Faculdade de Design,
Tecnologia e Comunicação
Universidade Europeia

2021

Winner Takes It All

GAME DESIGN DOCUMENT

VASCO CORREIA
TOMASZ WASILONEK

Summary

1. Introduction

This is a multiplayer turn-based game playable over the internet by 2 - 4 players.

Players are thrown into a medieval fantasy prison setting. There is a possibility to gain freedom if the convict takes part in a last-man-standing type of tournament and of course - wins it all. **Winner takes it all.**

Players enter a match where they must strategize their movement and actions to be the last one surviving.

It combines strategy and exploration of the environment to find the best spot on the map and good loot that will allow the player to defeat the other players.

The game will be created in Unreal Engine 4.27 and targets the Windows operating system.

Inspiration image taken from: https://www.adventurerooms.it/en/escape-medieval-prison-opens-pavia/

2. Game world and level design

The map will be made of tiles. The map size will be 200 by 200 tiles. (section 2.1 Tiles)

The map will replicate a big prison where the inmates (players) will fight.

For the vertical slice the team will create a level using cube tiles map, which will be outlined by wall tiles. In the future, if time allows it, the team would like to create a system for randomly generated maps using an algorithm with a defined set of rules.

Wall tiles will be used to create and define the structure of the level.

Placed around the level there will be treasure tiles that the player can go to collect loot. There will also be trap tiles that damage the player if they step on them. The last type of tiles will be elevated tiles, that will be harder to reach but if the player manages to get there, he will have cleared a lot more fog of war.

The map won't be flat, it will have slight elevations to add a more realistic feel to it.

Each map starts covered with **fog of war**, which makes most of it blacked out to the player.

Each player is placed in strategic starting positions on the map. The players don't see each other initially (because of the fog of war).

Levels will be designed in a way that gives players the possibility to find the best strategic positions for each map, that will allow them to use their available weapons.

2.1. Tiles

The tiles will be made of cubes. There will be initially 5 different tiles.

Each tile will have useful properties, like: type, size, effect, texture, position, attributes.

Reference image from: https://tanx.io/

2.1.1. Normal tile

The regular tile. Most of the map will be made of this tile. It will provide no extra effects to the player.

2.1.2. Obstacle tile

Wall tile is used to define the walls of the level. Player won't be able to walk, see or shoot through them. Provides no special effect.

2.1.3. Elevated tile

An elevated tile. When a player is on this tile, he can uncover a wider range of fog of war. (**Ex**. Normal: 3 tiles / elevated: 6 tiles).

Visually indicated by a different colour, i.e. yellow

Elevated tiles have a rate of 5% on the map.

2.1.4. Treasure tile

A tile with loot in it. When a player goes over this tile, some weapons will be added to his inventory.

Indicated by a treasure model on top.

If a player is close enough to the treasure tile, it will be highlighted even through fog of war.

To make it even more visible to the player, we will use some particle effects to indicate that something nice and useful lays there.

Treasure tiles have a rate of 10% on the map.

2.1.5. Trap Tile

A tile with a trap in it. When a player goes over this tile, he will lose 1 health point. These look like normal tiles until they are stepped on.

Trap tiles have a rate of 5% on the map.

3. Camera

The camera will be similar to a Real Time Strategy game. The camera intended to be like RTS games like Civilization V or Warcraft III.

It will be an isometric camera with a bird's eye view.

The player will be able to move the camera freely in the **movement phase** and scroll around the map.

Although the camera can be moved freely, it can also snap back smoothly to the player character if the player presses the spacebar key. Allowing them to move freely around the map to decide their next move.

In the **combat phase** the camera will snap to the player's character if he's attacked or if he becomes able to attack another player.

Camera from Warcraft III

4. Gameplay

The Player after connecting to a match will be randomly assigned to a predefined position on the map and from there the game starts for him.

