Mage Academy Documentation

Che Wang (Watcher) & Yaofei Wu (Villa)

Game Description:

Mage Academy is a fast-paced 2D action game where two players can play an intense magic duel by each controlling a mage that is capable of using a variety of unique magic spells.

The two mages can move in a 2D area in 8 directions (left, right, up, down and diagonal), magic spells will be launched at the current direction that the player is facing. The battles are fast-paced and real-time and although it's easy to learn the basic moves, it requires a considerable understanding of the effects of different spells to develop interesting strategies in order to defeat more powerful opponents. Effective keyboard control can also help a player to gain advantage.

The two players initially have identical health points and mana points. The mage can use a basic offensive skill called "arcane bolt" that can shoot out magic bolts that deal some damage and only cost a small amount of mana. But to use more powerful spells, the mage has to first use mana to generate magic orbs then convert those orbs into spells. The type of powerful magic that will be used depends on the key pressed as well as the number of each of the 2 types of magic orbs.

===
The spells are diverse in their types. There are four different categories of magic spells:

Basic offensive spell - shoot out 3 arcane bolts, each will deal a small amount of damage, does not require magic orb.

Powerful offensive spell - It consumes magic orbs which are generated by the player. If the player currently has more fire orbs, the spell is fireball, which creates a deadly burning projectile that will explode on contact. If the player currently has more ice orbs, the spell is ice shards, which creates 20 sharp and fast ice pieces that launch into the area the mage is facing. If the player currently has equal amount of two types of orbs, a shadow blade will not deal much damage but it is concealed in the shadows and hard to see. It has limited damage but can temporarily slow down the enemy mage and poison the enemy summoned creature, causing it to gradually lose health. Therefore, it's a powerful strategic spell that can be very useful when the mage is trying to escape from the an enemy or when the mage is trying to trap an enemy.

Summon creatures - Compared to offensive spells, it consumes more energy orbs. If the player generates more fire orbs, the creature summoned is the Fire Ghost. Fire Ghost constant seek and tries to approach the other mage, their fire deals damage over time when they are near a mage or enemy creature. If the player has more ice orbs, the creature summoned is the ice totem. Ice totems are stationary. They would automatically aim at the other mage or enemy creatures and shoot out ice blades from time to time. Also, they can block the way of the other mage so they can be used as strategic obstacles. A mage can use ice totems as barriers to escape from the other mage, or use them as shields to dodge bullets and attacks. If the player generated the same amount of fire and ice orbs, the creature summoned is the swallow. The swallow is a agile ranged unit that would fly swiftly to a random position from time to time and shoot out a missile which tracks and tries to hit the other mage. The missile deals a large amount of damage but can be shot down using spells and they only last a certain amount of time.

Wind Blast- A strategic spell that doesn't cost any mana but has a long cool down time. It can bounce back all attacking spells and can push away opponent mage and creatures. The wind blast also doubles the damage caused by the bounced back projectiles, making it a useful spell to save lives or even convert enemies' deadly attack against themselves.
==

The player can adjust his or her play style by choosing which spells to use in battle. For example, if the player focuses more on direct attack, he or she may want to throw more fireballs or ice shards. Or the player can summon more magic creatures and make an army of minions to crush the opponent. And essentially, players will have the chance to explore different combinations of spells and find out their best style. They will have a ton of fun in casting multiple magic spells, while strategizing their next moves in game.

How to run the program:

The game is designed for 2 players to play on the same computer.

Controlling keys:
Player 1 (red, on the left side)
- W (Mage Up), S (Mage Down), A (Mage Left), D (Mage Right)
- R (basic offensive spell), T (powerful offensive spell), Y (summon creatures)
- F (generate a wind blast (defensive spell))
- G (generate fire element orb), H (generate ice element orb)

Player 2 (blue, on the right side)
- Four arrow keys for Mage movement in 4 directions
- I (basic offensive spell), O (powerful offensive spell), P (summon creatures)
- K (generate a wind blast (defensive spell))
- L (generate fire element orb), ; <semicolon> (generate ice element orb)

When the game is end, press “0” to restart a magic duel.

Most interesting features about the programming:

- Randomness in magic effect and summoned creature behavior. It seems that some randomness can really make the game much more interesting. For instance, there is some randomness in the angle and speed of ice shards spell, makes their trajectory more diverse. The move pattern for summoned creatures also involve some randomness so their movements are more unpredictable.

- We spent quite some time designing the behavior of the magic orbs flying around a mage. Originally we created a force-directed layout, where the positions of the orbs are updated by applying attraction and repulsion forces. But later we changed to a different design: now they circle around the mage, each type of orb has a different radius and angular speed, giving a simple yet cool aesthetic effect.

- We used classes (Mage, Projectile, Creature, Effect) to store important information, every time a magic projectile, creature or effect is created, it will be added to a corresponding vector. For example, projectiles1 stores all the magic projects generated by player 1 and player 1's summoned creatures. Each class instance not just store the position, but also other attributes. For projectiles, this include the angle (direction) they are flying towards, their damage on contact, their speed etc. This decision gives more overhead but also offer a flexibility in changing their attributes in game with different magic effects. For instance, we can change the damage and speed of some of the fireballs on the field that are affected by some area of effect spell while keep others unchanged. We also used C++ enum extensively to differentiate different types of projectile, creatures or effects.

Challenges encountered:

A major challenge that we encountered is on structuring and organizing large amount of code. As we add more creatures, magic projectiles and effects into the game, our code becomes rather massive. So after the halfway deliverable we essentially rewrote half of the code, re-structured the code into many separate functions and classes. This decision helped us overcome the challenge and now making it very easy to add/change features in the game.

Another challenge is about playing sound, we decided that a magic duel game should have exciting sound effects in it, but when we first looked into audio usage, we looked in the wrong direction and tried some methods that are excessively complicated and didn't work. In the end, we realize that we can use SDL mixer for very effective audio playing, and that solved the problem.

Major design decisions that had to change:

We originally planned on having a second form of a mage: when a mage's health is critical, the mage transforms into a big magical monster, which leads to a boss fight. And we were thinking about making this part of the game content procedurally generated. However, in the end we found that this feature can have some problems with the balance and playability of the game. So in the end we decided to discard this feature and instead concentrate on making the current game very fun and cool to play.

Features we had in mind but not implemented:

- More magic. The original design was 4 elemental orbs types: fire, ice, wind and earth, but later we found that it is too complicated to control.

- Boss fight mode. A mode where the player can challenge an extremely powerful magic creature and its horde of minions, generated and unpredictable.

- Mage AI. It would be cool if we can a mage AI to support a single player mode.

- Skill updates: mage can gradually level up and gain extra skills/ enhance certain aspects of their magic.

Extra Notes:

The background dungeon image is found on the Internet. We also used some images we found on the Internet as raw material for creating our sprites.

Background music - Epic Blockbuster Trailer Music

Sound effects are mainly from freesound.org