

Final Game Project

The Mad King of Ur

Introduction	2
Storylines	2
Storyboard	4
Character Script	6
Diagrams	13
Class Diagram	13
Sequence Diagram	16
Target Market Research	18
Code and A.I. Processes	20
Ink Plugin	20
A.I. Processes	20
Developer Documentation	22
AIPlayer	22
DiceRoller	22
GameStatics	22
PlayerStone	22
SceneController	23
StateManager	23
Tile	24
TitleOptions	24
Tutorial	24

Introduction

As a digital version of The Royal Game of Ur, or Twenty Squares, The Mad King of Ur is a story-driven board game that takes the player back in time to the Babylonian era of Mesopotamia. While the actual game play is basically the board game Twenty Squares, the story is inspired by the history of Ur – speculated by historians to be the largest city of Mesopotamia during Babylonian era, with the earliest discovered game board hailing from 2,600 BC.

The part of history that inspired the story of The Mad King of Ur was from the city of Ur itself, where it was once the largest, busiest city of Mesopotamia. That is until the citizens revolted against the reign of a certain king. The king responded to the rebellion by cutting off the city's water source, redirecting an entire river away from the civilisation. The region became an uninhabitable desert until many hundred years later.

This piece of history became the essence of the story in The Mad King of Ur. You will be playing as a member of the rebellion, trying to best your tyrant in a game of wits on Twenty Squares, so that you may prevent something as catastrophic as the actual history of river diversion from happening, and prevent the destruction of an entire civilisation.

Storylines

You will meet 5 major characters in the game representing the 5 levels: Zimu, homeless man, palace goon, the magic door and finally the king, in that order.

Level 1: Zimu will reveal to you why you are an important asset to the rebellion – your heirloom Dice of Gods render you invincible, which implies that you have an absolute power on seeing the king if you demand so, and no one can stop you. You become the only hope to convince the king to stop the diversion of the river. However you can become vulnerable if you lose the Royal Game of Ur against someone.

In this tutorial level, Zimu trains you to play the Royal Game of Ur, to get you prepared and ready for the journey ahead. There is no game over in this level - Zimu will let you keep trying until you beat her. After you beat her, she will direct you to the closest horse cart

driver who can take you to Ur palace, for the journey from rebellion camp is too long on foot, and time is of the essence.

The rebellion camp is not a very ideal living space, and no one has game board nor paper, so Zimu will train you with the game board drawn on mud and random rocks as game pieces.

Level 2: After you leave the rebellion camp and reach the village's horse cart stop, you realised that you have no money to pay the driver for the journey. The driver recognised your Dice of Gods and makes a deal with you that, if you beat him in Royal Game of Ur, he'll give you a free ride to the palace. The game takes place on a piece of paper with the game board drawn on it, and again random pieces of rocks. If you lose to the driver, you'll be stuck in the village, resulting in a game over. If you win, you get to move on to the palace of Ur.

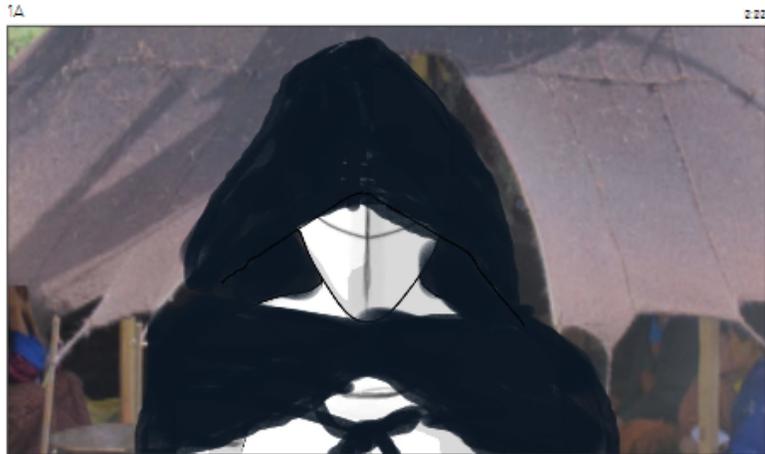
Level 3: You arrive at the palace entrance, summoning up bravery from your invincibility to walk straight up the front entrance. The palace guards threaten to attack you if you continue to approach, and when you do, the guards drive their pikes straight at you, but they break. They then came to a horrifying realisation that you possess the Dice of Gods. One of the guards challenges you to Royal Game of Ur, before he let you in. The game takes place on an actual wooden board with normal game pieces. If you lose, the guards will imprison you. If you win, the guard takes you inside the palace.

Level 4: The guard takes you to a grand stone door with many intricate engravings, and wishes you good luck, because this is the magic door that only allow those who beat it in Royal Game of Ur to go inside. If you lose, the magic door will kill you with, well, magic. If you win, the door opens and leads you directly to the king's chamber. This level takes place on a stone board that is actually part of the door. The door has near perfect AI that analyse the game moves very well.

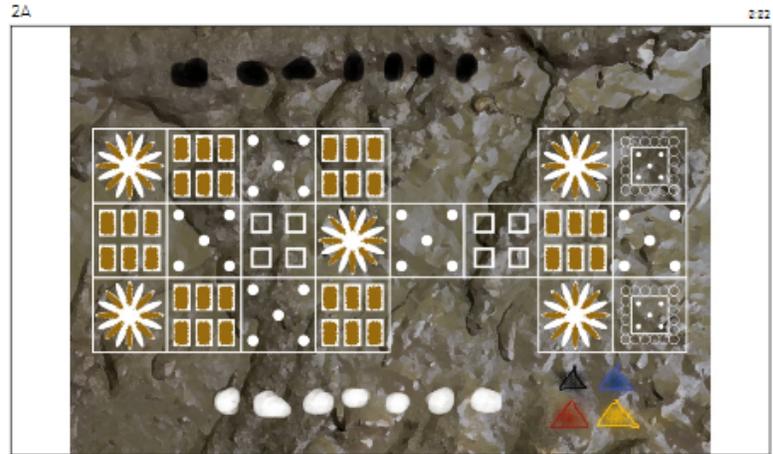
Level 5: Once you enter the king's chamber and meet the king of Ur, the king reveals that he too possesses his version of Dice of Gods, hence no one has been able to assassinate him or physically stop his tyranny. He will, however, have a final match with you to decide if he will stop the river diversion project, or have you executed. The game takes place on a golden game board, and the king will use his own dice. However, in addition to having very good AI, the king cannot roll zero, and he may occasionally cheat in the game, which you