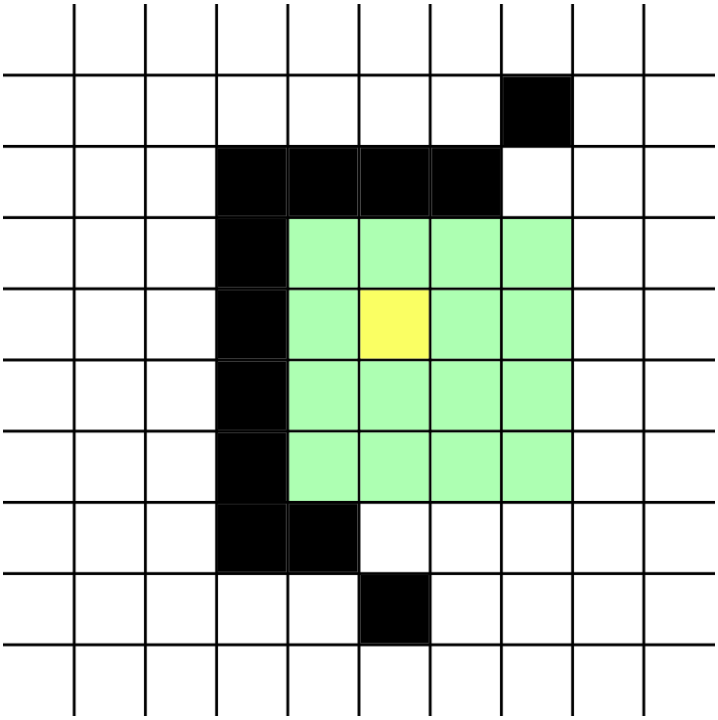
The core gameplay is organized into 2 stages that are repeating in a loop until only 1 player is left alive:

4.1. Movement phase

The movement phase is the first phase in the gameplay. Players will have a maximum amount of "Energy" that they can spend to move.

This stage lasts for **10 seconds** until every player has made his decision.

In this phase the player will be able to see too which tiles he is able to move with the energy he has. Tiles that the player can go to are highlighted in a green tint. Like in the example diagram below.



Possible movement example

1. Yellow tile is player position
2. Green tiles are the possible positions to move
3. Black tiles are walls and obstacles

While moving, the fog of war will be removed in a radius of 4 tiles around the player, thus revealing to the player the map itself.

He can now choose strategically what he wants to do before trying to engage in combat, and his weapons, can play a role in what he chooses.

The player should use this phase to:

1. take a good position to see other players or to hide from attacks if he can't see other players.
2. find treasure to acquire loot (and better weapons)

There are 3 major restrictions for the player:

1. The clock is ticking
2. He has only a certain amount of "energy" that he can use to take his position (i.e. how far can he go)
3. The player can’t walk on obstacle tiles

Examples of choices and trade-offs for the player:

1. Going to higher ground will dissolve more fog of war, and so he has a higher probability to see and attack other players (who might not even see him). However, this is a long journey, so it may take him more than 1 turn to arrive there, and he might need to resign from picking up loot nearby to save energy to get there.

2. If he decides to collect loot, he might get additional armour, better weapons, and maybe choose to hide behind a wall or rock, which will make it harder for enemies to engage and kill him.

After this phase is complete the game moves to the **combat phase.**

4.1.1. Energy

At the start of each movement phase each player will have 5 energy points to spend. The number of points he has is indicated in the UI under the Health Points.

The player is not forced to spend all his energy points

To move 1 tile the player will spend 1 energy point.

If a player kills another player he will have plus one point of energy per turn, permanently.

4.1.2. Pathfinding

When the player decides the position, he wants his character to go to, a pathfinding algorithm (A* algorithm) will run in order to find the best path from current position to desired position.

4.2. Combat phase

In the combat phase players will have the option to attack each other.

This stage lasts for 10 seconds **or** until every player has made his decision.

The players have different weapons. Some of them are close-range (knife) and others long ranged (throwable bricks, bombs).

The player has the option to attack a player or to do nothing in the case he prefers, for example, to remain hidden.

After the stage is complete (time passes or decisions are made), the players can see the animation of attacks and their results.

After the decisions are made a small number shows on top of the player indicating the amount of damage he took.

There are prerequisites to enter this phase:

1. At least 1 player must be able to see another player

The player should use this phase to:

1. Decide if they want to attack other players and with what kind of weapon.

Each player can only do one type of attack on each combat turn.

Ex. Player A sees Player B and Player C but can only attack one of them with one attack.

