

```
image trapCardimage;
  - Image campfireCardImage;
- Image campfireBuildingImage;
   - Image stakelmage;
   - Image staffImage;
   - Image bodyArmourImage;
   - Image helmetImage;
   - Image shieldImage;
   - Image goldImage;
  - Image oneRingImage;
- Image healthPotionImage;
   - Image allylmage;
   - Image doggielmage;
  - Image elanImage;
- Image sniperRifleImage;
    - Image heavyBulletImage;
    - Image andurillmage;
   - Image treeStumpImage;
    - MediaPlayer backgroundMediaPlayer;
   - MediaPlayer killingMediaPlayer;

    MediaPlayer shoppingMediaPlayer;
    MediaPlayer buttonClickMediaPlayer;

   - MediaPlayer getItemFromPathMediaPlayer;
   - MediaPlayer explosionSoundMediaPlayer;
    - MediaPlayer shottingSoundMediaPlayer;
   - DEFAULT_PLAYSPEED: double
   - currentlyDraggedImage: ImageView

    currentlyDraggedImage: Imageview
    currentlyDraggedType: DRAGGABLE_TYPE
    gridPaneSetOnDragDropped: EnumMap<DRAGGABLE_TYPE, EventHandler<DragEvent>>
    anchorPaneRootSetOnDragOver: EnumMap<DRAGGABLE_TYPE, EventHandler<DragEvent>>

    - anchorPaneRootSetOnDragOver: EnumMap<DRAGGABLE_TYPE, EventHandler<DragEvent>>
    - anchorPaneRootSetOnDragDropped: EnumMap<DRAGGABLE_TYPE, EventHandler<DragEvent>>
    - gridPaneNodeSetOnDragExited: EnumMap<DRAGGABLE_TYPE, EventHandler<DragEvent>>
    - gridPaneNodeSetOnDragExited: EnumMap<DRAGGABLE_TYPE, EventHandler<DragEvent>>
    - mainMenuSwitcher: MenuSwitcher
    - settingsSwitcher: MenuSwitcher

  - gridPaneSetOnDragDropped: EnumMap<DRAGGABLE_TYPE, EventHandler<DragEvent>>
- anchorPaneRootSetOnDragOver: EnumMap<DRAGGABLE_TYPE, EventHandler<DragEvent>>
  - anchorPaneRootSetOnDragDropped: EnumMap<DRAGGABLE_TYPE, EventHandler<DragEvent>>
- gridPaneNodeSetOnDragEntered: EnumMap<DRAGGABLE_TYPE, EventHandler<DragEvent>>
- gridPaneNodeSetOnDragExited: EnumMap<DRAGGABLE_TYPE, EventHandler<DragEvent>>
    - mainMenuSwitcher: MenuSwitcher
    - settingsSwitcher: MenuSwitcher
 + LoopManiaWorldController(LoopManiaWorld, List<ImageView>)
  + initialize(): void
  + startTimer(): void
 + pause(): void
  + terminate(): void
+ changePlaySpeed(ActionEvent event): void
+ addEntity(Entity entity, ImageView view): void
 + loadReward(String type): void
+ reactToEnemyDefeat(BasicEnemy enemy): void
 + onLoad(Ally ally): void
 + onLoad(Card card): void
 + onLoad(Item item, GridPane squares): void
 + onLoadUnequipped(Item item): void
 + onLoadEquipped(Item item): void
 + onLoad(BasicEnemy enemy): void
+ animateZombie(ImageView view): void
+ animateSlug(ImageView view): void
+ onLoad(Building building): void
+ buildNonEntityDragHandlers(DRAGGABLE_TYPE
 draggableType, GridPane sourceGridPane, GridPane targetGridPane): void
+ convertCardToBuildingByCoordinates(int cardNodeX, int cardNodeY, int buildingNodeX, int buildingNodeX, int buildingNodeY): Building + convertUnequippedToEquippedItemByCoordinates(int nodeX, int nodeY, int x, int y):Pair<Item,Item> + convertEquippedToUnequippedItemByCoordinates(int nodeX, int nodeY, int x, int y):Pair<Item,Item> + convertEquippedToUnequippedItemByCoordinates(int nodeX, int nodeY, int x, int y):Pair<Item,Item>
 + addDragEventHandlers(ImageView view, DRAGGABLE_TYPE draggableType,
 GridPane sourceGridPane, GridPane targetGridPane): void + removeDraggableDragEventHandlers(DRAGGABLE_TYPE draggableType,
 GridPane targetGridPane): void
+ handleKeyPress(KeyEvent event): void
+ switchToMainMenu(): void
+ switchToSettingsMenu(): void
 + loadGoals(JSONObject goals): void
  + setMode(String changeMode): void
  + switchToGame(): void
 + inHerosCastle(): boolean
  + sellDoggieCoin(): void
  + clickUnequipped(MouseEvent event): void
 + clickEquipped(MouseEvent event): void
  + buyltem(ActionEvent e): void
  + convertTrancedEnemy(BasicEnemy enemy, ImageView image): void
 + trackPosition(Entity entity, Node node): void
  + printThreadingNotes(String currentMethodLabel): void
  + updateLabels(): void
 + loadAllImages(): void
  + clickSnipe(MouseEvent event): void
 + playMusic(ActionEvent event): void
 + playKillMusic(): void
 + playShoppingMusic(): void
 + playButtonClickMediaPlayerMusic(): void
 + playGetItemFromPathMediaPlayerMusic(): void
  + playExplosionSoundMediaPlayerMusic(): void
 + playShottingSoundMediaPlayer(): void
 + getBackgroundMediaPlayer(): MediaPlayer
 + getKillingMediaPlayer(): MediaPlayer
  + getShoppingMediaPlayer(): MediaPlayer
  + getButtonClickMediaPlayer(): MediaPlayer
  + getItemFromPathMediaPlayer(): MediaPlayer
 + getExplosionSoundMediaPlayer(): MediaPlayer
  + getShottingSoundMediaPlayer(): MediaPlayer
  + displayGoals(ActionEvent event): void
```