can do nothing about. After beating the king, he dismisses you with a vague acknowledgement that he will stop the river diversion project. When you leave the palace, he utters under his breath: "for now..."

Storyboard



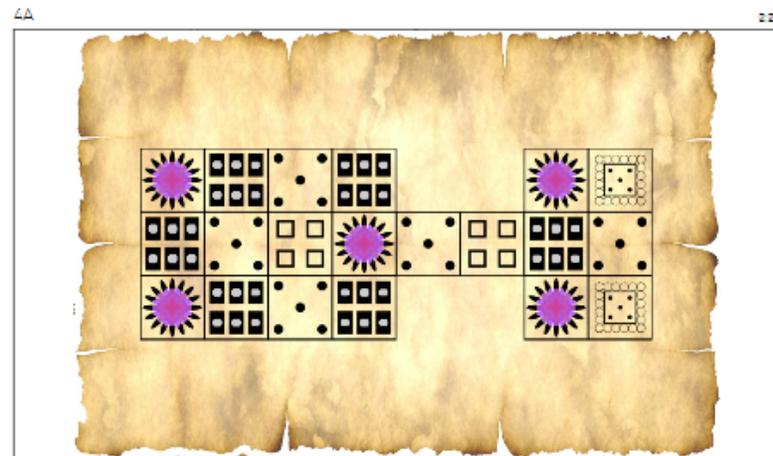
1A Zimu briefs you about you about the power of your dice and what the rebellion wants from you. Zimu talks in this scene before transiting to the game board.



2A The level against Zimu is designed to be a tutorial level where you get to replay if you lose. Here you are playing against Zimu with the game board drawn onto the ground, with rocks as game pieces and your Dice of Gods. The player is always at the bottom of the game board.



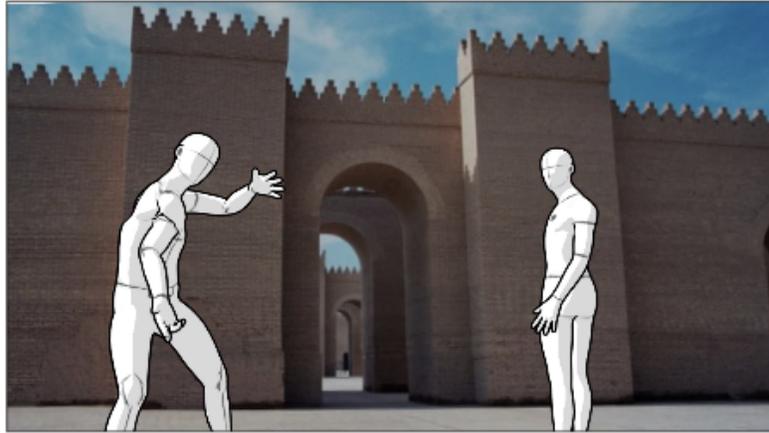
3A After you beat Zimu, she sends you to the town centre for a ride to the palace of Ur. The rider challenges you to a game match because you don't have any money on you, but he can't refuse your request due to your power with the Dice of Gods. The only way he could refuse is if he win against you, or he'll give you a ride if you win.



4A The game against the rider takes place on a parchment game board, which the rider kept with him for sentimental value. He used to play a lot with neighbours so he knew the game well, but is rusty now. If you lose, you'll be refused the ride and be stuck in town, resulting in a game over. If you win, he'll let you on the cart and you'll be on your way to the palace.

5A

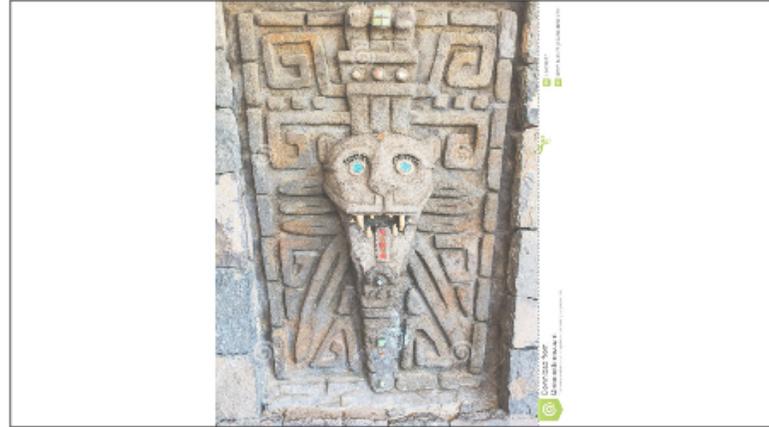
2:28



You reach the palace gate and is confronted by two guards. When you refuse to leave, they attack you with their weapons but the weapons broke on impact. Realising that you came with the Dice of Gods, one of the goons invited you to their guard post and challenged you to a game.

