UNIVERSITATEA ”ALEXANDRU-IOAN CUZA” DIN IAȘI

FACULTATEA DE INFORMATICĂ

LUCRARE DE LICENȚĂ

Got you 2D

propusă de

Darie Camelia Beatrice

Sesiunea: februarie 2023

Coordonator științific

Lect. Dr. Moruz Alex

UNIVERSITATEA ”ALEXANDRU-IOAN CUZA” DIN IAȘI

FACULTATEA DE INFORMATICĂ

Got you 2D

Darie Camelia Beatrice

Sesiunea: februarie, 2023

Coordonator științific

Lect. Dr. Moruz Alex

Avizat,

Îndrumător lucrare de licență,

Lect. Dr. Moruz Alex.

Data: ..... Semnătura: .....

Declarație privind originalitatea conținutului lucrării de licență

Subsemnatul **Darie Camelia-Beatrice** domiciliat în **România, jud. Galați, mun. Galați, Strada Traian, nr. 58, bl. D, et. 4, ap. 16**, născut la data de **21 iulie 2000**, identificat prin CNP **6000721170030**, absolvent al Facultății de informatică, **Facultatea de informatică** specializarea **informatică**, promoția 2018, declar pe propria răspundere cunoscând consecințele falsului în declarații în sensul art. 326 din Noul Cod Penal și dispozițiile Legii Educației Naționale nr. 1/2011 art. 143 al. 4 și 5 referitoare la plagiat, că lucrarea de licență cu titlul **Got you 2D** elaborată sub îndrumarea domnului **Lect. Dr. Moruz Alex**, pe care urmează să o susțin în fața comisiei este originală, îmi aparține și îmi asum conținutul său în întregime.

De asemenea, declar că sunt de acord ca lucrarea mea de licență să fie verificată prin orice modalitate legală pentru confirmarea originalității, consimțind inclusiv la introducerea conținutului ei într-o bază de date în acest scop.

Am luat la cunoștință despre faptul că este interzisă comercializarea de lucrări științifice în vederea facilitării falsificării de către cumpărător a calității de autor al unei lucrări de licență, de diplomă sau de disertație și în acest sens, declar pe proprie răspundere că lucrarea de față nu a fost copiată ci reprezintă rodul cercetării pe care am întreprins-o.

Data: 15.02.2023 Semnătura: 

Declarație de consimțământ

Prin prezenta declar că sunt de acord ca lucrarea de licență cu titlul **Got you 2D**, codul sursă al programelor și celelalte conținuturi (grafice, multimedia, date de test, etc.) care însoțesc această lucrare să fie utilizate în cadrul Facultății de informatică.

De asemenea, sunt de acord ca Facultatea de informatică de la Universitatea ”Alexandru-Ioan Cuza” din Iași, să utilizeze, modifice, reproducă și să distribuie în scopuri necomerciale programele-calculator, format executabil și sursă, realizate de mine în cadrul prezentei lucrări de licență.

Absolvent **Camelia-Beatrice Darie**

Data: 15.02.2023

Semnătura: 

Contents

Introduction

Card games are one of the most played games today. They are loved by many people worldwide for their simplicity and ease of play. If one might want to keep their children or friend entertained, having a deck of cards can be of great help, or at least it used to be. Nowadays most young adults are more interested in playing games on their laptop, rather than playing in real life. This is the reason many card games are now available online. Not only that, but a lot of online card games have been invented lately, such as Hearthstone[1] or Mythgard[6].

My interest in creating my own games was born during high school when most of my colleagues were playing various games like League of Legends[9], Call of Duty[10] or Hearthstone[1]. I wondered about how much work the developers put into those games, how rewarding it might feel to see people from all around the world enjoy a game that you have created. However, my passion only materialized during the *Game design* classes from the third year of university. After attending those classes I have decided that I want to make a game of my own. Card games have always been my favourite because they are always fun to play and most of them require you to have a strategy.

Contributions

The idea of the game came to me after playing *Hearthstone* for most of my high school years. It was, by far, one of the most popular game at that time, played by many people, including almost all of my classmates during breaks. While it is not the first game of this kind, it was the most interesting one.However, the game itself was completed by myself, with the help of my coordinator. To get a grasp of how games are done in Unity[2] I have watched many YouTube tutorials, most of them being about creating multiplayer card games. After getting an idea about what I wanted to do, I have started working on the cards, their attributes and what they should look like. After doing the card, I also had to add effects to them, because the game has both simple minions and special ones. The cards are modelled in such way, that I could always add new effects to the cards. Thins thing is extremely good because continuously adding new features, cards in my case, to the game is what keeps a game interesting to its audience. Another important thing that I had to do was structure the game in scenes.

