**Class Description**

1. **Singleton**
   1. **Application :** Quản lí chương trình, Update, Lấy thông tin từ trong cửa sổ Window
   2. **ResourceManager:** Quản lí tài nguyên của game: Texture, FrameData, DataMap, Font.Nhưng tài nguyên nào đã load rồi thì sẽ không phải load lại. Do có singleton nên tài nguyên đó có thể dùng được ở mọi class.
   3. **GameStateManchine:** Quản lí sự thay đổi của các GameState, cơ chế là dùng stack, push dần các State, rồi pop đi nếu muốn quay lại state cũ.
   4. **ItemManager:** Quản lí tạo Item, để cho dễ dàng, dùng ở mọi class, nên đã dùng singleton. Câu lệnh sinh item được đặt ngay trong Các class creep nên rất dễ dàng tạo item.
   5. **DataGamePlay:** Chứa thông tin trong game, số lượng quái đã tạo, số lượng quái đã chết.
   6. **BackGround:** Do Cả game chỉ dùng một hình nền, lúc thay đổi mỗi state sẽ vẫn tiếp tục các trạng thái của các vật. Trong class bao gồm các animation để trang trí.
2. **GameStateBase:** State cơ bản, cái mà mọi State nào cùng cần có. Chủ yếu dùng trong GameStateManchine để push và pop những state con của nó.
   1. **GSIntro:**  chứa giới thiệu clb, logo cá nhân.
   2. **GSMenu:** Chứa các button Play(Khi click vào sẽ push state gameplay vào stackState) và thoát application.
   3. **GSPlay:** Chứa Player, LevelManager
   4. **GSEnd:** Chứa 2 Button menu(Khi click vào sẽ pop 2 state trong listState là GSEnd và GSPlay để quay trở về GSMenu), và replay(Khi click vào sẽ pop 2 state trong listState là GSEnd và GSPlay để quay trở về GSMenu, rồi push Thêm GSPlay vào).
3. **Components**
   1. **AnimationComponent**
      1. **FrameData:** Hỗ trợ việc lấy thông tin Frame, ảnh
   2. **AttributeComponent:** Chứa thuộc của đối tượng như player, bomb, boss,… xem đối tượng đã chết chưa, còn bao nhiêu máu, sát thương, tốc độ,…
   3. **HitBoxComponent:** Tạo hình vuông để setVa chạm, đây là body của vật thể
   4. **HitCircleComponent:** Như HitBox nhưng mà là hình tròn
4. **Button:** Nút game, Cơ chế là dùng con trỏ hàm void để cài đặt các chức năng của button, nên mỗi button mình có thể cài đặt chức năng một cách dễ dàng
5. **Creep** 
   1. **BaseCreep:** quái cơ bản, chứa các thông tin của một con quái. Animation, AI, cơ bản là có 2 trạng thái. Là run và hit
      1. **CreepBat:** Con dơi
      2. **CreepBee:**  Con ong
      3. **CreepBlueBird:** Con chim
      4. **CreepSkull:** Đầu lửa
      5. **CreepSunDragon:** Con rồng

(Mỗi con quái đều có thuộc tính khác nhau, tốc độ di chuyển, Cách chi chuyển máu, sát thương)

* 1. **CreepManager:** là class cơ bản cho việc sinh các loại quái
     1. **CreepBatManager**
     2. **CreepBeeManager**
     3. **CreepBlueBirdManager**
     4. **CreepSkullManager**
     5. **CreepSunDragonManager**

**(**Việc sinh quái cho mỗi loại là cần thiết để dễ dàng cân bằng game, đã dùng thuật toán random để game cân bằng hơn**)**

* 1. **MonsterManager:** Một class thử việc sinh quái(Lúc quay video đầu tiên đã dùng class này để sinh quái, còn lúc hoàn thành thì không)