7A

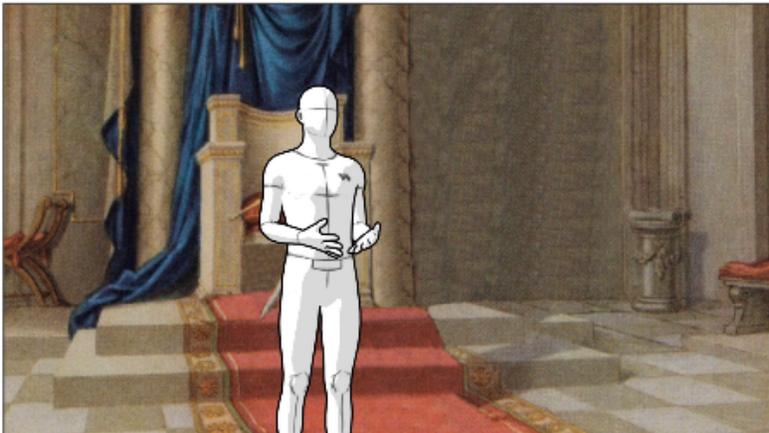
2:12



The way to the king's chamber is barred by a magical stone door with a game of Ur board on it. The goon explains to you that the door will only open to either the king or someone with Dice of Gods, and you would still have to play against the door to pass. The door has near perfect AI integrated to playing against you. He left you to your device and wished you good luck.nN

9A

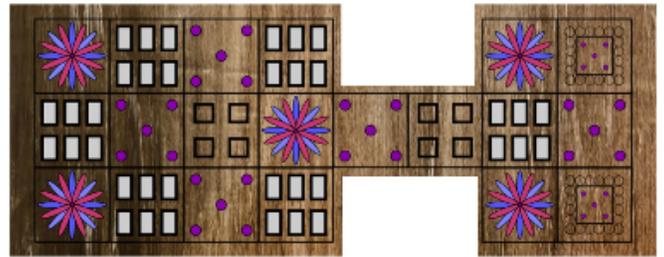
2:18



In this last level, you meet the king at last, who revealed that he has his own Dice of Gods as well, and will play against you before he listens to your demands, or he'll send you off to be executed.

6A

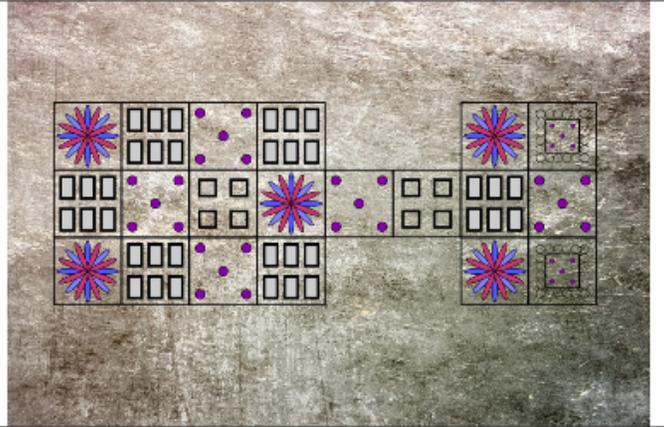
2:10



The goons often play Game of Ur during their breaks, so they have a standard wooden set ready to play. The goon plays the game regularly with others so his skill is fresh. If you lose against him, you'll be chased out of the palace and risk getting thrown into jail if you try to come back. If you win, he'll lead you to the inner chamber of the palace.

8A

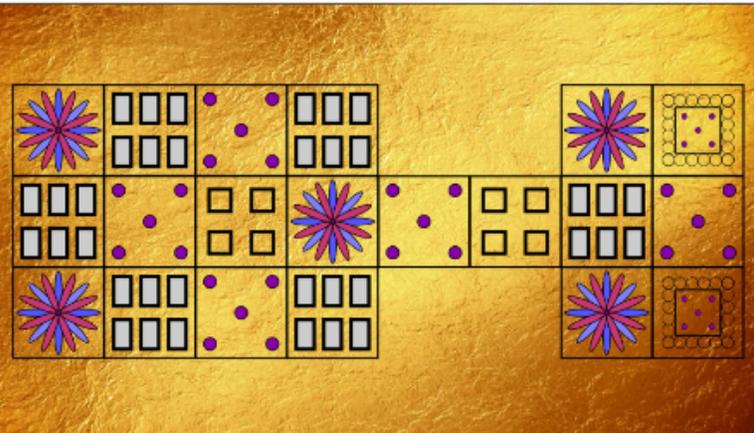
2:14



This level takes place on the magic stone door itself. If you lose against it, a lightning bolt will strike down to kill you. If you win, the door opens to the king's chamber.

10A

2:18



The game takes place on the king's own golden board. The king uses his own set of dice. In addition to being very good at the game, the king may also cheat, and he cannot roll a zero. Losing will result in your execution. Winning will result in him agreeing with your demand to stop the river diversion project. At least for now. The game ends.

Character Script

Level 1 - Meeting Zimu

Zimu: Listen... The king isn't just flooding your home. He's diverting the entire river of Ur to turn the whole kingdom into a desert. Your home just happen to be in the way of his diversion.

Zimu: Our entire civilisation is doomed if he pulls it off. Our agriculture, water source, medication, alchemy... All is gone if we lose the river.

Zimu: I brought you here because you have our only hope to stand up against the king. The Dice of Gods...

Zimu: As long as you have The Dice of Gods with you, no one can harm you physically. The only way you die is if you harm yourself, die of natural causes, or...

Zimu: If you lose Royal Game of Ur to someone and they kill you.

Zimu: That means knowing how to play Royal Game of Ur well, is essential to your survival.

Zimu: I understand that you haven't had much experience with this, so I'm here to train you.

Zimu: You can keep practicing with me. Don't worry, I won't kill you if you lose. I have no desire to become a tyrant like our king.

Zimu: You are a friend to us now. Even if you lose your power, we will keep our home open to you.

Zimu: Let us begin your training.

End of Level 1 - After Beating Zimu

Zimu: You've done well. I'm sorry for putting the burden of saving a kingdom on your shoulders, but you really are our only hope.

