

UNIVERSITY OF PIRAEUS

DEPARTMENT OF INFORMATICS

**Fantasy Auto Battlefield,**

an Auto Battler Card Game

This dissertation is submitted for the degree of

*Bachelor in Software Engineering and Intelligent Systems*

Supervisor

Prof. Themis Panayiotopoulos

Candidate

Konstantinos-Zois Syrios

Piraeus, February 2022

Abstract

The subject of this dissertation is the development of the game *Fantasy Auto Battlefield*. *Fantasy Auto Battlefield* combines elements from the *Card Game* and *Auto Battler* genres, and more information about those will be given in the introductory chapter. During the development of the program, special emphasis was given to the creation of an easy-to-learn and easy-to-use method of card creation, so that possible future developers may have the necessary tools to easily create new cards.

In the second chapter, there will be an analysis of the reasons for using the Unity 3D game engine and an overview of some of its features that were used during the development of *Fantasy Auto Battlefield*.

In the third chapter, an overview of the game rules and goals will be given. *Fantasy Auto Battlefield* is a game that pits the human player versus an AI player, with both players playing cards until at least one of the bases is destroyed.

The subject of the fourth chapter, will be the design of the user interface and what needs and choices lead to its development. Included there, will be an overview of the general user interface, as well as the card design.

In the fifth chapter, there will be a detailed overview of the code base. A short description will be given for every script developed and then a more comprehensive analysis of the main scripts will follow. Moreover, in this chapter the AI behavior logic will be analyzed and an overview of the used assets will be given.

Following the above, a game session will be showcased, accompanied by pictures as the player makes progress during a match of *Fantasy Auto Battlefield*.

Keywords

Unity, Design Patterns, Card Game, Auto Battler, Board Game, Hexagons

Contents

[Introduction 5](#_Toc98534788)

[The genre of the game 5](#_Toc98534789)

[Chapter 2 - Why use Unity? 6](#_Toc98534790)

[Chapter 3 – Rules and Gameplay 6](#_Toc98534791)

[Board Structure 6](#_Toc98534792)

[Terrain Types 7](#_Toc98534793)

[Playing Cards 7](#_Toc98534794)

[Units 7](#_Toc98534795)

[Buildings 8](#_Toc98534796)

[Spells 8](#_Toc98534797)

[Pawn Placement 9](#_Toc98534798)

[Frontline 9](#_Toc98534799)

[Mana 9](#_Toc98534800)

[Turn Structure 10](#_Toc98534801)

[Initiative 10](#_Toc98534802)

[Unit Movement 10](#_Toc98534803)

[Pawn Combat 10](#_Toc98534804)

[Victory Conditions 11](#_Toc98534805)

[Chapter 4 – User Interface 11](#_Toc98534806)

[Graphical Environment 11](#_Toc98534807)

[Heads Up Display 12](#_Toc98534808)

[Board UI 14](#_Toc98534809)

[Card UI 15](#_Toc98534810)

[Pawn UI 17](#_Toc98534811)

[Help Panel 18](#_Toc98534812)

[Chapter 5 – Implementation 19](#_Toc98534813)

[Overview of codebase 19](#_Toc98534814)

[Factories 19](#_Toc98534815)

[Info Containers 19](#_Toc98534816)

[Scriptable Objects 19](#_Toc98534817)

[Scripts 19](#_Toc98534818)

[Board Scripts 19](#_Toc98534819)

[Card Scripts 19](#_Toc98534820)

[Player Scripts 20](#_Toc98534821)

[UI Scripts 20](#_Toc98534822)

[HelpPanel 20](#_Toc98534823)

[MenuUI 20](#_Toc98534824)

[UI Observer-Subject 20](#_Toc98534825)

[Resources 20](#_Toc98534826)

[Analysis of main scripts 20](#_Toc98534827)

[Card Creation Method 20](#_Toc98534828)

[Presentation of the used assets 20](#_Toc98534829)

[Chapter 6 – A playthrough of *Fantasy Auto Battlefield* 21](#_Toc98534830)

[Chapter 7 – Conclusions, problems and future updates 22](#_Toc98534831)

[List of used assets (links) 23](#_Toc98534832)

[Bibliography 24](#_Toc98534833)

# Introduction

*Fantasy Auto Battlefield* was developed in the Unity 3D game engine using C#, and is a game that was built for the Windows operating system.

