# Spike Trap

The Spike trap is a common obstacle the player character will come across on their journey, this trap will be clearly visible to the player even when retracted as the design will be clearly different from normal floor tiles and should intuitively be something the player knows to avoid. The sharp nails of the trap will “spike” out and retract periodically or they could also permanently be out to provide a challenge to the player. Not only could this trap be on the floor it can also be on the walls and ceiling, making its application and potential diverse. Since the trap is either static or slowly periodically protruding/retracting, the consequence of the player touching the spikes would be instant death and reset to the checkpoint, regardless of the players health. This is because the “Traps effective area” is limited so the punishment should be more severe.

The design of this trap was inspired by lots of other games of a similar genre, the spike trap is a staple of the platforming genre and is required to increase the games difficulty. Without something to obstruct the player, the games difficulty would drastically reduce. As the setting of the game is inside a medieval castle, the appearance of iron spikes as traps, is a believable occurrence and helps to immerse the player in the world building of the game.

As a Player of the game “Plunderer”, I need to jump/get past the spike trap obstacles without touching them in order to achieve the current objective and progress the story of the game. Meaning that the successful implementation of the spike trap can be validated by having the player character touch the spikes. If the player character is killed and sent back to the checkpoint upon contact, then it is working as intended.

# Saw Trap

The Saw Trap is a rarer obstacle the player character will come across on their journey, this trap will be clearly visible and intuitive to the player as it will look different from the normal floor tiles/ game background. The Blade/Saw will run on its programmed tracks continuously back and forth at interchangeable speeds making it easier to adjust the difficulty of the game. Since this trap will be constantly moving around, likely at high speeds, the “traps effective area” is significantly larger than the spike trap, therefore the consequence of coming in contact with the trap for the player, will be less severe as it is easier to make mistakes as the player. The player will lose a portion of their health, and become invulnerable for a second, giving them time to move past the trap and decreasing the chances of being damaged twice by the same trap.

The Design of this trap was mainly inspired by the well-known platforming game “Hollow Knight”, with a trap that can have its speed and a large effective area adjusted to the designers whims, allows the creator of the levels more control over the difficulty of the level design. Since this trap is very versatile, and matches the design of the spike trap, it perfectly conforms to the medieval fantasy castle theme we are going with.

As a Player of the game “Plunderer”, I need to run/jump past the moving saw trap without touching them in order to achieve the current objective and progress the story of the game. Meaning that the successful implementation of the saw trap can be validated by having the player character touch the saw trap. If the trap is moving along its designed rout and reduces the players health upon contact, then the trap is working as intended.

# Initial Death Handling

Initial Death Handling is just the systems handling of the Playable characters death. In the event that the player characters health reaches zero, the player gets brought to the “Death Screen” where he will get score of their current playthrough and the choice to restart the game or from the last checkpoint the player has triggered. The Initial Death Handling of the player should be concise and intuitive, so the player will keep trying again after a fail. This is because the consequence of death means very little except a restart to the last checkpoint, which will encourage the player to keep trying again as the in-game characters death is a minor setback. Since this mechanic is strongly tied to the Health Bar Mechanic, it must be designed and implemented first before work on the Death Handling system can start.

The design of the Death Handling System needed to be carefully considered as it will servery effect the difficulty of the game, and the incorrect system could leave a bad impression on users and cause them to stop playing the game. In almost every game with a user played character, there are two main ways the game system deals with player death. The first method is the “hardcore/ironman death” meaning, the player will lose all progress completely and restart from the start. The second is the “checkpoint death” method, where the player is sent back to the last checkpoint, they have triggered losing all progress they had from the checkpoint to where they died. As “Plunderer” is a completely platforming game with considerable difficulty, the use of the “hardcore/ironman death” method, could cause the player to find the consequence of death too punishing, thus putting them off the game. Since that is the case we decided to go with the “checkpoint death” system as this allows us to make the level design and obstacles much more punishing without the user finding the game too difficult.

As a Player of the game “Plunderer”, I need to avoid getting my health reduced to Zero by the obstacles in the level in order to achieve my current objective. This means that the successful implementation of the Initial Death Handling, can be validated by the player being reset and brought to the death screen when their health bar reaches zero.