Zimu: You should be on your way to the palace now. Follow the path leading out of our campsite to the nearest town centre. A horse carriage travels between here and the capital city often.

Zimu: Don't worry about paying the rider. He won't be able to refuse an owner of The Dice of Gods on a quest. Save your money on food instead. None of us here can spare much, after all.

Zimu: Good luck, and may The Gods watch over you.

Level 2 - Catching a Carriage

Rider: Ey kid, ye catching a ride? That'll be 200 coins and I'm leaving in a mo...

Rider: What do ya mean you don't have money? Scram before I call the guards! Wait, what have ye got there?

Rider: Oh... The Dice of Gods... I see why ye got the guts to walk up to me like that. Ha, ye got me good.

Rider: Ye know, I used to play Royal Game of Ur everyday with neighbours. We were poor and drew our own board on a piece of parchment. Played off of that instead.

Rider: I ain't got time now since I started working as a carriage rider, but I kept the parchment with me. Like a memento of those good times, ye know?

Rider: Look, the drawing is just a lil faded, but still playable. What do ya say, a game to decide if you can catch my ride for free?

End of Level 2 - After Beating Rider

Rider: Alright alright, ye got me good kid. Ye got me good.

Rider: I'll take ya to the palace free of charge. I don't know what business ye got there, but I don't envy ya at all.

Rider: There are rumours about how the palace is full of death traps and magical curses... Of course no one dared to go inside to find out.

Rider: But if ye so determined, then just be careful in there, alright?

Rider: Get yerself comfortable on yer seat, I'll take off real soon.

Level 3 - Palace Entrance

Palace Goon: Halt! Who goes there?

Palace Goon: None may enter the palace without approval from the king!

Palace Goon: Hey, are you deaf? Why are you still standing there?

Palace Goon: If you aren't walking away now I'll make you go away myself!

Palace Goon: Well then, that's your choice!

(The palace goon aimed his halberd at your torso and charged. As the tip of the halberd almost made contact with your body, the halberd blade snapped off and flew onto the ground. The goon staggered.)

Palace Goon: Whoa what the... What trickery is this!?

Palace Goon: Oh... The Dice of Gods. You mean business with the king, are you?

Palance Goon: Well I'm not letting you in so easily just because I can't stab you.

Palance Goon: Royal Game of Ur just happen to be our favourite pass time when we're on a break. As powerful as the king is, he encouraged us all to play it, probably to prepare ourselves for pests like you who can't be killed.

Palance Goon: You'll either have to beat me to let you in, or you're going to scam!

End of Level 3 - After Beating Goon

Palance Goon: Ah well, you've won fair and square. Very well, I'll take you to the king's chamber.

Palance Goon: I'll warn you though, even I haven't been inside myself.

Palance Goon: The chamber is guarded by this freaky... door... thing.

Palance Goon: Only the king himself and those authorised by him directly had ever gone inside.

Palance Goon: Anyone who tries to go in without permission, well the door seems to do something that would make them never come back. Pretty scary if you ask me.

Palance Goon: Don't say I didn't warn you. I'm only taking you as far as that freaky door, then you're on your own from there. Got it?

Level 4 - Door to King's Chamber

(The palace goon left you at a massive, ominous stone door that must have been 5 floors tall. The door is filled with ornate carving, and a monstrous face in the middle.)

(As you got closer, the eyes on the door face glowed faintly. You also felt a subtle vibration coming from your dice.)

(You took The Dice of Gods out of your pocket. The whole floor started to shake, and a stone panel slid out from under the door face horizontally.)

(Above the stone panel, you saw a complete Royal Game of Ur board carved in, with playing stones sat on both sides of the board, ready to play.)

(The black stones on the door side of the board shook gently. It seemed like they would move with the door's magic, once the game begin.)

(Suddenly you lost hold of The Dice of Gods. They floated away from your hand and landed on the board. A rumble of thunder could be heard from inside this floor. There was no turning back. Your game with the door began.)

Level 4 - After Beating Door

(You have bested the magical door with seemingly perfect logic of the game programmed into its artificial stone mind.)

(The whole floor shook again, and this time, very slowly, the whole door sunk down into the floor. You grabbed The Dice of Gods from the game board before they were sunk along too.)

(After a moment waiting for the big, heavy door to sink all the way down, a massive, luxurious chamber decorated in red carpet, gold trimming and various ornamental pillars laid bare in front of you.)

(You've came a long way. You stepped inside to approach your final destination - the mad king himself.)

Level 5 - The Mad King

King: There you are. You've made it! Congratulations. You are at your last stop.

King: Surprised to see me so enthusiastic? Well, I have been expecting you.

King: I wouldn't be much of a king if I didn't know about The Dice of Gods, after all, right?

King: Do you even know how The Dice of Gods came to be?

King: Long ago, our ancestors warred among themselves. The bloodshed was so great that humanity was facing extinction.

King: The Gods could not bear to allow humanity to fall this way, so they created The Dice of Gods, crafted with precious stones from the deep earth.

King: They presented The Dice to our warring ancestors, and proposed that they should settle their differences over a game of dice and wit on Royal Game of Ur.

King: As our ancestors turned to Royal Game of Ur to settle their score, they forgot about violence and bloodshed. Hatred and vengeance might still remain, but The Gods have showed them that violence is not the solution.

King: The kingdom of Ur, and everyone, you and me, all exist today thanks to The Gods' intervention.

King: You might wonder how I knew all these.

King: The Gods made two sets of dice for both sides of our warring ancestors. One set was made with bloodstone.

King: The other set... was made with lapis lazuli.

(The king reached into his pocket and held a handful of items in his hand. Then he opened his palm in front of you. He was holding the lapis lazuli version of The Dice of Gods.)

(You also noticed something unsettling... one of the dice does not have an empty tip. This set of dice cannot produce zero.)