## The genre of the game

*Fantasy Auto Battlefield* combines elements from the *Card Game* genre and the more recent *Auto Battler* genre.

The first modern trading card game (TCG), *Magic: The Gathering* was published by *Wizards of the Coast*, Inc in 1993, and since then has inspired a great number of card games such as the more recent *Hearthstone* digital card game developed and publish by *Blizzard*.

Drawing inspiration from those predecessors, *Fantasy Auto Battlefield* supplies the player with a deck of cards, from which he draws cards and plays them by expending a resource called *Mana*. Each card may be a unit (or building) with its own stats, placed on the game’s board, or a spell card which affects the game’s board without leaving a “body” behind.

Some differences can be drawn from those games though, as the *Mana* resource here is generated by a flat amount each turn up to a maximum number that can be stored, whereas in the “*Magic: The Gathering”* for example the player needs to draw mana cards which after being played are replenished each turn. Another difference, is that in *Magic: The Gathering* and *Hearthstone* after the player draws all of his deck, they either lose or begin to take damage. In *Fantasy Auto Battlefield*, when the deck is empty, the player’s discard pile is shuffled into it.

*Fantasy Auto Battlefield* also draws inspiration from the *Auto Battler* genre. *Auto Battler* is a relatively new sub-genre of strategy video games that features chess-like elements where players place pawns on a board. The pawns after their placement, act autonomously according to some internal logic by moving and fighting opposing pawns.

The *Auto Battler* genre was popularized by games such as the *Dota Auto Chess* (which is actually a mod for the *Dota 2* game) and the *Teamfight Tactics* developed by *Riot Games.*

Similarly to those games, *Fantasy Auto Battlefield* has pawns that act autonomously although with more restraints as, for example, their movement is restricted on a single column of a hexagonal board.

# Chapter 2 - Why use Unity?

As mentioned before, the game engine that was chosen for the development of *Fantasy Auto Battlefield* is Unity 3D. Among its advantages, the Unity game engine boasts that it is royalty-free, a rich asset marketplace with even free assets, and, of course, the use of C# a modern and powerful Object-Oriented Programming (OOP) language.

Among the best features of Unity, I consider the *Monobehaviour* base class from which every Unity script derives. Using *monobehaviour’s* Awake(),Start(), and Update() methods, programming the behavior of game objects becomes effortless.

On the other hand, Unity also has two other features that greatly ease the usage of memory and those are the *Scriptable Objects* and the *Prefabs*. Both of those provide a way to implement templates for data fragments and game objects respectively, reminiscent of the *Flyweight* design pattern. Both *Scriptable Objects* and *Prefabs* were used extensively in the development of *Fantasy Auto Battlefield*.

Finally, a special mention should go to Unity’s extensive and comprehensive documentation which along with Unity’s forums, contain a wealth of information and problem-solving topics.

# Chapter 3 – Rules and Gameplay

In this chapter, the game design elements and rules of *Fantasy Auto Battlefield* will be explained.

A game of *Fantasy Auto Battlefield* takes place between a human player and an AI player. Both players take simultaneous turns that consist of five (5) phases. The goal of each player is to play cards and damage their opponent until their hitpoints reach zero (0).

## Board Structure

The board of *Fantasy Auto Battlefield* is made up of hexagonal tiles, each with its own unique terrain type.

For the dimensions of the board, after some experimentation, a five (5) width, eight (8) depth board was chosen. These dimensions provide a board of adequate size without slowing down the gameplay and offer interesting choices to the player.

The hexagons on the edges of the board in the depth axis, are of a special terrain type – the *base*. The *base* tiles represent the human player and their AI opponent.

When creating the rest of the board, each terrain type is allocated a percentage of tiles from the total board size, and then those tiles are placed randomly. This leads to a different map its game, which offers a strategic element to which the player needs to adapt differently in every game session.

## Terrain Types

*Fantasy Auto Battlefield*’s board is made up of the following terrain types:

* **Plains:** Plains have no effect.
* **Forest:** A pawn occupying a forest will have its defense increased.
* **Hills:** A pawn occupying a hill will have its attack increased.
* **Desert:** A pawn ending the turn on a desert tile will take damage.
* **Lake:**  A pawn ending the turn on a lake tile will heal a number of hitpoints.