Chapter 1

The Concept

1.1 Description

Got You 2D is a game played against the AI, in which the player has to attack the opponent and his monsters until wither his or his enemy's HP drops below 0. The player has to select a deck to play with before the game starts; his deck can contain simple minions, minions with effects or even spells.

Being a strategy game, the player has to be smart about his actions, decide what is better to attack at times, the opponent or its minions. Not only that, but also to know when it's the perfect time to play a specific card. While some are better played in early game, others have a bigger effect in late game.

1.2 Inspiration Source

As I have mentioned above in the Introduction, there are many games of this kind, however the biggest two games of this kind are, in my opinion, *Hearthstone* and *Magic: The Gathering*.

Even though they have resembling concepts, they are major differences between the two of them. In *Magic: The Gathering* mana is never a given, you only have as much mana as you have mana-generating cards in play, and if you want access to those cards, you'll have to draw them, but *Hearthstone* has a guaranteed system of mana because the player's mana pool starts at one and will be increase by one each turn, until it reaches a maximum of ten mana. Creature combat is another aspect where these 2 differ. In *Magic: The Gathering*, creatures attack the defending player, and that player

chooses whether to block with their own creatures or take damage. So, any time you attack, you should think about how your opponent could potentially block. That is not really possible in *Hearthstone*. While they are some exceptions, when a minion attacks another minion, the attacked minion's health will decrease inevitably. For example if 3(attack)/3(health) minion attacks a 2/2 minion, the 2/2 minion will die and the 3/3 one will become 1/3. Considering these big differences, I believe that **Got You 2D** takes more after *Hearthstone*.

Similarities between **Got You 2D** and *Hearthstone* include:

- Having the possibility to choose your deck in the Deck Builder. Knowing what cards to choose plays a really important role in a good player's strategy.
- The game ends when either opponent's or player's health gets to or below 0.
- The possibility to attack the enemy or his minions.
- Some of the cards in this game have the same effect as ones in *Hearthstone*. I chose to mirror some cards found in *Hearthstone* because I thought they were interesting and would be appreciated by the players.
- The cards are modelled in the same way, with minions having health, attack and mana. Their design is also a bit similar.
- Players have to pay with mana to play cards. The bigger the effect of a card, the more mana is costs.
- There are two type of cards in both games: minions and spells. While minions persist until their health gets to or below 0, spells can be used once and disappear after.
- The existence of the *Coin* card
- Most cards can attack only once. While in *Hearthstone* there are some cards that can attack twice, I chose not to implement this effect.

Differences between **Got You 2D** and *Hearthstone* include:

- *Hearthstone* Limits your deck to only have 2 copies of the same card maximum
- *Hearthstone* gives the Player vs Player option, while **Got you 2D** only offers the Player vs AI.

- **Got you 2D** also shows the players exactly how many cards are left in their deck and I also added a *Graveyard*; this shows how many cards a player has played so far.

- There are many cards whose effects I chose not to implement because I believe they are too complex.