King: You see, our ancestors fought among themselves because they did not have one leader uniting them.

King: Whoever possessed this set, was destined to be The King of Ur. The King would never have zero as a result.

King: I'm afraid you have inherited the inferior fate as The Gods intended.

King: But it is also The Gods' will to settle any grievance with a Royal Game of Ur, instead of violence, isn't that right?

King: Let's settle this now that you are here.

End of Level 5 - After Defeating the King

King: You did well, against all odds. You actually bested a king. Well done.

King: Go and tell your people that I am halting all river diversion project, effective immediately.

King: You have earned the demands of your people. Who knows, maybe you will join my council one day too? I'll let you have a think about it.

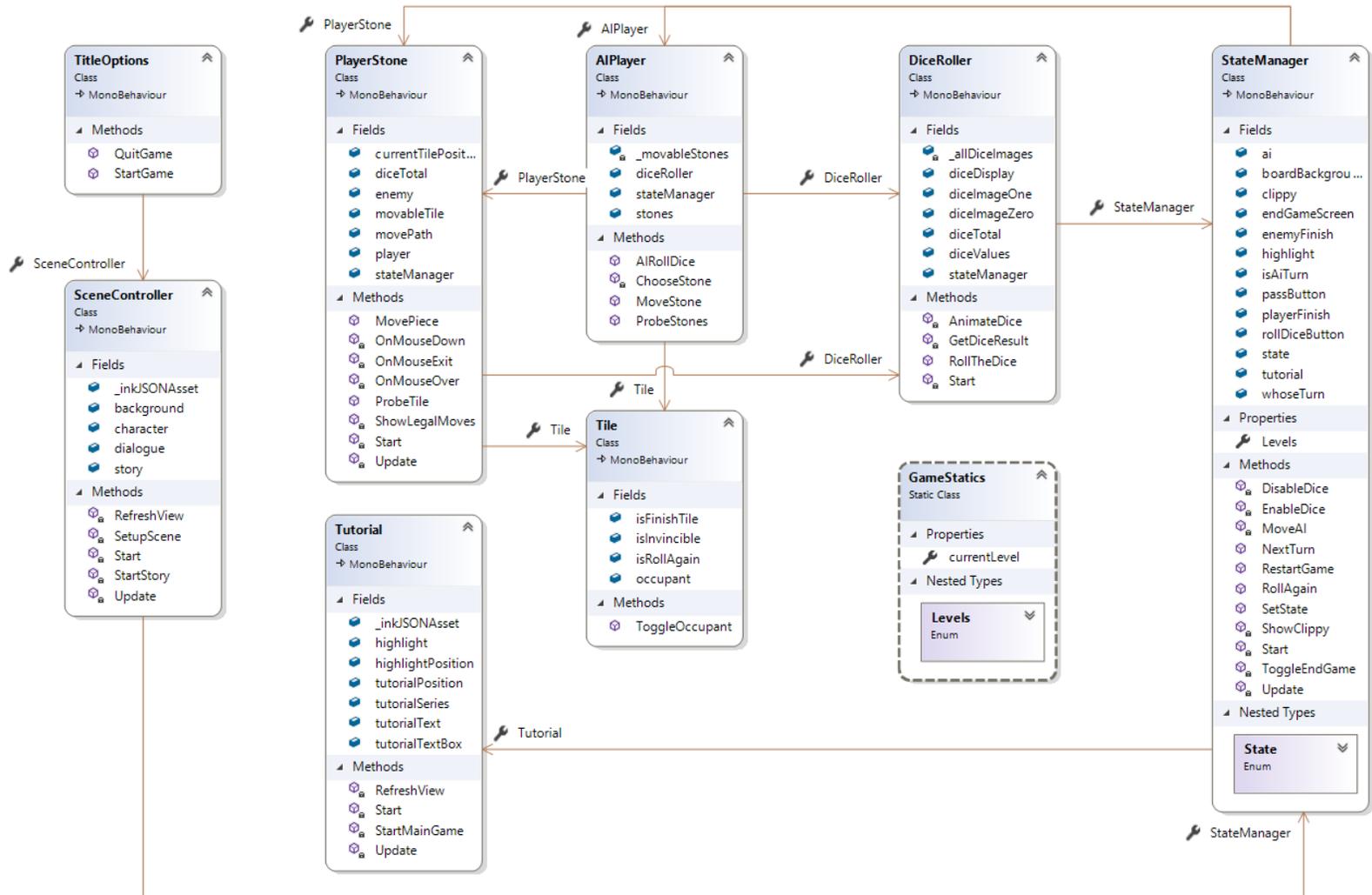
King: Rest assured, all of you will not have to worry anymore. You may go home with the good news and enjoy your well earned rest.

King: For now...

[END]

Diagrams

Class Diagram



The code design in The Mad King of Ur does not follow a strict object-oriented pattern. Instead it is component-based, which is a design pattern that Unity engine itself is built around. Component-based game programming treats each part of a game, such as a player, an enemy and a button as individual components that could interact with anything, with no restriction on object hierarchy or relation. In game programming, this offers great flexibility and simplicity to the code design.

PlayerStone

This is the object for a single playing piece in The Mad King of Ur, used by both the player and the AI (more on AI usage later). Each player stone contains dice roll result, which is used to calculate possible moves, as well as managed by StateManager game states to determine the current turn and state of the game (for example, stones can only be moved after the dice roll state). Each stone also keeps track of its own array of tiles for movement tracking.

AIPlayer

This class simulates the actions of a real human player by possessing the ability to call DiceRoller class methods directly, as well as making its own decision on stone and tile selection. It has its own array of PlayerStone objects that it controls during game play, which are equally managed by StateManager. AIPlayer contains more functions related to tiles to calculate best moves for each stone, after a dice roll.

