Kenny Hu Kewei
Andrew ID: keweih

Term Project Proposal

Project description

This term project will attempt to develop a python game called:
“King of the Hill, Man vs Machine”
The game is a 2d top-down game that involves a single player battling a team of AI opponents within a map with covers to control an objective point. Player and AIs will start the game at corners of the map while the objective point will be located at the centre of the map.
There will be a fog of war restricting players line of sight throughout the game. There will also be a cue system where players and AI are alerted to others performing actions (shooting, moving) at a certain location if they are within “earshot”.
There is a shooting element in this game, players and AIs can shoot another player as long as they are in line-of-sight. A player or AI dies when it has been hit 3 times. Killing an opponent grant the team points.
The player or AI can claim the objective point by standing on top of the control point when no one from the opposing side is also on the objective point. When a team is controlling the objective point, the team earns points.
The win conditions for each team (player and AI) is to either kill all members of the opposing team or end the round with more points

Competitive analysis

King of the hill has been a classic first-person shooter game mode for many years. Popular iterations of it have been included in popular titles like “Team Fortress” and “Overwatch”. Yet these games involves symmetric teams fighting over the objective while **“King of the Hill, Man vs Machine”** will include asymmetric combat between the player and the AI. There are also popular top-down king of the hill games like those in the Army Men series. Yet the emphasis of those games is real-time strategy while the emphasis of **“King of the Hill, Man vs Machine”** is that of a tactical shooter. Finally, the fog of war feature is a staple feature borrowed from dungeon explorer games.

Kenny Hu Kewei
Andrew ID: keweih

Structural plan

The game will utilise 2 **models** simultaneously

Model	What is included	What is it for
Coordinate	● Player and AI coordinate ● Bullet coordinate	● Smooth player and AI movement ○ Object collision logic ○ Draw players ● Unrestricted range of fire for player ○ Bullet hit registration
Grid	● Player, bullet and AI grid coordinate ● Map in grid coordinate	● AI pathfinding ● Fog of war generation ● Bullet and wall collision logic

Maps will be generated and stored in Grid from in an external file

Other game level objects/functions	What is it for
Points for each team	Used to decide the winner
Game timer	
Cues location	Display cues for player Help AI make gameplay decision
Capture point	Checks for players or AI who is currently the king of the hill
Mouse position	Track cursor position
RedrawAll	Draws all respective objects on canvas

Player Design	
Men attributes	**What is it for**
Health	
Coordinate position	
Grid position	An abstraction of the player's real coordinate obtained by a view to model function, player's small movements within a single grid

Kenny Hu Kewei
Andrew ID: keweih

	coordinate confine is ignored
List of “can see” grids	Grids that are visible to the player through the fog of war
Men functions	**What is it for**
Movement	Takes user input and checks against wall collision, updates models
Fog of war (stored in an external file)	Decides which grids can the player see

AI Design	
AI attributes	**What is it for**
Health	
Coordinate position	
Grid position	
Current objective	The objective (i.e. capture point, trap player) the AI is trying to achieve
Current destination	The destination for pathfinding
AI functions	**What is it for**
Ability to shoot player	Checks if the player is in AI line of fire
Pathfinding (stored in an external file)	Uses breadth-first graph search to find the shortest path to the destination
Objective deciding	Considers the current game situation to determine optimal objective for AI
Movement	Executes path from pathfinding function

Men and AI will be subclasses of larger player class, objects will be stored in an external file

Bullet Design	
Bullet attributes	**What is it for**
Bullet position	The coordinate position of bullets

Kenny Hu Kewei
Andrew ID: keweih

dy,dx of bullet	Change in bullet position
Bullet functions	**What is it for**
Bullet Generation	Creates a new bullet for player using cursor location relative to player
Bullet movement	Moves bullet using dy,dx
Bullet hit detection	Checks if bullet hit players, AIs by checking coordinate
Bullet wall collision	Check if the bullet has entered a wall grid and removes bullet if required