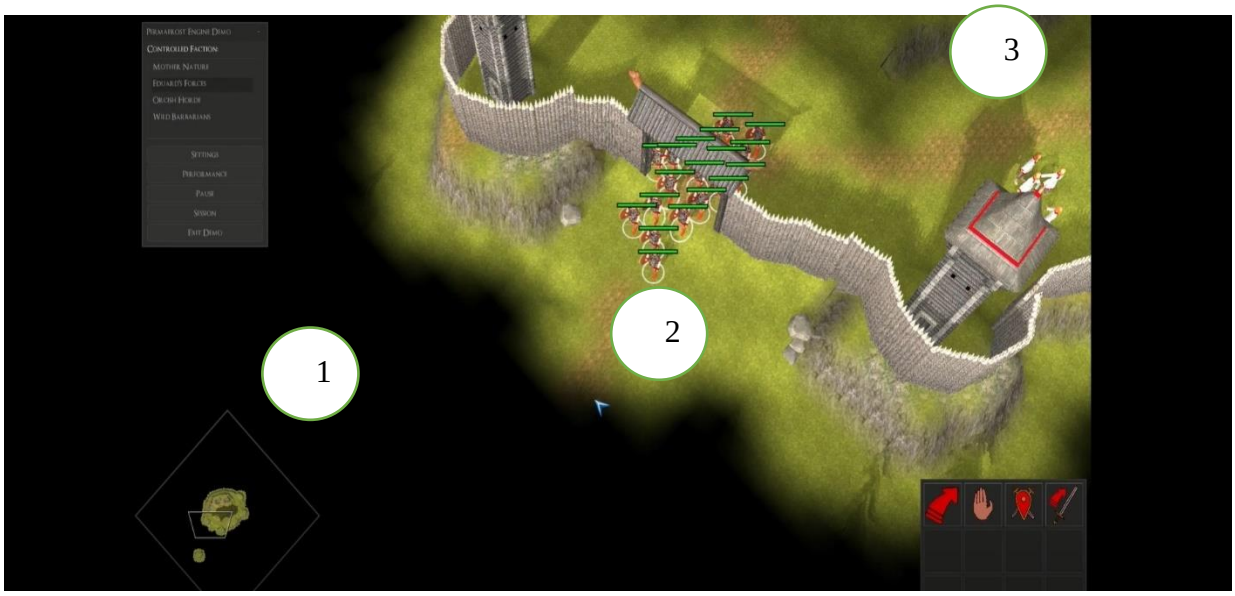
4.3. Fog of war

A core mechanic of our game is the fog of war. This will limit what a player can see and do. Removing as much fog of war as possible can prove to be a huge advantage for the player.

Fog of war can be thought of as all tiles of the map that the player still hasn't explored and thus, they are not visible. Like the image below shows when a character moves closer to tiles that he still hasn’t explored (coloured black) these are revealed and stay revealed for the remainder of the match.

Tiles that are out of the characters line of sight are greyed out.

Player can only see other player if they in his line of sight.

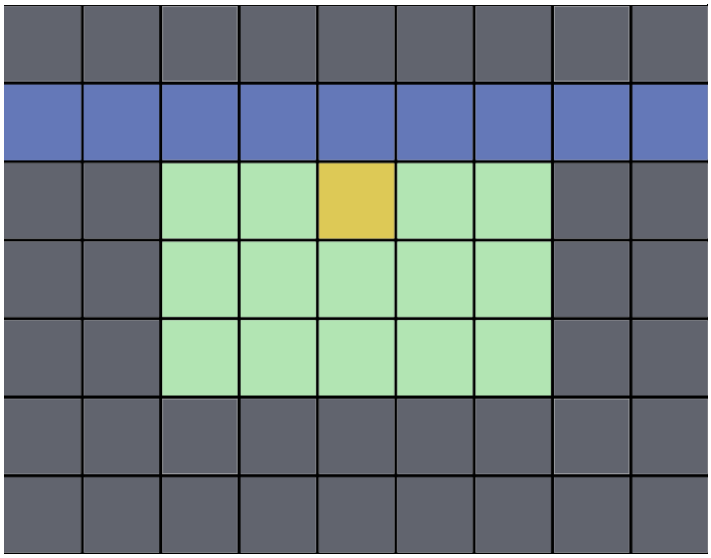


Reference image from Civilization

1. Unexplored tile (blacked out)
2. Explored tile (clear and visible)
3. Explored tile (clear and visible)

Fog of war is cleared in a radius of 4 tiles around the player. If there are walls between the player and some other tiles the fog of war will not be cleared because of the wall.

