

HearthClone

Philip Gooch

1703221

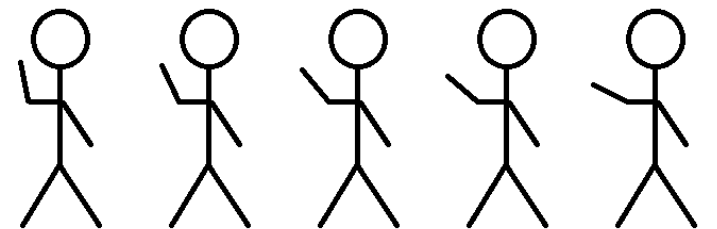
- The game is a clone of the well known game “HearthStone” by Blizzard.
- It is a turn based card game where you can play cards to summon minions that can fight for you.
- The objective is to kill the enemy hero by using your minions to strategically attack targets to improve your position and overwhelm the opponent.
- The game is designed so that it will fit any size monitor. You can re-size the window and all the shapes, sprites and collision detection will be updated.
- You can queue up attacks with your minions. You can attack with one and queue more attacks while the animation for the first is completing. I do this by creating a List of instructions that I then read from when animating. (This was very tricky.)
- Collision detection is pixel perfect apart from the ellipses for the minions where I used standard AABB detection for efficiency.
- I designed an algorithm to display the cards in an arc and to re-adjust their positions when cards are played or added. I am very pleased with how the cards “feel” with user interaction here.

Game controls

- The game is controlled by clicking and dragging minions and cards to the locations that you want.
- You can pause the game at any point by pressing the space bar.
- You can exit the game at any point by pressing the escape key.

Sprites / Animations

- I have very limited sprites as I am not artistic! I am much more interested in programming and problem solving. I have created a stick figure that has two animations to show my knowledge of how to create and animate them. Ideally there would be a separate sprite for each card and the game would be a lot more polished in general but I spent so much time on the coding.
- The shapes for the cards, heroes and minions are technically not sprites in that they are not images that I have imported. They are created using the ConvexShape class in SFML and the points are generated through code. I like this simplistic, geometric style.

