This section carries a total of 85 marks. However, the individual aspect of the game is evaluated on the report of it (see Section 3 Report).

This game should be built on the core game systems developed during tutorial sessions and must include implementations of the following:

1. A virtual camera that follows the player-controlled (hero) character at its centre (**10 Marks**)

可以通过写一个摄像机类实现

可以通过移动地图实现，固定角色在屏幕正中间，

如果是固定大小的地图，最外圈生成半径为屏幕大小一半的不可移动地形

1. A number of NPCs that attack the character (**17 Marks**)
2. **角色（基类）**

**公共的角色属性和方法。**

* 1. **变量**
     1. **角色名称**
     2. **生命值**
     3. **角色坐标**
     4. **速度**
  2. **方法**
     1. **Load角色外观**
     2. **Draw角色**
     3. **减少生命值**
     4. **碰撞检测（虚函数）**
     5. **移动（虚函数）**

1. **玩家角色 (PlayerCharacter)**

**包含两种攻击模式和一个特殊技能。**

**需要处理玩家与特殊地形的碰撞。**

* 1. **变量**
     1. **攻击模式**
     2. **特殊技能**
  2. **方法**
     1. **发射子弹**
     2. **范围攻击**
     3. **激活特殊技能**
     4. **移动**
     5. **特殊地形碰撞**

1. **非玩家角色 (NPC)：**

**在屏幕外随机生成，具有不同的外观、血量和速度。**

**三种 NPC 会向玩家笔直移动，进行近战攻击；一种 NPC 不移动但会发射子弹。**

**通过不同的子类分别实现每种 NPC 的独特行为。**

* 1. **抽象NPC类**
     1. **抽象移动NPC类**
        1. **实现可以移动的NPC**
           1. **三种不同的NPC进展**

**外观**

**血量**

**速度**

* + 1. **射击不移动NPC**
    - Generated randomly outside of camera view (2 Marks)
    - Their frequency increases over time (2 Marks)
    - At least 4 different character types that differ in appearance, health and speed (5 Marks)
    - General NPC behaviour that directs them directly towards the player (4 Marks)
    - One NPC behaviour makes it static but launches projectiles (4 Marks)

1. Collision system (**12 Marks**)
   * + Hero vs NPCs (3 Marks)
     + Hero vs impassable terrain (3 Marks)
     + Hero projectiles vs NPCs (3 Marks)
     + NPC projectiles vs Hero (3 Marks)
   * Clearly demonstrate how NPCs are implemented and handled
2. The hero attacks the NPCs with at least two different types of attack (**12 Marks**)
   * A linear attack that targets closest NPC (has cooldown) – the attack runs automatically all the time (3 Marks)
   * A special area of effect (AOE) attack that targets the top N max health NPC – triggerred by hero via separate button (can be instantanous but has significant cooldown). (7 Marks)
   * A powerup that increases either the speed of the linear attack and number N of NPCs targetted by the AOE (2 Marks)
3. A tile-based method (composed of a number of 32x32 pixel tiles) for displaying the background (**16 Marks**)
   * At least four different tile types (ie terrain types). One of which is impassable (eg water) for the hero. NPCs can traverse. (5 Marks)
   * Data driven level loading, i.e. load the game world’s tiles and map from a custom file format (5 Marks)
   * A version of the world which is infinite (with repeating tiles) (3 Marks)
   * A version with a fixed boundary (larger than what the screen displays) (3 Marks)
4. Game level runs for two minutes (**7 Marks**)
   * Show at least two levels with different maps (one infinite, one fixed) (5 Marks)
   * Score and FPS shown at the end (or during) (2 Marks)
5. The ability to save the current state of the game (and save to a file) at any point and reload it at the same exact point (**11 Marks**)

Please note that we will not be marking the game, rather we will mark the implementations of the above methods. Please use the report to explain your different solutions. You can show screenshots abstracted from the gameplay e.g. an image showing your collision detection system.

# Report (15 marks)

The report should be structured as follows:

* Introduction which introduces your game, and the technologies used
* A section on each of the technologies implemented (1-6 from Section 2) which contains details about how the technique works and how you implemented it. Half of this section should cover theory; half should cover the relevant implementation details for each method. You should include details here even if your implementation is partially complete.
* A short evaluation section – measure the FPS / ms per frame and see how this varies as the level complexity increases
* Limitations which cover the things you tried that did not work or if there are bugs that you know about but could not fix
* Conclusion which summarises the report in a single paragraph

Also add to the report a small section on how you would have approached this project differently if you had to start from scratch after this learning experience.

You are expected to include screenshots of the game running and implementations of game systems. For this you can use an application such as the “snipping tool” in Windows to directly paste the results in your document – this is very quick. Windows 10 has a new screen capture facility using Shift+MSkey+S. Similarly, most of the latest versions of Mac OSX permit the use of screen capture via shift-cmd-4 (various numbers provide different functionality).

Do not add the code to the appendix but provide it separately and in a format such that it can be compiled directly (see Section 1).

**Important:** All samples of code shown in the text need to be in text format, **not** a screen capture from your editor. Code in screen captures **will be ignored**.

加在window类中 用来检测鼠标的按下

// Check if a specific mouse button is pressed

bool Mousepressed(int mousebuttons) {

return mouseButtons[mousebuttons];

}