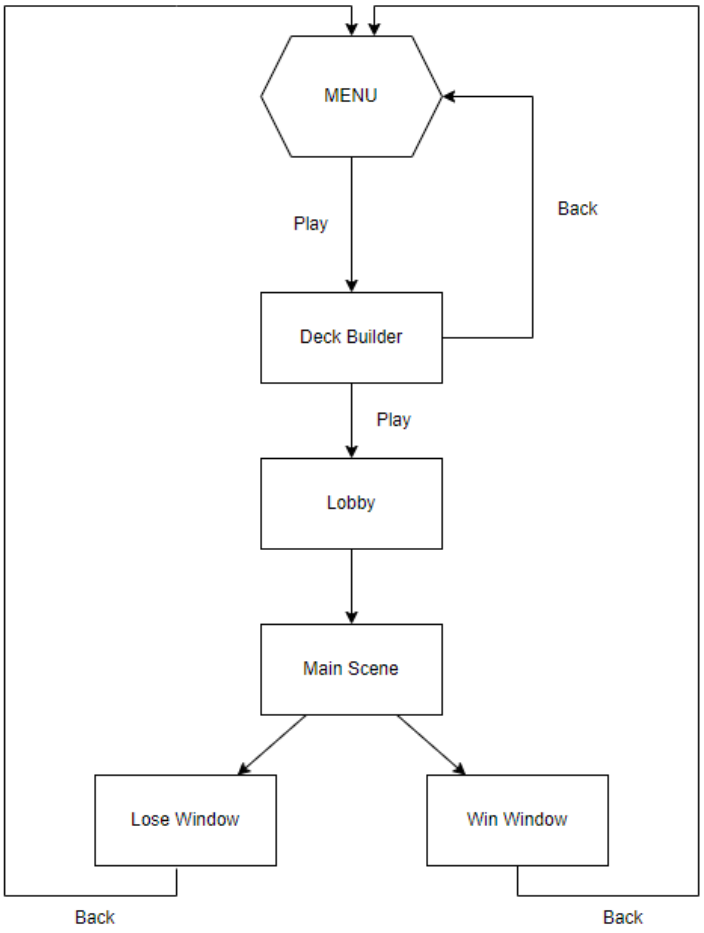
1.3 The course of a player in the game

When first starting the game, the player will find himself in the **Menu** scene. If he selects to play, he will be taken to the **Decks** scene. Here he could select the cards for his deck and he is also able to visualise the chosen cards. It being a strategy game, choosing the right cards is a crucial step. The player should keep in mind that there are some cards that are only good in late game, and some who have a bigger effect in the early game. Choosing only one type of cards, may very well result in the player losing because the player will either have a rough start or will be too weak in the late game. There are two buttons available in this scene, the player can either choose the *Back* option to return to the **Menu**, or he can press *Play* to start the game. After hitting *Play*, the player will be transferred to the *Lobby* where he will wait a few seconds for his game to start.Finally, he will find himself in the *MainScene* where the game begins. Both the player and his opponent receive **five** cards.

It is decided randomly who starts and the other person receives the **Coin**, a card which costs 0 mana and gives the user one more mana that turn only. I chose to give this coin to the player playing second for balance reasons.It is worth knowing that the first turn a minion is played, it can't attack usually, thus the person who starts playing first will have an advantage because he will be the first to attack. By using the coin, the second player could play a stronger minion. Now the battle itself starts. Players take turns to play minions or spells and to attack. As mentioned above, a monster can attack either the player or another minion belonging to the enemy and it can only attack once. The only card that can attack twice in the game is the "charge" card. After a while, one of the players' health will drop below 0 and the other will win.

If the player wins, the **Win Window** will activate, letting the player know of his win, otherwise the **Lose Window** will activate. After one of those windows activates, the player will have to press the *Back* button to go back to the **Menu**. To simplify it,

this would be the flowchart of the game:



1.4 Cards and their effect

Got You 2D offers a pretty big range of cards to choose from when building your deck. Here are some of them and the effects they have:

- Charge: this card can attack immediately, not having to wait for one turn to become active.
- Draw1 and Draw2: these cards give the player another one or two cards. The more cards you have, the bigger your range to choose from will be.
- Cards that increase your mana this turn only or even some that increase your maxMana permanently

- last but not least, there are many simple minions with different strength levels. I created them in such way that the player will be able to play with simple minions both in early and late game.

While it may not seem like there are many cards to choose from, all of this combined ffer the player a big enough pallette to choose from, depending on their strategy.

Chapter 2

Technologies and tools

In this chapter I will present the technologies I used and my personal approach towards them. I will talk about Unity and my resources.

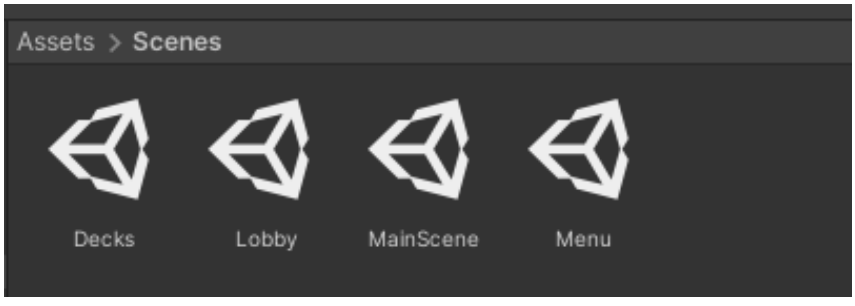
2.1 Unity

Unity is a cross-platform game engine developed that e can be used to create three-dimensional (3D) and two-dimensional (2D) games, as well as interactive simulations and other experiences.