## Playing Cards

The human player will have a deck filled with different cards of *Unit, Building* and *Spell* types. A card is drawn from the deck each turn and is played by paying its cost in *Mana*. When the player plays a card, it goes in the discard pile.

If the player’s deck is empty, and they try to draw a card then their discard pile is reshuffled into the deck.

The AI player does not use a deck, but instead chooses the cards that it will play from a tiered pool. More information on this process will be given in the fifth chapter, where a sub-chapter is completely dedicated to the AI’s behavior.

## Units

*Units* are one of the card types in *Fantasy Auto Battlefield*. When playing a *Unit* card, a new pawn is created at the chosen board tile representing the card. A *Unit* card has the following components/stats:

1. **Card Name**: The card’s name
2. **Mana Cost**: The card’s cost in mana
3. **Card Text**: The card’s special effect or flavor text
4. **Attack**: The card’s attack value
5. **Defense**: The card’s defense value
6. **Hitpoints**: The card’s hitpoints value
7. **Hex Pattern**: The card’s attack hex pattern (black: self, red: targeted hexagons)
8. **Card Type**: The type of the card

All *Units* also have the ability to move on the board. Each *Unit* moves one hex tile ahead in every move phase if that hex tile is empty.



Image - A Unit card

## Buildings

The second type of cards that the player will find in *Fantasy Auto Battlefield* are *Buildings*. *Buildings* have identical components and stats to the *Unit* cards, but most importantly they can not move. That makes the playing of *Building* pawns an important strategic decision, as placing them in the wrong board tiles may hamper the player.

## Spells

The third and final card type of *Fantasy Auto Battlefield* is the *Spell* type. *Spells* do not place a pawn on the board when played, and thus they do not have *Attack, Defense and Hitpoints* values. Instead, they have an immediate effect on the board in the form of drawing cards or destroying a pawn, etc.

## Pawn Placement

When the player decides to play a *Unit* or *Building* card, after paying its mana cost, they must decide where to place the resulting pawn. The player must always place their pawns on an empty board tile, adhering to the *Frontline* rule (see next sub-chapter). The players always have the choice of placing their pawns on the empty tiles of their base, as the opponent’s pawns never move onto them.

## Frontline

The *Frontline* rule dictates on which board tiles a player may place their pawns. The *Frontline* starts at the player’s base and expands as *Units* move on the board up to a maximum.



Image - Frontline depiction

For example, when the board is empty the player may place pawns only on base tiles (row 0). If the player owns pawns on the board that are on row one (1), he can also place pawns on that row. This effect maxes out at row two (2), as this was found to be a pretty balanced constraint for the mechanic. It also should be noted that pawns on greater rows (e.g., row 4), set the frontline to the maximum value.

## Mana

As mentioned before the main resource of the game and what the human player uses to play cards is the *Mana*. The player earns a number of *mana* at the beginning of their turn and can have stored up to a maximum of ten (10).

## Turn Structure

During a game of *Fantasy Auto Battlefield,* the players take turns that consist of 5 phases. Those are:

**Upkeep Phase:** The human player draws a card from their deck and gain mana, and also the initiative is changed (see next sub-chapter). Note: if the player’s deck is empty, the discard pile is reshuffled into it.

**Standard Phase:** The players decide which cards they will play. Each card decided upon must be paid for with mana for the human player. Then, the placement of the card’s pawn, or targeting of the card’s effect takes place, after which the card is moved to the discard pile.

**Move Phase:** The *units* on the board will now execute their auto-moves.

**Combat Phase:** The pawns now will attack any, and all possible targets.

**End Phase:** Effects that take place at the end of each turn are activated now (e.g., the desert tile terrain effect).

## Initiative

In case of ties during a game, such as *Units* of both players wanting to move in the same board tile, initiative is taken into account to determine priority. Only one player *has the initiative*, symbolized by the green coin, visible in *Image 2.* The *initiative* token changes hand every turn during the *Upkeep Phase.*

## Unit Movement

During the **Move Phase** all *units* will move one (1) board tile ahead in their column. The order in which the *units* will move is the following:

* The *units* of the player that has the *initiative* move first
* The *units* that have the greater distance from their base will move first
* No *units* may pass through friendly or enemy pawns
* No *units* may enter the enemy’s base tiles

## Pawn Combat

During the **Combat Phase**, all pawns attack simultaneously all available targets. Each pawn will compare its attack value with the defense of its targets. The attacker will deal as damage the differential between *Attacker’s attack* and *Defender’s defense* or a minimum of one (1), whichever is greater.