Bullet object and functions will be stored in an external file

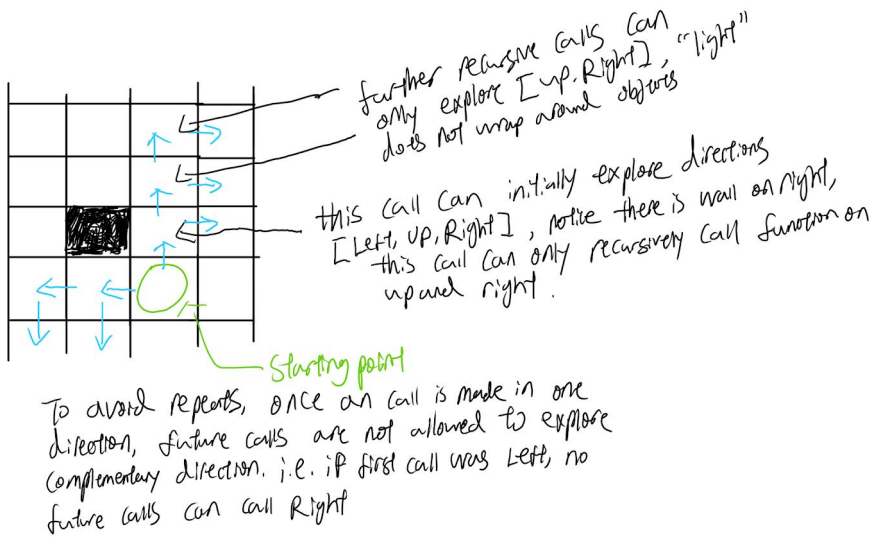
External Files overview:

- GameMap
- FogOfWar
- PlayerAndAI
- Pathfinding
- Bullet

Algorithmic plan

Fog of War	● Utilised modified flood fill algorithm ● Function calls flood fill on player position recursively ● The player can only see a predetermined distance, this is controlled by predetermining the max depth of recursion ● Results are stored as a list of grid cells that the player can see ● To emulate the properties of light not wrapping around objects and to improve efficiency, recursive call follows these rules:

Kenny Hu Kewei
Andrew ID: keweih

Pathfinding	• Breath first graph search
	• Start with an empty list
	• The result will be stored in a list of lists of moves
	• While true
	  ○ Pop the first element(list) from the main list
	  ○ Get legal moves from the current position
	  ○ For each legal move
	  ■ Do move by adding the move to the back of the list
	  ■ If the destination reached, break and return the list of steps
	  ■ Else, append the list to the back of the main list
	• The first returned list will be the shortest path to the destination

Timeline plan

Number	Feature	Planned completion
1	Map and model	20th Nov
2	Player movement	20th Nov
3	Fog of war	20th Nov
4	AI pathfinding	23th Nov
5	Bullet physics	24th Nov

Kenny Hu Kewei
Andrew ID: keweih

6	AI tactical decision making	25th Nov
7	Cue integration	26th Nov
8	Visual overhaul 1	26th Nov
9	AI strategic decision making	27th Nov

Version control plan

Each version of the program will be stored on google drive
The naming convention of each version will be as follows
V 0.(major feature completed).(date).(where I stopped working)
E.g. “V 0.3.2411.bulletGen” is the name of the version where AI pathfinding is done, saved on 24th Nov and last worked on function is bullet generation

For external files, the current working version will be the file being called by the main function, previous versions will be saved with version number with same naming convention

TP2 Updates

- Reload mechanic applied to both player and AI to balance shooting gameplay (included in tp2)
- AI now utilises same fog of war mechanic to only shoot player in line of sight(included in tp2)
- Dead AI will remain on screen as corpses, corpses subclass created (included in tp2)
- Implemented cheat, godMode with no fog of war and no reload time (included in tp2)
- Visual cues changed to footsteps, footstep object to be created (included in tp2)
- AI now also chases enemy shots as its attracted to gunfire (included in tp2)
- New maps to be included (partially included in tp2)
- Team fights may be included
- Gunshot graphics may be included
- Graphic to be improved
- UI to be added

TP3 Updates

- Improved UI and graphics
- Incorporated side scroll (userthon feedback)
- Allows rescaling of game (userthon feedback)
- Increased user tutorial cues (userthon feedback)
- Enhanced cues with tracks

Kenny Hu Kewei
Andrew ID: keweih