DiceRoller

This is the object that contains all dice throw related functions, including calculating the total, who the dice roll was for, and storing the dice result for the move calculation of a stone. It communicates dice results to StateManager as well for UI display and state changes.

StateManager

This class is the functional mastermind behind the game. It manages game states such as waiting for dice roll, waiting for post-dice roll move, and assigning either the player or AI to the current turn. It also handles win-loss checks, scene transition and UI display. It is responsible for giving PlayerStone the go-ahead to either roll a dice or make a move.

Tile

A Tile object represents an individual square on the game board. It contains information such as the function of the tile (for example roll-again, winning tile, invincible tile). A Tile object is passive, as in it does not initiate any action on its own. Instead it contains

information for AIPlayer to calculate best moves, and for StateManager to the state of next turn.

GameStatics

This is a static class that holds information on available level progression, and keeps track of current level with a static enumerated variable. In Unity, static objects are persisted globally. This helps all aspects of the game, including cutscenes and game board, to reference the level progression of the game in order to provide correct content.

Tutorial

This class handles the tutorial display at the start of the first level with Zimu. The tutorial is done as a separate class to call a unique dialogue dedicated to the tutorial, along with its own unique UI and interactions.

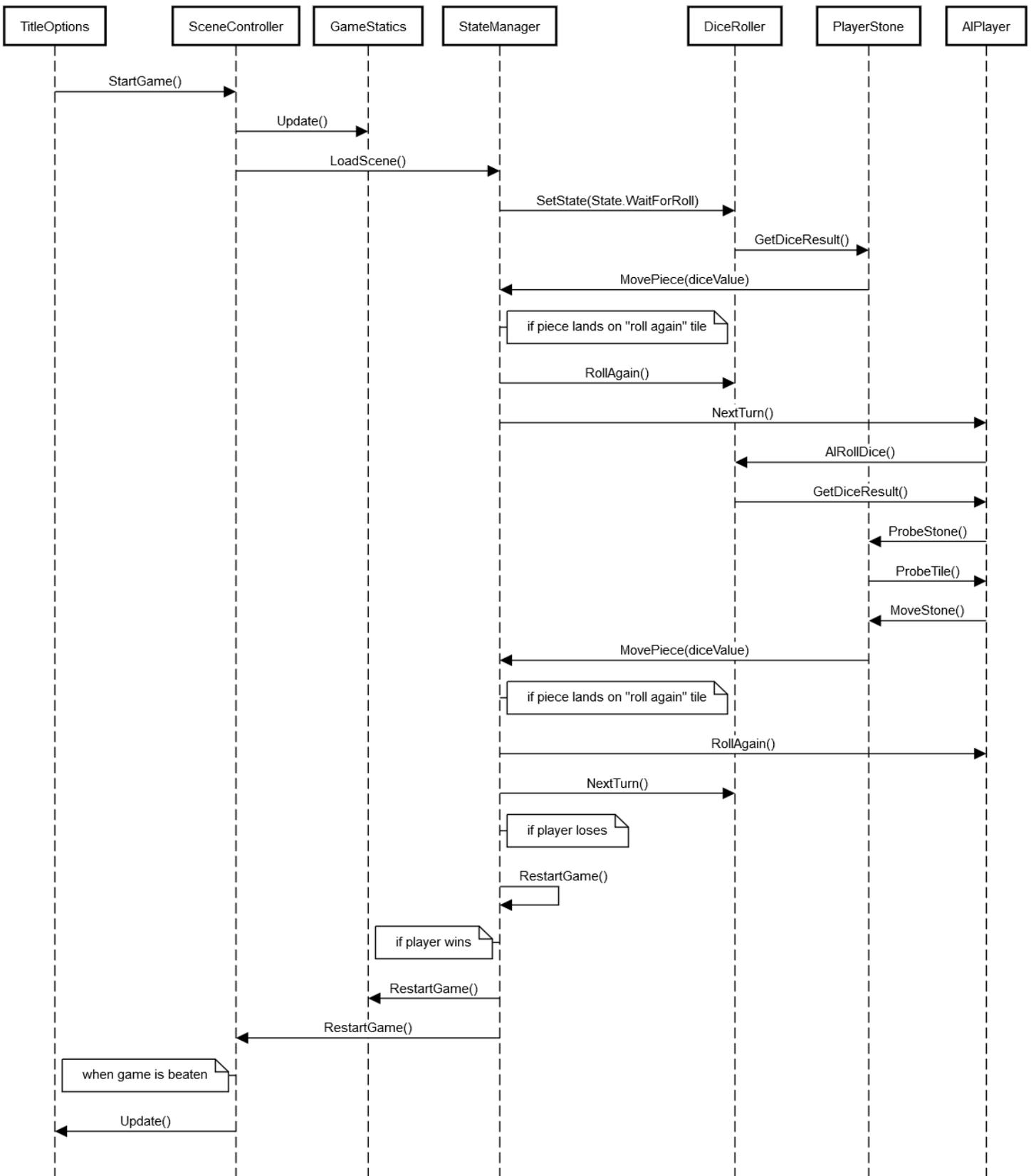
SceneController

This is the class that handles scene transition and dialogue display of all cutscenes. This is also where the plugin Ink is primarily used for showing dialogues, in addition to Tutorial (more information under *Code and A.I Processes*).

TitleOptions

A simple class to provide button methods for the title screen, which consists of Start Game and Quit Game buttons.