Check image below for a visual example.



Representation of the map

1. Yellow tile: player
2. Blue tile: Wall
3. Grey tile: Fog of war

4.4. Inventory

Each player will have an Inventory with 4 slots. There is no limit to the amount of ammunition a player can hold for each weapon.

This means that a player can have as much “bricks”, “bows” or other weapons as he collects. When a player collects a weapon that he already has, another “use” is added to that weapon.

4.5. Treasure

Treasure tiles are essential for a player's success. These tiles will contain weapon that will help the player defeat other players.

The player just needs to go to this tile and a random item will be added to his inventory.

Chest models taken from: https://sketchfab.com/mckinziemyra/collections/treasure-chests

5. Weapons

Weapons can be collected in treasure tiles. They are essential for the success of the player. When a weapon is collected it goes to the player's inventory. Apart from the club, weapons are only for a single usage, in other words, if a player uses a weapon - it is removed from the inventory.

If a player has already a weapon of one type and he collects another weapon of the same type, he will now have 2 usages of that weapon.

Ex: Player A has a brick in the inventory, he collects another brick he now has 2 bricks.

Below are listed all types of weapons planned to be implemented.

5.1. Close-range:

5.1.1. Club (default)

The default weapon. All players start with this weapon in the beginning of the match, and it has unlimited uses.

Damage	Range	Drop Chance	Sprite
1 unit	1 tile (must be neighbour tiles)	0%	

5.1.2. Knife (prison-made)

Damage	Range	Drop Chance	Sprite
3 unit	1 tile (must be neighbour tiles)	50%	

This weapon can only be used at a distance of 1 tile, meaning that for a player to attack another player using this weapon they must be on a tile right next to his opponent.

It causes 1 health point of damage.

5.2. Long-range

5.2.1. Throwable bricks

An efficient weapon that can attack at a distance.

Damage	Range	Drop Chance	Sprite
2 unit	3 tiles	25%	

This weapon has a reach of 3 tiles in any direction.

It causes 2 health point of damage.

5.2.2. Hand bomb

A high damage weapon that can be used at a medium distance.

Damage	Range	Drop Chance	Sprite
3 unit	2 tiles	25%	

This weapon has a reach of 2 tiles in any direction.

It causes 3 health points of damage.