Every pawn that reaches zero (0) hitpoints be removed from the board.

The pawns have a *hex pattern* that they are able attack to. For example, a pawn may be able to attack the board tiles directly in front of it and behind it. Another pawn may be able to attack the three frontal board tiles.

When a pawn reaches the enemy base, it will deal 1 damage each turn to the base, even while attacking other pawns.

## Victory Conditions

A game of *Fantasy Auto Battlefield* ends when at least one of the players’ bases reaches zero (0) hitpoints.

# Chapter 4 – User Interface

The two main purposes that drove the decisions for the User Interface’s (UI) design, were simplicity and readability. In this chapter, the UI will be showcased using screenshots and explaining some of the logic behind its design.

## Graphical Environment

For the graphical environment of the game, a feeling of playing a board game was desired. To achieve this, a 3D room was created containing a table on which the game board is set up. Plenty of lighting in the room and warm colors on the wooden and fabric textures lead to a comfortable and enjoyable experience during the game.

The player can fly around the room using the camera controls (that will be explained in the next chapter) bound only by some constraints -free movements is only allowed within the bounds of a cube, smaller than the room and above the game board’s height- so that the camera movement does not interfere with the gameplay experience.



Image 3 - Graphical Environment

## Heads Up Display

The Heads Up Display (HUD) of *Fantasy Auto Battlefield* is split into three (3) main parts.

Firstly, there is the upper part of the HUD. It contains the information about the hitpoints of the human player and how many mana points they have. It also states in which game phase the game is in.



Image 4 - HUD upper part

The second part is where most of the information useful for the player is displayed. This part can also be split into three (3) parts of its own:

1. The Card Info section where information will be displayed when the player clicks on a pawn on the game board.



Image 5 - HUD, Card Info section

1. The Tile info section where information will be displayed when the player clicks on a tile on the game board.

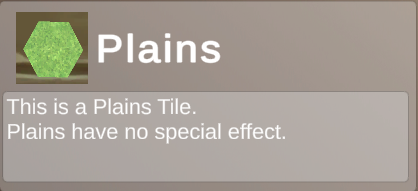


Image 6 - HUD, Tile Info section

1. The menu section where the following buttons are displayed:
   1. **Toggle Hand Button:** it shows or hides the player’s hand of cards.
   2. **Play Card Button:** when the player has selected a card and wants to play it, they will click on this button.
   3. **End Phase Button:** this button is used to move to the next phase of the turn.
   4. **Open Menu Button:** this button opens the menu that the player can use to exit the game or view the *Help Panel.*

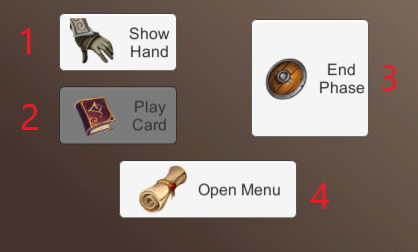
****

Image 7 - HUD, Button section

And last but not least, there is the game log where the progress of the duel is recorded in text format. The game log can also be minimized using the upper right button as displayed below:

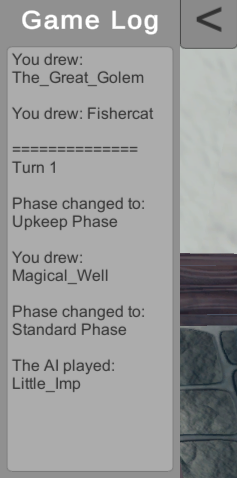


Image 8 - HUD, Game Log

## Board UI

The *Board UI* is a rather small category that contains the display of the bases’ hitpoints, the human player’s deck counter and the *Initiative Token.*



Image 9 - Base's Hitpoints



Image 10 - Deck Counter



Image 11 - Initiative Token

## Card UI

The components of a card’s UI have been previously shown in *Image 1*, so here instead there will be a comparison between a *Unit/Building* and a *Spell* card.