Sequence Diagram



-
1. Player presses the start game button which triggers `StartGame()`.
 2. `SceneController` plays the dialogue, then updates the level in `GameStatics` and starts the game board for that level, initiating `StateManager`.
 3. `StateManager` sets the game state to `WaitForRoll` to enable dice rolling.
 4. After the dice roll, `DiceRoller` passes the result to `PlayerStone` to evaluate legal moves of each stone.
 5. Player clicks on a `PlayerStone` object to move a piece to its legal tile, triggering `MovePiece()`.
 6. `StateManager` decides the next turn. If the piece lands on a roll-again tile, the sequence repeats from step 3. Otherwise, `StateManager` calls `NextTurn()` to pass the turn to `AIPlayer`.
 7. `AIPlayer` triggers dice roll via `AIRollDice()`.
 8. `DiceRoller` passes the dice roll result back to `AIPlayer`.
 9. `AIPlayer` inquires for `PlayerStone` to collect all legal moves information.
 10. `PlayerStone` runs `ProbeTile()` on all AI pieces and return stones with legal moves to `AIPlayer`.
 11. `AIPlayer` makes a move with `MoveStone()`.
 12. `PlayerStone` moves AI's chosen piece and informs `StateManager`.
 13. `StateManager` repeats the check at step 6.
 14. After every turn, `StateManager` also checks if either side has won (having all 7 pieces on the finish tile). `RestartGame()` is called when this is the case, and the result depends on who won. If AI wins, `StateManager` resets the game on the same level. If player wins, `StateManager` updates `GameStatics`, and transits to `SceneController` for the next level dialogue.
 15. When the game has been beaten, `SceneController` transits back to title screen.

Target Market Research

In 2017, market research firm Arizton released a report¹ on board games industry forecast for 2018 - 2023 in Europe. The key takeaways from this report that support the choice of creating The Mad King of Ur are as follow:

- *"Fantasy-based TV shows are driving the demand for games in the Europe board games market during the forecast period"* - This point is referring to shows like Game of Thrones, Supernatural, The Witcher, Castlevania and more, though with Game of Thrones being the major success in this genre. The success of these current fantasy TV shows piqued interests in old-fashioned antiques, including traditional board games like Royal Game of Ur.
- *"Table top games to boost critical thinking among end-users in the Europe board games market"* - Unlike action-oriented video games, which utilise mental agility and reflexes, board games are structured with greater emphasis on long-term thinking. It's been proven many times in scientific researches that brain exercises slow the effect of dementia and aged memory loss². Board games provide such brain exercise with much lower bar of entry than reflex-based games.

While this research supports board games in general, The Royal Game of Ur was popularised by a video³ from British Museum, featuring the curator Irving Finkel who deciphered the rules for the ancient board game, and a famous YouTube star Tom Scott. The video accumulated nearly 3 millions views, with a thumbs-up and thumbs-down ratio of 127,000:951 at the time of writing (which means it takes every ~134 viewers to have someone dislike it). The video is clearly in majority's favour, with new comments still being added to this day.

1

<https://www.researchandmarkets.com/reports/4603486/board-games-market-in-europe-industry-outlook#rela0-4244537>

² <https://www.webmd.com/alzheimers/guide/preventing-dementia-brain-exercises#1>

³ <https://www.youtube.com/watch?v=WZskjLq040I&t=1026s>

On the other hand, one of the largest PC game distribution platforms, Steam, currently does not feature as many games tagged as Board Games. The board game genre stands in comparison with the following popular genres at the time of writing:

Steam Game Tag	Number of Games
Adventure	23441
Simulation	16116
Strategy	14803
RPG	13430
Puzzle	4674
Sports	3398
Racing	2707
Platformer	2652
Visual Novel	1882
First Person Shooter	1805
Sandbox	1631
Board Game	379

This shows that, despite the popularity The Royal Game of Ur has received, and the growing interest in board games due to successful fantasy TV shows, **there's a large untapped potential for board game genre** in the video game industry.

In conclusion, the decision to create The Mad King of Ur as a video board game is supported by a great potential, nesting between the positive forecast for board games and lack of board game releases in the video game market. Video board games in general is a worthy product to be made for the market, and The Royal Game of Ur in particular is a fantastic choice for complementing the historic fantasy genre enjoyed by TV audiences today.

Code and A.I. Processes

Ink Plugin

The Mad King of Ur utilises a free Unity plugin for distributing dialogues in the cutscenes and tutorial: Ink by Inkle Studio. Ink is its own choose-your-own-adventure scripting language for creating branching dialogue text games, but the Unity plugin allows scripted dialogues written in Ink to be directly imported into Unity, with helper functions to continue dialogues and fetch story variables.

A.I. Processes

The Mad King of Ur is a board game, therefore A.I. plays a heavy role in being the player's direct opponent. The Royal Game of Ur was much easier for implementing an A.I. than chess, and can be tweaked with various levels of complexities in order to implement different levels of difficulty.

The game code contains a standalone object AIPlayer to imitate all player actions. Anything that the player can do, AIPlayer can also call the functions to perform the same. This includes rolling dice, choosing a stone to move, roll again and kicking a player stone off.

```
public void AIRollDice()
{
    diceRoller.RollTheDice();
}
```

AIRollDice() is called when it is the AI's turn.

```
public void ProbeStones()
{
    foreach (var stone in stones)
        if (stone.currentTilePosition + diceRoller.diceTotal < 16)
            if (stone.ProbeTile(diceRoller.diceTotal) != null)
                _movableStones.Add(stone);
    Debug.Log(_movableStones + " " + diceRoller.diceTotal);
}
```

After AIPlayer has rolled the dice, it will scan through all its stones for ones that has eligible move, and put them into a list.

```
private PlayerStone ChooseStone()
{
    _movableStones.Sort((y, x) =>
        x.currentTilePosition.CompareTo(y.currentTilePosition));
}
```

```
foreach (var stone in _movableStones)
{
    var movableTile = stone.ProbeTile(diceRoller.diceTotal);

    if (movableTile.isRollAgain) return stone;
    if (movableTile.isInvincible) return stone;
    if (movableTile.occupant!=null&&movableTile.occupant.player ==
stone.enemy && !movableTile.isInvincible) return stone;
    if (movableTile.isFinishTile) return stone;
}

var rand = new Random();
var chosenStoneIndex = rand.Next(_movableStones.Count);
var chosenStone = _movableStones[chosenStoneIndex];
return chosenStone;
}
```

The above code is where the “brain” of the AIPlayer lies. After getting a list of stones with legal moves, it sorts the stones by current position – the stone that is the most ahead is considered first, for the closer a stone is to the finishing point, the closer you are to your winning objective, hence that’s always top priority.