I have chosen this engine this engine because it is by far the most popular one, therefore it is extremely easy to find tutorials for different game and most of it's functionalities are explained really well.

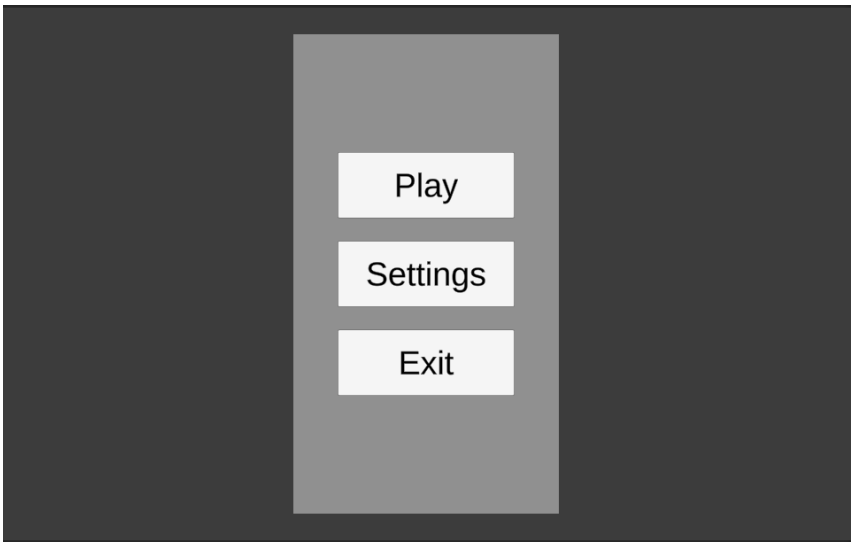
2.1.1 Scenes

Scenes are where you work with content in Unity. They are assets that contain all or part of a game or application[3].These are the scenes that I used:

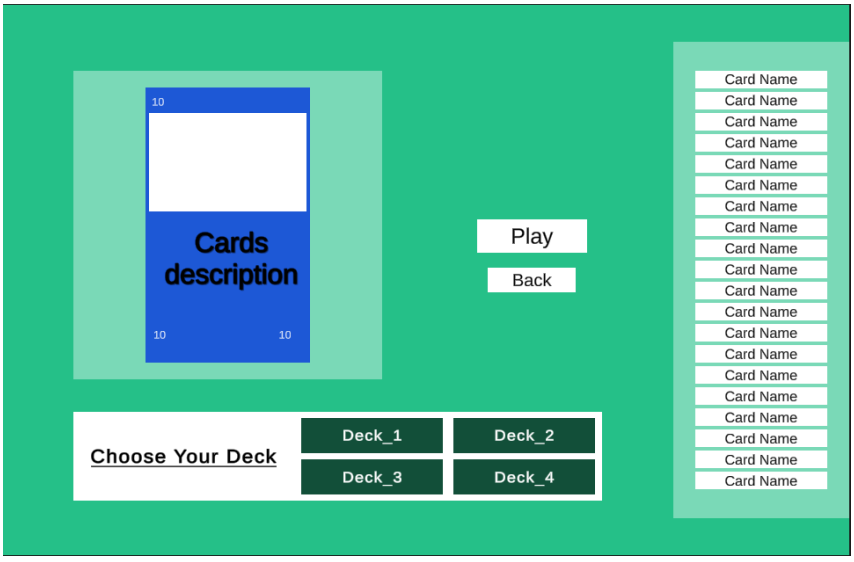


Now we will take each of the scene and present it.

- Menu: is is the scene where the game starts. It gives the player 3 options: to exit, to change the settings of the game (not completed yet) or to play.



- Decks: here the player can make his own deck and select it to play with it. He has the possibility to create 4 different decks and to play with whichever he pleases.



- Lobby: this is an intermediate scene, this prompts the player to wait till the game starts. The player does not have to do anything here, as the game will automatically transfer him to the next scene

- Main scene: here the player will have to face his opponent and play to win. Here players play cards, take and give damage. Here, in blue we have the players' mana, in purple we can see the health of each player, the green text represents the cards remaining in the deck and lastly, with red we can see how many cards there are in the Graveyard of the players.