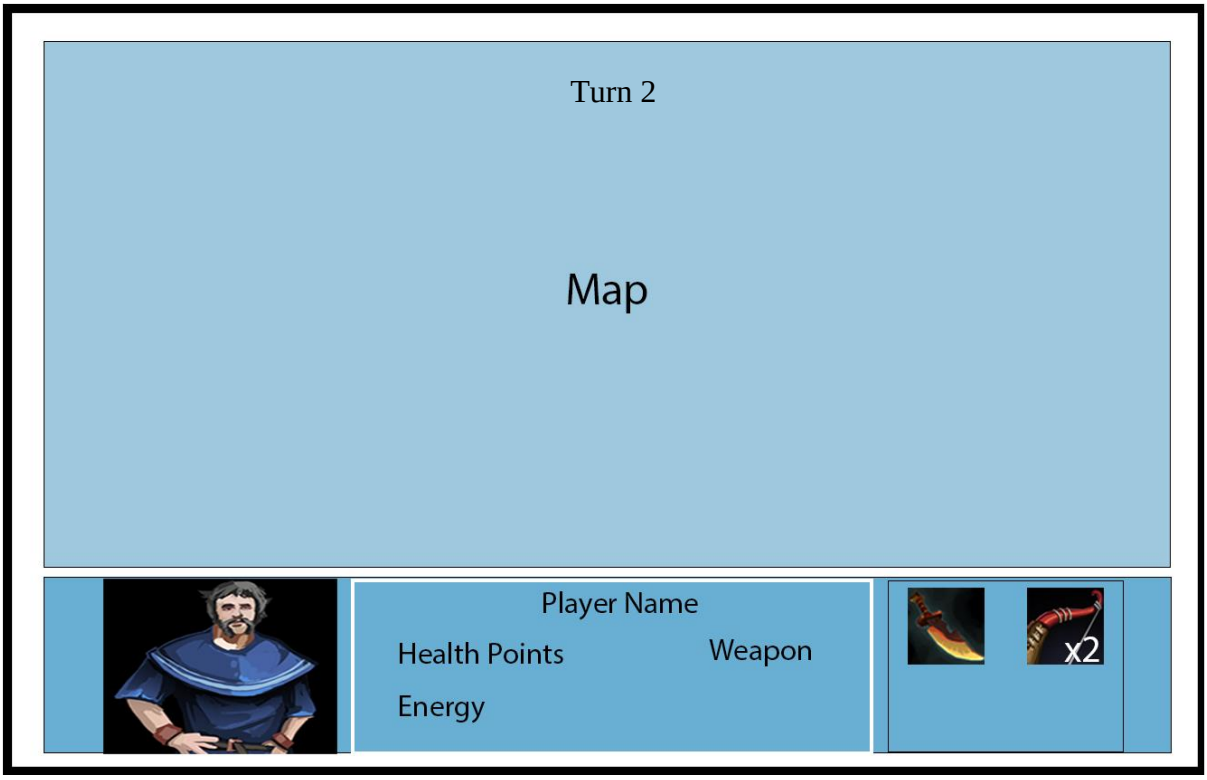
6. Front End

The UI will be divided in 4 sections: The map, player image, character information and inventory.

All of the gameplay occurs in the map section, this is where the level will be, player character, tiles, enemies and so on. On the top we can also see which turn the match is currently on.

In the inferior section of the map the rest of the UI will be present, this includes a picture of the player character, player information (name, current health points, current energy and weapon) and inventory where the player can see the weapons he has and how many uses.

Below there is a mock-up of the intended look.



In-game UI mock-up

At all times during gameplay on top of the player character there will be a health bar showing the current amount of HP the character has.

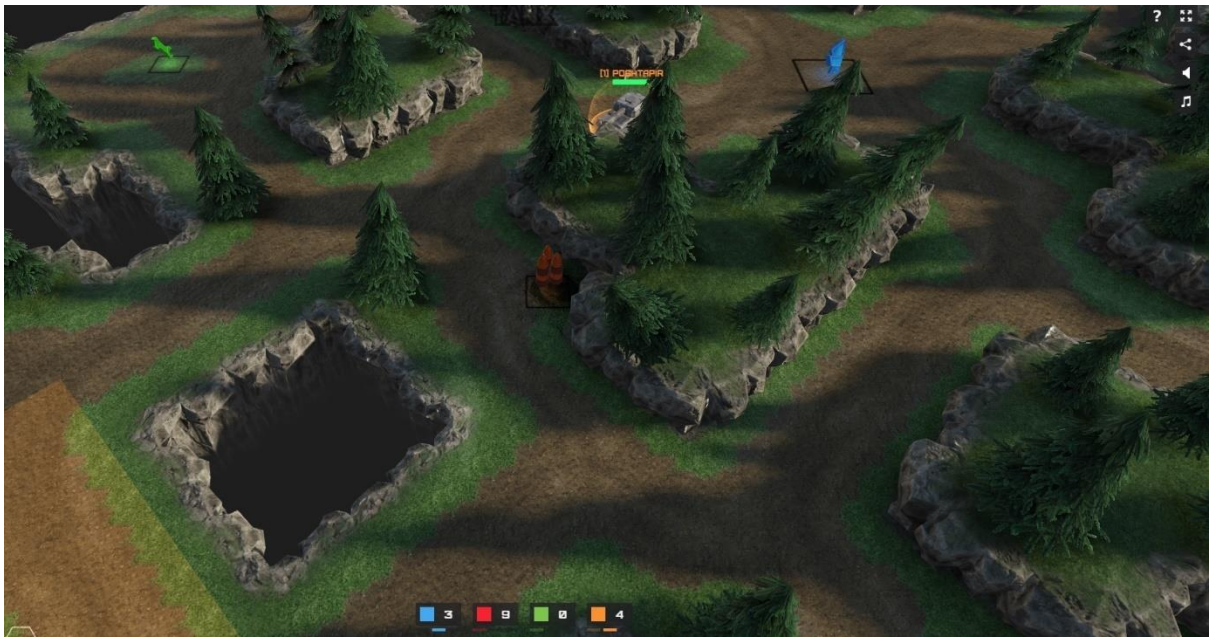
Reference image from Warcraft III

7. Graphics

The graphics are intended to have a slightly dark theme since the setting is a prison. All 3D models will be low polygon to increase the performance of the game.