1. **Item:** class item cơ bản, chứa thông tin cũng như chức năng;
   1. **AttributeItem:** Thuộc tính của item
   2. **ItemAllowThrowBomb:** Cho phép ném bomb, chứ không chỉ là đặt bomb
   3. **ItemCoolDownThrowBomb:** Giảm thời gian nữa 2 lần ném bomb
   4. **ItemCoolDownThrowBullet:** Giảm thời gian nữa 2 lần bắn đạn
   5. **ItemCoolDownTimeBomb:** Giảm thời gian nổ bomb
   6. **ItemDamgeBomb:** Tăng Damge của bomb
   7. **ItemDamgeBullet:** Tăng Damge của đạn
   8. **ItemHeart:** tăng máu
   9. **ItemJump:** Tăng lực nhảy
   10. **ItemPowerThrowBomb:** Tăng lực ném bomb
   11. **ItemPowerThrowBullet:** Tăng Lực bắn đạn
   12. **ItemSpeed:** Tăng tốc độ chạy
2. **MovingDirect:** Lấy thông tin của 2 điểm trên trục tọa độ, có góc, và vector đơn vị
   1. **MouseDirect:** Lây thông in hướng của con trỏ chuột và một điểm.
3. **GUI**
   1. **HeartBar:** Thanh máu
4. **ExposionEffect :** Tính lực nổ, lực bay ra, của các vật khi bomb nổ, hoặc đạn bắt chúng
5. **Bomb:** Sử dụng State machine để chuyển các trạng thái
   1. **IBomb:** Chức năng của class bomb, hỗ trợ lấy thông tin từ class bomb
   2. **IBombState:** chức năng của các State, để dễ dàng thay đổi state
   3. **BombOffState:**  Trạng thái 1
   4. **BombOnState:** Trạng thái 2
   5. **ExplosionState:** Trạng thái 3

**(**mỗi trạng thái sẽ có 1 biến animation khác nhau**)**

* 1. **BombManager:** Quản listbomb, cài đặt lực tác dụng vào quả bomb( hiệu ứng ném)

1. **Rocket:** Tương tự như BombManager. Nhưng đạn sẽ không chịu tác dụng của trọng lực, còn bomb thì có.
   1. **IBulletState**
   2. **IBullet**
   3. **Bullet**
   4. **FireState**
   5. **ExplositonState**
2. **Player:** Con nhân vật chính, cơ chế chuyển trạng thái vẫn là dùng State Machine. Quản lí mọi thứ của player. Bắn đạn, chuyển Weapon,…
   1. **IPlayer**
   2. **IPState**
   3. **RunState**
   4. **JumpState**
   5. **JumAniticipation**
   6. **IdleState**
   7. **Hit**
   8. **GroundState**
   9. **FallState**
   10. **DeadHit**
   11. **DeadGround**
3. **Boss:** Kế thừa base Creep, nhưng có AI khác. Và có thêm 1 trạng thái là bắn ra đạn quái vật.
   1. **CreepBullet:** Đạn của boss là một con quái
   2. **BulletBossManager:** Quản lí đạn
4. **World:** Tạo thế giới box2D, nơi mọi vật thể trong đó. Hỗ trợ tính các hoạt động vật lí
5. **EffectWindow**
   1. **LayerShadow:** Hiệu mờ dần, sáng dần(Bản dễ dùng hơn của ShadowWindow)
   2. **ShadowWindow**
6. **TileMap:** Vẽ map, class này khá mạnh, hộ trợ vẽ map từ file json đc lấy từ phần mềm tiledMap, dùng thư viện khác để đọc file json. Vẽ xong sẽ tạo mặt đất lấy từ thông tin các khối. Có thể vẽ được tile Map đa lớp nhưng trong con game này chưa có cơ hội thử,
7. **Entity:** Chủ yếu chưa body của vật thể
8. **Decoration:** Các animation dùng để trang trí cái background trông ngầu hơn.
   1. **BackTree**
   2. **Cloud**
   3. **ParallexCloud:** Hiệu ứng mây chạy vô tận.
   4. **WaterReflect**
9. **Config**
   1. **GameConfig:** Chứa một số thông tin game dùng cho cả project
   2. **ItemConfig:** chứa thông tin của các item, hỗ trợ việc thay đổi các giá trị 1 cách dễ dàng
   3. **CollisionManager:** Thông tin logic game, vật nào được phép chạm vào vật nào. Vật nào không được phép.

**Hàm Core Game:**

* **MovingDirect:** Dùng toán học cơ bản để tính được vector đơn vị và góc của đường thẳng A, B với trục Oy. AI của mọi con quái đề phải dựa vào class này để tìm được hướng di chuyển.
* **ExposionEffect:** Tính lực tác dụng lên mọi vật khi nổ. Tính khoảng cách, tính hướng tác dụng. Do box2D có hỗ trợ chỉ lấy nhưng vật trong 1 hình vuông. Nên mình không phải duyệt hết tất cả các vật thể trong thế giới. Khi tính toán xong. Ép lực vào tâm body của vật.
* **Các Class quản lí:** Giúp game trở nên tối ưu hơn rất nhiều