2.1.2 Prefabs

Unity's Prefab system allows you to create, configure, and store a GameObject complete with all its components, property values, and child GameObjects as a reusable Asset. The Prefab Asset acts as a template from which you can create new Prefab instances in the Scene.[4]

In my case, I transformed the Card into a prefab, because I also need it in the *Main Scene*, but also in the *Decks* scene

2.1.3 Update() method

Update is a function which is called every frame,but only if the MonoBehaviour is enabled. I use it many times, but a really important example is in the *CardController* script where I model cards behaviour.

```
void Update()
{

    if (canAttack && targetEnemy == true && attacking && attackingC
    {

        playerManager.GetDamagae(currentAttack, 1);
        summoningSickness = true;
        canAttack = false;
        attacking = false;
        targetEnemy = false;
        print("attack");

    }
```

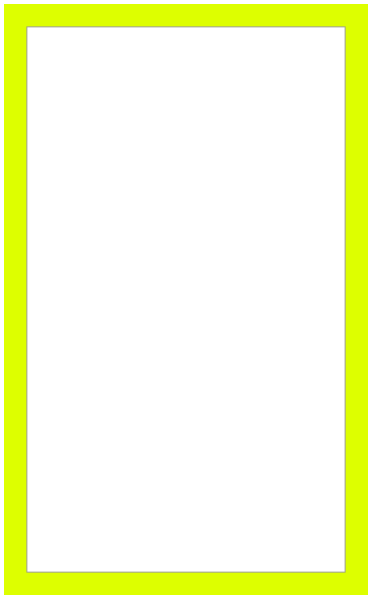
```
        if (isSummoned)
            if (isYourEnemy == false)
            {
                if (this.currentHealth <= 0)
                {
                    border.SetActive(false);
                    enemyBorder.SetActive(false);
                    graveyardManager.AddCard(card[0], 0);
                    Destroy(this.gameObject);
                }
            }
            else
            {
                if (this.currentHealth <= 0)
                {
                    border.SetActive(false);
                    enemyBorder.SetActive(false);
                    graveyardManager.AddCard(card[0], 1);
                    Destroy(this.gameObject);
                }
            }
    }
```

2.1.4 TextMeshPro

”TextMeshPro is the ultimate text solution for Unity. It’s the perfect replacement for Unity’s UI Text and the legacy Text Mesh.TextMeshPro provides Improved Control over text formatting and layout with features like character, word, line and paragraph spacing, kerning, justified text, Links, over 30 Rich Text Tags available, support for Multi Font & Sprites, Custom Styles and more.”[5]

2.2 Resources

When working of the card's behaviour during attack, I wanted to add borders, one red to signify that card is attacked and a yellow one to show that this card is attacking. I have soon realized that because of the way mt card prefab worked, I couldn't really do that easily. After some brainstorming I realized that T could use *GIMP*[11] to create the border. This is what it looks like:



It's dimensions are 160x260 and the card's are 150x250. I made the border in such way that 150x250 of it is transparent with a border of 10.

Chapter 3

Game mechanics

3.1 Base rules

Playing a game like this could not be possible without there being some rules enforced. Some of these were already mentioned above, but just to make sure that the the main rules of the game are known, here they are:

- When minions are first played, they can not attack.
- One player starts and the other receives the coin.
- The game ends when either the player or the enemy has a HP of or below 0.
- When a minion attacks another minion, it also takes damage equal to the other enemy's attack.
- When a minion's health drops below 0, it dies.
- A minion can only attack once per turn.
- When a card is played, it costs mana. The more important is the game, the more mana it will cost to play.
- When the game first starts, every player gains 1 mana. Each turn their max mana is increased.
- At the start of the turn, the person who's turn it is receives a new card from the deck.

3.2 Cards

This is what every card looks like. Each one has these five components: mana, health, attack, name and description.

- Mana: it is used to play cards; most cards cost mana to play, but there are some exceptions.
- Health: every minion has health. However spells do not have health, therefore once they are played, they disappear.
- Attack: this component defines how much damage a card deals.
- Name: is the name of the card.
- Description: a brief text that explains what the card does so the player can know what will happen if he plays the card.

