • part 25h (portals - queued sub slice rendering)	To get the rendering fixed, I needed to give the portal characteristic to the faces (instead of the map cells). Furthermore I split off a filter function from the DDA algorithm, to get more grip on the DDA and how the results thereof are rendered. Finally, I rewrote the recursive sub slice renderer into a version that uses a queue.	

Ideas for future work

Joseph21
July 24, 2023

(in random order)

1. a fractional map cell has a height of < 1.0f. I want to extend this concept to have map cells that have a bottom that is elevated instead of being at height 0.0f within its level;
2. object rendering contains bugs w.r.t. rotation and scaling. Needs to be debugged and possibly refactored;
3. object rendering can be extended with more sophisticated forms of animated sprites. This is also necessary to create more sophisticated NPCs;
4. I want to be able to place objects on other layers than the default layer (0);
5. wall, ceiling, roof or floor texturing can be animated as well;
6. the dynamic map cell concept can be elaborated on;
7. The map file concept provides a lot of flexibility in building all sorts of maps, and combining them with portals. It becomes increasingly tedious to do this all by hand. At one moment I need a separate tool to build map files;
8. Also change the orientation (at discrete angles 90, 180 or 270 degrees) at a portal to get into the other map.
9. Create player dynamics and physics – velocity, acceleration, gravity, possibility to jump;
10. …