To achieve the fog of war the use of shaders might be necessary, but the team still isn't 100% sure on the best way to accomplish this in Unreal Engine 4.

A good simple reference for the style of graphics for Winner Takes It All is this game: https://tanx.io/



Reference image

7.1. Particle systems

Some particle systems will be used to achieve certain effects, below there is a list that might increase in size in the future:

1. Hand bomb explosion
2. Shiny particles in treasure tiles
3. Dust particles around the map to create a prison effect

8. Assets (3D, 2D)

Most of the assets for the game will the acquired via the internet, maybe only the tiles will be modelled via Blender since they are easy geometrical shapes.

8.1. Character

Ideally 4 different character models would be necessary for the game, so that each player is assigned to one. But in order to reduce the workload maybe only 1 model, with animations will be acquired and then we change its colour to differentiate each player.

8.2. Weapons

Ideally, we will need a model for each of the weapons in the game. Additionally, we will need 1 sprite for each weapon so that they can be identified in the inventory (these are already acquired).

So, we need:

1. Club model
2. Knife model
3. Throwable brick model
4. Hand bomb model

8.3. Game World

For the game world we will need a texture or material for each of the tiles.

The mandatory model is the wall. Due to time constraints other models like rocks, cells and decoration will only be implemented if there is enough time.

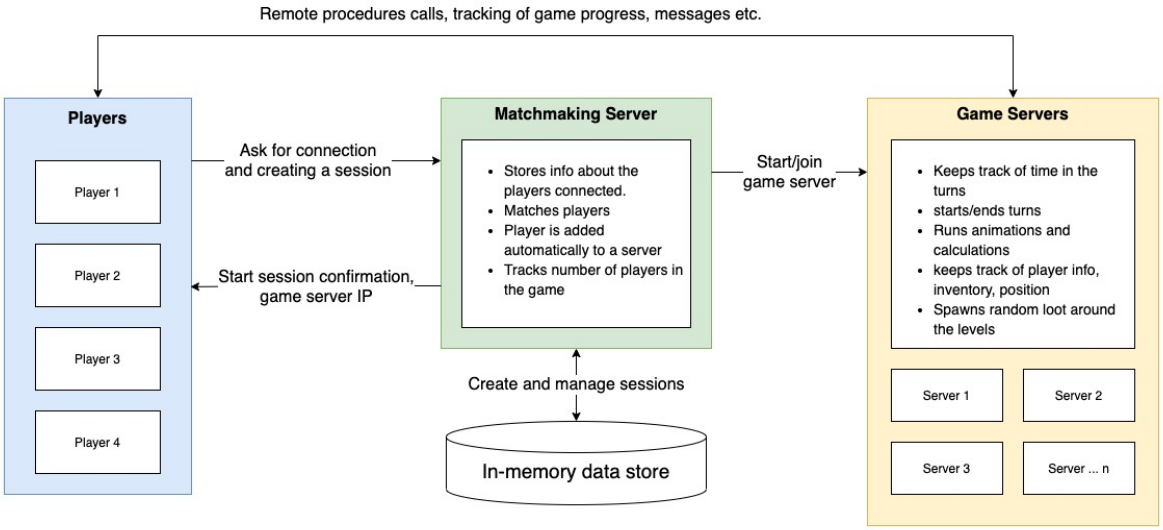
Ideally, we would also need a treasure model to put in the treasure tiles.

8.4. Music/Sfx

The following table shows the mandatory music and SFX for the game. If time allows more SFX might be added.

Music/Sfx
Main Menu music
In-game music
End of turn sfx
Damage taken sfx
Trap tile sfx
Treasure tile sfx
Defeat music
Victory music

9. Networking, matchmaking, multiplayer online spec



The multiplayer logic will be divided into three different parts:

- Client-side
  - The game package that players install and play
  - Each player's package will connect to the Matchmaking service through the internet and ask for connecting to a game session
- Matchmaking service
  - Receives incoming requests from the Clients
  - It checks for existing Game servers that have free slots for new players
    - If it can find one, it sends that server's IP address to the player, and allows for connection
    - If it can't find one, it spawns a new Game server, and sends it IP to the Client.
- Game server
  - Starts / Ends game
  - Manages all game-related logic that needs to be synchronized (position of the players, amount of world discovered, collected loot etc.)