After the list of stones is sorted, it goes through each stone and ask these questions in the following order:

1. Will the stone end up in a rosette tile (roll again)?
2. Will the stone end up in an invincible tile (the rosette tile in the centre)?
3. Will the stone end up on an enemy stone who isn’t on an invincible tile (therefore able to knock the enemy stone off)?
4. Will the stone end up in the finishing tile?

At any point if the above question is yes, that stone is chosen to be the move the AIPlayer makes.

If all of the above is no for the current stone in the list iteration, the next stone in the list is selected to ask these questions again.

If no stone in the list can say yes to any of the above questions, a completely random stone is chosen to be the AIPlayer’s move.

Developer Documentation

AIPlayer

Class for controlling AI player.

AIRollDice ()	Rolls the dice for AI turn.
ChooseStone ()	The main AI process for deciding which stone to move after a dice roll. Returns: A PlayerStone object from AI side.</returns>
MoveStone ()	Picks an AI piece to move. For the king's level, if AI rolls 0, it gets to roll again.
ProbeStones ()	Checks legal moves for each AI pieces after a dice roll.

DiceRoller

Main class for handling dice rolls.

AnimateDice ()	Simulates dice roll animation by rotating the dice sprites as a coroutine, then calls GetDiceResult() at the end.
GetDiceResult ()	Sums up the final dice result and updates game state to allow the player/AI to pick a move.
RollTheDice ()	Adjusts UI displays after dice is rolled, and begins the coroutine to animate the dice.
Start ()	Sets up dice array.

GameStatics

Static class to keep track of current level progression.

currentLevel	Static property to keep track of current level.
---------------------	---

PlayerStone

Main class for an individual stone, or piece.

MovePiece (int diceValue)	Moves a piece to its destination, if it has a legal move. Then moves onto the next turn. Parameter: Final dice roll result
OnMouseDown ()	Moves the selected piece with legal move on left mouse click.
OnMouseExit ()	Un-highlights legal move indicator when mouse cursor leaves a piece.
OnMouseOver ()	Displays the legal move of each piece on mouse over, after dice roll. Calls ShowLegalMoves(...) upon hovering over a piece.
ProbeTile (int diceValue)	Checks the property of the destination tile for a piece. Parameter: Final dice roll result Returns: Destination tile object and its properties.

<code>ShowLegalMoves (int diceValue)</code>	Calculates the selected piece's move based on dice roll result. Parameter: Dice roll result</param>
<code>Start ()</code>	Sets up the starting positions of pieces.
<code>Update ()</code>	Provides real time update of pieces depending on their location.

SceneController

Main class for handling scene transitions in the game. Used to transit between cut scenes and game board.

<code>RefreshView ()</code>	Grabs next dialogue line and displays it.
<code>SetupScene (string bgImage, string chImage)</code>	Shortcut method for placing background image and character image correspondent to the current level in the cut scene. Parameter "bgImage": Background image for the cut scene. Parameter "chImage": Character sprite for the cut scene.
<code>Start ()</code>	Sets up the cutscene based on current level progression.
<code>StartStory (GameStatics.Levels level)</code>	Begins the current level's cut scene dialogue. Parameter: The current level progression in game.
<code>Update ()</code>	Detects left mouse click during cut scene to continue dialogue, or moves to next level / game board after the dialogue is finished.

StateManager

Main class for managing various states during game play.

<code>DisableDice ()</code>	Hides the Roll Dice button.
<code>EnableDice ()</code>	Turns on Roll Dice button if it is the player's turn to roll dice, or asks the AI to roll the dice.
<code>MoveAI ()</code>	Commands the AI to proceed if it is the AI's turn.
<code>NextTurn ()</code>	Switches between player and AI turn, and enables dice roll.
<code>RestartGame ()</code>	Progresses the game after the match has finished. Either moves to the next level if the player has won, or restarts the level if the player has lost.
<code>RollAgain ()</code>	Allows the player of the current turn to roll again, without switching player.
<code>SetState (State newState)</code>	Updates the current state of game play, and calls methods correspondent to the state. Parameter: The state to switch to
<code>ShowClippy (bool x)</code>	Displays the ever-so-helpful Clippy to remind the player how to play. Parameter: To display text or not
<code>Start ()</code>	Sets up the game board based on current level progression. Also starts the tutorial sequence if current level is the first level.

State	List of possible states during game play.
ToggleEndGame (bool win)	Finishes the game and displays a button to continue to the next step. Parameter: If the player has won or lost
Update ()	Updates various UI displays on the game board in real time. Also detects if the game has finished (with either side reaching 7 finished pieces).

Tile

Main class for an individual tile and stores information about a tile.

ToggleOccupant (PlayerStone player)	Keeps track of the tile's occupancy. Parameter: The PlayerStone object that is occupying the tile
--	--

TitleOptions

Main class for the title screen.

QuitGame ()	Exits the game.
StartGame ()	Sets up new game after Start Game button is pressed, starting from the very first level.

Tutorial

Main class for running the game tutorial at first level.

RefreshView ()	Continues the tutorial dialogue if it is not finished yet.
Start ()	Detects if the current level is the first level, then sets up the tutorial display and dialogue.
StartMainGame ()	Begins the main game after tutorial dialogue is finished, and hides the tutorial UI.
Update ()	Continues the tutorial dialogue if left mouse button is clicked.