**DS**

**Group Course (Group XX): Mid-Term Report**

|  |
| --- |
| Introduction and Objectives |
| Our group project comes from the second topic “Tomb Raider”. A game bases on the story in that a girl driving a scooter fights with monsters in the maze and tries to save her missing boyfriend. The Unity5 engine is used to create this game and our game is in a 3D mode. The characters and scenes are built by voxel modeling.  Our objectives are to build an interesting game which can be most interesting group project incidentally with upgrading our skills. We will stick to the basic requirement to finish the first step of our game, then to create more funny and delicate components to make it more wonderful. |
| **Requirements and Functionality** |
| 1. Characters and Scenes Building  The single-minded girl, 3 kinds of monsters, a skilful boss and mazes will all built in voxel modelling.  2. Moving with Camera Following  The moving and camera following will all follow TPS game type. This will make the game more modern.  3. Background Music  We have found some suitable music to cooperate with the game.  4. Life Value  Player will have a life value to show how much remaining life with the range from 100 to 0. The value will decrease when hit by monster and increase when meet props.  5. The Frequency the Monsters Show in  When the player moves to the special place, the monster will show out only one time.  6. Three Different Kinds of Monster  The first type is the guard, they will keep chasing the player and want to touch player. The second one is Necromancer, they will use magic to make hurt. The last kind is Ninja from remote Japan, they will randomly choose their direction to run straight forwards to hit player and then suicide.  7. Monster Life Value  Monster also have life value, but commonly will not show out.  8. Gun shooting  Player use gun to hit monsters, but the number of bullets is limited in a accurate number. And we will make the special effects of shooting and blood when monster dead. When monsters are hit by the bullet, its life value will decrease.  9. Resurrection  When the life value of player become zero, player will dead and resurrection in the start point in the current map.  10. Shadow  We use the part of the engine to make the shadow to make the game more real.  11. Props  The bullet bags and blood bags will appear in same special places. When player meet this props, the number of bullets or life value will increase.  12. GUI  Building with 3D GUI.  13. Boss  There is a final boss that exists in our game with detailed information about its blood value and the decrease of its blood value during attack. Its first skill is fire hurt, it can burn through the wall to attack the player behind any wall successfully. This skill enforces the player not to hide behind the wall for any longer time. The second skill is chalk hurt, it can hurt the player out of a long distance but can not fly through the wall. It is powerful because of the huge number of the chalks. |
| **Logic Flow (Preliminary version) pseudo-code** |
| **Moveing:**  Int v,w;  Float movespeed;  Get the value of back or forward from keyboard(-1or1) to v;  Get the value of left or right from keyboard(-1or1) to w;  Use vector3 to the direction and speed which have be input.  **Monster show out**  Transform [] points;  Gameobject monster;  Bool collisionover=false;  Catch the transform of the point to points;  When(something make collision and the tag is player and the collisionover==false)  {  For(Int monsternumber and will wait a short time )  Random create the gameobject in the points;  collisionover= true;  }  **Props creating**  Transform [] points;  Gameobject props;  Bool collisionover=false;  Catch the transform of the point to points;  When(something make collision and the tag is player and the collisionover==false)  {  Random create the gameobject in the points;  }  TriggerEnermy.cs  //the receive for the hurt from the player  public abstract class Trigger : MonoBehaviour {  protected abstract void Start ();//initialization  protected abstract void OnTriggerEnter(Collider collider);//the judgment of the hurt  protected abstract void Des();//the logic flow after being attacked by the player  }  public abstract class EnermyMove : MonoBehaviour {  protected GameObject playerObj;  protected Transform modelTrns;  protected abstract void move ();//the logic of the movement of the enermies  protected abstract void attackPlayer(AttackSkill attackSkill);//the logic of the hurt behavior of the enermies  } |
| **Relation between Data Structure** |
| 1、Unity have the special date type: texture, material, gameobject, they are store in folds, can be used in every scene.  2、To find Game Object, we will use the tag or the name of the game object to do the HashMap to find the instance playing in the project.  3、We use the queue to manage the display and the instance of the monsters.  4、During the procedure of the guards to chase the player, there is Dijkstra algorithm used to realize this method. |
| **Plan and Schedule** |
| We divide the whole project into Game Manager part, Enermy part, Bullet and props part, Player part, scene part and video part. With the accurate part to continue our project. |