3.3 AI behaviour

The algorithm I used for the AI is not really complex. Every turn it will play exactly one card and will also attack the player's health will all his available minions. I considered that, for now, attacking the player will be more efficient and more helpful for the AI. In the AI.cs file I call the *CalculateDmg* method in order to establish how much damage can the AI inflict upon the player. This is what the method looks like:

```
    public int CalculateDmg()
{
    int x = 0;
    for (int i = 0; i < enemyZone.Count; i++)
    {
        if (enemyZone[i].summoningSickness == false)
            x += enemyZone[i].card[0].cardAttack;
    }
    return x;
}
```

To better understand the code:

- enemyZone: it represents how many cards the AI has on the table right now, these are the cards that he will attack with
- summoningSickness: this variable shows if a card can attack or not. When it is set to *False*, the card can attack.
- cardAttack: it represents the attack of the card

Chapter 4

Making it available to the world

4.1 Target audience

My main target would be young adults, with ages between 10 and 26. I believe that anyone under 10 would not have the mental capacity to play and appreciate the game and anyone above 26 would not have the time to play such a complex game. They should be interested in card games, strategy games and collectible games.

4.2 Advertising it to the public

My opinion is that what decides if a game will be successful or not is the way it is advertised. Even if a game might be good, it won't reach its full potential unless it is properly promoted. Before publishing the game, I will try to collaborate with both big and small content creators on YouTube. To promote it in Romania I would get in contact with CreativeMonkeyz[12] and to reach a larger audience I would try to get a paid partnership with PewDiePie[11].

However, I don't think that would be enough so I will also advertise it on Facebook[13], Instagram[14] and Twitch[15] by using targeted ads.

Conclusions and future plans

While **Got You 2D** is not very complex yet and it's algorithms are quite simple, I would say that it is a good game, developed with passion and hard work. It could certainly use some improvements,but it also has many features and strength points.

Its **strengths** include:

- It is interesting. This is extremely important when talking about a game. People are more inclined to play it if they find the concept intriguing. And the game actually being interesting and putting their mind to work is what keeps them playing the game and even telling their friends about it.
- The option to play against an AI. We all have played game where it was taking so long for another player to join, that you just gave up. Always being able to play is a really big plus, I believe.
- Clear rules. In this day and age people are always on the rush, always needing things to go fast and straight to the point. Them understanding the rules make sure that they enjoy the game, can try their actual best and don't quit out of frustration.
- Many cards to choose from. This being a collectible game, the main attraction are the cards. The bigger their range is, the more interesting it keeps the game.

Improvements that I plan to implement from now on:

- Better graphics. While it may have not mattered this much decades ago, at this moment, the graphics are one of the biggest selling point of a game. Whenever a new game drops a teaser or a trailer, the quality of the graphics is the second thing they comment about, after the concept.
- The option of multiplayer. It is very rare for people to play alone, most of them think of video games as way to do something while spending time with friends.

And if they do not have friends to play, they try to make some bu chatting with people they find in the game.

- Even more cards, more complex. Even though there are already some interesting cards in the game, there could always be more. This will be one of the min thing that I will focus on, when improving the game.
- Different level of difficulties. Some players will eventually become better than the most others, therefore will want new challenges and to play with people who have the same skills as them.
- A better AI. As I have said before, the algorithm is not very complex yet. I will change it so the AI can play more cards and to be able to also attack the player's minions.

To conclude, I believe that the game has real potential and many people would be interested in playing it. It has many strengths and we were already proven that this kind of concept attracts many people.

Bibliography

1. Blizzard Entertainment, *Hearthstone*, 2014
2. Wikipedia, Unity (game engine), *https://en.wikipedia.org/wiki/Unity_(game_engine)*
3. Unity Manual, Scenes, *https://docs.unity3d.com/Manual/CreatingScenes.html*
4. Unity Manual, Prefabs, *https://docs.unity3d.com/Manual/Prefabs.html*
5. Unity Manual, TextMeshPro, *https://docs.unity3d.com/Manual/com.unity.textmeshpro.html*
6. Rhino Games, Mythgard, 2017
7. Niantic, Pokemon Go, 2016
8. TSR, the Coast, Dungeons & Dragons
9. Riot Games, League of Legends, 2009
10. Infinity Ward, Call of Duty, 2003
11. GIMP Development Team, GIMP, 1998
12. Wizards of the Coast, Magic: The Gathering,1993