Image 12 - Building Card

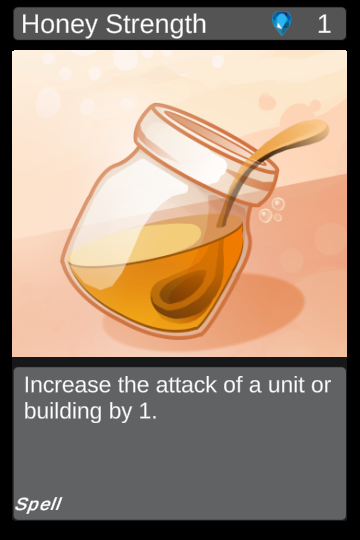


Image 13 - Spell Card

As you can see the *Building* card has virtually no differences with *Unit* card. The difference between *units* and *buildings* is that the former can move.

On the other hand, the *spell* cards are “empty” in comparison. They only display the name, the cost and the effect of the card.

## Pawn UI

Each pawn is represented on the board with the game object depicted in the following image. The most prominent feature is the card’s artwork and the only thing that is displayed are the card’s current stats. This makes the state of the board extremely readable and easy to understand what card is each pawn.

The pawn’s stats are displayed with the same order as on the cards (i.e., Left: *Attack,* Upper Right: *Defense,* Down Right: *Hitpoints*).

*Note: The color of the pawn’s stats changes to green, white, and red when the current value is higher, equal, or lower than the original values respectively.*



Image 14 - Pawn

## Help Panel

Finally, the player has access to a *Help Panel* in which information about the game can be read, accompanied by useful images.



Image 15 - Help Panel

The player can browse through the entries of the *Help Panel* using the buttons at the bottom of the screen. The *Help Panel* was coded using Scriptable Objects for the entries, in order to be very flexible and extendable.

# Chapter 5 – Implementation

## Overview of codebase

TODO mention extendibility, use of patterns, etc. – Alphabetical order by folder

|  |  |
| --- | --- |
| Factories | |
| BuildingFactory | is method is used to execute the effects of hills and forests, i.e the tile types that change pawn stats.  /// For those effects coordinates and the pawn is required, in order to determine if the pawn enters or leaves a tile of those types,  /// and to change its effects |
| SpellFactory | change pawn stats.  /// For those effects coordinates and the pawn is required, in order to determine if the |
| UnitFactory |  |
| Info Containers | |
| CardType |  |
| GamePhases |  |
| HexDimensions |  |
| HexPatternCodes |  |
| TileHeight |  |
| TileType |  |
| Scriptable Objects | |
| BuildingCardData |  |
| SpellCardData |  |
| TileData |  |
| UnitCardData |  |
| Scripts | |
| FlyCamera |  |
| GameManager |  |
| GameManager\_PartialHelpers |  |
| SFXController |  |
| Board Scripts | |
| BoardGenerator |  |
| BoardManager |  |
| HexTile |  |
| Card Scripts | |
| Building |  |
| Card |  |
| IEffectOnDeath |  |
| IEffectWhenSpawning |  |
| IEffectWithTargetWhenSpawning |  |
| PawnStats |  |
| Spawnable |  |
| Spell |  |
| Unit |  |
| Player Scripts | |
| AIPlayer |  |
| Deck |  |
| DiscardPile |  |
| Hand |  |
| HumanPlayer |  |
| Player |  |
| UI Scripts | |
| BoardUI |  |
| CardInHand |  |
| GameLog |  |
| MainUI |  |
| HelpPanel | |
| HelpPanel |  |
| HelpPanelEntry |  |
| MenuUI | |
| CreditsPanel |  |
| MenuBackground |  |
| MenuButtons |  |
| UI Observer-Subject | |
| IObserverUI |  |
| SubjectUI |  |
| UIEvent |  |
| Resources | |
| CardCatalogue |  |

## Analysis of main scripts

## Card Creation Method

## Presentation of the used assets

# Chapter 6 – A playthrough of *Fantasy Auto Battlefield*

# Chapter 7 – Conclusions, problems and future updates

# List of used assets (links)

# Bibliography