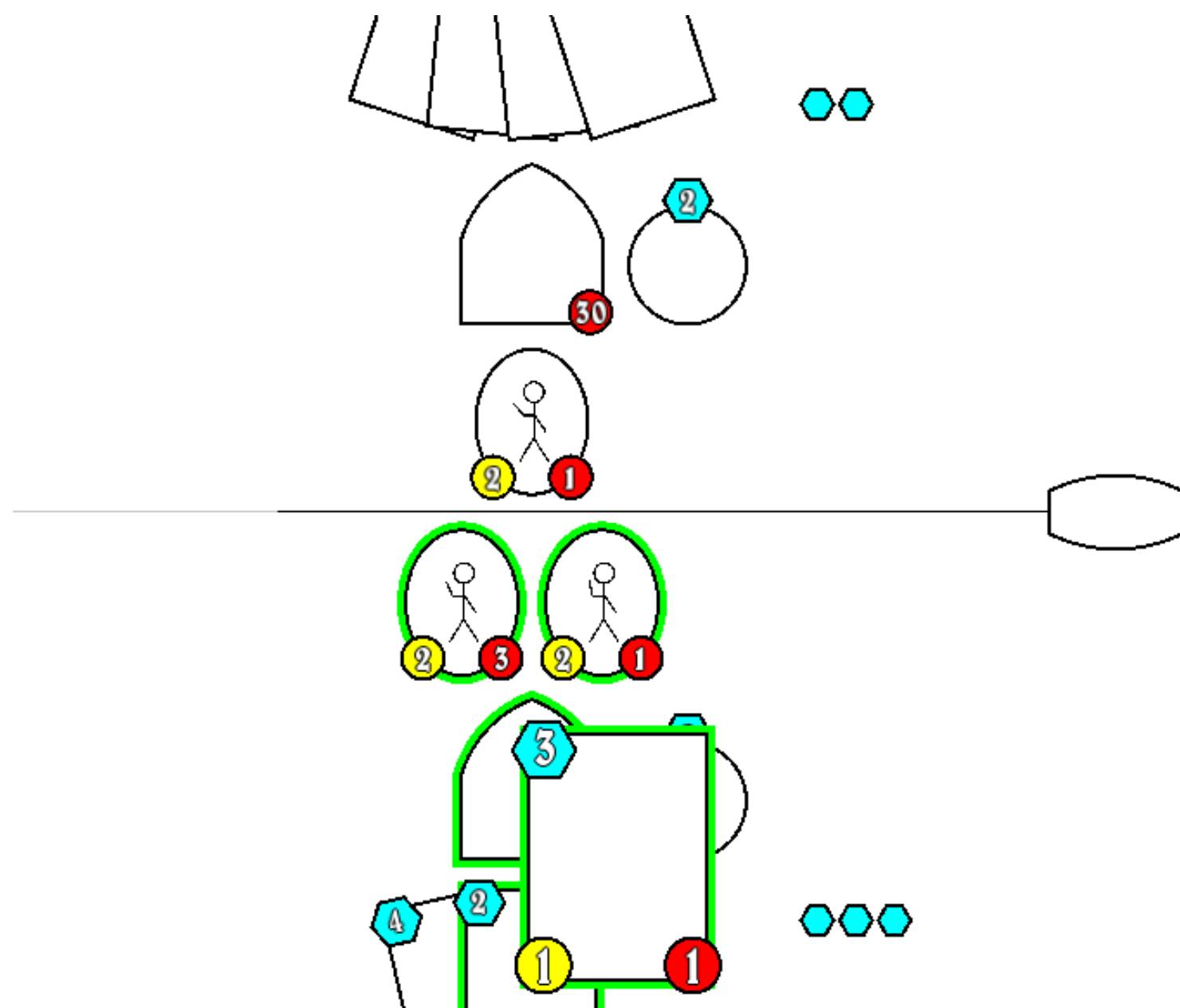


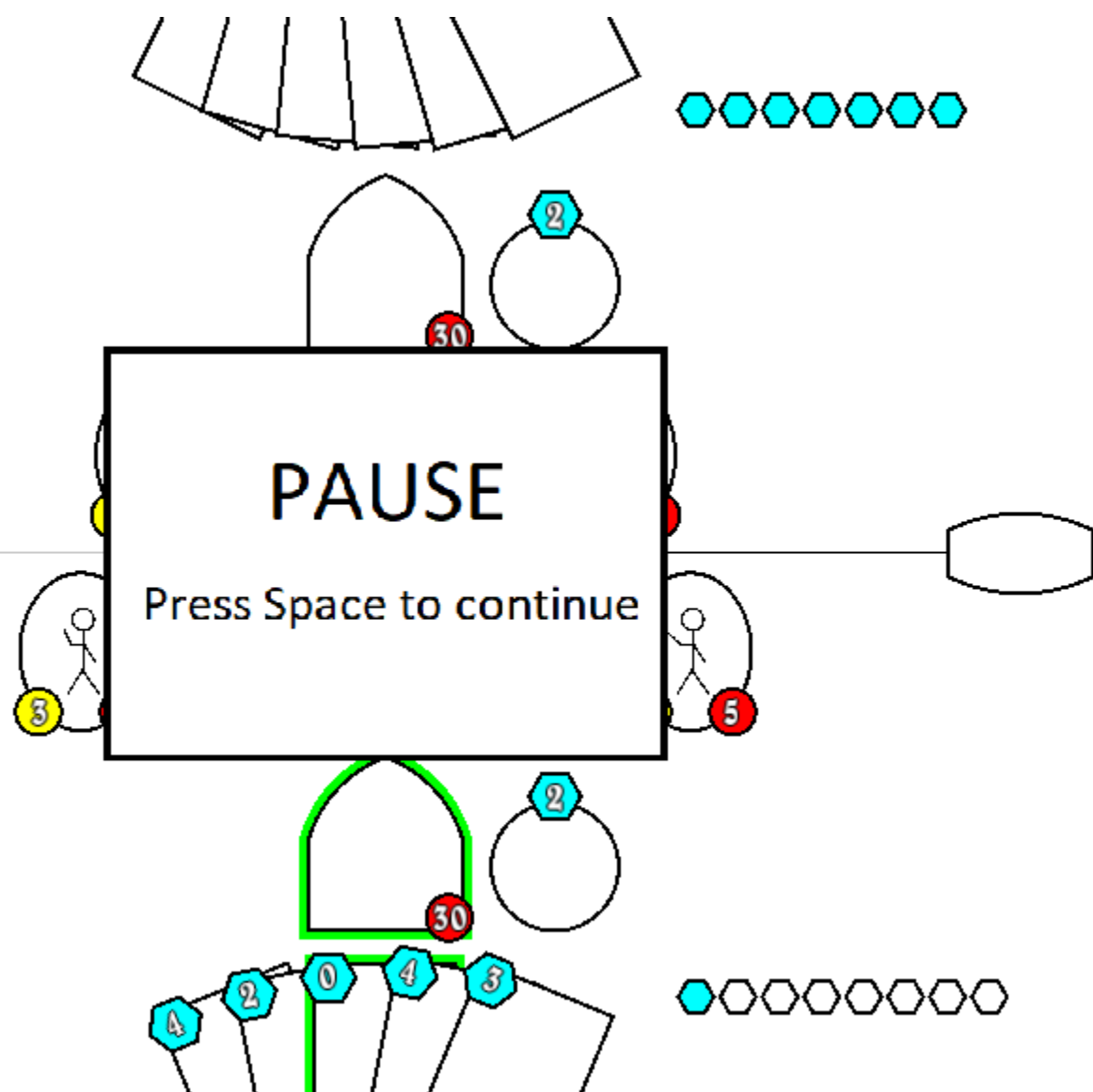
Screenshots

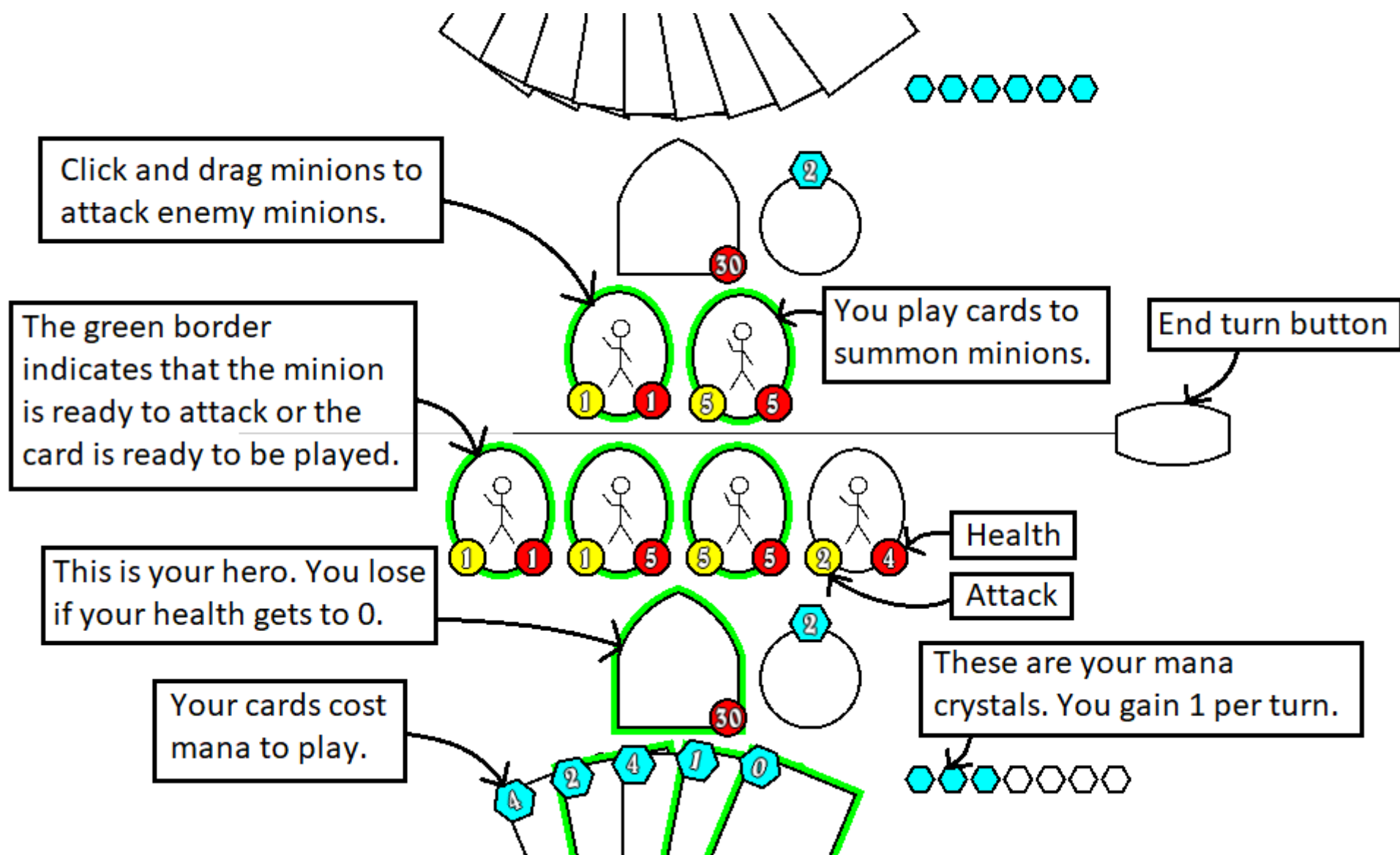
HEARTHCLONE

Play

Tutorial







Collision Detection

- For my cards I wanted them to be able to rotate and be stacked on top of each other. I found AABB collision to be inaccurate for this purpose. I used a technique where I first calculated the point of each corner of the card. I then found the equation of the line between the two points. I then checked if each of the four lines of the card intersected the line between the point $(0, 0)$ and the point of the mouse position. I could then tell if the mouse was inside the card by counting how many times the lines intersected. If it was an odd amount it meant it was inside and if it was even it was outside. After realising that in a rectangle it was only possible for two of the lines to cross at a time, I optimised the code so that if two were intersecting, it would break out of the loop.

- For my end turn button and my hero portrait I used a combination of circle and AABB collision.
- For the minions colliding with each other, because of the design of my game, I simply had to check if the sprite was higher (or lower) than the one it was attacking.

Game logic

- I have implemented a queue for the attacks. When you tell a minion to attack another one, the minions' health is adjusted. The IDs of these minions are stored in a list that is then read when animating the minions. When one animation is finished it is deleted from the front of the list and the next one starts.
- There is an end turn button that swaps whose turn it is. There is also a timer that counts down. When it reaches the end it changes turn.
- I designed an algorithm to calculate the positions of the cards that updates when a card is played.
- When hovering the cards over the battlefield the minions separate to fit in the new one.
- When the minions move, their velocity is affected by acceleration to create a smooth animation.

Audio

- I have the original HearthStone music running in the background.
- I have the sound of breaking glass each time there is a minion collision.

References

- This is entirely my work apart from the provided “AudioManager”, “input” and “Animation” classes. I am proud to say I made the stick figure myself!
- The only external asset I used was the music that I got from <https://playhearthstone.com/en-us/media/> and converted it to a .ogg file using VLC player.